Information Propagation in Complex Networks
Structures and Dynamics

Märtens, Marcus

DOI
10.4233/uuid:b1a1ead7-a631-4f05-b9a9-17a1be6e15e1

Publication date
2018

Document Version
Final published version

Citation (APA)
Märtens, M. (2018). *Information Propagation in Complex Networks: Structures and Dynamics*. [Dissertation (TU Delft), Delft University of Technology]. https://doi.org/10.4233/uuid:b1a1ead7-a631-4f05-b9a9-17a1be6e15e1

Information Propagation in Complex Networks

Structures and Dynamics

Information Propagation in Complex Networks

Structures and Dynamics

Proefschrift

ter verkrijging van de graad van doctor
aan de Technische Universiteit Delft,
op gezag van de Rector Magnificus prof. dr. ir. T. H. J. J. van der Hagen,
voorzitter van het College voor Promoties,
in het openbaar te verdedigen op maandag 8 januari 2018 om 15:00 uur

door

Marcus Märtens

Master of Science in Computer Science, Universität Paderborn, Duitsland,
geboren te Aschersleben, Duitsland.

Dit proefschrift is goedgekeurd door de

Promotor: prof. dr. ir. P. F. A. Van Mieghem
Copromotor: dr. ir. F. A. Kuipers

Samenstelling promotiecommissie:

Rector Magnificus,	voorzitter
Prof. dr. ir. P. F. A. Van Mieghem,	Technische Universiteit Delft
Dr. ir. F. A. Kuipers,	Technische Universiteit Delft
Onafhankelijke leden:	
Prof. dr. M. J. G. van Eeten	Technische Universiteit Delft
Prof. dr. ir. M. J. T. Reinders	Technische Universiteit Delft
Prof. dr. ir. P. H. M. Spronck	Tilburg University
Prof. dr. ir. A. Iosup	Vrije Universiteit Amsterdam
Dr. L. Douw	Vrije Universiteit Medisch Centrum Amsterdam

Keywords: information propagation, functional brain networks, toxicity, multiplayer online games, network epidemics, epidemic spreading model, complex networks, symbolic regression

Printed by: Ipskamp Printing

Front & Back: Inference and propagation of waterwaves during rain.

ISBN 978-94-028-0907-7

An electronic version of this dissertation is available at
http://repository.tudelft.nl/.

To my family.

Contents

SUMMARY

Knowledge and information have been a powerful force of success and development in human societies from the beginning. However, in the digital age, our capabilities of collecting and transporting information have grown into new dimensions. Modern communication networks like the Internet span the whole earth and start to reach out even into space. Online social networks are a huge machinery used by billions of people on a daily basis, to receive news, post memories and organize activities. Although we were fast to adapt these new technologies, we do not fully understand all the consequences of this high level of connectivity.

This thesis is a contribution to a deeper understanding of how information propagates and what this process entails. At its very core is the concept of the network: a collection of nodes and links, which describes the structure of the systems under investigation. The network is a mathematical model which allows to focus on a very fundamental property: the mutual relations (links) between information exchanging agents (nodes). This simplicity makes networks elegant, as no specifics of any supporting hardware are needed to reason on this high level of abstraction. The developing field of network science led to countless applications of the network model to all sorts of complex systems in nature and technology. Naturally, it became an essential part of many multi-disciplinary research projects. Therefore, understanding how information propagates in networks enables us to learn and conceivably control the intricate processes, which we observe in complex systems. Since complex systems are the driver for this research, the first three chapters of this thesis are studies based on data collected from vastly different application domains, after more fundamental research is addressed in the later parts.

Chapter 2 deals with the interaction of players of a popular multiplayer online game. Due to the competitive design of the game, teams are formed ad-hoc and compete with each other for victory. Some of the players exhibit anti-social behavior towards their teammates, which is known as toxicity. We analyze how toxicity in player networks emerges by developing a toxicity detector, highlighting possible triggers and analyze the disposition of players towards toxic teammates. Furthermore, we show how toxicity is linked to game success.

Chapter 3 continues with a study of the human brain as a functional network. Information processing in the brain is measurable with technologies like magnetoencephalography. From such measurements that were collected from a group of subjects, the phase transfer entropy is computed as a quantity that reflects information exchange. When associated with the links between brain regions, unusual high numbers of certain substructures are observed in this network. We find that one of these substructures, the bidirectional two-hop path, to be highly abundant and robust within different frequencies bands, which highlights its importance for the propagation of brain activity. A clustering of the network based on these frequent substructures reveals a spatially coherent organization of important brain regions.

A common symbol of propagation is the virus, which is at the center of the third data-driven analysis of this thesis in Chapter 4. More precisely, we research the digital version of the virus, the computer worm, and analyze its propagation by epidemic network models. With epidemic models, the state of the nodes in a network can be described as susceptible or infected. An infection process and a curing process determine how the nodes are changing between those states. We extend on the standard epidemic models, the SIS model, by a time-dependent curing rate function to reflect the changes in the effectiveness of the active worm removal. Once we set the curing rate function, the empirical worm data are fitted and analyzed on multiple scales from the global over the country down to the autonomous system level. The fitted model explains how computer worms or similar self-replicating pieces of information might change in their effectiveness over long periods of time.

The SIS model returns as a central piece in Chapter 5 again. Although spreading processes are frequently modeled in isolation, the dynamics of many real-world applications are often driven by the interaction of multiple of such processes. These interactions can range from viruses that compete for susceptible nodes to viruses that mutually reinforce their propagation. We study the special case of superinfection, in which one dominant virus spreads within the infected population of a weaker virus. We highlight the conditions for which a co-existence of both viruses is stable and show that extinction cycles become possible if the infection rate of the dominant virus becomes too strong. Furthermore, we show that some of the possible outcomes of a superinfection are difficult to approximate with common mean-field techniques. However, the second largest eigenvalue of the infinitesimal generator of the underlying Markov process is potentially linked to co-existence and thus stability.

Chapter 6 is a study on the capabilities of symbolic regression for network properties. We develop an automated system based on Genetic Programming which is able to be trained by families of networks to learn the relations between several of their properties. These properties can be features of the networks like the eigenvalues of their adjacency or Laplacian matrices or network metrics like the network diameter or the isoperimetric number. We show that the system can generate approximate formulas for those metrics that often give better results than previously known analytic bounds. The evolved formulas for the network diameter are evaluated on a selection of real-world networks of different origins. The network diameter bounds hop-based information propagation and is thus of high importance for designing network algorithms. A careful selection of training networks and network features is crucial for evolving good approximate formulas for the network diameter and similar properties.

Finally, the thesis concludes with Chapter 7 which revisits the concepts that were developed and provides some critical assessment on their potential and limitations.

SAMENVATTING

Kennis en informatie zijn altijd een drijvende kracht geweest achter de ontwikkeling en het succes van menselijke samenlevingen. In het digitale tijdperk is ons vermogen om informatie te vergaren en te versturen tot nieuwe proporties gegroeid. Moderne communicatienetwerken zoals het internet omvatten de hele wereld en reiken zelfs tot in de ruimte. Online sociale netwerken zijn enorme machines die dagelijks door miljarden mensen worden gebruikt om nieuws te lezen, herinneringen online te plaatsen, en activiteiten te organiseren. Hoewel we deze nieuwe technologieën snel omarmd hebben, begrijpen we alle consequenties van deze hoge connectiviteit nog niet volledig. Dit proefschrift draagt bij aan een beter begrip van hoe informatie zich verspreidt en wat dat om het lijf heeft. Aan de basis van dit werk ligt het concept van een netwerk: een verzameling knopen en lijnen die de structuur beschrijven van het te bestuderen systeem. Een netwerk is een wiskundig model dat ons in staat stelt om ons te richten op een fundamentele eigenschap: de wederkerige relaties (lijnen) tussen de entiteiten (knopen) die informatie uitwisselen. Deze eenvoudige voorstelling maken netwerken elegant; men hoeft niets te weten van de onderliggende hardware van een communicatienetwerk om op een hoger abstractieniveau over deze netwerken na te denken. De ontwikkeling van netwerk theorie als een onderzoeksgebied heeft tot ontelbare toepassingen van het netwerk model geleid, op allerlei natuurlijke en technologische complexe systemen. Het is als vanzelfsprekend een essentieel onderdeel van vele multidisciplinaire wetenschappelijke projecten geworden. Als we begrijpen hoe informatie zich verspreidt kunnen we de ingewikkelde processen die we zien in complexe systemen doorgronden en misschien zelfs sturen. Aangezien complexe systemen de drijfveer achter dit onderzoek zijn, bevatten de eerste drie hoofdstukken van dit proefschrift onderzoeken gebaseerd op data uit verschillende domeinen. Daarna zal in de verdere hoofstukken meer fundamenteel onderzoek behandeld worden.

Hoofdstuk 2 beschrijft de interactie tussen spelers in een populair online spel. Omdat het spel aanzet tot competitie, vormen zich ad-hoc teams die met elkaar strijden om de overwinning. Sommige spelers gedragen zich asociaal naar hun medespelers, dit gedrag staat bekend als toxicity. We analyseren hoe toxicity ontstaat in het netwerk van spelers door een toxicity-detector te ontwikkelen. We benoemen mogelijke triggers en analyseren hoe spelers zich opstellen naar toxic medespelers. Ook tonen we aan de toxicity gerelateerd is aan succes in het spel.

Hoofstuk 3 vervolgd met een onderzoek naar het functionele netwerk van het menselijk brein. Met technieken zoals magnetoencephalography is het mogelijk om de activiteit in de hersenen te meten. De berekende fase overdracht entropie in de metingen van een groep patiënten wordt geïnterpreteerd als een grootheid die staat voor de mate van informatie overdracht. Wanneer die weer wordt geïnterpreteerd als een verbinding tussen gebieden in de hersenen wordt een ongebruikelijk hoog aantal typische substructuren zichtbaar. We hebben ontdekt dat één van die structuren, een bi-directioneel

twee-stap pad, zeer veel voorkomt en in verschillende frequentiebanden. Hieruit blijkt dat die structuur belangrijk is voor informatie overdracht. Door het netwerk te clusteren met behulp van deze veel voorkomende structuren wordt een spatieel samenhangende organisatie van hersen gebieden zichtbaar.

Een virus is een goed voorbeeld van een zich verspreidend fenomeen en is het onderwerp van het derde op data gebaseerd onderzoek in dit proefschrift ,gepresenteerd in hoofdstuk 4. We onderzoeken een digitale versie van een virus, een computerworm, en analyseren het gedrag daarvan met behulp van epidemische netwerk modellen. In een epidemisch netwerkmodel zijn de knooppunten of vatbaar voor het virus ol geïnfecteerd. Een infectieproces en een helingsproces bepalen hoe de knooppunten van toestand veranderen. We breiden het standaard epidemisch model, het SIS model, uit door het helingsproces tijdsafhankelijk te maken. De tijdsafhankelijkheid reflecteert dat het verwijderen van een computer worm eenvoudiger wordt naar mate er meer van bekend is. Eerst stellen we een functie vast voor het tijdsafhankelijke helingsproces en daarna analyseren we de gemeten verspreiding van een computerworm op verschillende schaalgroottes: van globaal tot landelijk tot netwerk niveau. Het model verklaard hoe computer wormen en andere zichzelf kopiërende informatieprocessen van effectiviteit veranderen in de tijd.

Het SIS model komt ook terug als onderwerp in hoofdstuk 5. Hoewel verspreidingsprocessen vaak in isolatie beschreven worden, is er in de realiteit vaak interactie tussen verschillende processen. Deze interactie loopt uiteen van virussen die dezelfde gezonde knopen proberen te infecteren tot virussen die wederzijds elkaars effectiviteit versterken. Wij onderzoeken een speciaal geval van superinfectie waarin een dominant virus zich verspreidt in de geïnfecteerde populatie van een ander, zwakker, virus. We tonen aan onder welke condities beide virussen stabiel kunnen samenleven, en dat er cycli van uitsterving mogelijk zijn als het dominante virus te sterk wordt. Verder laten we zien dat sommige van de mogelijke uitkomsten van superpositie slecht zijn te benaderen met mean-field technieken. Daarentegen is de op twee na grootste eigenwaarde van de infinitessimal generator van het onderliggende Markov proces waarschijnlijk gerelateerd aan het voorkomen van samenlevende virussen en daarmee aan stabiliteit.

Hoofdstuk 6 is een onderzoek naar de toepasbaarheid van symbolische regressie om netwerk eigenschappen te bepalen. We ontwikkelen een geautomatiseerd systeem gebaseerd op Genetic Programming dat getraind kan worden op families van netwerken om de relaties tussen de netwerkeigenschappen te leren. Deze netwerkeigenschappen kunnen kenmerken zijn zoals de eigenwaardes van de adjacency matrix of de Laplaciaan, of kengetallen zoals de diameter of het isoperimetrisch getal. We tonen aan dat het systeem benaderingsformules voor die kengetallen genereerd die vaak betere resultaten geven dan bekende analytische grenzen. De geëvolueerde formules voor de netwerkdiameter hebben we geëvalueerd op een selectie van "real-world"netwerken. De netwerkdiameter begrenst via-via informatie verspreiding en is daarom van groot belang bij het ontwerpen van netwerk algoritmes. Het is belangrijk om de training netwerken en de te trainen eigenschappen zorgvuldig te selecteren om goede benaderingsformules te kunnen evolueren voor de netwerkdiameter en soortgelijke eigenschappen.

Ten slotte sluiten we dit proefschrift af met hoofdstuk 7 waarin we terugkeren naar de concepten die naar voren gebracht zijn en we kritisch kijken naar het potentieel daarvan, en de beperkingen.

1

Introduction

1.1. A Network Perspective

From the dawn of the digital age to our present time, the people of our world have never been more tightly connected on so many layers. The increased volume and velocity at which information is reaching us can be overwhelming and managing the streams of knowledge has become a key skill of many higher professions. From the fundamental works on the small world problem by Travers and Milgram [1] up to the breakthrough work of Watts and Strogatz [2] we have learned by now that we are living in a small world, separated on average only by a few hops. The basic picture of an information exchange incorporating a message that is sent between a sender to a receiver over some channel has been scaled up to whole populations of senders and receivers, which use multitudes of networks of communication channels to broadcast and interact rapidly. Understanding information propagation on a large scale is thus also a question of understanding networks.

Easley and Kleinberg write in their book [3] that "in the most basic sense, a network is any collection of objects in which some pairs of these objects are connected by *links*". The power of this simple definition is its ability to provide context to a large variety of situations. By translating our world into the abstract structure of nodes and links, we are able to focus on the relation between objects, rather than being distracted by their details.

The network perspective does not only help us to find our own place as humans living in a digitized world, but also proves to be a much more fundamental tool, particularly for science. Molecular interactions between proteins [4], structural connections and functional correlations in brains [5], traffic flows of ships [6] and airplanes [7], electrical power grids [8] and financial transactions [9] are only a small sample of applications, which can be understood as networks. Even the causal relationships of space-time, the fabric of our universe, have already been subject to a network model [10].

Part of the appeal of the network perspective might be in the visual qualities of the network representation that often allows us to obtain a bird's eye perspective on the system at hand. After all, many of us already connected some dots with lines as children to marvel at the complete image. However, the direct visual representation of a network by drawing links as lines between nodes has its limitations if confronted with the scale of the complex systems that have grown in nature or were assembled by our hands. If one was to represent the neurons of the human brain as a network, one would have to draw around 86 billion nodes [11], each potentially connected to thousands of others. But already smaller networks can appear cluttered and “messy”, unless sophisticated visualization techniques are applied. Although our visual understanding of networks might be limited when it comes to size, the network still exists as a mathematical object which is amenable for analysis. The goal of such an analysis remains the same: we strive to understand the big picture which is painted by all the abstract small relations.

Looking through the lens of network science means to look at natural phenomena as the result of dynamic processes involving many small connected entities, who influence each other by mutual exchange of information. The goal of this thesis is to apply this lens to a selection of different application domains and focus on the relation between network structure and information flow.

1.2. STRUCTURES SUPPORTING PROPAGATION

When it comes to structure, a network without any connections at all will clearly not facilitate any information flow. On the other hand, a network in which all nodes are connected with each other could be said to allow for a maximum propagation. Both cases are unlikely extremes, which are rarely observed in nature. Typically, man-made or natural grown networks avoid to connect all nodes with each other, as each new connection usually requires some form of investment. Still, we find sparse and tightly-knit connections assembling structures to support the purpose of the corresponding networks. A frequent observation is the emergence of hubs as they are found in scale-free networks [12]. There exists only a minor number of hubs in such networks, but they connect a larger number of other nodes like bridges. As such, we expect hubs to be of importance for information flow. Although important, hubs are not the only structure which requires attention. Sparse connected networks deliver a variety of features worth studying with respect to information propagation.

The structure on the smallest possible level is the link that connects two nodes with each other. There are only a few possibilities at this scale: a link between two nodes might either exists or it might not. In directed networks, a link might be oriented from one node to the other and in weighted networks a real number might be associated with the link. Either way, the link remains the most fundamental building block of any network.

We can observe richer structures if we look at more than two nodes at the same time. Small subsets of nodes (typically three to four), can have different combinations of links between them. Each such combination is called a motif [13] and their frequency in a network can be counted. If there is an overabundance of a certain motif in a network (in comparison to a randomized null-model), it is often argued that this micro-structures did not arise by chance but must carry an important function for the network or the underlying process. We give evidence for this hypothesis in Chapter 3, which analyzes the motif structure of the human brain network.

Following links from one node to the next will create other sub-structures in networks: walks and paths. Particularly shortest paths, which connect two nodes by the least amount of intermediary hops are often argued as likely candidates for an increased information propagation, as short paths might be faster than detours through the whole network. Taking all shortest paths in a network, we can determine which one requires the highest amount of hops. This quantity is called the network diameter and presents a worst case bound for hop-based propagation.

The network diameter is also an example for a network metric, a characteristic number of a network which is generally computed by taking the entirety of nodes and links into account. Consequently, network metrics are often a condensed reflection of some structural properties. Chapter 6 will show how the diameter and other network metrics may be learned from spectral network features.

1.3. The Dynamics of Propagation

While the structures introduced in the last section describe the network features which are used to transfer information, the dynamics describe the rules for the process that takes place. This usually includes how information is stored, how it can change and by which means it is transported.

The most frequent place to store information in a network is in the nodes. For example, in the Susceptible-Infected-Susceptible (SIS) model, each node has one bit of information which determines whether it is in the infected or the susceptible state. Links allow for this information to spread: with a certain probability, a susceptible neighbor of an infected node will become infected as well. The dynamics of this model are at the core of Chapters 4 and 5 of this thesis, which will introduce extended variants and elaborate on further details. The SIS model is known as a compartmental epidemic models, and has been developed to describe the spread of infectious diseases, typically caused by a pathogen like a virus. The virus itself is not explicit in the SIS model, but is represented by the fraction of the node population in the infected state.

The concept of the virus has long surpassed its biological domain and has found its electronical manifestation in the computer virus. But already the idea of self-replicating information that infects hosts in order to propagate is been used as a metaphor, mostly for news or digital content, which are shared amongst individuals and groups. Going "viral" is a synonym for success, outreach, speed, popularity and resilience. While there are certain factors which influence the virality of content [14], we seem to be only at the beginning to understand how those large-scale information cascades come to existence. However, once they appear, they are like an avalanche that rolls through our networks. For the worse, similar to an avalanche, some of those cascades can also be harmful and cause damage.

It is the goal of this thesis to contribute to the knowledge of spreading, but not for the sake of engineering viral marketing strategies, who seek to maximize profit by maximizing spread, but to give society the intelligence to prevent harmful information to grow out of our control. The automated systems and networks which we deployed into our society and even in our social lives have become bigger than many of us would have ever imagined and it is important that we use these technologies responsibly. People need protection from exploitation by our networks and providers of such services need to be educated on how to prevent or diminish possible negative effects. Consequently, undesirable and harmful processes will be frequently at the focus of this thesis, e.g. the spread of anti-social behavior which is the center piece of Chapter 2. Also the spread of a harmful computer worm will be analyzed in Chapter 4, for which the SIS model will yet again be extended to reflect the impact of possible counter-measures over time.

1.4. THESIS OUTLINE

This thesis will begin with three studies of information propagation from different application domains. All of them are based on real-world data and highlight different approaches to analyze the structure and dynamics of the corresponding processes. Chapter 2 is a study of anti-social behavior which is encountered in online gaming communities. First, we develop a system to detect the anti-social behavior before we analyze a potential spread. Chapter 3 continues with a study of the human brain, for which we have a closer look into its functional connectivity network. A clustering based on information flow motifs reveals a higher order organization of the brain, highlighting the close connection between structure and dynamics of information flows. Chapter 4 studies an extended SIS model which introduces time-dependent components. To show its applicability, data from a highly persistent computer worm is analyzed with the corresponding model, giving insights into the effects of deployed counter-measures.

After these specific studies, the following two chapters investigate more general concepts and their implications. Chapter 5 introduces "superinfection" which is an extension to the SIS model that enables us to consider nested spreading processes, for which one virus spreads inside the population of another. Conditions for extinction cycles and stable co-existence of both virus populations are shown by extensive simulations. Lastly, Chapter 6 is a study on the intricate relations between the representation of a network and its properties. By training a genetic programming model on spectral network features, we infer approximate formulas for several important properties, including the network diameter, which provides a lower bound for any hop-based propagation scheme.

Finally, Chapter 7 concludes this thesis by reflecting on the common concepts that have shown to be useful throughout several chapters. The main contributions of the thesis will be discussed, possible limitations are shown and future research is suggested.

2

Detection and Spreading of Toxicity in Multiplayer Online Games

Social interactions in multiplayer online games are an essential feature for a growing number of players world-wide. However, this interaction between the players might lead to the emergence of undesired and unintended behavior, particularly if the game is designed to be highly competitive. Communication channels might be abused to harass and verbally assault other players, which negates the very purpose of entertainment games by creating a toxic player-community. By using a novel natural language processing framework, we detect profanity in chat-logs of a popular Multiplayer Online Battle Arena (MOBA) game and develop a method to classify toxic remarks. We show how toxicity is non-trivially linked to game success.

This chapter is partially based on a published paper [15].

2.1. Introduction

Multiplayer Online Battle Arena (MOBA) games have been growing increasingly popular and captivate their player base in virtue of complex game mechanics and competitive nature. Riot's League of Legends claims to have over 100M monthly active players [16] and grosses over 1 billion US dollars of revenue yearly [17]. With 20M US dollars in 2016, the largest price pools in the history of eSports for a single tournament was crowdfunded almost entirely by the player base of Valve's Dota 2 [18].

MOBAs are played in independent n vs n matches, typically with $n = 5$, in which the players of each team need to closely cooperate to penetrate the other team's defences and obtain victory. Players who refuse to cooperate and act without considering their own team are easy targets and get killed more frequently, which diminishes the team's chances. Together with the intricate and sometimes counter-intuitive strategic nature of MOBAs, this gives rise to conflict within the teams. Triggered by game events like kills or just simple mistakes, players begin to turn sour. The communication channels that were meant to coordinate the team effort can then be used to verbally assault other players, often by using offensive terms and heavy insults.

Possible consequences are resigned players, whom might no longer be interested in competing for the win. But even if the match is won eventually, players could still feel offended, abused and might regret their decision to play the game in general. In this way, the *mood* of a communication could qualify as a social Quality of Experience (QoE) metric [19].

Collecting bad game experiences like this is harmful for the community, as it can bias a player's attitude towards engaging in cooperation even when confronted with fresh opponents and new teammates in later matches. The perceived hostility in a player community is frequently referred to as *toxicity*. Toxicity imposes a serious challenge for game designers, as it may chase active regular players away. Jeffrey Lin, the head of the internal research team at Riot Games, has reported that players who experience in-game toxicity are up to 320% more likely to quit playing League of Legends [20]. It might also prevent new players from joining the game, because a toxic base appears as unfriendly and hostile to newcomers, which seems enough of a threat to Valve to change the matchmaking system for Dota 2 newcomers to ensure a "good social experience" [21]. Despite the efforts of some of the biggest game developing companies in the worlds, toxicity is still prevalent in their game communities and demands more research. This chapter is a study on how toxicity can be detected, its impact on game success, possible causes and its contagious potential.

Our main contribution is the introduction of an annotation system for chats of multiplayer online games capable of detecting toxicity (Section 2.3). We apply the system to a large dataset (Section 2.2) collected from a representative game of the MOBA genre and propose a method based on machine learning that uses the annotation system to predict the outcome of ongoing matches (Section 2.4). We furthermore apply a topic model to the chat data and show how the latent topics in the players language align with our previous annotations (Section 2.5). Given the offensive nature of toxicity, we gather evidence for its likelihood to influence the behavior of players confronted with it and possibly even spread within player networks (Section 2.6). We end by relating our study to previous work (Section 2.7) and conclude with a summary of the results

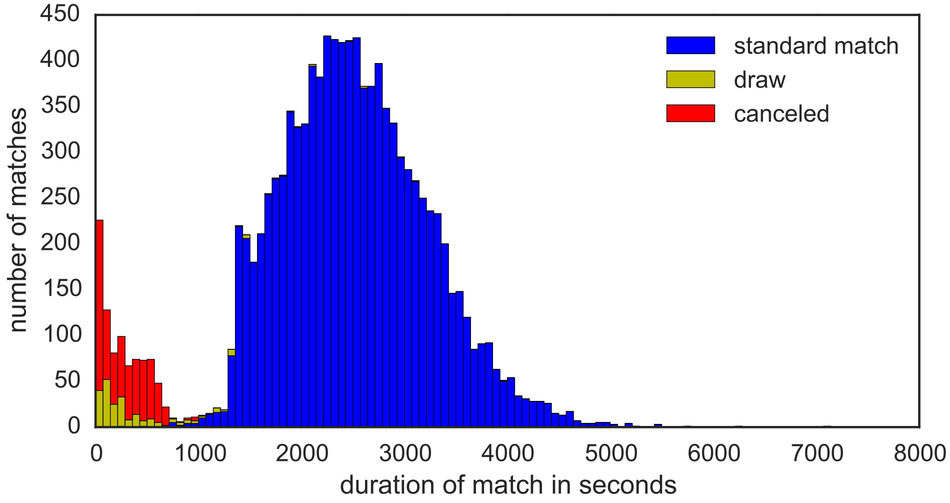


Figure 2.1: Distribution of match duration in the DotAlicious dataset.

and a perspective on how our system might help to improve the gaming experience (Section 2.8).

2.2. Data

2.2.1. Data Sources

All data used in this work are based on one of the ancestors of all MOBA games: Defense of the Ancients (DotA). This game started as a custom map for the real-time strategy game Warcraft III, but soon became so popular that community platforms emerged that allowed for players to register, get profiled and being matched up against each other based on their skill. One of these platforms was DotAlicious, from which we crawled our data.

The website of DotAlicious is no longer available online, as DotA has been substituted by newer MOBAs like League of Legends, Heroes of the Storm or Dota II. The core game principles have not been changed much by DotA's successors, but the accessibility of replays, chat-logs and player-related information for them is more limited due to several privacy concerns of the developing companies. Also, alternative means of information exchange, like protected voice-chats, make it more difficult to obtain a record of comprehensive inter-team communication. Hence, we believe that our data from DotA are suitable for our purpose, while still being representative for the game genre in general. Additionally, it allows us to study toxicity without harming a *live* community.

2.2.2. Data Cleansing and Match Outcome

Our DotAlicious dataset consists of replays from 12923 matches, spanning the time between the 2nd and the 6th of February 2012.

The duration of matches in the dataset is distributed bi-modally, indicating that a small fraction of the matches ended prematurely. We used information from the hosting-bot of DotAlicious to determine matches that resulted in a draw or were canceled by the players early on. In total, out of 12923 matches, 1653 were aborted before game start, 706 were canceled after game start and 241 resulted in a draw by mutual player agreement (see Figure 2.1).

For the remaining matches, there are two possible outcomes: either one team destroys the other team's main structure (victory condition) or all players of one team forfeit, which results in a collective surrender (loss condition). We have identified 10305 matches with a well-defined winning team, of which 6082 matches ended by the victory condition and 4223 matches by surrender. 18 matches needed to be excluded as their outcomes were unclear.

2.3. Game Communication Modelling

2.3.1. Annotation System Design

For all matches, we extracted all chat-lines used by the players and applied a tokenization based on simple white-space splitting. Symbols like “!” or “?” remained part of the words, as long as they were not separated by white spaces. The case of the letters was unchanged to analyze the use of capitalization as a stylistic figure (shouting).

Overall, the language used is extremely abbreviated, elliptical, full of spelling-errors and barely following grammatical structures. Consequently, standard techniques from Natural Language Processing (NLP) like part-of-speech recognition, spelling-correction and language detection were either not applicable or performed poorly. On the other hand, we observed little variety in the topic of the chat, resulting in a rather restricted and repetitive set of vocabulary. We thus devised a novel annotation system to classify the most frequent words together with their miss-spelled variants.

The most dominant language in the corpus is English, which is used as a *pidgin language* for non-native speakers to communicate with each other. To classify the most frequently used words in this work, we do not consider words from any other language. Consequently, non-English words will be either “unannotated” or classified as “non-latin” (for example in the case of Chinese, which is easy to detect).

To classify the semantics of a word, we apply sets of simple rules to them. There are three different classes of rules that we use:

1. **pattern**: the word includes or starts with certain symbols,
2. **list**: the word is member of a pre-defined list (also known as “dictionary”), and
3. **letterset**: the set of letters of the word equals the set of letters of a word from a pre-defined list.

The letterset class is useful to capture unintentionally or intentionally misspelled words, if no meaningful recombination of their letters (like anagrams) exist in the corpus. For example, the set of letters used to spell the word “noob”[1] is {“n”,“o”,“b”}, which is

[1] “noob” is a common insult in video games. It is derived from the word “newbie”, which comes from “newcomer”. Thus, it implies that someone has the lowest possible level of skill and knowledge of the game.

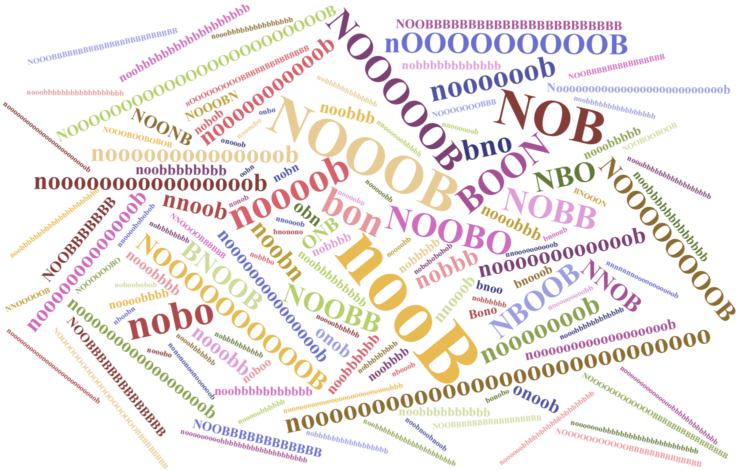


Figure 2.2: Wordcloud of the most frequent case-sensitive variants of the word “noob” from the datasets chat-corpus (“noob” and “NOOB” excluded). Size is scaled logarithmically by word frequency. Wordcloudgenerator by Jason Davies (https://www.jasondavies.com/wordcloud).

the same set as used for words like “NOOOOOOOOb”, “boon”, “nooobbbbb” or “noonb” which were actually used in the chats (see Figure 2.2). In total, the letterset method allowed to capture 224 (case-sensitive) different ways of writing “noob” that were used in the dataset. On the other hand, no other meaningful English word that could be built using this set (for example “bonobo” or “bonbon”) was found in the corpus. Also for other words than “noob”, the amount of introduced *false positives* due to the letterset-method was negligible for our dataset.

Table 2.1 shows the rule classes for each annotation category together with a short description, some examples, their precedence and their absolute prevalence in the text corpus. The sources for the word-lists and patterns for each different category are listed in Appendix A.

The text-corpus consists of 7042112 words in total, of which 286654 are distinct. Each distinct word is checked against our rules and annotated accordingly. If no rules apply, the word is “unannotated”. If multiple rules apply, we break the tie by choosing the category with the highest precedence. Considering the set of all distinct words in the corpus, our annotation system covers around 16% of them. However, many of the most-frequent words are annotated, so that on average over 60% of all (non-distinct) words used per match have an annotation.

2.3.2. Different Chat-modes

Our data allows us to investigate two fundamentally different chat-modes for each match: in the all-chat, a player can broadcast a message to each other player that participates in the match. In the ally-chat, the message is only sent to players in the same team as the sender. We observe that on average 90% of all messages are exchanged in the ally-chat and only 10% are broadcasted to all players. Private player-to-player communication

Table 2.1: Annotation categories

category	description	rules	examples	precedence	unique count
nonlatin	special character, foreign language	pattern	文章	500	20133
praise	acts of courtesy, kindness, sport spirit or gratitude	list	gj, gg, thx, hf	100	295
bad	profanity, swear words, inappropriate language	list, letterset	noob, idiot, fuck	90	4881
laughter	acronyms expressing laughter	letterset	HAHAHAHA, lol, ROFL	60	2158
smiley	emoticons, symbols resembling faces or emotions	pattern, list	:D, ;), oO, -_-	50	1110
symbol	symbols or numbers	pattern	?, 1, ..., ??!?, /	40	3181
slang	DotA-specific terms, team coordination	list	ursa, mid, back, farm, bkb	30	10046
command	control words to trigger actions	pattern	!ff, !pause, -swap	20	2513
stop	English stop words	list	was, i, it, can, you	10	1322
timemark	time-stamps, inserted in pause-mode	pattern	[00:05], [01:23]	5	223

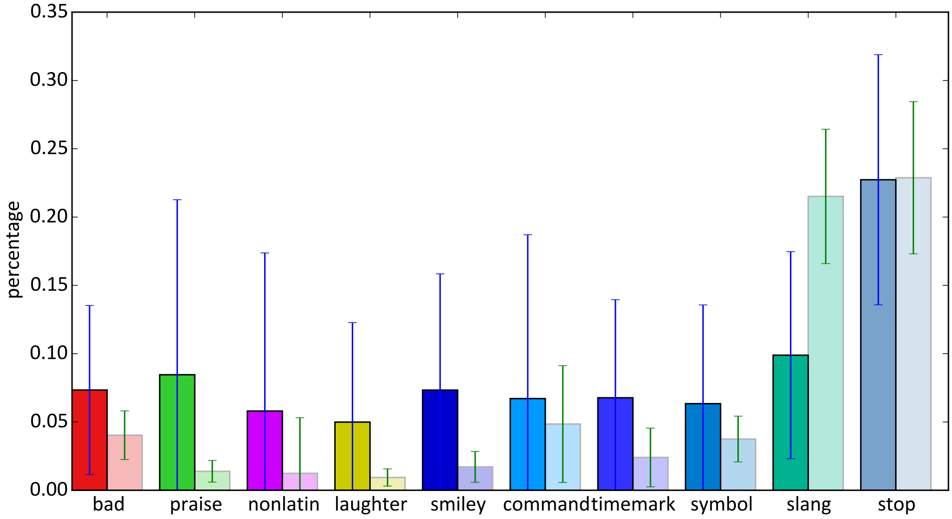


Figure 2.3: Average use of annotated words per chat-mode. Chat-mode depicted as solid bars (all-chat) and as transparent bars (ally-chat). Error-markers show one standard deviation. Category "unannotated" was omitted.

is also possible, but was not recorded within our data. Figure 2.3 shows the relative amount of annotated words averaged over all matches for both chat modes. Note that the words from the "stop"-category are used almost equally in both chat modes, meaning that our selection of stop-words is context-independent. The usage of words from the "slang"-category is twice as high in the ally-chat, since slang is mainly used to transfer sensitive information to coordinate the team in its battle. The heavy relative use of slang in the ally-chat creates a bias in almost all other annotation categories towards the all-chat.

2.3.3. TOXICITY DETECTION

For the purpose of our investigation, we define toxicity as the use of profane language by one player to insult or humiliate a different player *in his own team*. As such, the use of "bad" words is a necessary, but not sufficient condition for toxicity. For example, bad words can also be used just to curse without the intent to actually insult someone else. Profanity is also used in ironic or humoristic ways. For example, some players use self-deprecating remarks to admit in-game mistakes: "sry, I am such a noob - lol". Thus, detection of toxicity can not be based on words alone but needs to take the current context into account.

We are using n-grams to distinguish toxicity from ordinary profane language. An n-gram is a contiguous sequence of *n* words that appears in a context. The context in our case consists of all words in the chat-line that contained the "bad" word plus all words from all chat-lines that were sent by the same player to the ally-chat not more than 1 second before or after.

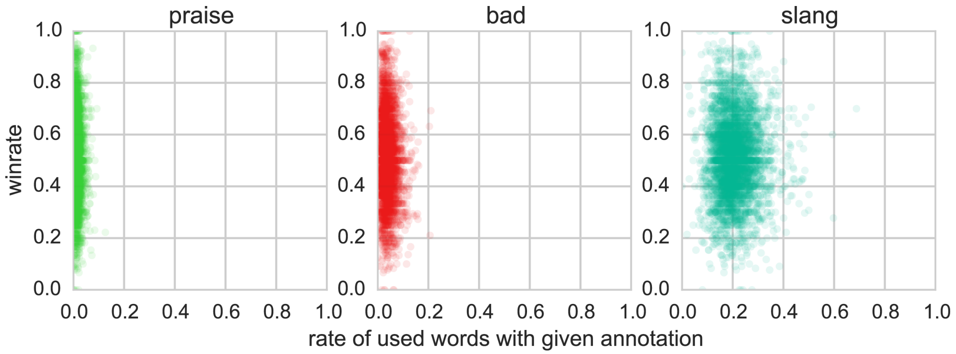


Figure 2.4: Correlogram between annotation categories and winrate.

For all players who participated in at least 10 matches, we search for all “bad” words they use, construct their contexts and count each n-gram that contains at least one “bad” word for $n = 1,2,3,4$. Afterwards, we look at the 100 most frequently used n-grams for $n = 1,2,3,4$ and manually determine which of them are toxic and which are not. Our criterion for toxicity is the following: for unigrams ($n = 1$) we consider them toxic if they could be understood as an insult. For example “crap” is no insult, but “moron” is. For n-grams with $n = 2,3,4$, we consider every context toxic that includes an insult directed towards a person. Examples include “fucking idiot”, “shut the fuck” and “i hope u die”. On the contrary, profane language that we do not classify as toxic includes n-grams like “fuck this”,“cant do shit” and “dont give a fuck”. In total, we deem 45 unigrams, 21 bigrams, 32 trigams and 36 quadgrams as toxic. The list of these n-grams is provided in Appendix B.

2.4. Analysis of Game Toxicity and Success

2.4.1. Triggers of Toxicity

Table 2.2: Kill-events before toxicity

	kill-events	**from killer**	**from victim**
toxicity ($\Delta = 5s$)	2219	23	849
random ($\Delta = 5s$)	1488	74	478
toxicity ($\Delta = 10s$)	5285	124	2559
random ($\Delta = 10s$)	3176	200	1042

Our method detects at least one toxic remark in 6528 out of the 10305 matches. In 90% of all toxic matches, there are at most 5 toxic remarks detected. Several outliers exist in the data, the strongest contains 22 toxic remarks in a single match. The total number of toxic remarks was 16950. We expect that certain game events trigger players to act toxic. One possible game event is a *kill* where one player (killer) temporarily eliminates the character of another player in the opposing team (victim). There is a reaction time Δ involved between the actual kill-event and the time a player needs to submit a response to the chat. We look for each toxic remark if there was a kill-event taking place not earlier than Δ seconds before. For comparison, we also choose 16950 random chat-lines (distributed over all matches) and look for a kill-event in their recent past as well. It turns out, that toxic remarks are more frequently preceded by kill-events than random remarks. Table 2.2 reports the absolute number of kill-events and how many of them were submitted by the killer or the victim. Especially victims of kill-events tend to become toxic, potentially blaming their teammates for their own fate.

2.4.2. Game Success and Profanity

We have the hypothesis that with diminishing chances to succeed in the game, the level of profanity raises. To test our hypothesis, we compute the winrate for each player as the amount of matches won divided by the amount of matches played in total. We restrict the analysis to players who participated in at least 10 matches, which leaves 4009 distinct players in our dataset. Next, we count how many words the players used for our annotation categories “bad”, “praise” and “slang”. Normalized by the total number of words, we correlate this number with the winrate, and plot the results in Figure 2.4. Surprisingly, there seems to be no strong linear correlation in either case, which is confirmed by the correlation-matrix given by Table 2.3.

Table 2.3: Pearson Correlation between winrate and use of words

	winrate	**bad**	**praise**	**slang**
winrate	1.0	0.0739	-0.0161	0.0059
bad	-0.0739	1.0	0.0454	-0.1540
praise	-0.0161	0.0454	1.0	0.1152
slang	0.0059	-0.1540	0.1152	1.0

An analysis based on absolute word-counts with focus on whole teams (rather than single players) reveals a different picture: for each “bad” word used by a winning team, we determine the point of time in the match when it was submitted to the chat. As different matches vary in duration (recall Figure 2.1) we normalize time to the interval $[0, 1]$ on the horizontal axis, with 1 indicating the end of a match. Out of this data we construct a histogram using 100 equally distributed bins. We overlap this histogram with a second histogram, constructed the same way but for words used by the losing teams. As winning and losing teams use a slightly different absolute number of words per bin, we normalize each bin accordingly to eliminate bias. Figure 2.5 reports on the vertical axis the fraction of words used in each bin over all words used by the respective winning or losing team while Figure 2.6 shows a histogram based on toxicity detected by our toxic n-grams.

As we can see from the top part of Figure 2.5, after a short initial period, in which it is uncertain to the players whether they might be losing or winning, we observe that teams that will lose the match in the end tend to use relatively more bad words than

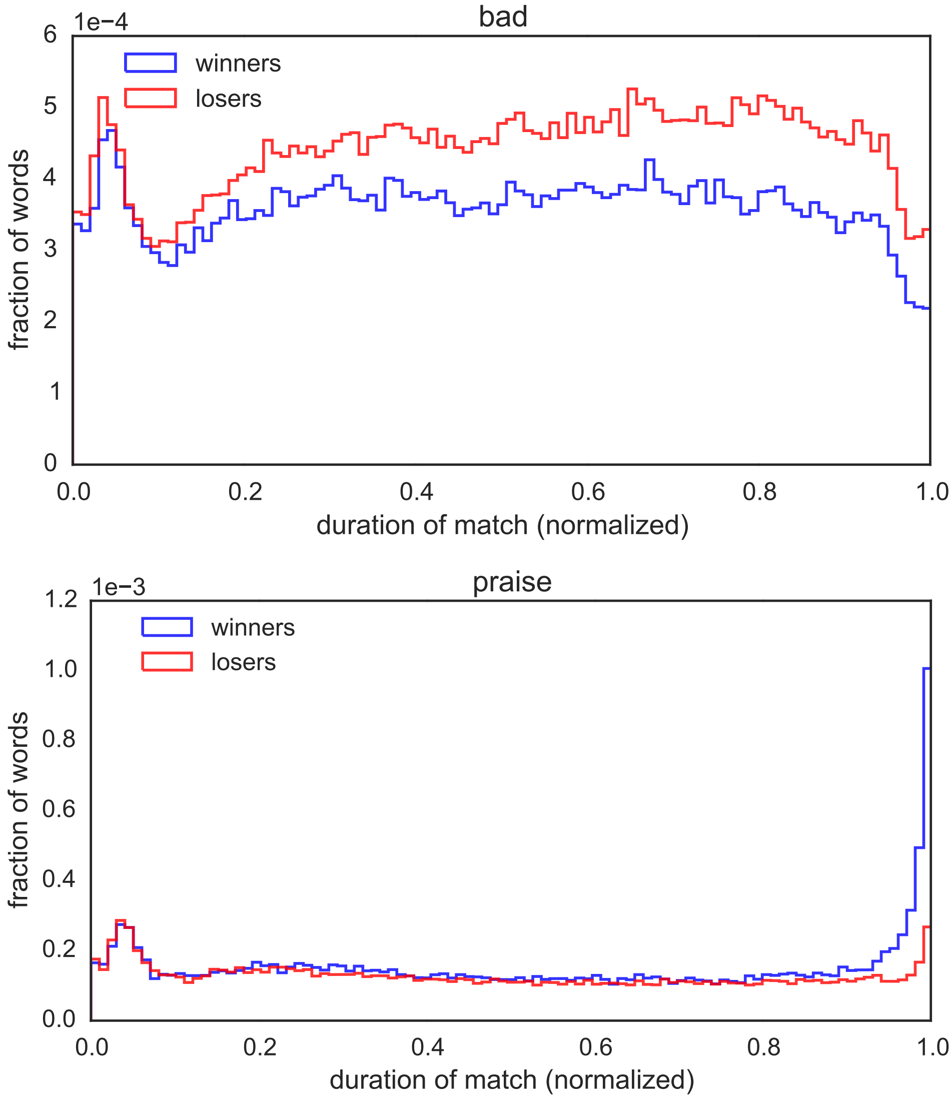


Figure 2.5: Overlapping histograms, comparing winning and losing teams in their usage of words from categories “bad” in comparison to “praise”.

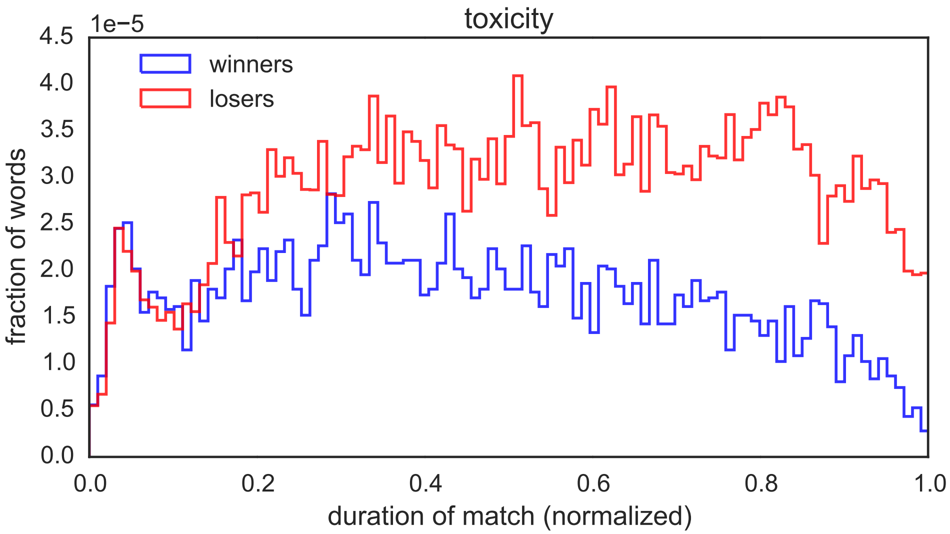


Figure 2.6: Overlapping histogram, comparing winning and losing teams in their usage of toxic n-grams.

teams that will win the match. This difference is even bigger if toxicity is considered. More interestingly: while the usage of bad words is somewhat consistent throughout the match, the usage of toxicity varies more. It seems that the winning teams use less toxicity at the late stages of the match, as it becomes apparent that they will be victorious. The need to shame and blame teammates seems to be significantly higher for the losing team than the winning team at this point in time. Another interesting aspect is the usage of the category "praise" which seems consistent for most of the matches but peaks clearly for the winning team by the very end. This effect is due to the traditional phrase "gg" (good game) which is a word from the "praise" category and often used just before the match finishes. Winning teams use this phrase significantly more, probably as they might perceive the match as more enjoyable.

2.4.3. Predicting Match Outcome

As we have shown, toxicity appears only in 60% of all matches and is thus too infrequent to be used for predicting match outcome in general. Therefore, we analyze the predictive power of all words with respect to their annotations, including the category of "bad" words. We train a linear support vector machine (SVM), which is a supervised learning model, to predict the winning team. Our features are constructed as follows: first, we count all words that were used in the ally-chat. These counts are then used to determine the TF-IDF (term frequency inverse document frequency) for each word. TF-IDF is a standard weighting technique frequently used in information retrieval [22] that relates the occurrence of words in one document (term frequency) with their occurrence over all documents of the corpus (inverse document frequency). Formally, Given a document d out of collection of documents D, the term-frequency $\mathrm{TF}(t,d)$ is the absolute count of occurrences of a term t in d divided by the total number of terms in d. For a set of documents D, the inverse document frequency $\mathrm{IDF}(t,D)$ is defined as:

$$\mathrm{IDF}(t,D) = \log\left(\frac{|D|}{|\{d \in D : t \in d\}|}\right). \tag{2.1}$$

The TF-IDF is the product of both:

$$\mathrm{TFIDF}(t,d,D) = \mathrm{TF}(t,d) \cdot \mathrm{IDF}(t,D). \tag{2.2}$$

In our case, the set of documents D is the set of all chats over all considered matches. As there are two teams per match competing with each other, we have two ally-chats that give us two different sets of features, which we both use together as one document. The SVM is trained using a stochastic gradient descent algorithm on the TF-IDF to predict the winning team. For all computations, we use Scikit-learn [23] with its default parameters for all algorithms and do not undertake any effort to optimize them. The idea is *not* to create the most accurate classifier possible but rather to use the accuracy of the classifier to measure the importance of words with respect to match outcome.

The outcome should become more certain with the progression of the match, which should be reflected in the words used by the players. We introduce the parameter t to control the amount of chat history that is given to the classifier. For example, for $t = 1.0$ the classifier is trained (and evaluated) on the complete ally-chats of each match, whereas for $t = 0.5$ it only knows what was written until the middle of the matches. The classifier itself has no notion of time: the TF-IDF features are purely based on frequencies (bag of words model) and reflect neither the order of words nor the specific time they were submitted to the chat.

As each word corresponds to one feature, we can partition all features with our annotation system. We use the classifier 1) for all words regardless of their annotations, 2) for all words but words from the "command" category, 3) for no words except from the "bad" category and 4) for no words except from the "slang" category. The reason for excluding words from "command" is to avoid to provide the classifier with information if a player forfeited, which is announced by typing the command word "!ff" in the chat.

Figure 2.7 shows the average accuracy and the 95% confidence interval of the classifier for these scenarios under a 10-fold cross-validation. The number of used features and the accuracy scores for $t = 0.5$, $t = 0.75$ and $t = 1.0$ are presented in Table 2.4.

While words from the "bad" category (which constitute a precondition for toxicity) have some predictive power, it is significantly lower than using just *all words* or words from "slang" alone. We find it also remarkable that "slang" uses the least amount of features but gives still fairly good predictions. This might be due to the importance of team coordination which is covered mostly by key words from this category. It seems reasonable that their usage shows not only the game expertise of players, but also engagement and an increased interest to improve the team-play, which could result in a better chance to win the match. The occurrence of "bad" words however seems to be much less indicative for either winning or losing, suggesting only a weak link to game success. Consequently, also toxicity might not be the best indicator to determine if a game is going well for a team or not. Profanity will appear either way.

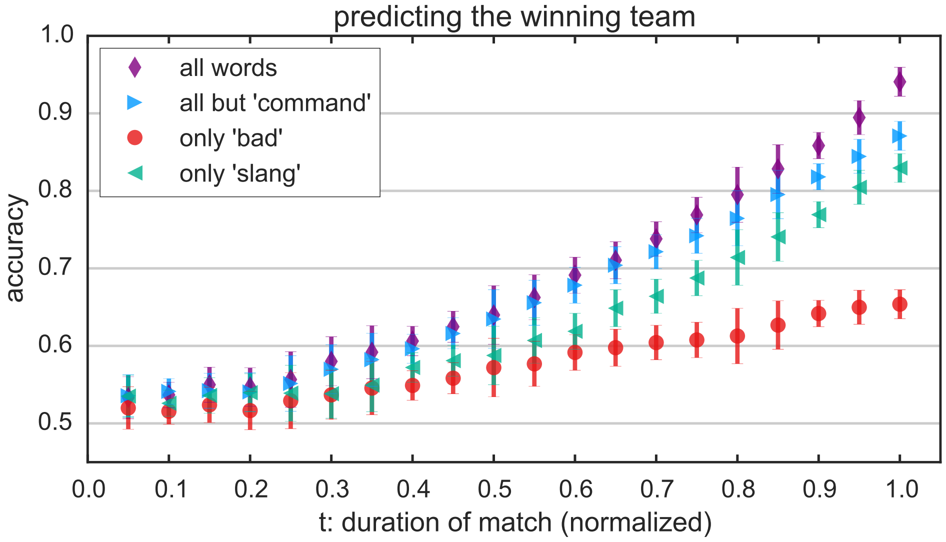


Figure 2.7: Accuracy of the linear SVM on ally-chats.

Table 2.4: Accuracy and number of features used by the classifier.

		#features	avg accuracy	std accuracy
t = 0.5	all words	127612	0.6399	0.0140
	all but “command”	126900	0.6346	0.0103
	only “bad”	1442	0.5720	0.0137
	only “slang”	880	0.5877	0.0189
t = 0.75	all words	170063	0.7689	0.0092
	all but “command”	169298	0.7421	0.0099
	only “bad”	1767	0.6077	0.0096
	only “slang”	908	0.6875	0.0114
t = 1.0	all words	208598	0.9407	0.0048
	all but “command”	207758	0.8708	0.0070
	only “bad”	2020	0.6538	0.0108
	only “slang”	921	0.8295	0.0093

2.5. TOPIC MINING OF PLAYERCHATS

A frequent approach for data exploration in Natural Language Processing (NLP) is the application of *topic models* [24]. Given a set of documents, composed out of words, the task of a topic model is to infer latent “topics” which are represented in the documents. Thus, each document is typically understood as a mixture of topics, while certain combinations of words classify each topic. Once a topic model is fitted, it allows to classify documents with respect to their topics and discover new relations. We hypothesize that the categories of our annotation-system from Section 2.3 should find some reflection within the topics that arise from the conversations between players. In particular, we expect typical game-chat to be a mixture of some of our pre-defined categories (slang, bad, praise, etc.). We are also interested to analyze how strongly the topics differ between players for which toxic statements were observed in contrast to players for which no toxicity was detected by our method.

We limit the analysis to the subset of players that played at least 10 matches and sent a minimum of 100 words over all their matches to the ally-chat. We collect for each single player all words submitted in a “bag–of–words” model. Thus, each document in our analysis corresponds to exactly one player, but is composed from words that were used throughout multiple matches. The complete corpus consists out of 5732154 (not different) words distributed over 3911 documents. We further eliminate words belonging to the timemark, stop, symbol and nonlatin category, which reduces the total number of words by roughly 30%. In a next step, we eliminate words that occur in less than 1% and more than 90% of all documents. Note, that our words are case-sensitive, so we make a distinction between “Noob”, “noob” and “NOOB”. As a result, we obtain a 3911×4456 document word matrix, whose cells represents the term-frequencies $\mathrm{TF}(t,d)$ for word t in document d.

Next, we apply Latent Dirichlet Allocation (LDA) [25] to obtain $T = 20$ topics, with a per-document topic distribution prior $\alpha = 0.01$ and a per-topic word distribution prior $\beta = 0.001$. The LDA is an unsupervised learning algorithm, which groups words based on their co-occurrence into topics, and assigns a sparse distribution over all topics for each document. Thus, for each topic, a word occurs with a certain probability while each topic has a certain probability to be part of a document. Table 2.5 shows the 10 most probable words for each topic.

While there is no straightforward interpretation as a semantic topic, we observe some noticeable patterns:

- Topic 3 has a strong focus on game coordination with important terms from the “slang” category.
- Topic 5 has a strong focus on “command” words.
- Topics 12 reflects the usage of smileys and acronyms like “lol”.
- Topics 1, 6 and 7 seem to consist out of mostly unannotated words, which might be stop words from foreign languages (e.g. topic 6 consists of German stop words).
- Topics 9 contains many words written with an apostrophe, which is unusual in chat communication.
- Topics 10 and 18 have high probabilities to produce upper-cased words.
- Topic 20 has the highest probability to generate a word from the bad category (≈24%) while Topic 1 has the lowest (≈3%).

topic	10 most frequent words	pr. bad
1	de la le tu il si c lol ma pas	0.027
2	!ping -water miss re -ii -swap -weather na care man	0.031
3	top bot care smoke push farm solo wards rosh tp	0.032
4	plz top bot care ye well ulti then gj farm	0.034
5	-clear -hhn -ii -don -water -CLEAR -weather moonlight random pls	0.037
6	ich du :D ^^ die das der und top ja	0.039
7	:D da ne ti e se si na mi je	0.045
8	top bot lol gj care :) pls k re gank	0.046
9	I don't it's i'm can't top bot gj i'll didn't	0.048
10	xD SS MISS MID lol TOP care top gj such	0.049
11	!silence -swap top c -random off -ms bot re !resume	0.049
12	:D -.- lol xD XD :) xd gj ^^ gg	0.052
13	pls top bot gang care ffs wp farm gj ty	0.054
14	^^ well your top farm cant game then fucking rofl	0.068
15	!pa -swap care omg lol plz w8 push gj pls	0.069
16	ur game im cant play fucking win didnt fuck shit	0.083
17	lol top fucking wtf kill fuck bot im push gank	0.083
18	U GO B I FUCKING YOU ME gg MID FUCK	0.100
19	game noob fucking farm fuck gang idiot ulti play cant	0.112
20	noob omg ff gg fucking lol noobs team retard idiot	0.241

Table 2.5: The 10 most probable words per topic. Words are ordered by decreasing probability within the topic from left to right. The topics are ordered by the overall probability of generating a word from the bad category from top to bottom. Colors indicate the different annotation categories (compare Figure 2.3).

We furthermore observe that some words like "top" (which refers to a part of the game map) are so common for DotA chats, that they appear across topics with high probabilities, and can be thought of as stop words of the DotA-language. We also note that words of acknowledgement like "gj" (good job) and "ty" (thank you) appear more likely in topics that put less probability on the bad category in general, which makes sense as they carry quite the opposite sentiment.

How are the topics related to toxicity? As the use of profanity (bad category) is a pre-requisite for toxicity, it is expected that the language of toxic players consists out of topics with a high frequency of bad words. We confirm this bias by sorting the 3911 players into two groups: players for whom there was no occurrence of toxicity detected within our data (1351 players) and players with at least one occurrence of toxicity over all their matches (2560). For both groups, we average and normalize the probabilities to be contained within a certain topic. Figure 2.8 confirms that there is a trend for toxic players to prefer topics with a high density of bad words.

2.6. The Influence of Toxic Teammates

Toxicity is a provocative behavior and tends to trigger reactions from people. We are interested to see whether a player becomes toxic when confronted with toxic teammates, i.e. whether toxicity is contagious. For a definite answer to this question, one would need to thoroughly examine what causes toxicity in the first place. A first attempt to find causes for toxicity was done in Section 2.4.1, which showed a correlation between kill-events and toxicity. However, other in-game behavior, i.e. a perceived lack of performance,

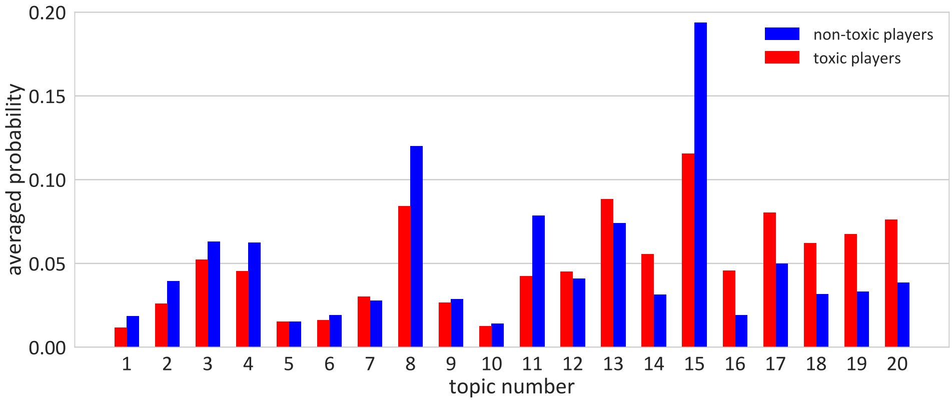


Figure 2.8: Averaged topic probabilities for different groups of players. The topics are ordered by their probability of generating a word of the bad category. As bad words are a prerequisite for toxicity, the vocabulary of toxic players in comparison to non-toxic players is more likely to be based on topics with a high topic number.

idling or missing opportunities might be hard to observe, quantize and relate to toxicity. Instead of studying all potential factors and relations in detail, this section will apply an information theoretic approach. In particular, we model toxicity of players as a discrete stochastic process and use the information theoretic notion of transfer entropy to quantify the amount of influence that toxic teammates exhibit on each other.

2.6.1. Transfer Entropy

Transfer Entropy was first derived to quantify the directional information exchange between two processes evolving over time [26]. It presents an alternative to the use of mutual information that is able to detect asymmetric influences between complex systems. Transfer entropy has become popular in brain research to model information flow and effective connectivity for magnetoencephalographic data [27, 28] but has also found application in social network analysis [29]. This measure will reappear as phase transfer entropy in Section 3.2.1 of this thesis. For notation purposes, we will briefly introduce related information theoretic concepts and define the basic transfer entropy here.

Let X be a discrete random variable over sample space $\{x_1, \ldots, x_n\}$. The (Shannon) entropy $H(X)$ is defined as

$$H(X) = -\sum_{i=1}^{n} \Pr[X = x_i] \cdot \log_2(\Pr[X = x_i]). \tag{2.3}$$

The entropy can be interpreted as the amount of uncertainty contained within a random distribution. If $H(X) = 0$, then the outcome of X is perfectly predictable. The base of the logarithm is a scaling factor for the entropy, which will be neglected in the following.

Let Y be a second discrete random variable over a sample space $\{y_1, \ldots, y_m\}$. We can condition the entropy of X by our knowledge of Y and define the conditional entropy

$H(X \mid Y)$ as

$$H(X \mid Y) = -\sum_{i=1}^{n}\sum_{j=1}^{m} \Pr[X = x_i, Y = y_j] \log\left(\frac{\Pr[X = x_i, Y = y_j]}{Pr[Y = y_j]}\right). \tag{2.4}$$

The conditional entropy has the property that $H(X \mid Y) = 0$ if and only if X is completely determined by Y. Additionally, it can be shown that $H(X \mid Y) = H(X)$ if and only if X and Y are independent.

Next, we want to look at a discrete stochastic process, which is a sequence of discrete random variables $X(t)$, where $t \in \{t_1 < t_2 < \ldots\}$ can be interpreted as observations in time. If we assume a history of h hops, the conditional entropy

$$H(X_t \mid X_{t-1}, \ldots, X_{t-h}) \tag{2.5}$$

describes the amount of uncertainty in predicting X_t given observations in the past of $X_{t-1}, \ldots, X_{t-h}$. For two stochastic processes Y_t and X_t and two time windows h_1 and h_2, the transfer entropy from Y to X is defined as

$$\mathrm{TE}_{Y \to X} = H(X_t \mid X_{t-1}, \ldots, X_{t-h_1}) - H(X_t \mid X_{t-1}, \ldots, X_{t-h_1}, Y_{t-1}, \ldots, Y_{t-h_2}). \tag{2.6}$$

Thus, $\mathrm{TE}_{Y \to X}$ measures the reduction of uncertainty for predicting X_t from a history $X_{t-1}, \ldots, X_{t-h_1}$ by additionally taking the history of $Y_{t-1}, \ldots, Y_{t-h_2}$ into account. If $\mathrm{TE}_{Y \to X} > 0$ some information from process Y_t must exhibit an influence on process X_t. Note that the transfer entropy is asymmetric and is thus able to characterize directed information transfer.

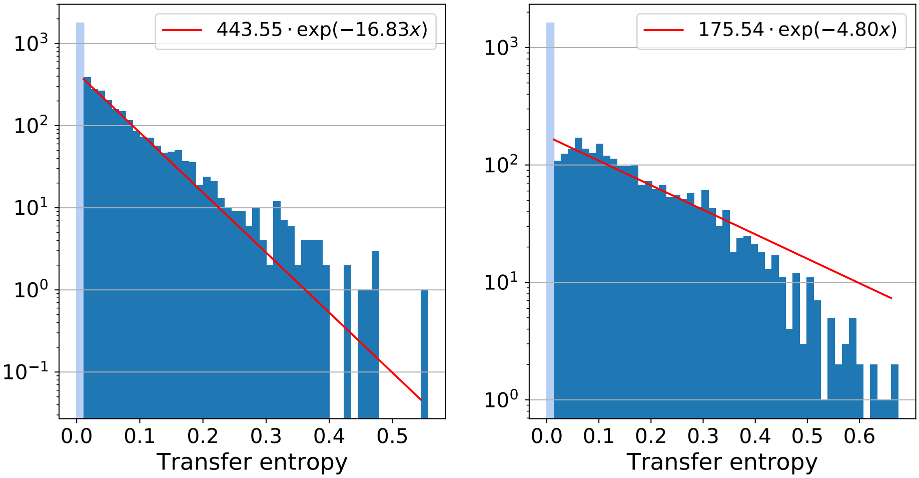
2.6.2. Toxicity as Transfer Entropy

We define the history of a player p as the sequence of matches p has played. For a match $m(t)$ we define $X_t = 1$ if we detected a toxic statement in the chat of this player and $X_t = 0$ otherwise. Thus, toxicity is modeled as a stochastic process by X_t. By moving through the complete history of player p and binning the observations, we can estimate the probabilities to compute the first term of the transfer entropy.

Since each team consists of 5 players, player p will have 4 (potentially different) teammates in each of his matches. We model the toxicity of these teams as a sequence of random variables $Y_t \in \{0, 1, 2, 3, 4\}$, which count how many of the teammates say something toxic in the corresponding matches from the history of player p. This allows us to compute the second term of the transfer entropy.

By moving through the history of the player, we can estimate the joint and conditional probabilities needed to compute the transfer entropy. For simplicity, we use the same time window $h = h_1 = h_2$ for both the player and the team mates' past matches. We analyze only the set of players that played at least 10 matches. Figures 2.9 shows a histogram of the computed transfer entropies for time windows of size $h = 1$ and $h = 2$.

We observe a large number of players with a transfer entropy of 0, which is mainly due to the effect that a large number of players (1443 out of 4013) have no toxic statement in any of their matches detected. Consequently, the transfer entropy of those players must be 0. Although, a transfer entropy of 0 does not imply that a player never said something toxic within his history, this case rarely happens within our data (87 players).



(a) window size: 1 match

(b) window size: 2 matches

Figure 2.9: Histogram of transfer entropies in 50 equal sized bins and log-scaled number of players on vertical axis. Red lines are fitted exponential functions for non-zero transfer entropies. The pale blue bin consists of players with zero transfer entropy.

We find that the distribution of the non-zero transfer entropy can be described by an exponential function. The corresponding fitted parameters are given in the legend of Figure 2.9. The exponential distribution suggests that the susceptibility towards a toxicity transfer from toxic teammates is strongly heterogeneous. While a large majority of players have a low transfer entropy and are thus less likely to be influenced by their teammates, there is also a minority who become significantly predictable in their toxic statements, once information about their teammates is considered.

The quality of fit for the exponential distribution degrades if longer match windows of $h > 2$ are considered (not shown), while the computational effort for computing the transfer entropy sharply increases. Additionally, the number of bins needed to describe the probabilities increases exponentially with h, which demand much longer match histories to avoid a sparse population of the bins. Lastly, we find the transfer entropy only mildly correlated with the toxicity rate (i.e. fraction of matches in the history of a player in which he said something toxic) with a correlation coefficient of ≈ 0.562. This moderately high correlation is expected because of the previously described relation between a zero toxic rate and a zero transfer entropy. It is reasonable to assume that for a large number of those non-toxic players either not enough matches were recorded to contain toxic statements or bad statements where too ambiguous for our conservative detection to be flagged. Thus, given more data and a sharper detection algorithm would significantly lower the size of this group of players and consequently reduce correlation. In fact, we suspect that transfer entropy and toxic rate should be almost uncorrelated.

2.7. Related Work

Antisocial behavior in virtual environments has been investigated in the field of computer sociology, most commonly under the term cyberbullying [30]. The impact of profane language on video games [31] and in a wider sense also on social media [32] is a vital area of research. Suler [33] shows psychological factors explaining the *online disinhibition effect*, giving *toxic disinhibition* as a negative example. This effect is a possible explanation why we observe such high levels of bad behavior online in general.

While toxicity in MOBAs can be understood as an act of cyberbullying, it is notoriously hard to define and blends with other abstract behavorial concepts. Most similar is *griefing*, the act of disrupting other player's game enjoyment by unacceptable behavior. While both toxicity and griefing might encompass verbal harassment ("flaming"), griefing is also associated with scamming, power imposition and theft of ingame items. Griefing was scientifically investigated for virtual worlds like Second Life [34] and MMORPGs like World of Warcraft [35]. Behavorial patterns reflecting griefing and verbal abuse have also been reported in the MMO World of Tanks by a work from Esmaeili et al. [36].

Most of the time, toxicity is studied with respect to player retention. Birk et al. [37] give a study with focus on player experience, social exclusion potentially resulting hostile cognitions. Tyack et al. [38] study motivations and reasons to start and to stop playing MOBA games via a survey and interviews, reporting (amongst other) that 12.18% of the particpants quit the game due to "unsportsmanlike players". Shores et al. [39] define a toxicity index based on peer evaluations of players facilitated by some third-party app for League of Legends, giving evidence that player retention is negatively affected by toxicity.

An excellent case study for toxicity in MOBAs is given by the works of Blackburn and Kwak [40, 41]. The authors use crowd-sourced decisions from the *tribunal*, a player-based court that was once provided by Riot Games as a mechanism to pass judgments on reported incidents in matches from League of Legends.[2] While our definition of toxicity is tied to profane language only, the authors additionally consider certain in-game actions (i.e. "intentional feeding") as toxic. They develop a classifier to assists or even substitute the crowd-sourced decisions of the tribunal, which are whether an accused player is guilty of toxic behavior or not. As only cases submitted to the tribunal are considered, the authors have access to a ground truth for toxicity which is not present for our data. However, this might also create a selection bias, as typical matches will not end up on the tribunal.

Shim et al. [42] describe a different system based on the Pagerank [43] algorithm to filter out "bad players" in MOBAs. Our approach is orthogonal, as it uses natural language processing on the player chats instead of relying on player's complaints submitted via a report function. Institutions like the tribunal would not work without players reporting others, while our approach does not need any explicit player feedback to detect and monitor toxicity.

There has only been little scientific work regarding the spread of toxicity between players, though some initial ideas are given by the works of Ki et al. [44] and Woo et al. [45].

[2] The tribunal together with its data has been disabled by Riot Games since early 2014.

2.8. Chapter Summary

We have developed a methodology to annotate frequently used expressions in written chat communication of Multiplayer Online Games. While our method is tested in this work only with data from DotA, we believe that it can be adapted to other MOBAs and possibly even games of different genres. To use the full system, one would need to update the pre-defined lists and patterns of game-specific terms to match their equivalents from the new game. Although this requires some degree of game-knowledge, the detection of profanity itself is largely independent from any game-specifics, as it is based on profanity used in the English language, enriched by a few terms commonly used in computer games. It would be insightful to compare the accuracy of our classification with actual player experiences, i.e. by interviewing players to confirm that they felt offended by certain statements.

Surprisingly, the win-rate of players seems to be largely unaffected by toxicity, which highlights that toxicity is a phenomenon that exists independently of the end result (winning or losing) of a match. A possible reason could be the complex nature of the game which allows for unexpected comebacks and sudden swings in favor of one team or another. The training of our classifier showed that bad words are also a bad predictor of success, while features from the slang category appear much more powerful, possibly reflecting better team-play and a more effective communication. It is remarkable that such a classifier can achieve a reasonable accuracy by just taking textual features into account. Extending the classifier with actual game features (e.g. kills, deaths, experience and gold earned, etc.) might provide an even better system which could be helpful for analysts and commentators of live streams, who are supposed to accurately describe the current odds for each team.

The probabilities obtained by the LDA topic model provide a high-level view on the language of the players: while the toxicity detection with n-grams only takes a contextual time-window of 10 seconds into account, the LDA is constructed from the complete vocabulary of a player, considering everything he ever said (in the ally-chat). Thus, we see a potential in topic models for monitoring individual players over longer periods of time, while the toxicity detection is helpful to judge short-termed situations within matches. Given long-term data, the success of certain counter-measures (like temporal bans, low-priority queues, etc.) against presumably toxic players could be evaluated by computing the shift of a players probability distribution between certain topics. Thus, we envision the toxicity detection and topic models like LDA to be essential building blocks for such systems.

Instead of only monitoring, the right decisions about possible counter-measures need to be taken if one wishes to cure a toxic community. Since competitive triggers for toxicity like kill-events are simply part of the game and cannot be avoided, the most critical aspect is to prevent escalation and the corruption of players who are in principle not hostile. One has to keep in mind that there may be plenty of unknown triggers and confounding factors for toxicity, so it is unreasonable to assume that every active toxic statements is the result of a previous statement received as an insult from a teammate. However, the distribution of the players transfer entropies shows that some players are more receptive to be influenced by offensive language than others. In particular, it would be most advisable to protect players with a high transfer entropy to be exposed to highly toxic teammates.

To summarize, our analysis shows that toxicity is fueled by the inherent competitiveness (i.e., killing each other) of MOBA games but is only weakly linked to success. If players can be successful despite being toxic, they need a different incentive to cease insulting and behave more pleasantly. On the other hand, the matchmaking systems that ensemble the teams could be altered to take toxicity into account to avoid creating a social powder keg. Even if we might not be able to prevent toxicity entirely, a better control will benefit newcomers, experienced players and game developers alike.

3

Information Flow Clustering in Functional Brain Networks

Recent work has revealed frequency-dependent global patterns of information flow by a network analysis of magnetoencephalography data of the human brain. However, it is unknown which properties on a small subgraph-scale of those functional brain networks are dominant at different frequencies bands. Motifs are the building blocks of networks on this level and have previously been identified as important features for healthy and abnormal brain function. In this chapter, we present a network construction that enables us to search and analyze motifs in different frequency bands. We give evidence that the bi-directional two-hop path is the most important motif for the information flow in functional brain networks. A clustering based on this motif exposes a spatially coherent yet frequency-dependent subdivision between the posterior, occipital and frontal brain regions.

This chapter is based on the published papers [46, 47].

3.1. Introduction

The application of network science to neuroscience has provided a new research perspective on the organization of brain networks from healthy subjects and patients suffering from neurological disorders [48, 49]. A recent study by Hillebrand et al.[50] observed frequencydependent global patterns of information flow based on magnetoencephalography (MEG) data of healthy subjects. However, little is known about the underlying mesoscale level in terms of network motifs at which these flows occur.

To analyze information flow, the pairwise measure of transfer entropy (TE) has often been applied [26]. For a pair of time series X and Y, TE quantifies the improvement in predicting the future of X when considering both the current value of X and the current value of Y, compared to only using the current value of X. At the level of brain regions, the TE value is classified as a measure of effective connectivity between two regions.

Recently, an extension of the TE that is based on phase information [51], the Phase Transfer Entropy (PTE), has been proposed in order to lower the computational costs and complexity [52, 53]. After calculating all pairwise PTE values, functional brain networks with nodes representing brain regions and link weights inheriting their pairwise effective connectivities, can be constructed so that the topology of these networks can be characterized.

Based on the pairwise PTE values, Hillebrand et al.[50] observed that for higher frequency bands, *alpha1*, *alpha2* and *beta*, the global information flow was predominantly from posterior to anterior brain regions, whereas the pattern was opposite for the low frequency *theta* band. The latter, an anterior-to-posterior pattern, was also discovered in electroencephalography (EEG) data [54]. It was hypothesized that the information flow in resting-state networks is likely driven by the strong posterior structural hubs and their high levels of neuronal activity [50, 55, 56]. However, the opposite directions of information flow are not yet fully understood.

Another biological explanation for the reverse patterns could be the Default Mode Network (DMN), which is the network of brain regions that are active during resting-state. The DMN consists of two interacting subsystems: the temporal system, which is responsible for memory, and the fronto-parietal system, which is essential for self-relevant mental simulations [57]. These two subsystems seem to exist in parallel, though at different frequencies, and their interaction represents an integration mechanism for brain functions [58]. This hypothesis is strengthened by results from invasive animal recordings of the visual cortex [59, 60], where the opposite directions of information flow have been connected with the process of memory consolidation [61].

In this chapter we investigate the information flow patterns with regard to a smaller scale for different frequency bands. On the mesoscale level of brain networks, network motifs have been identified as a valuable feature by many previous studies [62–64]. Motifs are frequently occurring subgraphs of networks, typically consisting of three or four nodes [13]. Previous studies were able to link structural and functional brain networks with regard to their motifs to describe flexibility in switching between different brain functions [65] and for coupling of brain dynamics [66]. Furthermore, changes in the motif frequencies of so-called progression networks for patients suffering from Alzheimer's disease have been discovered [67], showing that motif analysis may provide potentially powerful new biomarkers.

The importance of motifs has not only been studied for brain networks, but also for various others, like biological transcription networks [68], food webs [69] or transportation and mobility networks [70]. In order to link motifs to the modular organization often present in such networks, Benson et al.[71] proposed a new algorithm for motif-based clustering. Since this algorithm identified clusters of functional importance in the neuronal network of the *C. Elegans*, it appears to be a promising approach to analyze the higher-order organization of human brain networks.

The remainder of this chapter is structured as follows: Section 3.2 gives the necessary background information on the brain data, the computation of the PTE, the motif search and the clustering algorithm. Section 3.3 outlines the network construction from a related variant of the PTE, the directed phase transfer entropy (dPTE), the results of the motif search and the clustering when applied to the most important motif, providing a first proof of concept. Section 3.4 conducts a complementary study based directly on the PTE, which demands a slightly different network construction. Furthermore, the analysis is extended by comparing observations from two different frequency bands. Section 3.5 concludes the chapter.

3.2. Background

3.2.1. Measuring the Information Flow in the Brain

MEG measures the magnetic field fluctuations induced by neuronal activity [72]. The data for our analysis is based on MEG recordings in 67 healthy subjects from a preceding study [56] and was used to show the frequency-dependence of the global information flow in the brain. In particular, it was shown that the *alpha2* band at 10-13 Hz has a strong back to front information flow, while the *theta* band at 4-8 Hz has a strong front to back information flow [50]. For this reason, we base our analysis on the *alpha2* and *theta* band as well. Figure 3.1 gives a schematic overview of our processing pipeline, from an example time series of source level MEG data towards obtaining the PTE matrices for the *alpha2* frequency band (*theta* frequency band data follows a similar processing).

From the MEG measurements, we obtained[1] phase time series [51] from 78 different cortical regions of interest (ROIs) based on the Hilbert transform. We denote a possible value of the instantaneous phase of the signal of region X at time t by x_t and abbreviate the probability that the phase of X equals x_t at an arbitrary time point t to

$$\Pr[X_t = x_t] = \Pr[x_t]. \tag{3.1}$$

The information flow between two ROIs, X and Y, is then quantified by the Phase Transfer Entropy [52]

$$\mathrm{PTE}_{XY}(h) = \sum \Pr[x_{t+h}, x_t, y_t] \times \log\left(\frac{\Pr[x_{t+h} \mid x_t, y_t]}{\Pr[x_{t+h} \mid x_t]}\right) \tag{3.2}$$

[1] The MEG data were recorded using a 306-channel whole-head MEG system (Elekta Neuromag Oy, Helsinki, Finland) during a no-task, eyes-closed condition for five consecutive minutes. A beamformer approach was adopted to project MEG data from sensor space to source space [73] and the automated anatomical labelling (AAL) atlas was applied to obtain time series for 78 cortical regions of interest (ROIs) [74, 75]. For each subject, we extracted the first 20 artefact-free epochs of 4096 samples (3.2768 s).

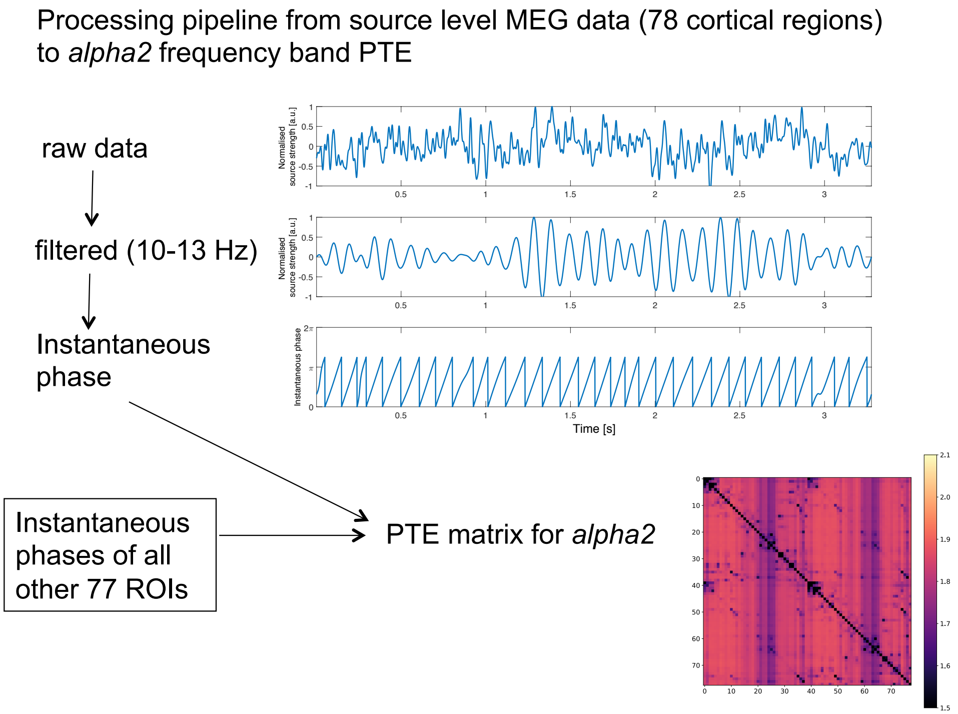


Figure 3.1: Processing pipeline from source level MEG data (78 cortical regions) to *alpha2* frequency band PTE matrix. The figure shows an example time series of a single ROI for a single epoch. In order to calculate the PTE matrices, we need the instantaneous phases of all 78 cortical regions. The PTE matrix entry (i, j) corresponds to the PTE value from region i to j.

for a certain time delay h, where the sum runs over all possible values x_t, x_{t+h} and y_t of the instantaneous phases of the signals. The (joint) probabilities are determined over histograms of their occurrences in an epoch [52]. Following the methodology of Hillebrand et al. [50], we fix h at

$$h = \frac{N_s \cdot N_{ROI}}{N_\pm}, \tag{3.3}$$

where $N_s = 4096$ and $N_{ROI} = 78$ are the number of samples in an epoch and the number of ROIs, respectively, and $N_\pm$ counts the number of sign changes for the phase across time and ROIs. For clarity, h will be omitted from the notation and we use only PTE_{XY} instead of $\mathrm{PTE}_{XY}(h)$ in the remainder. It should be noted, that the PTE of two regions X and Y is asymmetric, so $\mathrm{PTE}_{XY} = \mathrm{PTE}_{YX}$ does not hold in general. In order to remove individual bias of the measurements, all pairwise PTE values are averaged over all subjects and all epochs. A histogram of those averaged PTEs is shown in Figure 3.2 for the *alpha2* and *theta* band.

In the work of Hillebrand et al. [50], a normalized version of the PTE, called the directed Phase Transfer Entropy (dPTE) is used. The dPTE for two ROIs X and Y, is

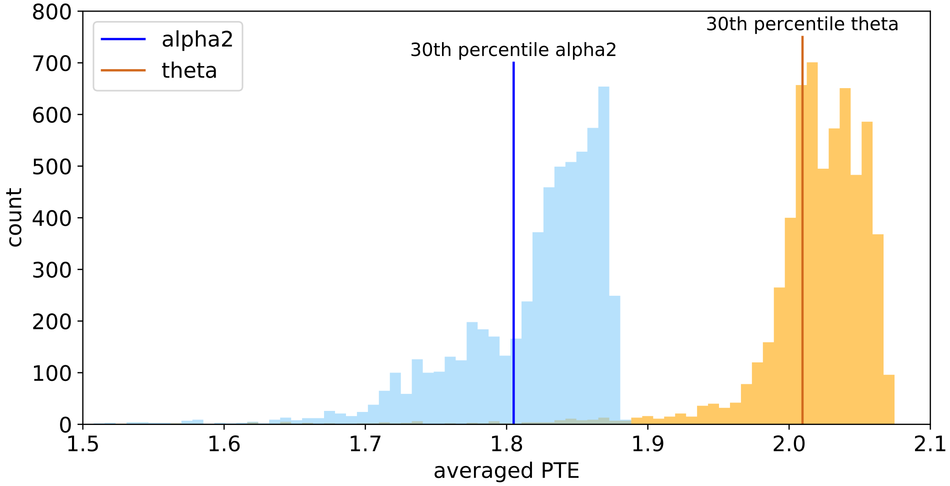


Figure 3.2: PTE between each possible pair of ROIs averaged over all subjects and measurement epochs. In total, 6006 average PTEs are displayed as a histogram with 100 bins for each of the two frequency bands. The *alpha2* frequency band (shown in blue) has on average lower PTEs than the *theta* frequency band (shown in orange). The vertical lines mark the 30th-percentile of each distribution.

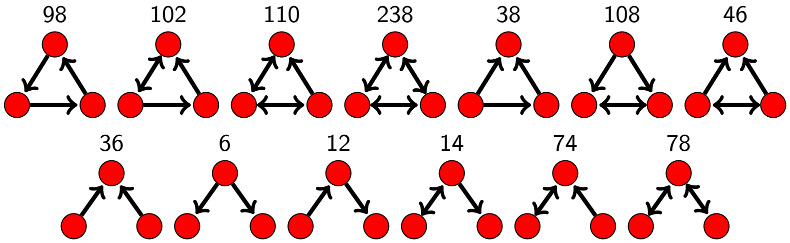


Figure 3.3: All 13 possible connected directed 3-motifs. The motif ID in binary represents the 3 × 3 adjacency matrix of the motif.

defined as

$$\mathrm{dPTE}_{XY} = \frac{\mathrm{PTE}_{XY}}{\mathrm{PTE}_{XY} + \mathrm{PTE}_{YX}}. \tag{3.4}$$

Since the PTE can only take positive values, this definition of dPTE is well-defined and its value ranges from 0 and 1. In contrast to the PTE, the dPTE is a measure of the preferred direction of information flow. If the predominant flow of information is from X to Y, then $0.5 < \mathrm{dPTE}_{XY} < 1$, else $0 < \mathrm{dPTE}_{XY} < 0.5$. Since, $\mathrm{dPTE}_{XY} + \mathrm{dPTE}_{YX} = 1$, the dPTE allows to determine for each pair of ROIs, which is a stronger receiver or a stronger sender respectively.

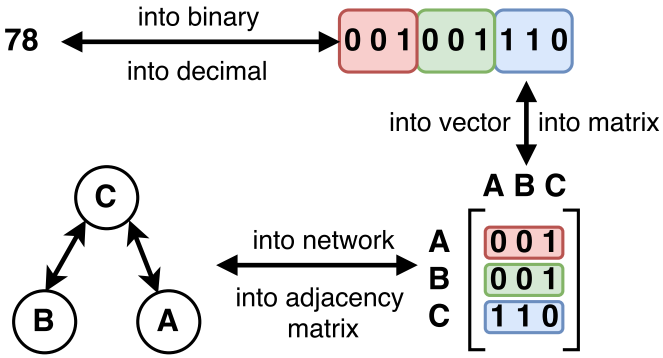


Figure 3.4: Motif ids explained by motif 78 as an example. The decimal representation 78 encodes the binary adjacency matrix of the motif.

3.2.2. Motif Search

Our motif search is performed with the *mfinder* software version 1.2 [76]. We focus our investigation mainly on the 13 different 3-motifs as shown in Figure 3.3. Each motif is identified by a number whose binary representation translates to the adjacency matrix for the corresponding motif, consistent with the notation used by *mfinder*. Figure 3.4 gives an example of this conversion, using motif number 78 (the bi-directional 2-hop path).

For any given network G (to which we refer as "original network"), the *mfinder* software performs two tasks: first, it counts the frequency $J_{G,M}$ of all motifs M in G and second, it generates a number of random networks with similar properties as the original network and determines the motif frequencies in each of them as well. For every original network, *mfinder* generates 1000 random networks using the switching algorithm described in [77] with 100 switches. We use the default parameters for *mfinder*, which preserve the degree sequence of the original network and the number of bi-directional links.

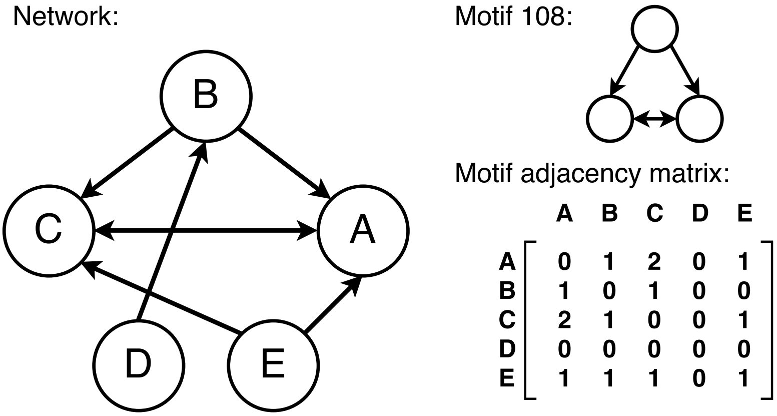
The random networks serve as a null model to determine which motifs are *overexpressed* in the original network. More precisely, we adopt the criteria given in the supplemental material of Milo et al.[13]. These criteria are:

1. The probability that a motif in a random network occurs more or an equal amount of times as in the original network is smaller than 0.01.

2. The motif appears in the original network at least 4 times with a distinct set of nodes.

3. The ratio between the motif frequency of the original network and the average number of occurrences of the motif in the random networks is at least 1.1.

Given the mean $\mu(J_{rand,M})$ and the standard deviation $\sigma(J_{rand,M})$ of the motif frequency in the random networks, the magnitude of overexpression of motif M in G is given by its z-score



$$\begin{array}{c|ccccc} & A & B & C & D & E \\ \hline A & 0 & 1 & 2 & 0 & 1 \\ B & 1 & 0 & 1 & 0 & 0 \\ C & 2 & 1 & 0 & 0 & 1 \\ D & 0 & 0 & 0 & 0 & 0 \\ E & 1 & 1 & 1 & 0 & 1 \end{array}$$

Figure 3.5: Example for the construction of a motif adjacency matrix based on the motif 108. In this example network, motif 108 can be found twice: 1. nodes {A, B, C}, 2. nodes {A, C, E}. Each instance of motif 108 is only counted once per set of nodes (node-disjoint motifs). The outgoing link from node D is still in the network, but is not part of any motif instance 108. The node-pair {A,C} is part of two different (node-disjoint) instances of motif 108, which is why there is a value of 2 at the corresponding cells of the motif adjacency matrix.

$$z_{G,M} = \frac{J_{G,M} - \mu(J_{rand,M})}{\sigma(J_{rand,M})}. \tag{3.5}$$

A motif which is not overexpressed may still occur quite frequently in the original network, though it arises at a similar frequency by a random link rewiring process. Thus, it can be argued that overexpressed motifs carry some functional importance for the underlying system since they do not arise merely by chance.

3.2.3. MOTIF-BASED CLUSTERING

Benson et al. [71] developed a clustering algorithm that partitions a network G based on a motif M. The main idea of their algorithm is to construct clusters by "cutting" through the minimum possible number of motif instances, while maintaining a high density of motif instances within each of the clusters. In this section, we summarize only the basic concepts (including the algorithm) necessary to understand how the clustering of the networks was achieved. Details about the performance, complexity and additional applications can be found in the supplemental material of Benson et al. [71] together with a comprehensive analysis of the algorithm.

MOTIF ADJACENCY MATRICES

Let G be a directed network with a set of nodes $\mathcal{N} = \{1, 2, \ldots, N\}$. Two motif instances are called *node-disjoint* if their set of nodes are not identical, i.e. they have at least one node not in common. For each pair of nodes i, j let w_{ij} be the number of node-disjoint motif instances in which i and j participate together. Then, the $N \times N$ symmetric matrix W_M with elements w_{ij} is called the *motif adjacency matrix*. The elements d_{ij} of the *motif* *diagonal degree matrix* D_M are given by

$$d_{ii} = \sum_{j=1}^{N} w_{ij} \tag{3.6}$$

and the *motif Laplacian* by

$$L_M = D_M - W_M. \tag{3.7}$$

The clustering algorithm uses the eigenvector belonging to the second smallest eigenvalue of the *normalized motif Laplacian*, which is defined as

$$\mathscr{L}_M = I - D_M^{-\frac{1}{2}} W_M D_M^{-\frac{1}{2}} \tag{3.8}$$

where I denotes the identity matrix. Figure 3.5 illustrates the construction of a motif adjacency matrix.

Motif conductance

Given the motif adjacency matrix W_M of a network G, and a partition of the nodes $N = |\mathscr{N}|$ into two disjoint subsets $\mathscr{N}_1$ and $\mathscr{N}_2 = \mathscr{N} \setminus \mathscr{N}_1$, we define the motif conductance $\phi_G(\mathscr{N}_1, \mathscr{N}_2)$ of that partition as

$$\phi_G(\mathscr{N}_1, \mathscr{N}_2) = \frac{\mathrm{cut}_G(\mathscr{N}_1, \mathscr{N}_2)}{\min\{\mathrm{vol}_G(\mathscr{N}_1), \mathrm{vol}_G(\mathscr{N}_2)\}} \tag{3.9}$$

with

$$\mathrm{cut}_G(\mathscr{N}_1, \mathscr{N}_2) = \sum_{i \in \mathscr{N}_1, j \in \mathscr{N}_2} w_{ij} \tag{3.10}$$

and for $a = 1, 2$

$$\mathrm{vol}_G(\mathscr{N}_a) = \sum_{i \in \mathscr{N}_a} \sum_{j}^{N} w_{ij} = \sum_{i \in \mathscr{N}_a} d_{ii}. \tag{3.11}$$

Thus, the motif conductance $\phi_G(\mathscr{N}_1, \mathscr{N}_2)$ equals the ratio between the number of motif-instances cut by the partition $\{\mathscr{N}_1, \mathscr{N}_2\}$ and the lowest number of preserved motif-instances in one of the two partitions.

Motif-based clustering algorithm

A low conductance is often a desirable quality for a network clustering [78]. However, finding the minimum conductance of a network is a well-known NP-complete problem [79] which directly translates to the complexity of finding the minimum motif conductance ϕ_G^*. [71] present a polynomial-time algorithm that finds a nearly optimal partition $\{\mathscr{N}_1, \mathscr{N}_2\}$ with motif conductance

$$\phi_G(\mathscr{N}_1, \mathscr{N}_2) \leq 4\sqrt{\phi_G^*} \tag{3.12}$$

for 3-motifs. In practice, the runtime is largely dominated by the computation of the motif adjacency matrix, which is still efficient for the motifs of size three that we consider for this work.

The algorithm from [71] is a generalization of the classical spectral clustering algorithm [80, 81], which makes use of the Laplacian matrix of a network. The eigenvector corresponding to the second smallest eigenvalue of this matrix is known as Fiedler's vector [82] and by ordering its elements, a node partition of a low (link-based) conductance can be devised.

The main steps of the algorithm from Benson et al. [71] consist of computing the motif adjacency matrix W_M from which the normalized motif Laplacian $\mathscr{L}_M$ is constructed and the second smallest eigenvalue is computed. Afterwards, the corresponding eigenvector is used to create a partition $\{\mathscr{N}_1, \mathscr{N}_2\}$ according to the smallest motif conductance. The motif conductance is only defined for nodes that participate in at least one instance of the motif M. Thus, if a node does not participate in any motif instance of M, it can neither be a member of $\mathscr{N}_1$ nor $\mathscr{N}_2$. We refer to such nodes as *non-participating nodes* and show them as a third, separate group in later visualizations, in case they occur.

The complete algorithm is listed as Algorithm 1 in pseudocode. We implemented the algorithm in Python (using NumPy and NetworkX) and double-checked our results with the implementation available on the SNAP-platform [83].

Algorithm 1 Motif-based clustering algorithm

```
**Input:** Directed, unweighted network $G$ and motif $M$
**Output:** Motif-based clusters $\mathscr{N}_1$ and $\mathscr{N}_2$ (subsets of nodes)
$W_M \leftarrow$ motif adjacency matrix of $G$ with respect to $M$
$D_M \leftarrow$ diagonal degree matrix of $W_M$
$\mathscr{L}_M \leftarrow I - D_M^{-\frac{1}{2}} W_M D_M^{-\frac{1}{2}}$ normalized motif Laplacian
$z \leftarrow$ eigenvector corresponding to second smallest eigenvalue of $\mathscr{L}_M$
$\sigma_i \leftarrow$ index of vector $D_M^{-\frac{1}{2}} z$ with $i$th smallest value
$\ell \leftarrow \arg\min_{i=1,\dots,\mathscr{N}} \phi_G(\{\sigma_1,\dots,\sigma_i\},\{\sigma_{i+1},\dots,\sigma_N\})$
$\{\mathscr{N}_1, \mathscr{N}_2\} \leftarrow \{\sigma_1,\dots,\sigma_\ell\},\{\sigma_{\ell+1},\dots,\sigma_N\}$
**return** $\{\mathscr{N}_1, \mathscr{N}_2\}$
```

3.3. Information Flow Analysis for dPTE

3.3.1. Network Construction Based on dPTE

The pairwise dPTEs over all ROIs can be interpreted as a weight matrix of a fully connected network. Since the data is from 67 subjects each over $k = 20$ epochs, we have 1340 weighted networks to begin our construction. We apply a procedure to thin out links and induce a directionality per link instead of a weight. After this transformation, which we call "sparsificiation", we obtain a sparse directed (unweighted) network for each subject, which is amenable for motif search and analysis.

The sparsification consists of two steps: First, we discard all links whose weights are in close proximity to 0.5. More precisely, every link whose average weight (over all

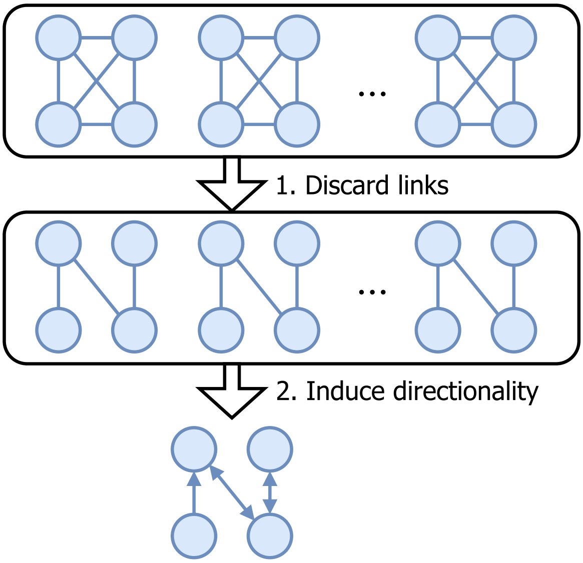


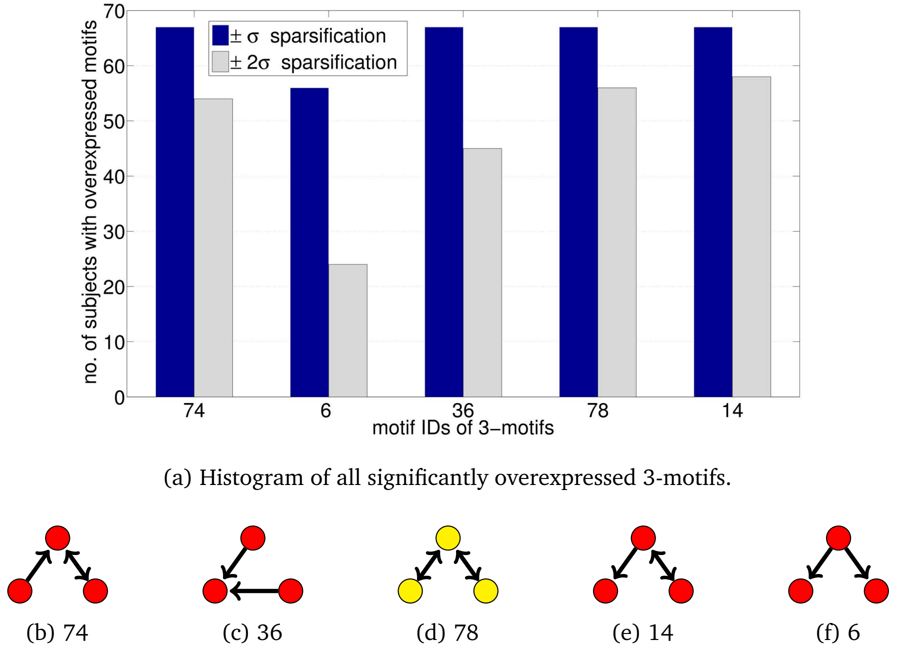
Figure 3.6: Schematic overview of the two steps for constructing the directed network (sparsification): (1) discard links close to 0.5 (2) induce directionality for remaining links.

epochs) is within the closed interval $[0.5 - \alpha\sigma, 0.5 + \alpha\sigma]$ will not be considered, where σ is the standard sample deviation taken over all epochs over all pairs of nodes and α is a positive real control parameter. Under the assumption of a normal distribution with mean 0.5, the 3σ-rule states that this procedure will remove approximately 68% for $\alpha = 1.0$ and 95% for $\alpha = 2.0$ of all links. A schematic overview of the sparsification procedure is given by Figure 3.6.

In a second step, we determine for each remaining link whether it should be bi- or uni-directional, and in case of the latter, in which direction the links should be oriented. Clearly, all remaining link weights are now bounded away from 0.5, though it is possible, that for different epochs a single link weight might be lower or higher than 0.5, which makes it ambiguous which member of the node pair is the dominant sender and which the dominant receiver. Let k^+ (k^-) be the number of epochs that the dPTE$_{XY}$ is above (below) 0.5 where $k = k^+ + k^-$ is the total number of epochs for a subject. If $k^+/k \geq 0.75$, we assume X to be a dominant sender and thus we induce a uni-directional link from X to Y. Contrary, we assume X to be a dominant receiver if $k^+/k \leq 0.25$ and point the link from Y to X. If neither applies ($0.25 < k^+/k < 0.75$), we assume that X and Y frequently change roles between dominant sender and dominant receiver. Thus, we induce a bi-directional link between them.

3.3.2. OVEREXPRESSED MOTIFS

With sparsification, we generate one directed network for each of the 67 subjects for our motif search. Additionally, we construct an averaged effective connectivity network (short: averaged network) by considering all epochs of all subjects together. This con-



(a) Histogram of all significantly overexpressed 3-motifs.

(b) 74

(c) 36

(d) 78

(e) 14

(f) 6

Figure 3.7: **(a)** Frequency of significantly overexpressed 3-motifs over all regarded subjects after the $\pm\sigma$ and $\pm 2\sigma$ sparsification, respectively. **(b)-(f)** All significant 3-motifs over all subjects together with their motif ID. The yellow motif with ID 78 is also overexpressed in the averaged network.

struction results in a "virtual" subject with $k = 1340$ instead of $k = 20$ epochs. We set α to 1.0 and 2.0 to compare on different levels of sparsity.

Since the complexity of motif search increases dramatically with the size of the motif, we restrict our search to subgraphs of 3 and 4 nodes (further called 3-motifs and 4-motifs). The motif search is performed as outlined in Subsection 3.2.2. We report the motifs which are overexpressed with a z-score of at least 2.

Overexpressed 3-motifs

For both variants of the sparsification method ($\alpha = 1$ and $\alpha = 2$), we find the same overexpressed 3-motifs over all subjects meaning that those motifs are more frequent in our analyzed networks than in the null model (see Figure 3.7). Those five motifs are not triangular, as they include at least one pair of nodes, which is not connected by any link. The absolute frequency of those motifs is displayed as a histogram in Figure 3.7a for the $\pm\sigma$ and the $\pm 2\sigma$ sparsification, respectively. The analysis on the averaged effective connectivity network confirms the over-representation of the motif with ID 78, the bi-directional, which is the only significantly overexpressed motif that has been found for different sparsification methods (z-scores: 88.25 for $\pm\sigma$ sparsification and 82.7 for $\pm 2\sigma$ sparsification).

Overexpressed 4-motifs

In Figure 3.8a we present a histogram of all significantly overexpressed 4-motifs with the two different sparsification levels. Twelve 4-motifs were found overexpressed in all 67 subject networks. Considering the averaged network at ±σ sparsification, we we find 3 overexpressed motifs (marked yellow in Figure 3.8, z-scores: 203.74 for ID 13260, 111.89 for ID 4382 and 14.85 for ID 4698). There were no overexpressed motifs in the averaged network found at the ±2σ sparsification. The two 4-motifs with number 13260 and 4382, the bi-directional ring and the bi-directional star, respectively, have the highest z-scores in the averaged effective connectivity network and are a subset of the overexpressed 4-motifs found for every individual subject. The overexpression of those two motifs cannot be explained by the higher number of bi-directional links in the effective connectivity network since the null model contains the same number of bi-directional links.

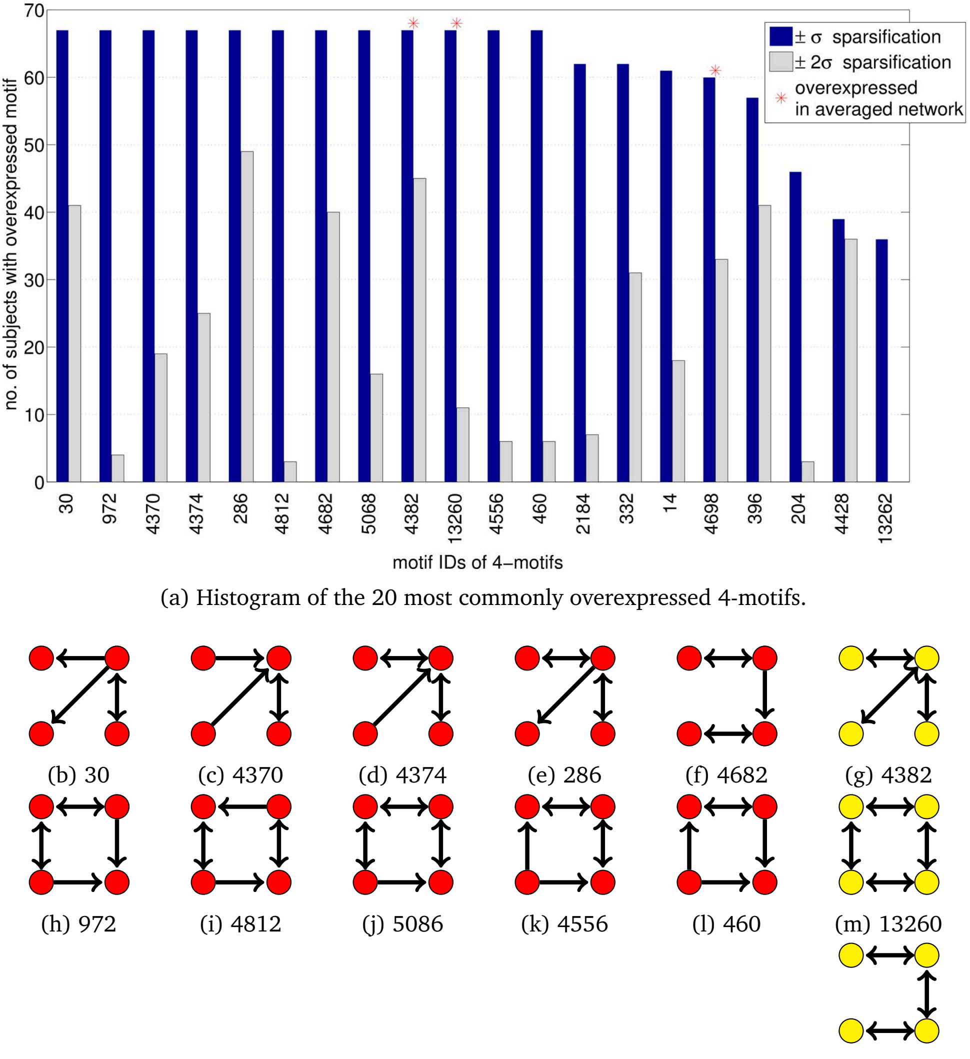
3.3.3. Motif-based Clustering

We apply Algorithm 1 from Section 3.2.3 to the averaged effective connectivity network. Since for both sparsification methods, the 3-motif with ID 78 was significantly overexpressed in the averaged effective connectivity network and in every subject network, we cluster according to this motif. Figure 3.9a shows the two clusters we find for the sparsified network for ±σ plotted on the template brain. The frontal brain regions seem to be consistently part of the same cluster (red) and the distribution of the clusters across the two brain hemispheres shows a strong symmetry. The sparser network resulting from the ±2σ sparsification method was disconnected. Consequently, we could only obtain a motif-based clustering of the largest connected component. The result is shown in Figure 3.9b. For comparison, we also applied an link-based spectral clustering algorithm [81] to see whether the motif-based clustering gives an advantage. Figure 3.9c shows the result.

3.3.4. Discussion

The fact that the motif-based clustering reveals a strong symmetry between the two brain hemispheres is remarkable, given that the algorithm does not work on the physical geometry of the brain but merely on a topological level. This supports the idea of a higher-order organization of the effective connectivity brain network. In comparison, the result of a standard spectral clustering algorithm (link-based conductance) as shown in Figure 3.9c gives a much weaker symmetry and a more disconnected spatial distribution of the two clusters. Thus, we conclude that a motif-based clustering may provide meaningful structures.

However, a rather dense network (±σ) seems to be necessary for the emergence of a higher-order structure since the clustering for the sparser averaged network (±2σ) appears to be frail (see Figure 3.9b). This demands a further investigation which level of sparsity results in an optimal clustering. So far, the sparsification is based on the choice of a parameter α to eliminate nodes which cannot be identified by the dPTE as dominant sender or dominant receivers. However, the actual directionality of each link, i.e. if it is a uni-directional or bi-directional link, depends on the number of epochs in which the dPTE is above or below 0.5.



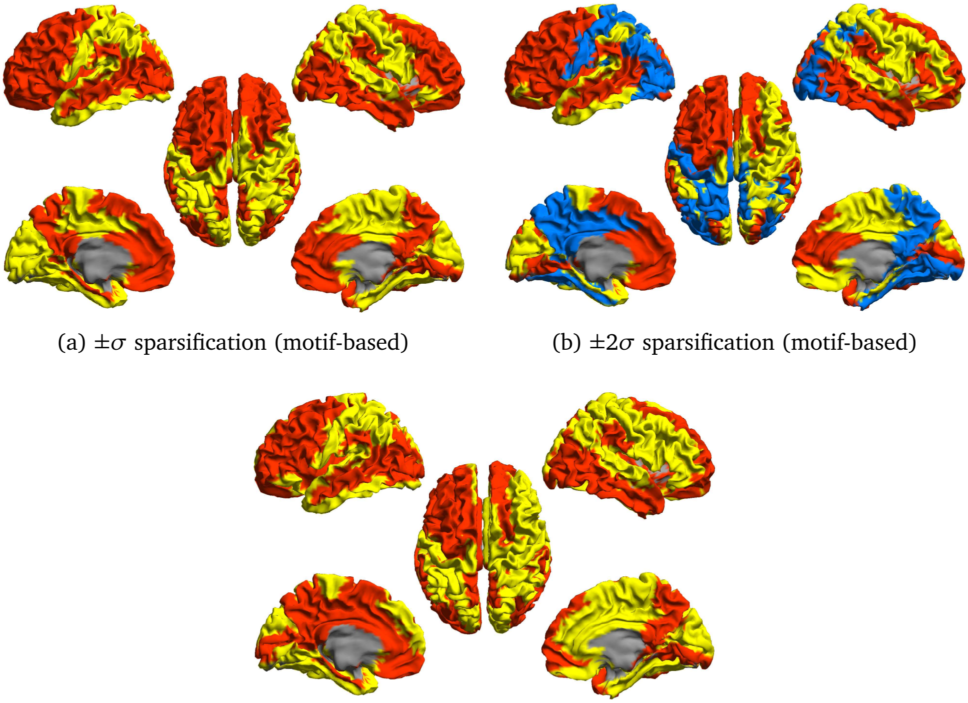
(a) Histogram of the 20 most commonly overexpressed 4-motifs.

(b) 30 (c) 4370 (d) 4374 (e) 286 (f) 4682 (g) 4382

(h) 972 (i) 4812 (j) 5086 (k) 4556 (l) 460 (m) 13260

(n) 4698

Figure 3.8: **(a)** Histogram of the 20 most commonly overexpressed 4-motifs over all subjects after the $\pm\sigma$ and $\pm 2\sigma$ sparsification, respectively. An asterisk marks the motifs that are also overexpressed in the averaged network. **(b)-(m)** The twelve 4-motifs that are overexpressed after the $\pm\sigma$ sparsification in every subject with their motif ID. The yellow motifs **(g)**, **(m)** and **(n)** are overexpressed in the averaged network.



(a) ±σ sparsification (motif-based)

(b) ±2σ sparsification (motif-based)

(c) ±σ sparsification (spectral clustering)

Figure 3.9: **(a-b)** Two Clusters (red and yellow) based on motif 78 on the template brain for different sparsifications. Blue colored regions in **(b)** consist of non-participating nodes (nodes that are not part of any motif 78 instance). **(c)** A spectral clustering on the same network as in **(a)** breaks the symmetry between brain hemispheres and seems less coherent.

In the following section, we complement our study with a different approach, which does not emphasize on the roles of dominant senders and receivers, but on the magnitude of (potentially bi-directional) information flow. Consequently, we have to revise the network construction and base it on the PTE directly, for which directionality does not need to be induced by deviations over different epochs, but emerges naturally. Moreover, instead of fixing the sparsification parameter to some fixed $\alpha\sigma$, we will investigate a larger interval of possible thresholds, which results in different densities for the network. In addition to the *alpha2* frequency band, we will also expand the study with data from the *theta* frequency band to see whether different frequencies give rise to different clusters in the brain.

3.4. Information Flow Analysis for PTE

3.4.1. Network Construction Based on PTE

The pairwise PTE values between all 78 ROIs imply a fully connected network G_{PTE} where each ROI is a node and the PTE is the weight of each link. In order to filter out noise and

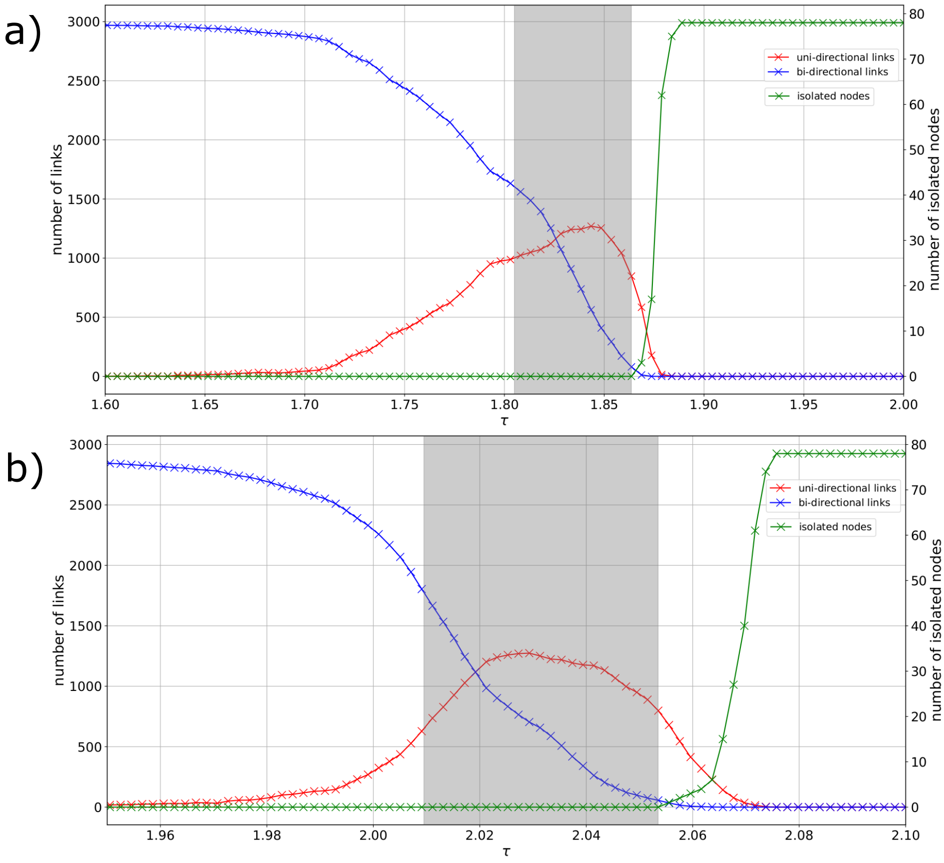


Figure 3.10: Left axis: change in link density (in absolute number of links) with respect to different values of τ. Our selection of τ_{min} corresponds to a relative link density of 0.7 (4204 links, counting each bi-directional link as two links). Right axis: number of isolated nodes. Our selection of τ_{max} is the highest possible τ for which there is still 1 weakly connected component (i.e. the network has no isolated nodes). The grey shaded areas indicate the resulting interval $[\tau_{min}, \tau_{max}]$ for a) *alpha2* and b) *theta* band.

focus on the most important connections possessing the highest PTE values, all links with a PTE below or equal a certain threshold τ are discarded (set to zero) and all links above τ remain without a weight (set to one). This procedure eliminates weak connections which might otherwise obscure the inherent topology induced by significantly stronger connections. If (for a fixed h) $\text{PTE}_{XY} > \tau$ and $\text{PTE}_{YX} > \tau$ for two ROIs X and Y, a bi-directional link between X and Y is set. Similarly, for $\text{PTE}_{XY} > \tau \geq \text{PTE}_{YX}$, only a uni-directional link from X to Y is set. Thus, by selecting an appropriate threshold τ, the fully connected weighted network G_{PTE} is transformed into a sparser, directed and unweighted network $G(\tau)$, also known as binary directed network.

Finding an appropriate threshold τ is a challenge in itself [84], which we will not undertake, since one singular value for τ will not be needed in our approach here. Instead, we consider a class of networks $G(\tau)$ created by sampling τ from an interval

Table 3.1: Network properties of $G(\tau)$ for τ at the endpoints of the interval $[\tau_{min}, \tau_{max}]$. For *alpha2* we have $[\tau_{min}, \tau_{max}] = [1.8050, 1.8636]$ and for *theta* $[\tau_{min}, \tau_{max}] = [2.0095, 2.0535]$.

	alpha2		***theta***	
	$G(\tau_{min})$	$G(\tau_{max})$	$G(\tau_{min})$	$G(\tau_{max})$
#uni-directional links	1006	848	648	799
#bi-directional links	1601	81	1776	56
average degree	53.949	12.949	53.846	11.679
assortativity	-0.105	-0.129	-0.351	-0.062
link density	0.700	0.168	0.700	0.152

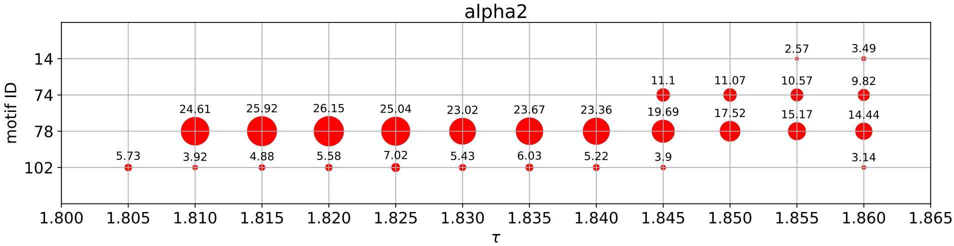


Figure 3.11: Overexpressed 3-motifs for *alpha2* band data in the interval $[\tau_{min}, \tau_{max}]$. The area of the circles scales with the z-scores. The numerical value of the z-scores is plotted on top of each circle for better comparison. Note that motif 78 has consistently the highest z-score.

$[\tau_{min}, \tau_{max}]$. Setting $\tau = 0$ results in a fully connected network whereas setting τ to the maximum of all PTE values results in an empty network of 78 isolated nodes. Clearly, these extreme thresholds provide networks that lack structure and present no insight. To avoid constructing such degenerate networks, we pick a narrower interval as follows:

We set τ_{max} to be the smallest threshold at which the obtained network is still weakly connected, i.e. has no isolated nodes. To avoid too many weak connections, τ_{min} is set to the 30th-percentile of the PTE distributions (see Figure 3.2). This value eliminates a fair amount of weak connections while the majority of the strongest connections persist.

The networks within $[\tau_{min}, \tau_{max}]$ are all connected, but sparse enough to resemble complex structures. At τ_{max} itself, the link density is 0.168 for *alpha2* and 0.152 for *theta*, whereas the 30%-percentile of τ_{min} corresponds to networks with a link density of 0.7. This allows to cover a large variety of different networks in $[\tau_{min}, \tau_{max}]$, each representing a different perspective on the underlying data. For example, we observe that the assortativity [85] for *theta* frequency band data ranges from -0.351 to -0.062 and that the ratio between uni-directional and bi-directional links is changing as well. Table 3.1 contains the exact values of τ_{min} and τ_{max} together with some properties of networks at the interval endpoints. Figure 3.10 shows how the number of links is changing for various sampled values of τ, including the interval.

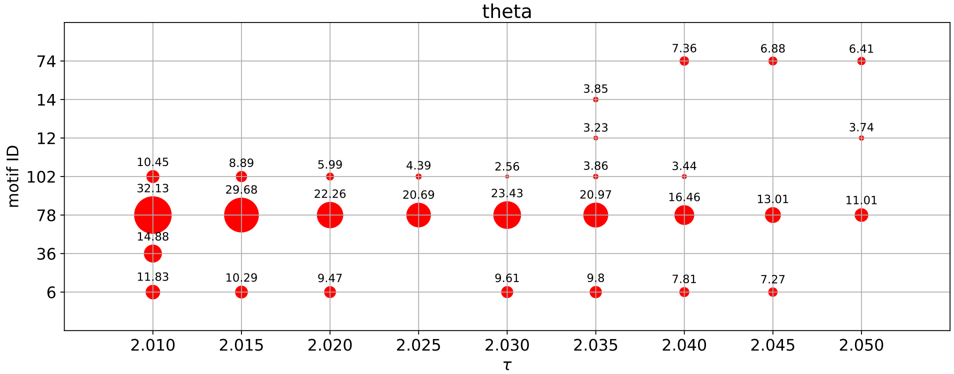


Figure 3.12: Overexpressed 3-motifs for *theta* band data in the interval $[\tau_{min}, \tau_{max}]$. The area of the circles scales with the z-score. The numerical value of the z-score is plotted on top of each circle for better comparison. Note that motif 78 has consistently the highest z-score.

3.4.2. Overexpressed Motifs

We sample the interval $[\tau_{min}, \tau_{max}]$ with a step-size of $\Delta = 0.005$, for both *alpha2* and *theta* band data. For each sampled threshold τ, we construct $G(\tau)$ and regard $G(\tau)$ as the original network for *mfinder* in order to determine all overexpressed motifs. Figure 3.11 shows the overexpressed motifs for *alpha2* and Figure 3.12 for *theta* band data together with the corresponding z-scores.

We observe that motif overexpression depends on the chosen threshold τ. For example, in the *alpha2* band motif 74 and motif 14 were only detected in very sparse networks close to the connectivity threshold τ_{max}. Moreover, there are gaps at certain ranges of τ in which a motif does no longer fulfill all overexpression criteria, e.g. motif 102 at $\tau = 1.85$ and $\tau = 1.855$ for *alpha2* or motif 6 at $\tau = 2.025$ and $\tau = 2.050$ for *theta*.

From all overexpressed motifs, motif 78 stands out for the following reasons: Firstly, motif 78 is overexpressed in both, *alpha2* and *theta*, for a large part of the interval $[\tau_{min}, \tau_{max}]$ without gaps between our sample points. Secondly, the z-scores for this motif are always higher than the z-scores of any other overexpressed motif for the corresponding thresholds. Hence, we select motif 78 as our motif M for the motif-based clustering in Section 3.4.4.

3.4.3. Apex-ratio and Overlap with Hubs

Motif 78 encodes a pattern in which one central node is bi-directionally linked with two otherwise disconnected nodes. The node at this central position of motif 78 is known as *apex* and has been shown to be related to brain dynamics in previous studies [86–89]. In Figure 3.4, the node labeled “C” is at the apex-position. The apex-ratio of a node is the ratio between the node occupying the apex-position divided by its total participation in instances of the complete motif 78. For example, an apex-ratio of 1 corresponds to a node that is always at the apex-position of motif 78, and never at a different position. Figure 3.13 shows a mapping of the average apex-ratio to the template brain for both frequency bands. The average was taken over equally distributed sample points, taken

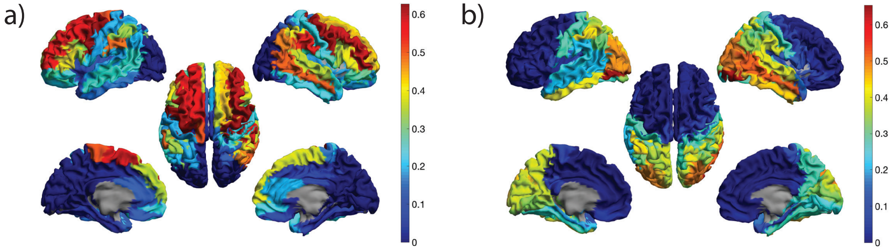


Figure 3.13: Average apex-ratio mapped to the template brain. The average was taken over equally distributed sample points of a) $[\tau_{min}, \tau_{max}] = [1.8050, 1.8636]$ for *alpha2* and b) $[\tau_{min}, \tau_{max}] = [2.0095, 2.0535]$ for *theta*. The step-size for sampling was $\Delta = 0.005$. The nodes with the highest apex-ratio in the *theta* band are found in posterior brain regions, where for *alpha2* the apex-ratio is lowest.

from the corresponding $[\tau_{min}, \tau_{max}]$ with a step-size of $\Delta = 0.005$.

Following the study by Sporns et al.[64] conceptually, we are interested in the relation between the apex-ratio of a node and its degree. A node is a *high-degree node*, if its degree (number of incoming + outgoing links) is at least as large as the average degree of the network plus one standard deviation. Figure 3.14 shows that most of the nodes with the highest apex-ratio are also high-degree nodes in both the *alpha2* and *theta* band for τ fixed to $\frac{1}{2}(\tau_{min} + \tau_{max})$. While the apex-ratio and the number of high-degree nodes change with τ, we observe (not shown) a similar relation for different values of τ as well. More specifically, when considering the sample points between τ_{min} and τ_{max} described in the previous paragraph, the Pearson correlation coefficient between the apex-ratio and the degree for all nodes with a positive apex-ratio lies within $[0.53, 0.86]$ for *alpha2* and within $[0.55, 0.95]$ for *theta*.

3.4.4. Motif-based Clustering

The first step to apply the motif-based clustering to the brain is to fix a motif M. In Section 3.4.2, we identified motif 78 to be of high importance: it is prominent in both, the *alpha2* and *theta* band and provides continuously the highest z-score of all motifs, which designates it as the strongest candidate. Moreover, motif 78 is most robust against changes in τ as it was overexpressed at almost all sample points taken within $[\tau_{min}, \tau_{max}]$. However, it is not obvious, which of these sample points would result in the best possible network representation to create a meaningful clustering. To circumvent the selection of a fixed single threshold, we define a set of different thresholds T, each of them related to a different network and thus to different motif adjacency matrices. This is similar to the analysis done for Figure 3.11, where we sampled $[\tau_{min}, \tau_{max}]$ with a step-size of $\Delta = 0.005$, resulting in a set

$$\mathrm{T}^* = \{\tau_{min} + k \cdot \Delta \mid k = 1, \ldots, 12\}. \tag{3.13}$$

While this set is sufficient to get an idea about the impact of a changing τ on motif counts and makes for some compelling visualizations, equally distributed sample points result in a bias, since the change in the networks (i.e. their numbers of links) does not scale

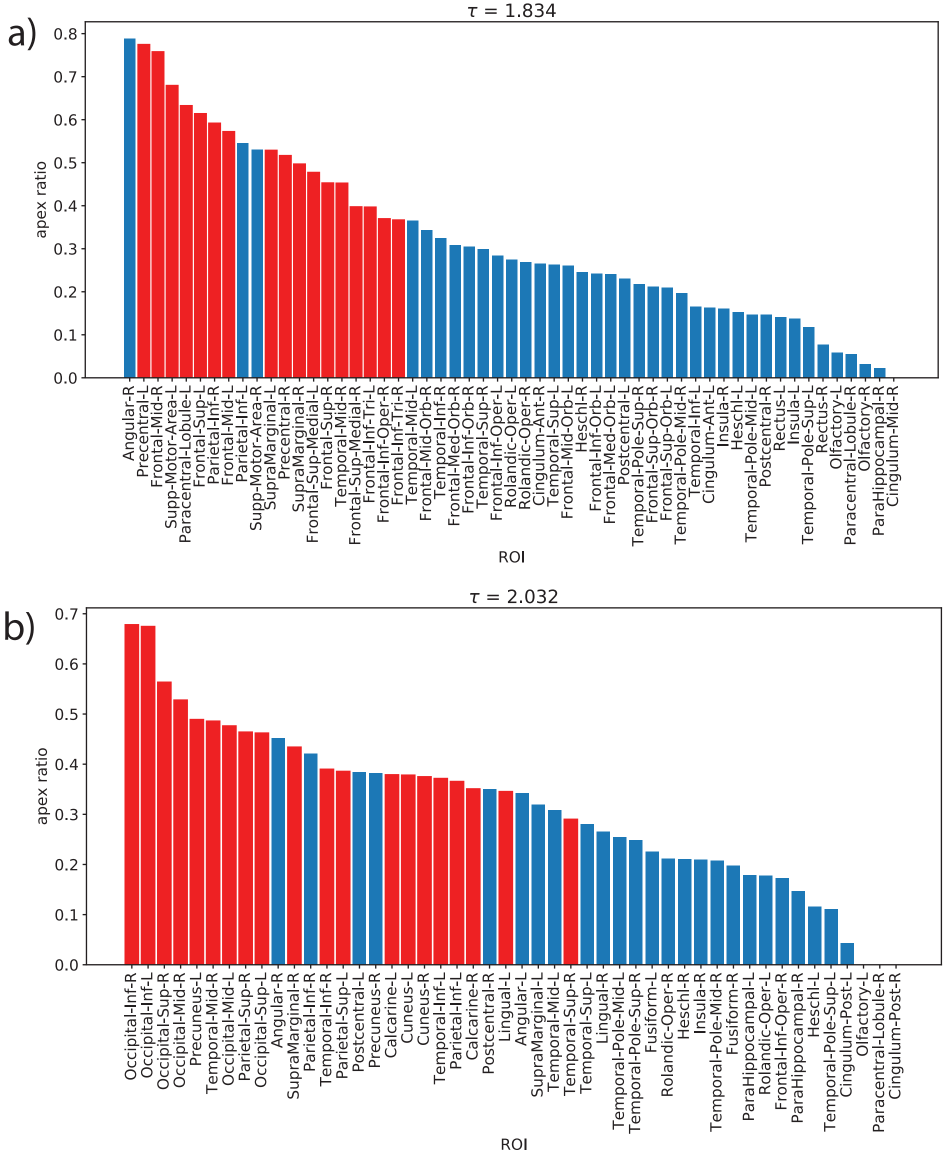


Figure 3.14: ROIs sorted in decreasing order by their apex-ratio. Red bars mark high-degree nodes, i.e. nodes with a degree higher than the average degree plus one standard deviation. a) *alpha2* band for $\tau = 1.834$, b) *theta* band for $\tau = 2.032$.

linearly with τ as shown in Figure 3.10.

To avoid this bias, we pick the sample points T such that between each two consecutive sample points the corresponding networks change by the same amount. The smallest amount of change between two networks is the existence (or absence) of a single link. If we begin with the network $G(\tau = \tau_{min})$ and slowly increase τ by ε until $G(\tau)$ and $G(\tau + \varepsilon)$ differ by exactly one link, we add $\tau + \varepsilon$ to our set T of sample points and continue this procedure until we eliminate the next link and so on. Thus, T consists of all thresholds τ within $[\tau_{min}, \tau_{max}]$ at which the corresponding networks change by one link[2], creating an unbiased sample of high resolution.

For a graph $G(\tau)$ based on a threshold τ, let the corresponding motif adjacency matrix be denoted by $W_M(\tau)$. Then, summing the motif adjacency matrices over all networks generated by the elements in T results in an *aggregated motif adjacency matrix*

$$W_{M_{agg}} = \sum_{\tau \in \mathrm{T}} W_M(\tau) \tag{3.14}$$

for each frequency band. Applying the motif-based clustering algorithm to the aggregated motif adjacency matrix given by Equation (3.14) constructs a partition that takes the structure of different networks into account. Motifs consisting of strong links (i.e. with weights close to τ_{max}) will be part of many of these networks, giving them more importance when searching for a partition of low motif conductance. In contrast, motifs with weak links (weights close to τ_{min}) receive less consideration accordingly.

Although the aggregation avoids to base the complete analysis on a single fixed threshold, it introduces another difficult choice: the sample interval $[\tau_{min}, \tau_{max}]$. Our reasoning to set τ_{min} to the 30th-percentile of the PTE-distribution and τ_{max} to the weak connectivity threshold has been discussed already in Section 3.4.1. To add to this reasoning, we want to point out that in general, a small change to the endpoints from $[\tau_{min}, \tau_{max}]$ will only result in small changes to aggregated clusterings, while a small change to a clustering based on a single threshold is comparably more sensitive. Ultimately, setting the interval $[\tau_{min}, \tau_{max}]$ must, to some extent, remain a matter of *preference*, as it reflects which of the measurements (PTE values) are expected to be meaningful.

The results of the partition of the brain into 2 clusters are shown in Figure 3.15 for the *alpha2* band data and in Figure 3.16 for the *theta* band data, based on our preference for $[\tau_{min}, \tau_{max}]$.

3.4.5. Discussion

Overexpression of motif 78

Concerning network motifs, we observed that the overexpression of motif 78 we have shown for the dPTE-based networks of Section 3.3 persists for the PTE-based network construction as well. Two other motifs, 14 and 74, which can be regarded as degenerated forms of motif 78 missing one uni-directional link, have also been identified as overexpressed in both cases (dPTE and PTE). Due to the overview over a range of thresholds (Figure 3.11 and Figure 3.12), we can explain the origins of the overexpression of these

[2] Note that the values in T are exactly the PTE values of the links that get removed by this procedure.

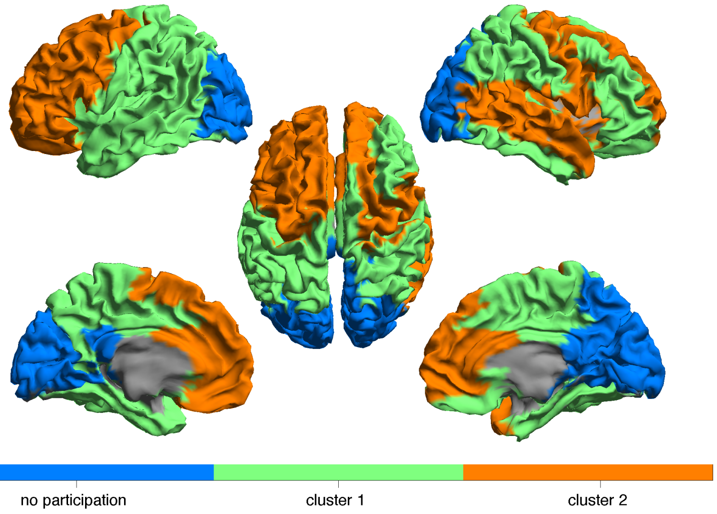


Figure 3.15: Partition of brain networks into two clusters of nodes based on motif 78 for the *alpha2* band. 15 out of 78 nodes did not participate in any motif 78 instance and are shown as a separate third cluster.

related motifs: Since motifs 14 and 74 are only overexpressed for higher thresholds τ and, thus, only for sparser networks, their appearance seems to be a direct consequence of the applied threshold removing the weakest link in motif 78. Thus, motifs 14 and 74 are most likely consequences of the applied threshold not representing new triangular relations but supporting the overall dominance of motif 78.

The overexpression of motif 78 is also in line with previous research stating the same result for the structural brain networks of the macaque and the cat [62]. Gollo et al. [90] applied neural mass models on the macaque connectome and identified motif 78 as an important motif for the dynamic core of the brain network. Furthermore, a recent study by Wei et al.[91] singled out motif 78 as an important motif for the information transfer in functional brain networks. In particular, a node at the apex position of motif 78 acts as a *bridge* for the information flow between its neighbors and the overexpression of motif 78 could represent the basic principle of segregation and integration at the macroscopic level of brain regions [62]. The principle of segregation and integration originates from neuronal dynamics where signals from spatially segregated neurons are integrated with each other into one coherent signal [92–94]. Further, Honey et al. [63] showed that the participation of a node in motif 78 has a high correlation with being a hub of the network. The overexpression of motif 78 together with its close relation to hubs confirms previous findings identifying hubs as drivers for the integration of information flow [64, 90, 95]. In addition, the overexpression of motif 78 in both frequency bands, *alpha2* and *theta*, strengthens the claim even further that motif 78 is a general building block of effective connectivity networks and therefore an important feature for the information flow in

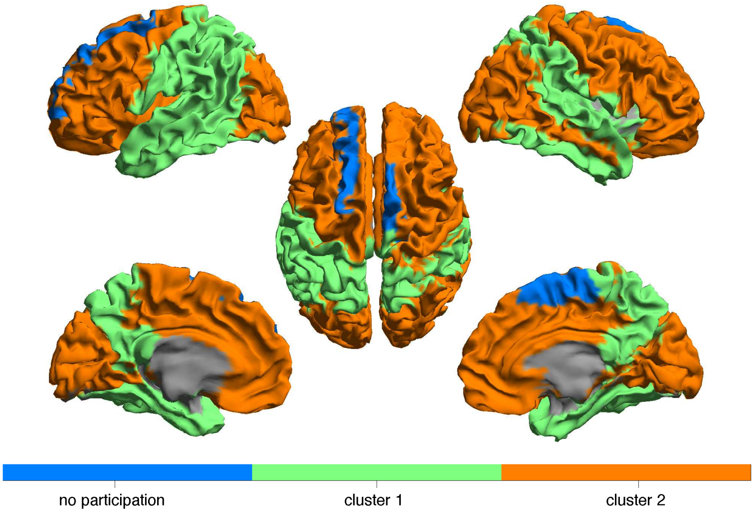


Figure 3.16: Partition of brain networks into two clusters of nodes based on motif 78 for the *theta* band. 2 out of 78 nodes did not participate in any motif 78 instance and are shown as a separate third cluster.

brain networks.

We showed that the hubs of the effective connectivity network often take on the apex position of motif 78. This hub-apex relation has previously been shown by Sporns et al. [64] for structural brain networks. We now extended this finding to the effective connectivity networks, identifying another shared feature of brain structure and function. The effective connectivity hubs seem to be located more in the front for the *alpha2* band and in posterior regions for the *theta* frequency band (Figure 3.13). Considering these opposite locations together with the opposite directions of information flow that have been discovered by Hillebrand et al. [50], these effective connectivity hubs seem to be the targets of the global information flow. Thus, one could argue that their target position in the global information flow patterns makes these hubs “slaves” of the information flow, which is line with a previous study by Gollo et al. [90]. These findings support earlier studies by Moon et al. [55] and Meier et al. [96], which showed that hubs play an important role for the global network dynamics, and extend them from the structural to the functional domain.

Clusters of the functional brain network

When analyzing the global intertwined organization of motif 78, we identified spatially coherent clusters in both frequency bands. Overall, the motif-based clustering algorithm split the brain in three major parts, the frontal lobe, the occipital lobe and the rest corresponding to a joint cluster of temporal and parietal lobe. Without including any spatial information in the construction of the directed networks or any restriction on locations for the performed clustering, we were able to recognize this well-known global spatial organization of the human brain in our obtained clusters.

As a commonality between the *alpha2* and *theta* band, the frontal regions seem to be nearly consistently together in one cluster. Moreover, in alignment with the recent study of Hillebrand et al. [50] we also observe differences in the global patterns between high and low frequency bands. Whereas in the *theta* band, the posterior regions belong together with the frontal lobe in one cluster and thus participate in motif 78 together with the frontal lobe, the posterior regions in the *alpha2* band do not participate in motif 78. For the *theta* band, the frontal and the occipital lobe apparently share many interactions in the form of motif 78 because the clustering algorithm does not split them. This strong higher-order interaction between posterior and frontal brain regions could relate to the previously described global pattern of information flow between frontal and posterior regions in the *theta* band [50, 54].

The non-participating regions in the *alpha2* band consist mainly of strong hubs in posterior brain regions, which in our constructed networks have no in-degree but a significant out-degree. These nodes cannot participate in any instance of motif 78 as they would need at least one incoming link. Thus, the previously described pattern of information flow from the posterior to the frontal regions in the *alpha2* band is more likely based on the strong sending links, and less on this particular motif. However, the high density of motif 78 in the frontal regions might still play a role for the integration of the received signals from the posterior regions.

3.5. Chapter Summary

The motif search for different frequency bands resulted in the dominant overexpression of motif 78 in networks generated over a wide range of thresholds. This motif, which was also observed in previous studies and our dPTE-based networks, seems to represent a general building block for the information flow in functional brain networks resembling the organizational principle of segregation and integration. The motif-based clustering revealed the higher-order organization of effective connectivity on a global scale. The differences between higher and lower frequency bands could be traced back to the interaction pattern between the posterior regions and the frontal regions. In the *theta* band, the frontal regions participated in many instances of motif 78 together with the posterior regions, pointing towards a strong integration of information flow between those spatially segregated areas. In the *alpha2* band, the posterior regions are no longer part of any cluster as they miss necessary bi-directional links to participate in motif 78, although the segregation between the frontal regions and the remainder of the brain is still observable.

4

Epidemics with Time-dependent Rates

Epidemic models like the SIS or SIR model enable us to describe simple spreading processes over networks but are often not sufficient to accurately capture more complex network dynamics as exhibited by sophisticated and malicious computer worms. Many of the common assumptions behind epidemic models do not necessary hold if the process under investigation spans big networks or large scales of time. We extend the standard SIS network model by dropping the assumption of a constant curing rate in favour of a time-dependent curing rate function, which enables us to reflect changes in the effectiveness of the active worm removal process over time. The resulting time-dependent mean-field SIS model allows us to study the evolution of the size of computer worm bot-nets. We exemplify the complete procedure, including data-processing, needed to obtain a reliable model on data from Conficker, an extremely resilient computer worm. Using empirical data obtained from the Conficker sinkhole, we fit long time periods of up to 6 years on multiple scales and different levels of noise.

This chapter is based on a published paper [97].

4.1. Introduction

Computer worms have become a plague in today's Internet. While their local mechanisms of propagation can be reverse engineered and are well understood, their global impact remains hard to estimate. Armed with the potential to spread indefinitely, not even the authors of the worms might be able to predict how many machines will end up compromised as part of a worm's botnet.

While mathematical models for epidemics can be applied to estimate the size of botnets over time [98], [99], many of them rely on strong assumptions which might not be fulfilled by the networks or the worms under consideration. This makes it especially hard to fit empirically obtained data from measurements with those models. First of all, most worms start spreading undiscovered and apply camouflaging techniques, so that data from the early infectious periods are often lacking. Since infected machines are subject to different up and down-times found in various environments, they might appear or disappear in networks at different points in time. Worms can also hibernate undetected on media like USB-sticks, possibly allowing them to reinfect even cleaned up machines. To complicate matters even more, the spread of computer worms is additionally influenced by humans that apply various sorts of counter-measures, like patches, blocking of certain IP address ranges or re-routing and filtering network traffic, sometimes in very disruptive manners.

All these complex behaviours were observable for one of the prime examples of a long-lasting continous battle against a maliciously growing computer worm: *Conficker* [100]. At its highest peak, Conficker was estimated to had infected over 9 million of Windows machines worldwide [101], creating one of the largest botnets in the history of the Internet. The command and control structure of this botnet was disrupted by using a sinkhole [102], a server that intercepted all calls from infected machine originally addressed to reach the bot-masters. The log-files of the sinkhole allow us to view Conficker's spread on very different levels of granularity by filtering the infected IP addresses by their autonomous systems (AS) and, for example, aggregating them again to the level of individual countries.

Especially on AS-level, the data can be noisy and sometimes exhibit unexpected patterns, as the worm was removed with varying effectiveness over time. To properly address this data, an extension of the traditional epidemic models is needed, which is able to describe the evolution of a worm over long periods of time. Our main contribution is to propose a new time-dependent mean-field SIS-model and apply it to the case of Conficker. In particular, this chapter is structured as follows:

- In Section 4.2, we describe some traditional epidemic models used for computer worms and propose our new and general time-dependent mean-field SIS-model, which takes the aspects of reinfection and applied countermeasures into account.
- Section 4.3 explains how the general model needs to be further adjusted to the specific case of Conficker, how the sinkhole data was processed and critically reviews the legitimacy of our underlying model assumptions when put into practice.
- Afterwards, we fit our model to the actual Conficker data in Section 4.4. We show that our model deals better with the inherent noise of the data by providing

high quality fits comparable to previously introduced models. For special cases in which the decline of the computer worm does not follow a strictly monotonically decreasing pattern, our model is still able to give a reasonable explanation of the data as it allows for changes in the effective worm removal process, which is not possible for monotonous models.

- Finally, Section 4.5 relates our model to previous work on computer worm research and Section 4.6 concludes with a discussion on possible applications of the new model and ideas on future research.

4.2. Epidemic Models

This section describes classical epidemic models, which we will later use as a reference, and introduces our main contribution: the time-dependent mean-field SIS-model.

4.2.1. The Population-based SIR-model

The population-based Susceptible-Infected-Removed (SIR) model, originally described by Kermak and McKendrick [103], describes a spreading process in a fully mixed population, for which the corresponding underlying network is the complete graph, in which each of the N individuals can be in one out of three possible compartments: I for *infected*, S for *susceptible* (to infection) or R for *removed*. The dynamics are described by the following set of differential equations

$$\frac{dS}{dt} = \frac{-\beta SI}{N}, \quad \frac{dI}{dt} = \frac{\beta SI}{N} - \delta I, \quad \frac{dR}{dt} = \delta I \tag{4.1}$$

where β is the infection rate and δ is the rate at which infected individuals are removed from the population. Both β and δ are assumed to be constant in classical SIR theory, in which case set (4.1) was already solved analytically [103] in 1927.

Individuals in SIR either stay in compartment S or make the transition

$$S \rightarrow I \rightarrow R.$$

Consequently, the number of susceptible hosts over time is always monotonically decreasing within this model. Similar to the SIS-model, which we introduce next, also the SIR model can be generalized to contact networks [104]. In this work however, we will use the simple original SIR model defined over fully mixed populations as a base-line for comparison with our more sophisticated time-dependent model.

4.2.2. The Network-based SIS-model

The network-based Susceptible-Infected-Susceptible (SIS) model [105–107] is a Markovian model that describes a spreading process with possible reinfection for an underlying contact network. Each node of the network can be in two possible compartments: I for *infected* or S for *susceptible* (to infection). A network of N nodes can thus be in 2^N different states.

Usually, two independent Poisson processes, each with constant rate, determine the transitions between these states. The *infection process* determines for each susceptible node its transition to the infected compartment dependent on the number of infected neighbors. A node can only become infected if it shares a link to an already infected node. Each infected neighbor contributes with an infection rate of β to the infection. The *curing process* determines for each infected node its transition from the infected to the susceptible compartment with a corresponding curing rate δ.

As the state-space of the SIS Markov model grows exponentially in N, computing the probabilities of infection per node becomes quickly intractable for large networks. Mean-field approximations [105, 106, 108] are a common technique to reduce the size of the governing equations and make them amenable for analytic solutions. The N-Intertwined mean-field approximation model [109] (NIMFA) is currently the best continuous-time, first-order mean-field approximation. Given a fixed network represented by an adjacency matrix A, in which each entry a_{ij} indicates the existence of a link between nodes i and j, NIMFA approximates the probability $v_i(t)$ that a node i is infected at a certain time t by

$$\frac{dv_i(t)}{dt} = -\delta v_i(t) + \beta(1 - v_i(t)) \sum_{j=1}^{N} a_{ij} v_j(t). \tag{4.2}$$

These equations can be solved to determine the steady-state infection probability $v_{i\infty}$ of each node i, where $\frac{dv_i(t)}{dt} = 0$ and from which the average steady-state fraction of infected nodes

$$y_\infty = \frac{1}{N} \sum_{i=1}^{N} v_{i\infty} \tag{4.3}$$

can be computed.

If an r-regular graph[1] is considered as the underlying contact network, the infection probability is the same for each node due to symmetry: $v_i(t) = v(t) = y(t)$. Thus, Equation (4.2) simplifies to

$$\frac{dy(t)}{dt} = \beta r y(t)(1 - y(t)) - \delta y(t) \tag{4.4}$$

The particular Equation (4.4) was studied by Kephard and White [110], who gave the solution

$$y(t) = \frac{y_0 y_\infty}{y_0 + (y_\infty - y_0)e^{-(\beta r - \delta)t}} \tag{4.5}$$

where the evolution of the fraction $y(t)$ of infected nodes is described by the initial fraction y_0 of infected nodes and the steady-state fraction of infected nodes

$$y_\infty = \lim_{t \to \infty} y(t).$$

While Equation (4.4) only holds for regular networks, it gives an excellent starting point for the development of a time-dependent model, as we will see in the next subsection.

[1] A graph G is r-regular if each node in G has degree r.

4.2.3. The Time-dependent Mean-field SIS-model

If the source for an infection or for curing is not constant, the fixed rates β and δ have to be transformed into functions $\beta(t)$ and $\delta(t)$ to describe the rates for the infection and curing processes at any time t. The time-dependent extension of Equation (4.4) is

$$\frac{dy(t)}{dt} = \beta(t) r y(t)(1-y(t)) - \delta(t) y(t). \tag{4.6}$$

While the exact Markovian SIS dynamics seem to be impossible to solve for time-dependent rates, even for highly symmetric cases as the complete graph, Van Mieghem [111] shows that the differential equation (4.6) can be solved exactly to determine the evolution of the fraction of infected nodes $y(t)$ over time by

$$y(t) = \frac{\exp\left(\int_0^t (r\beta(u) - \delta(u))du\right)}{\frac{1}{y_0} + r\int_0^t \beta(s)\exp\left(\int_0^s (r\beta(u) - \delta(u))du\right)ds}. \tag{4.7}$$

A convenient short-hand is to define the net dose as

$$\rho(t) = \int_0^t (r\beta(u) - \delta(u))du \tag{4.8}$$

which equals the net average number of infections reduced by all curings in a time interval $[0,t]$ for a particular node in the r-regular graph. Using the net dose (4.8), Equation (4.7) becomes

$$y(t) = \frac{e^{\rho(t)}}{\frac{1}{y_0} + r\int_0^t \beta(s)e^{\rho(s)}ds}. \tag{4.9}$$

The main quantities in (4.9) are the degree of the regular network r, the initial fraction of infected nodes y_0, the time-dependent infection rate function $\beta(t)$ and the curing rate function $\delta(t)$. We describe in subsection 4.4.1 how the parameters $\{r, y_0\}$ and the functions $\{\beta(t), \delta(t)\}$ can be determined to match the infection curve of the Conficker worm. For the remainder, we will refer to this model in short as the time-dependent SIS-model.

4.3. Methodology

This section outlines the necessary steps before applying any epidemic model (like the ones introduced before) to measured data from a sinkhole. We will first describe the Conficker data, how it was processed and why we think that the time-dependent SIS-model is a reasonable choice to describe the propagation of the worm.

4.3.1. Datasets

All data of the Conficker worm is based on logfiles from the sinkhole. The sinkhole was used to disrupt the update mechanism of Conficker, which connected to 250 pseudo-randomly generated URLs in order to get payload (i.e. instructions, malware or new functionality) from its original authors. By registering the domain names before the botmasters, and redirecting every access to a central server (the *sinkhole*), the worm was effectively cut off from its authors.

There were some partially successful attempts of the botmasters to regain control over the Conficker botnet, made possible due to mistakes during the sinkholing process. This resulted in new variants of the worm, which employed a more sophisticated update mechanism. However, after April 2009 the botnet remained under control. From this perspective, Conficker provides an interesting case study of the propagation of an unaltered computer worm over a reasonably long period of time.

In total, the sinkhole logfiles provide us data from February 2009 to September 2014 and contain over 178 million unique IP addresses. With the help of GeoIP-databases [112] and IP-to-ASN-lookup [113], these IP addresses were associated with the corresponding ISO country code and autonomous system (AS). Thus, the data can be viewed at different levels of granularity:

- **global:** all unique IPs for the complete sinkhole worldwide
- **country:** all unique IPs belonging to a specific ISO country code
- **autonomous system:** all unique IPs belonging to a specific AS

In total, the IPs belong to 241 different ISO country codes and to over 34.000 different autonomous systems.

4.3.2. Preprocessing

Botnet size estimation Accurate estimations of the amount of infected machines is a difficult problem (see Abu Rajab et al. [114]), as long as our only way for identification of a machine is via its IP address. On the one hand, it is possible to *undercount* because multiple infected machines might share a common IP address due to Network address translation (NAT). On the other hand, a single infected machine might be represented by multiple IP addresses due to different ISP policies. To avoid this *overcounting*, the number of IP-addresses needs to be corrected by a DHCP-churn rate, which varies over countries and ISP. Determining accurate DCHP-churn rates is a challenge in itself (see Moura et al. [115]), which we will not undertake here.

Instead, we aggregate the unique IP-addresses over short time slots of one hour. We consider the DHCP-churn effect on this time scale to be minimal. Using short time slots introduces another source of undercouting because not every infected machine might be contacting the sinkhole every hour (for example they might not be powered). The hourly values are then averaged out over a time slot of a week, eliminating biases introduced by diurnal patterns. Accordingly, together with the NAT-effects, our estimate of the infected machines should be considered as a lower bound.

Data cleansing While analyzing the sinkhole data, missing measurements become apparent as there are several periods ranging from a few hours to a few days in which the number of IPs drops down to zero. We account technical malfunctions of the infrastructure (i.e. downtime of the sinkhole) for these artifacts. Consequently, we remove these outliers before applying any model fitting by the following procedure: for each datapoint z of week w, we compute the difference between z and the median of all datapoints in a time window spanning ±10 weeks from w. From all datapoints, we exclude the 10% with the highest differences. This procedure does not remove all outliers for all cases, but reduces their impact on the fitting procedure considerably.

Normalization In order to apply the time-dependent epidemic model (4.6), the data need to be normalized, because $y(t)$ describes the average fraction of infected nodes and not the number of infected nodes in the networks. An accurate normalization would use the amount of vulnerable machines, which is not known to us. In fact, the Conficker worm spreads only in unpatched versions of all major Microsoft Windows versions up to Windows Vista and Server 2008, for which we have not found reliable estimates. Instead, we use the *peak point* of infection to generate a relative scaling. After the aggregation and data cleansing, we determine the maximum number p_{max} of infections over the whole infectious period and compute the scaling factor s_y by

$$s_y = \frac{k}{p_{max}} \tag{4.10}$$

where k is a real number between 0 and 1. For our fitting procedure, a value of $k = 0.9$ proved to be sufficient and was used if not stated otherwise. The scaling factor s_y can be used to fit the original data as we will discuss in Subsection 4.4.1.

We use a bin size of one week to count the unique IPs, resulting in 280 bins for the complete infectious period. This period is linearly transformed on the horizontal axis so that the starting point of the infectious period maps to 0 and the end point maps to 1.

4.3.3. MODEL ASSUMPTIONS

The dynamics of the Conficker-spread are heterogeneous, because different infection vectors are invoked to infect new machines and networks. For example, a person might obtain the virus by plugging out his USB-stick from an infected computer. Much like an actual biological disease, this person could traverse large amounts of space and time before he triggers a new infection with his USB-stick on a different machine. This and similar effects make the construction of an actual (dynamic) contact network impossible. Considering the extremely large scale of the Internet and the long period of time (6 years), it seems completely unreasonable to assume that any (simple) model would be able to reflect this degree of complexity.

However, the time-dependent mean-field SIS-model (4.6) can be refined to capture the basic observable infection patterns. In order to justify the refined model, we review the basic assumptions and argue to which extent they are adequate in the case of Conficker.

Constant spreading rate As Conficker was disconnected from its authors by the sinkhole, its code remained largely constant for the whole infectious period. There exist some updated versions of Conficker (named Conficker C, D and E) at the very beginning of the logged infectious period (up to April 2009), but these updates were used to improve the command and control structure of the virus and to add a scareware payload to it. The main infection vectors (NetBIOS vulnerability, USB-sticks and Shared Folders) remained largely unaffected by these changes. It is thus safe to assume, that the spreading rate of Conficker remained constant. For this reason, we will set $\beta(t) = \beta$ in Equation (4.6).

Time-dependent curing rate function Contrary to the infection rate $\beta(t)$, we assume that the curing and removal forces were not constant. In general, the clean-up of Conficker was regarded to be rather involved as the worm possessed several countermeasures. The curing rate function $\delta(t)$ of our model (4.6) reflects the combined effort that was spent to fight Conficker, i.e. by patching the vulnerability, use of removal tools and also the replacement of infected machines. As some of the countermeasures did not provide complete immunity, reinfection with Conficker was possible and is well documented [100]. This effect is reflected by the basic SIS dynamics. Next to SIS, our model is able to simulate removal as in the SIR-model [106] by increasing the curing rate of a node. Once a node's curing rate is very high, its infectious periods become very small, which can be regarded as a removal or immunization effect. Although our model (4.6) never *explicitly* removes nodes from the network, the time-dependent curing rate blends both SIS and SIR-dynamics and thus captures effects like permanent removal of machines or an acquired immunity, for example by system upgrades without excluding reinfection dynamics like SIR.

Network topology The Equation (4.6) of the time-dependent mean-field SIS-model demands an underlying and constant contact network of degree r. This is a necessity from a model point of view, as computation would quickly become intractable otherwise. While the Internet is clearly not constant in its size, we justify the regularity assumption by one of the infection vectors of Conficker. The technical reports (see Porras et al. [116]) suggest that Conficker used a scan-and-infect subroutine that occasionally scanned random IP-addresses[2] for new victims. The worm did not flood the complete IP-space but concealed itself by connecting only to a limited amount of possible new victims. Thus, for the fixed allocated time-slots that we investigate, we assume that there is an upper bound on the possible scan-attempts for an infected machine, which is independent of the configuration or network properties (e.g. bandwidth). This upper bound translates into an estimate on the degree of the underlying contact network, which is our parameter r.

4.4. Model Application

4.4.1. Modeling the Spread of Conficker

The key to a good epidemic model of Conficker is to determine the time-varying parts of Equation (4.9), namely the spreading rate function $\beta(t)$ and the curing rate function $\delta(t)$. As argued before, we assume $\beta(t) = \beta$ as constant, so that the net dose $\rho(t)$ in (4.8) simplifies to

$$\rho(t) = r\beta t - D(t) \tag{4.11}$$

[2] Due to a bug in the pseudo-random number generator of Conficker, only one fourth of the complete IP-address space was susceptible for this attack vector.

where $D(t) = \int\limits_0^t \delta(u)du$ is the *accumulated curing dose*. If we assume $D(t)$ to be an analytic function, there exists a Taylor series that allows us to express $D(t)$ precisely. The Taylor expansion is truncated after d terms to retrieve a polynomial approximation

$$D(t) = \int\limits_0^t \delta(u)du \approx \sum_{i=0}^{d} a_i t^i \tag{4.12}$$

with $a_d \neq 0$. We use the last two equations to transform Equation (4.9) as

$$\begin{aligned} y(t) \quad &\approx \quad \frac{\exp\left(r\beta t - \sum\limits_{i=0}^{d} a_i t^i\right)}{\frac{1}{y_0} + r\displaystyle\int_0^t \beta \exp\left(r\beta s - \sum_{i=0}^{d} a_i s^i\right) ds} && (4.13)\\ &= \quad \frac{\exp(-a_0)\cdot\exp\left(r\beta t - \sum\limits_{i=1}^{d} a_i t^i\right)}{\frac{1}{y_0} + \exp(-a_0)\cdot r\beta\displaystyle\int_0^t \exp\left(r\beta s - \sum_{i=1}^{d} a_i s^i\right) ds} && (4.14)\\ &= \quad \frac{\exp\left(r\beta t - \sum\limits_{i=1}^{d} a_i t^i\right)}{\frac{\exp(a_0)}{y_0} + r\beta\displaystyle\int_0^t \exp\left(r\beta s - \sum_{i=1}^{d} a_i s^i\right) ds}. && (4.15) \end{aligned}$$

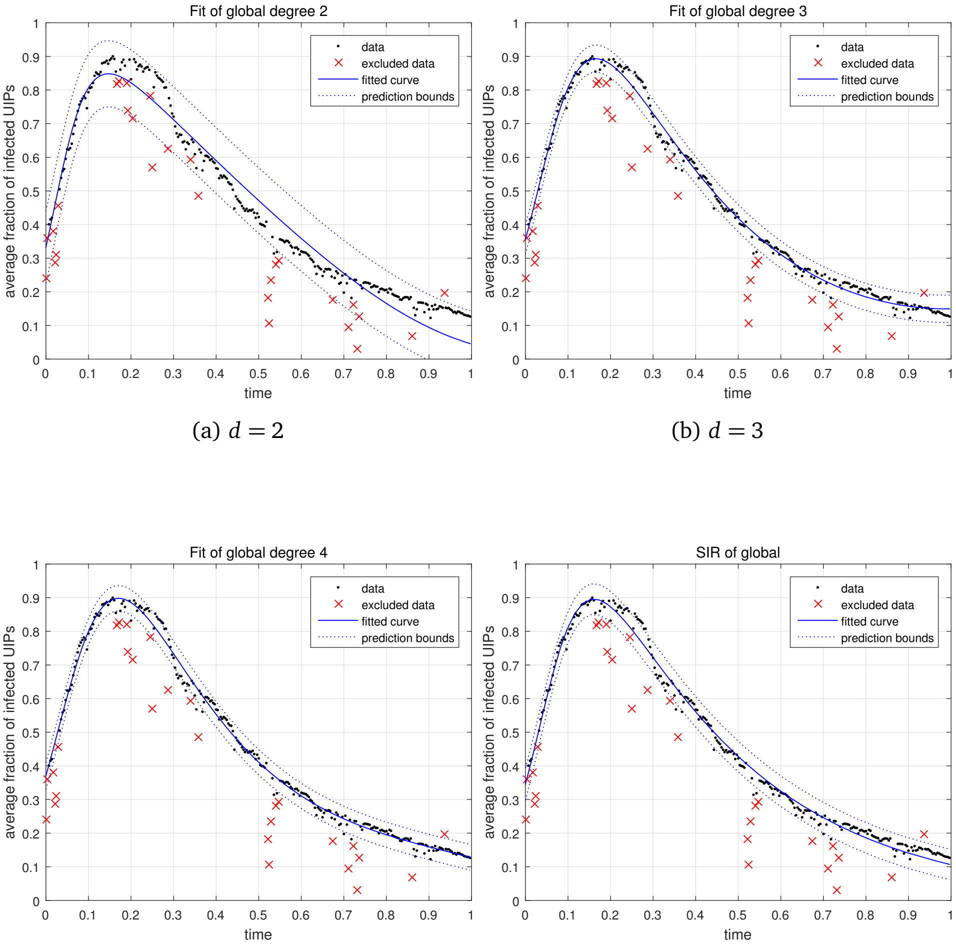
We define the products $y_0^a = y_0 \cdot e^{-a_0}$ and $\beta_r = r\beta$ to simplify the model further. Thus, the free parameters of our model are y_0^a, β_r and $a_d, \ldots, a_1$. In order to apply the model to absolute values rather than fractions, we apply the scaling-factor from Equation (4.10):

$$y(t) = \frac{e^{\widetilde{\rho(t)}}}{s_y\left(\frac{1}{y_0^a} + \beta_r \int_0^t e^{\widetilde{\rho(s)}} ds\right)} \quad \text{with} \quad \widetilde{\rho(t)} = \beta_r t - \sum_{i=1}^{d} a_i t^i. \tag{4.16}$$

Since we are interested in the dynamics and not in absolute values, we will set $s_y = 1$ for simplicity. Consequently, in all the Figures showing model fittings, the maximum value will always be found at $y = 0.9$ (recall Section 4.3.2, Normalization). In summary, the model is given by

$$y(t) = \frac{e^{\widetilde{\rho(t)}}}{\frac{1}{y_0^a} + \beta_r \int_0^t e^{\widetilde{\rho(s)}} ds} \quad \text{with} \quad \widetilde{\rho(t)} = \beta_r t - \sum_{i=1}^{d} a_i t^i. \tag{4.17}$$

Given the preprocessed and cleansed data as input, the trust-region non-linear least squares method provided by MATLAB is used to fit our model. The initial values for



(a) $d = 2$

(b) $d = 3$

(c) $d = 4$

(d) SIR

Figure 4.1: Global-level fitting of Conficker in time-dependent SIS and SIR model. Dotted curves are the 95% prediction bound

Table 4.1: Fitting parameters for global model on Conficker, 95% Confidence bounds in parenthesis.

d	**2**	**3**	**4**
β_r	23.5 (19.55, 27.45)	12.03 (10.98, 13.08)	7.304 (5.656, 8.952)
v_0^a	0.333 (0.287, 0.379)	0.3554 (0.3396, 0.3712)	0.3703 (0.3556, 0.385)
a_1	-1.17 (-1.64, -0.692)	-5.572 (-5.813, -5.331)	-7.597 (-8.23, -6.964)
a_2	16.06 (14.1, 18.02)	23.29 (22.48, 24.11)	31.05 (28.61, 33.49)
a_3	-	-10.33 (-10.79, -9.863)	-28.07 (-33.43, -22.7)
a_4	-	-	9.646 (6.713, 12.58)

all parameters were picked randomly between zero and one. Then, the polynomial of degree 3 was fitted first. The parameters found for this fit were used as initial guesses to fit the polynomial of degree 4, which significantly improved the convergence speed. The same procedure was applied to create the fit for degree 5, based on the fit of degree 4.

4.4.2. Quality of Fits at Global Scale

The degree of freedom of the proposed model for Conficker is $2 + d$, where d is the order of the polynomial curing rate function (4.12). Thus, we are able to trade the complexity of the model with its accuracy. In a first experiment, we examine which value of d is useful for modelling the underlying data. We use the global aggregation of all virus infections as input, because it has less noise than data from country- or ASN-level granularity. Figure 4.1 shows how the quality of the fit and of the prediction bounds improves by using higher degree polynomials. The adjusted R^2-value is 0.96 for $d = 2$ and 0.99 for $d \geq 3$, indicating a good fit overall. However, a visual inspection reveals that the fit for $d = 2$ is not good enough to accurately describe the decline of the virus. On the other hand, $d = 4$ does not provide much more quality, but requires an additional fitting parameter. Thus, guided by parsimony as a modelling rule, we believe that $d = 3$ is a good choice for this particular case. Table 4.1 gives the actual values of the fitted parameters.

Fixing $d = 3$ results in 5 parameters that need to be determined by the fitting procedure. This is one parameter more than we would use to fit the population-based SIR-model, which is also shown in Figure 4.1d for comparison. More specifically, the parameters used for the SIR-model are the size of the population N, the constant infection rate β, the constant curing rate δ and the initial number of infected individuals v_0. As the adjusted R^2-value for SIR is with 0.99 qualitatively very high and on the same level as for $d = 3$, SIR is an even more parsimonious model that works very well on global scale.

4.4.3. Quality of Fits at Subglobal Scales

Moving from global-level to country-level, we have more noise and variation in the data since not all countries were affected by Conficker in the same way. Out of the 241 different ISO countries, we picked a subset of 40 countries (belonging to the OECD and the European Union) to analyse. We used again SIR as a baseline and compared it with fits of order $d = 3, 4$ and 5. We chose the adjusted R^2-value and the sum of squared errors (SSE) as indicators for the quality of fit. Figure 4.2 shows the distribution of those indicators for the 40 chosen countries sorted by their corresponding quality in the SIR-model. The fits of the Conficker-model are of high quality and only in 3 cases worse than the corresponding SIR-model. We believe that the fitting procedure converged sub-optimally in those cases, as 7 parameters needed to be determined, while lower order fits were still better than SIR or equally good. A visual inspection of the fits showed that the time-dependent SIS-model is able to fit the tail better than SIR. The latter is forced to monotonically decline in this area while the time-dependent SIS-model can better adapt to nearly constant viral levels which are observable in the tails of the data for some of the countries.

Moving down to the ASN-level imposes a bigger challenge, since the number of infected IPs becomes so low that the influence of noise grows more significant. To circumvent this problem, we selected the 30 ASNs with the highest number of infected unique IPs out of 34000 available for our analysis. Still, the data for some of these ASNs is considerably more distorted than any country-level data. While gaps of missing data on country-level usually span a couple of weeks, they can spent months or even years for some of the ASN datasets. We expect that some sort of ISP-wide countermeasures were used to prevent infected machines during those times to connect to the sinkhole, though we did not find evidence for this claim.

The data-cleansing procedure is not sufficient to remove all outliers completely, so they inevitably impact the quality of the models. Figure 4.3 shows the results in the same way as we did for the country-level. Some of the fits are not visible as they are of so low quality ($R^2 < 0.9$ or SSE > 2) that they are outside our scale. We ordered all data after the quality of the SIR-fit nevertheless. The variation in quality is much higher than it was for the country-level: 4 out of 30 ASNs were so degenerated that every model produced only poor fits. Similar to the country-level, we observe the time-dependent SIS-model to be better than SIR, though with a higher relative qualitative difference than on country-level. This is not unexpected as models with more degrees of freedom adapt more easily to noise in general. A visual inspection of the fits on the ASN-level showed that in not degenerated cases (i.e. high *jitter* or very large gaps) the models still give a fairly good description of the spreading pattern.

4.4.4. Determining the Effectiveness of Worm Removal

While fitting the absolute or relative number of infected IPs gives a comprehensive overview about the prevalence $y(t)$ of Conficker, the time-dependent SIS-model allows for an additional perspective. To gain more insight in the worm removal, we look closer to the curing rate function $\delta(t)$ and understand, what it actually means. While changes in the curing rate can be easily interpreted as changes in the applied countermeasures against the worm, we have to keep in mind that the most effective countermeasures were already in place *before* the sinkhole recorded its data. More precisely, the NetBIOS vulnerability was patched very fast in November 2009, 4 months before our data collection starts. However, Conficker was remarkably resilient, raising the question why the worm could survive despite the patch for years?

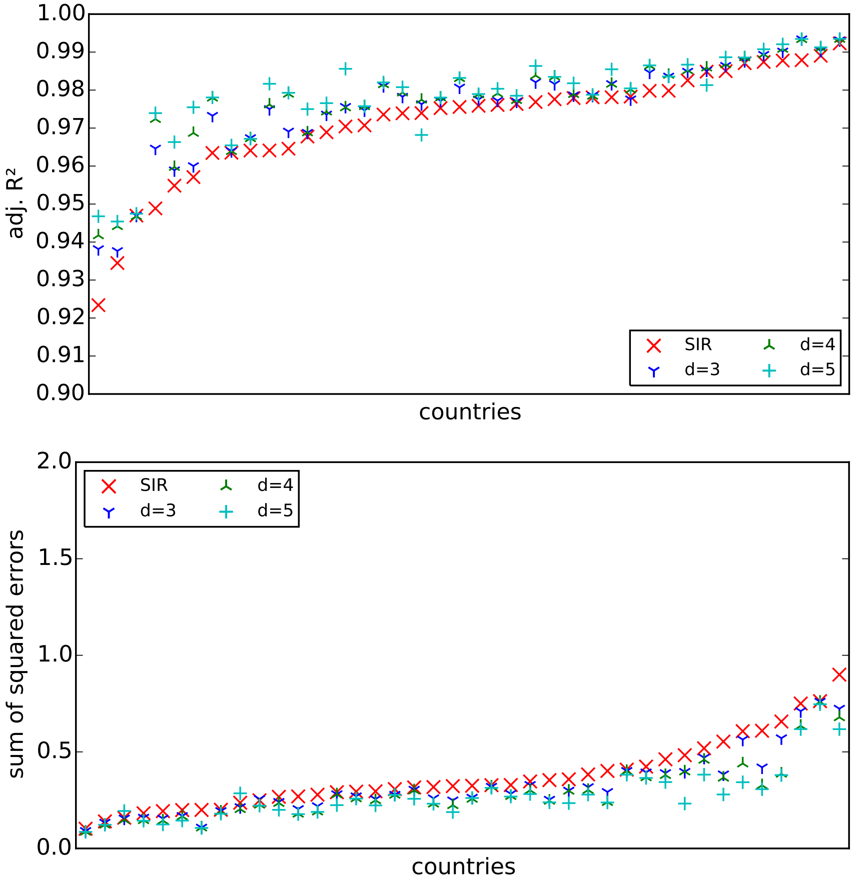


Figure 4.2: Overview of Country-level fitting quality, ordered by SIR. On top: adjusted R^2 value, on the bottom sum of squared errors.

If we assume that the slow decline of the worm is, to a large amount, not caused by security patches but by long-term effects like failure or substitution of infected machines with newer ones, it seems reasonable that the population-based SIR-model gives a fairly good description of the process. By looking at the curing rate function $\delta(t)$ and the accumulated curing dose $D(t)$ obtained by the fits of the time-dependent SIS-model, we can analyse at which periods in time the removal of the worm changed. We call the time pattern given by $\delta(t)$ the *effective worm removal*.

Figure 4.4 shows $\delta(t)$ and $D(t)$ obtained by the fits on the global aggregation level. It is interesting to see that $\delta(t)$ seems to approach a sigmoidal shape in the interval $[0, 1]$ once we increase the degree of the polynomial. A sigmoidal effective worm removal starts very low, describing a time-period in which the worm is persistent and spreads unhampered. However, this is followed by a sharp increase in curing rate, which leads to a rapid decline in the virus prevalence. Finally, the sigmoidal shape reaches a saturation

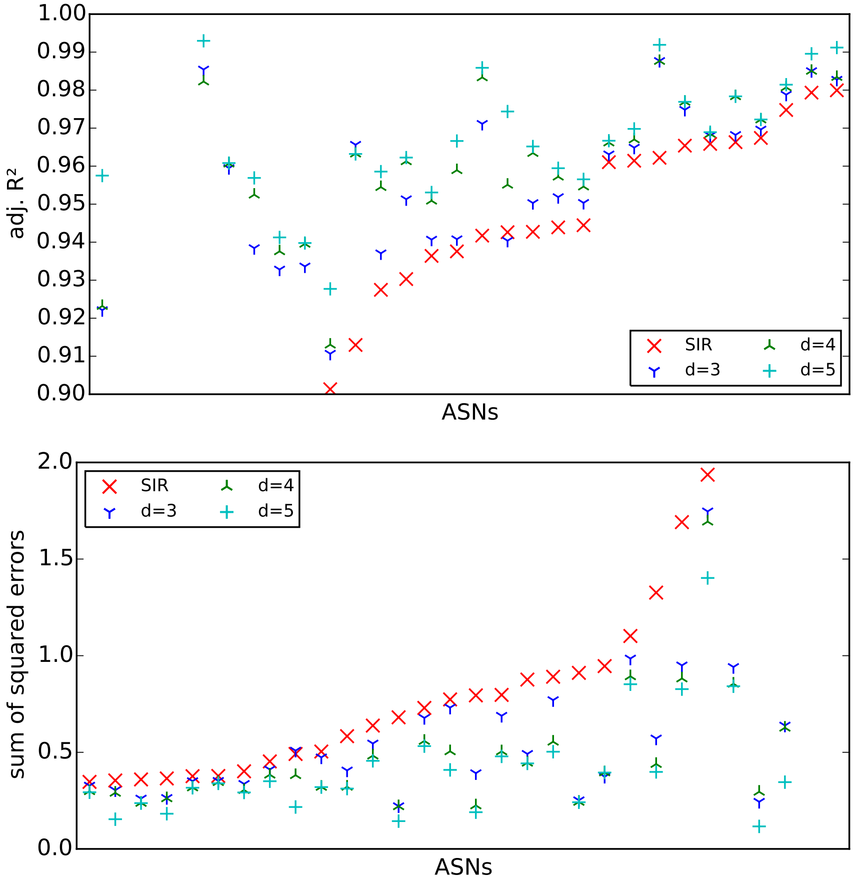


Figure 4.3: Overview of ASN-level fitting quality, ordered by SIR. On top: adjusted R^2 value, on the bottom sum of squared errors.

of high curing rates, which explain the low levels of persistence in the later phases of the worm evolution and the long period of final decline.

4.4.5. Sensitivity of Parameters

For $d = 3$, the time-dependent SIS-model is determined by 5 parameters: a_3, a_2, a_1, β_r and y_0^a. Since $y(0) = y_0^a$, the meaning of y_0^a is clear: it defines the initial fraction of infected IPs and thus the starting point for the spreading process. However, it is not obvious how the course of the infection is influenced for $t > 0$ by a_3, a_2, a_1 and β_r? To investigate this further, we collected all values that occurred for those parameters while fitting each country-level dataset. To avoid outliers, we computed for each parameter the 10% and 90%-percentile. The range between both values was divided linearly so that we end up with 5 evenly spread out values for each parameter, which are representative for the fits. We used the median of those 5 values to define a reference curve and adjusted

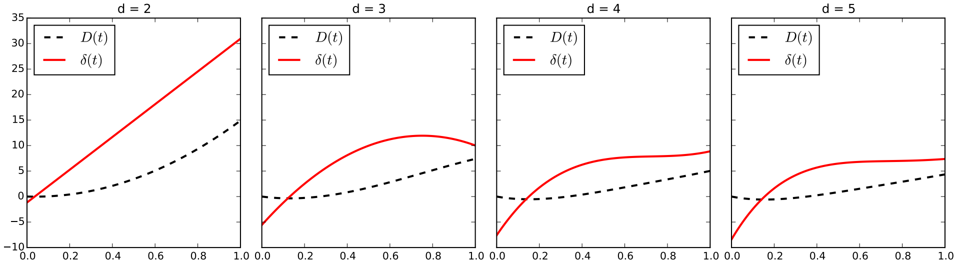


Figure 4.4: $\delta(t)$ and $D(t)$ for global-level fitting with polynomial degrees 2,3,4 and 5.

each parameter separately to see how sensitive the model is to changes. Results are shown in Figure 4.5.

The coefficients a_3, a_2, a_1 have dominant influence on different phases, with a_3 being dominant at the later stages, a_2 in the middle and a_1 at the beginning. The curvature of the infection itself is strongly influenced by β_r, which regulates the height of the peak and the decline of infection. During the fitting procedure, these 4 parameters are balanced out against each other. For example: increasing β_r makes for a much flatter decline in virus prevalence, but also moves the peak higher. If the data suggests a flat decline but a low peak, β_r should be high but also with a high a_1 to correct the peak.

An intriguing property of the time-dependent SIS-model is the fact that it is not monotonous unlike the SIR model: by decreasing a_3 or increasing a_2, it is possible to have an increase in virus prevalence *after* the maximum peak. While this behaviour is - for the case of Conficker - not observable on country-level, there are some rare occurrences on AS-level that suggest such a behaviour may occur in practice. Figure 4.6 shows with AS8452 such an example and shows the corresponding fits of the non-monotonous time-dependent SIS model in comparison to the monotonous SIR model.

4.5. Related Work

In a previous work by Asghari et al. [117], the same dataset of the Conficker worm was analyzed to show the effectiveness of anti-bot net campaigns in different countries. The focus of our current work is different: we develop a very general epidemic model to describe time-dependent propagation and use the Conficker data as an example to show the applicability of such an approach. In contrast, the previous work started with the data of Conficker and developed a non-epidemic descriptive model to extract features of the worm prevalence on country-level. Those features were then correlated with different institutional factors (broadband access, operating system market shares, software piracy, etc.) to explain regional difference between the countries.

Epidemic models have already been investigated for describing spread of computer worms [99, 118–120], mostly by extending SIS or SIR-models. A frequent approach is addition of new *compartments* (see Faryad et al. for a general multi-compartmental framework [121]) in order to describe different states of the machines under consideration, for example SEIR, SIRS, SEIRD (see Peng et al. [122]). We avoid a comparison with those model as we lack of reliable empirical data for additional compartments. Of-

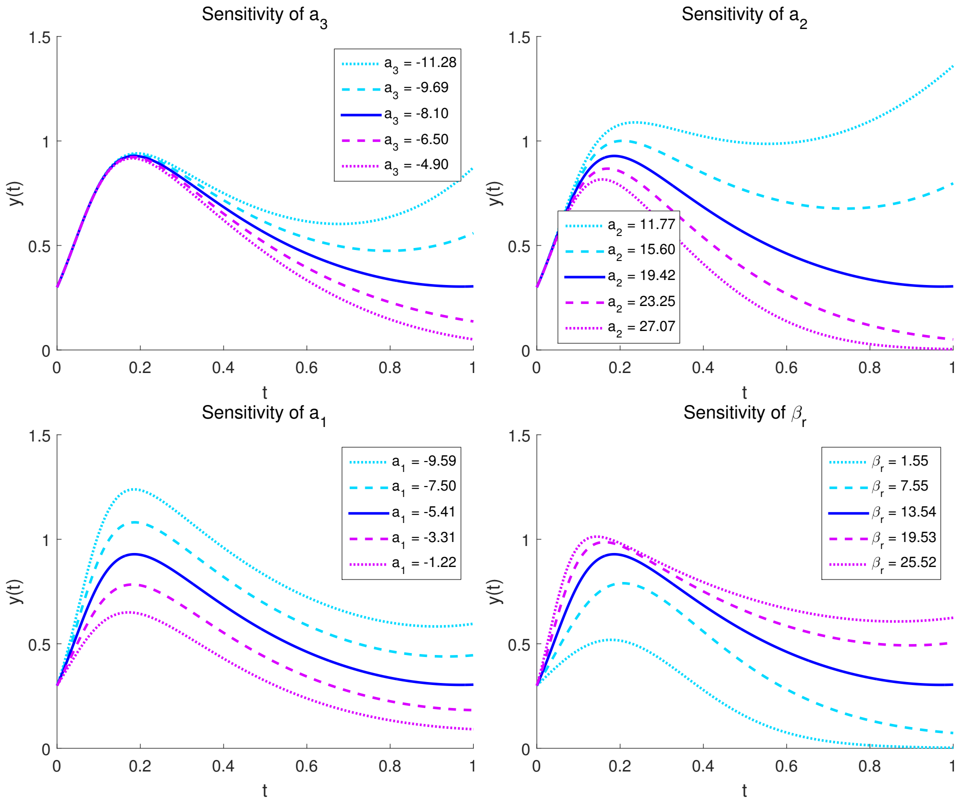


Figure 4.5: Influence of the polynomial coefficients a_3, a_2, a_1 and β_r on the virus prevalence. The solid curve corresponds to $a_1 = -5.41$, $a_2 = 19.42$, $a_3 = -8.10$ and $\beta_r = 13.54$ in each subfigure and is used as a reference to study the change of its shape by adjusting each parameter separately.

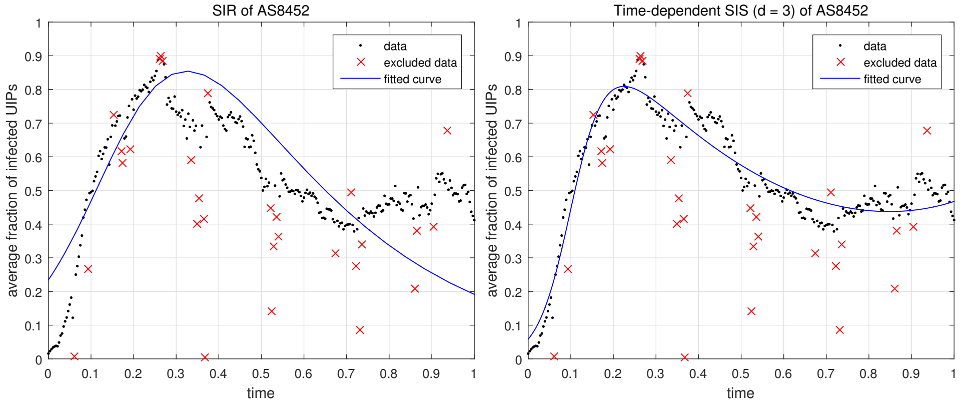


Figure 4.6: Left: The SIR model has difficulties to fit the noisy and non-monotonous data from AS 8452. Right: The time-dependent SIS model with $d = 3$ already provides more flexibility to address those issues.

ten, only the infected and the susceptible compartment can be measured or estimated without imposing further assumptions. Our model is only based on the susceptible and infected compartment.

We also do not compare with models that emphasize strongly on topology like the WPM model by Peng et al. [123], which uses 2d-cellular automatons to describe malware propagation, which is based on a different set of assumptions.

Zou et al. [124] introduce a two-factor worm model, which fits the propagation pattern of the Code Red computer worm well. Interestingly, a time-dependent infection process $\beta(t)$ was introduced for similar reasons as our curing rate function $\delta(t)$. It was argued that the massive spreading of Code Red congested buffers of Internet routers which resulted in a decline in infection rate. This assumption is unlikely to be valid for more subtle and hideous viruses like Conficker, that spread over years, while Code Red was literally *exploding* within a day. Moreover, the two-factor worm model needs 4 different compartments and 6 free parameters, which make the actual fitting for this model much more involved than for the time-dependent SIS and the simple SIR model. As the two-factor worm model is an extension of SIR, it also suffers from the drawback of being monotonous, despite its change in infection rate.

Zhang et al. [125] extend the basic SIR model to a hybrid epidemic model, which is evaluated on data from Conficker as well. Their model considers multiple ways of infection (globally and locally) and thus does not assume - like our and many other works - that Conficker only spread via its scan-and-infect subroutine. However, the counter-measures and the curing was not in their focus, as no change in curing rate was incorporated. In general, we do not compare ourselves with hybrid epidemic models, though our model could be used as part of one in order to describe one possible way of infection based on scan-and-infect behaviour of computer worms.

4.6. Chapter Summary

In this chapter, we showed how epidemic models can be designed to describe the complex propagation and decline patterns of long-lasting computer worms with Conficker as a showcase. While the assumptions on the spreading process for the SIR model are very simplistic and debatable for the case of Conficker, this simple model still provides a surpringsly good approximation of the worms propagation pattern. We consider SIR a good choice if the computer worm can be regarded as not changing over long periods of time, if the data is smooth and a clean monotonous decline is expected. If any of these conditions are violated, the time-dependent mean-field SIS model should be preferred, since it allows for more flexibility by adapting the degree of the polynomial curing rate function to increase the quality of the model fit. It is also possible to exchange the polynomial with *any* integrable function that might represent the effective worm removal over time.

If we assume that future worms will have the ability to adapt (for example by some evolutionary process), they might also develop an immunity against certain counter-measures in which case a non-monotonous curing rate function could be used to model this behavior. Although we assumed that Conficker was not changing its spreading rate over time, it is possible to apply the time-dependent SIS model also for worms for which this assumption is not true. In this case, one would need to define a time-dependent infection rate function $\beta(t)$ analogously to $\delta(t)$.

From a theoretical perspective, the fact that both the SIR and the time-dependent SIS model can fit the same data is surprising as well and might suggest SIR being a special case of SIS with time-dependent curing or infection rate functions. Possible future work could investigate the relationship between both models. Because of the monotonicity of SIR it is clear that not every propagation pattern of the time-dependent SIS model can be reproduced by SIR. However, it might be possible that there exists a mapping from the parameters of the SIR model to the parameters of the time-dependent SIS model in a way that the SIR propagation patterns can be (approximately) reproduced.

On a more practical level, our work could provide some building blocks for the development of novel worm tracking systems that would monitor the current effective removal over time. There might also be suitable applications outside the domain of computer worms for which a time-dependent epidemic model like ours could be applicable. Examples could include diffusion of technologies, spread of memes, or fighting darknet websites – where various measures and counter-measures over time influence the diffusion pattern. While very different scenarios, it would be interesting to see whether network epidemic models are still applicable or not.

5

Epidemics of Superinfection

We introduce an extension of the SIS epidemic model that describes infection, mutation and curing for a whole hierarchy of viruses, resembling a nested spreading process. In our model, high level viruses are only allowed to spread to nodes that have acquired a lower level of infection before. The simplest case of two viruses, in which one "superinfects" the other, shows already rich dynamics that are difficult to predict by common mean-field approximation techniques in certain cases. We derive an exact Markovian description for superinfection in the complete network and the star network showing that the steady state of the epidemic process is highly sensitive to the spreading rate of both viruses. Taking the spreading rates into account, we outline conditions for epidemic outbreaks, coexistence of both viruses and extinction cycles.

This chapter is based on a published paper [126].

5.1. Introduction

Epidemic spreading in networks is a process that models the propagation of information, the spread of rumors, the contagion with diseases and similar phenomena. SIS-like models [105, 106] are amenable to analytical understanding and enable us to describe and study effects like endemic outbreaks and extinction of viruses. We interpret a virus as a mere metaphor for the entity or property that is transferred from one node to others in the network.

Superinfection is a medically inspired term that originally described the process of infecting an already infected individual with a second, usually more severe virus. This can be due to a weakness of the immune system caused by the first virus that allows for the infection of the second virus, which in return will dominate the first one. Hence, superinfection can be regarded as an interaction of competing viruses, in which the second virus is dominant, but also strongly dependent on the (pre)-infection of the first virus.

The interactions between HIV and the Herpex Simplex Virus type 2 (HSV-2) are a well known medical example of superinfection: Acquisition of HSV-2 significantly increases the chances of getting superinfected with HIV and to transmit HIV to others [127].

Apart from the medical context, we see superinfection as an embedded epidemic spreading process. This view is applicable if two conditions hold: first, there needs to be a pre-condition for the infection with the dominant virus and second, the pre-condition itself spreads like a virus.

We expect that several applications can be better understood if described as a superinfection. One example is marketing: The knowledge about features (or the mere existence) of a product is a pre-condition for people to make a buying decision. This knowledge spreads via advertisement in social networks, increasing brand-awareness, but not necessarily inducing a buying decision for each infected individual. However, once several persons buy the actual product, first-hand experiences start spreading via self-written reviews over social networks as well. A possible consequence is that people pre-disposed by advertisement will now make a buying decision and start to actively promote the product as well.

A different use case is observed in cybersecurity, in which a security vulnerability is the precondition to acquire a piece of malware like a computer worm. Sometimes, even a computer worm can be a regarded as a precondition to become infected by another worm. For example, the Blaster computer worm [128] was able to spread in late 2003 to over 100.000 Windows machines, where it caused significant damage. Shortly after Blaster, another computer worm named Welchia [129] appeared. It spread via the same system vulnerability as Blaster, but was designed to eliminate Blaster and then to patch the whole system. While most-likely released with best intentions, Welchia ultimately might have caused more damage than Blaster, which shows how easy the interactions of computer worms are misjudged.

While Welchia was not a full success, the general idea of *active defense* provides a promising alternative to traditional countermeasures that rely on classical anti-virus software only. Instead of deploying a resource-hungry scanner on each machine on the network, a network operator might rather design an *antiworm* as an epidemic control unit, which allows the removal of undesired software. This worm would only spread to infected machines and would eventually restore these systems back to a healthy state once protection is no longer needed.

In order to realize these applications, this chapter introduces the fundamental theory of superinfection by the use of epidemic models. After discussing related work in Section 5.2, we describe the standard SIS-model and our model of superinfection in Section 5.3. In Section 5.4, we show that the interaction of 2 viruses is already so complex that standard analysis techniques like mean-field approximation produce deviations. In order to understand the interactions between the viruses and their conditions for spreading and survival, we analyze the exact Markov process for small complete networks and small star networks in Section 5.5 in order to conclude in Section 5.6.

5.2. Related Work

Nowak and May are among the first authors who focused on modeling superinfection in their pioneering work [130]. Their research is motivated by parasite evolution for which they provide a simple model based on an immigration-death process. Similar to our work, a virus hierarchy is introduced in which stronger viruses dominate less virulent strains. In particular, only the strongest virus is considered active, which means that it is the only virus that spreads. In contrast to our work, mixed populations are assumed and no underlying contact network is taken into account.

The case where multiple viruses are active is known as *coinfection* and was also explored and modeled by Nowak and May [131]. Mosquera et al.[132] show that superinfection can be a limit case for a coinfection process and give conditions for coexistence of multiple viruses. Also this work does not consider different contact networks and assumes fully mixed populations instead.

Multiple other works exist that investigate the existence and interaction of multiple viruses in the setting of competing viruses [133–136], in which infection with one virus provides immunity to the other (*cross-immunity*).

Newman and Ferrario [137] study an SIR-like epidemic model, where the infection with one virus is a prerequisite for the infection with a second virus. While they call their model a coinfection, it fits our definition of a superinfection very well. The authors use a general configuration network model and evaluate their spreading process on two examples: one network with a Poisson degree distribution and one with a power-law degree distribution. Their model is different from ours as they consider a sequential process, in which both viruses spread at well separated times rather than in parallel. Also, our work is a generalization of the SIS-model and not of the SIR-model.

Wu et al. [138] study a different superinfection model by means of linear stability analysis and extensive computer simulations on networks with power-law degree distributions. Conditions for coexistence in terms of the reproductive numbers of both viruses are given.

Superinfection has been used as a feature in many models in the field of computational biology. Prado et al. [139] show - based on computer simulations - how coevolutionary cycles between pathogen virulence and sociality for hosts in contact networks are influenced by the possibility of superinfection. Leventhal et al. [140] show analytically and with simulations how the topology of an underlying contact network may impact the spread of competing viruses in the SIS-model. In their work, the second virus appears after the first virus has reached an endemic state, but can only spread to the subpopulation of still susceptible nodes, as they assume cross-immunity. Our work can be regarded as complementary, as we instead restrict the virus to spread only in the subpopulation of nodes already infected by the first virus.

A similar effect to superinfection appears in information diffusion processes between different pieces of information (contagions) that traverse the network. Myers and Leskovec [141] show that interaction between contagions can change their relative spreading properties, having a major impact on the diffusion process. Similar effects might trigger information cascades [142] and are observable as interacting waves in networks [143].

5.3. Modeling Superinfection

This section is divided in two parts: First, we present the standard Markovian SIS-model of epidemic spreading and point out some of its properties. Second, we show how this model can be generalized to describe superinfection by introducing mutations and additional viruses.

5.3.1. The SIS-model

The standard SIS-model is a Markov chain in which each node in a network can be in two possible compartments: I for *infected* or S for *susceptible* (to infection). Two underlying Poisson processes govern the transition between these compartments: the *curing process* is a nodal transition that changes the node's compartment from I to S (curing) with a fixed rate of δ. The *infection process* is a link based transition and changes the compartment of a node from S to I (infecting) with a fixed rate of β for each link of a susceptible node to an infected neighbor. Both rates β and δ are assumed to be fixed and global constants for the whole network. The fraction $\tau = \beta/\delta$ is called the *effective infection rate*.

Depending on τ, different behaviors of the SIS process are observed. A very low τ (for a specific topology) results in a relatively short survival time of the virus, as the average hitting time of the absorbing state of the Markov process is very low as well. However, there exists an epidemic threshold τ_c for which the virus becomes *endemic* and infects a constant fraction of nodes for a considerably longer period of time [106].

Given N nodes and two compartments, the network can be in 2^N different states, which define the *state space* of the Markov chain. The exponential growth of the state space makes an exact computation infeasible for larger networks. To deal with this issue, graph-automorphisms of the network can be used to "lump" certain states together (see Simon et al. [144]). This allows for polynomial sized representation for networks with symmetric infectious states like the complete network (shown in Figure 5.1) or the star network. However, this approach is not feasible for all networks. In order to analyze the epidemic process in these networks as well, mean-field approximations are frequently used, which we describe further in Section 5.4.

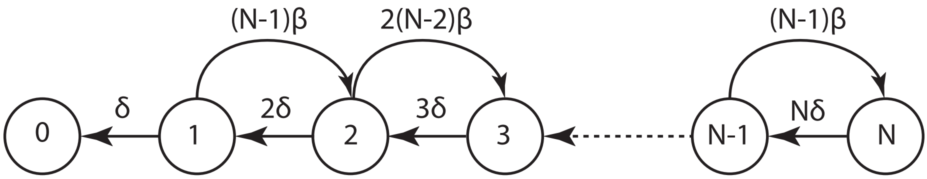


Figure 5.1: Markov graph of the SIS-model for the complete network of N nodes. Each node here represents one possible state of the Markov chain and is labeled with the total number of infected nodes of the underlying network. Links represent the transition rates between neighboring states. State 0 is an absorbing state, as there are no new infections possible once all nodes are cured.

5.3.2. The Superinfection SIkS-model

We propose a natural extension of the standard SIS-model which we call the SIkS-model. In this model, there exists a total number of k infection compartments $I_1, \dots, I_k$ linked to viruses 1 to k, which constitute an *infection hierarchy*: S is only susceptible to infections by virus 1, nodes in compartment I_j are only susceptible to infections by the next higher virus $j+1$, for $j = 1, \dots, k-1$. The infection rates $\beta_1, \dots, \beta_k$ describe the rate at which these infections occur. See Figure 5.2a for an illustration of these link based transitions.

5

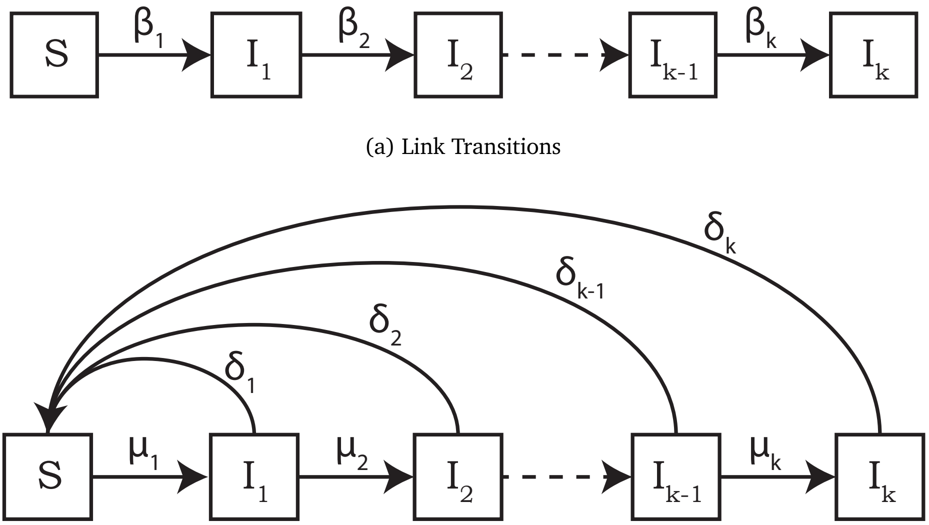
Depending on the current compartment, there are k curing rates $\delta_1, \dots, \delta_k$ which are nodal transitions back to S. We also introduce nodal *mutation* with rates $\mu_1, \dots, \mu_k$ as a second force of infection, which allows a node to switch to the next higher infection level without exposure to the next higher virus by a neighbor. Figure 5.2b illustrates these node based transitions.

The SIkS-model reduces to the SIS-model for $k = 1$ and $\mu_1 = 0$ and to the ε-SIS-model [107, 145] (which is an SIS-model that allows for self-infection) for $\mu_1 = \varepsilon$ for some small $\varepsilon > 0$. The size of the state space for the SIkS-model is $(k+1)^N$.

In order to study the effect of superinfection, we confine ourselves to $k = 2$ for the remainder. In this case, the complete process is governed by 6 parameters, namely: $\beta_1, \beta_2, \delta_1, \delta_2, \mu_1$ and μ_2. We will occasionally call β_2 *superinfection rate*, in contrast to β_1 which we simply call *infection rate*.

Finding a succinct representation of the Markov chain for the complete network in the SIkS-model for $k = 2$ is more complex than for the SIS-model. As we have two different viruses in the network, we can no longer identify a state with the total number of infected nodes. Additionally, there exist new transitions due to the introduction of mutation. Figure 5.3 shows a succinct representation of these transitions in a Markov chain of size $\frac{1}{2}(N+1)(N+2)$, where N is the total number of nodes. The state space counts all possible combinations for the total number of infected nodes in the network per virus.

In the star network, the spread of viruses is dominated by the compartment of the center node. If the center node is in compartment S, mutation is the only way of infecting new leaf nodes, as the only neighbor of a each leaf node is susceptible. Similarly, the only way of virus 2 to spread towards leaf nodes is to infect the center node, which is only possible once it is in compartment I_1. The compartment of the center node is like a switch that determines which virus may currently spread towards leaf nodes, which are the majority of all nodes in the star network.



(a) Link Transitions

(b) Nodal Transitions

Figure 5.2: Transition rate graphs for the SIkS-model.

Hence, the state space of the Markov chain of the star network is partitioned into 3 groups according to the 3 possible compartments of the center node. For each group, the states are further ordered similarly to the complete network by all possible combinations of the number of infected leaf nodes by each virus. Figure 5.4 shows a succinct representation of all possible transitions in the star network. The size of the state space for this Markov chain is $\frac{3}{2}N(N+1)$, where N is the number of leaf nodes (thus the total number of nodes is $N+1$).

5.4. Mean-Field Approximation

The exponentially growing state space of the Markov process imposes a hard challenge for the analysis of epidemics, especially for larger networks. In order to reduce the size of the governing equations, mean-field approximations are frequently applied in literature [105, 108, 146]. The use of mean-field approximations allowed for the discovery of some interesting results for the standard SIS-model, like a lower bound for the epidemic threshold by the inverse of the spectral radius of the adjacency matrix [109].

The recently introduced GEMF-model [121] is a generalized mean-field approximation of epidemic processes with multiple compartments in multilayer networks. We adopt the following GEMF-notations: each compartment is labelled by a number from 1 to M. The state of a node i at time t is $x_i(t) = e_m$, if i is in compartment m at time t. The vector e_m is a unit-vector, which is 0 at every position besides position m, where it is 1. As each entry of $x_i(t)$ is a Bernoulli random variable, the expected value of $x_i(t)$ is

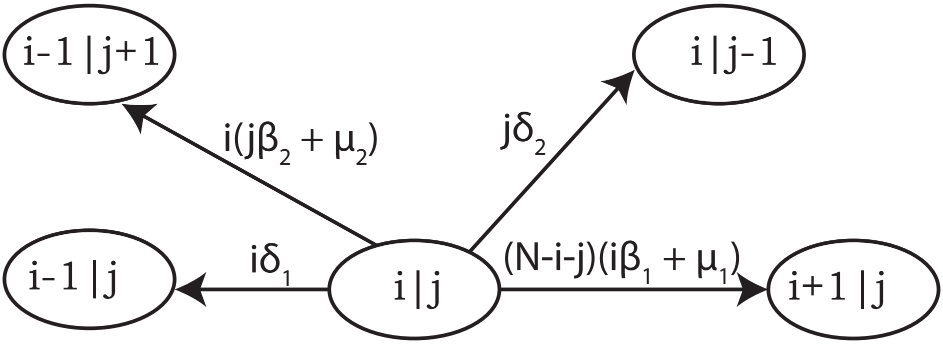


Figure 5.3: Superinfection in the complete network: the state space of the Markov process consists of tuples (i, j), where i nodes are infected with virus 1 and j nodes are infected by virus 2, leaving $N-(i+j)$ nodes in compartment S. Shown are all the possible outgoing state transitions from state (i, j) with the according rates.

given by the compartment occupancy probability vector:

$$v_i(t) = E[x_i(t)] = [\Pr[x_i(t) = e_1], \ldots, \Pr[x_i(t) = e_M]]^T \qquad (5.1)$$

We are particularly interested in the *average fraction of nodes* that belong to a compartment m. This is equivalent to $v_m(t)$ for any time t for the complete network.

In our model of superinfection, we have $M = 3$ compartments labelled with S, I_1, I_2. We use two layers[1] to describe the spreading process of the viruses. Each layer has an *influencer compartment*, which determines the spreading condition for the link based transitions. The influencer compartment of layer 1 is I_1, meaning that a node which has a neighbor in compartment I_1 on layer 1 undergoes a link transition with the specific rates in the transition matrix of layer 1, which we name A_{β_1}. Similarly, the influencer compartment of layer 2 is I_2, so any node with a neighbor in compartment I_2 on layer 2 undergoes a state transition with the specific rates in the transition matrix A_{β_2} of layer 2. In addition, there exists a third transition matrix A_δ, which describes all the nodal transitions and thus needs no influencer compartment. For our model of the superinfection, these matrices are given by

$$A_{\beta_2} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & \beta_2 \\ 0 & 0 & 0 \end{bmatrix}, \quad A_{\beta_1} = \begin{bmatrix} 0 & \beta_1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

$$\text{and} \quad A_\delta = \begin{bmatrix} 0 & \mu_1 & 0 \\ \delta_1 & 0 & \mu_2 \\ \delta_2 & 0 & 0 \end{bmatrix}.$$

[1] The GEMF-model allows for different contact networks on different layers, so it is possible to have two viruses that spread in the same population of nodes but over different links. In our case, both viruses use the same network.

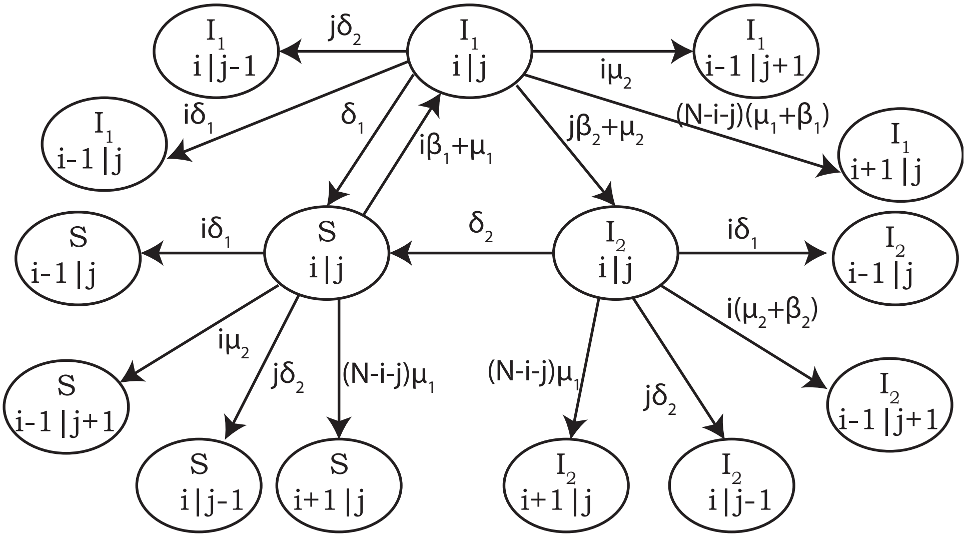


Figure 5.4: Superinfection in the star network with N leaf nodes: the state space of the Markov process consists of triplets (C,i,j), where C describes the compartment of the central node, i the number of leaf nodes that are infected by virus 1 and j the number of leaf nodes that are infected by virus 2. Shown are all the possible outgoing state transitions from the states $(S,i,j),(I_1,i,j)$ and (I_2,i,j) with their corresponding rates.

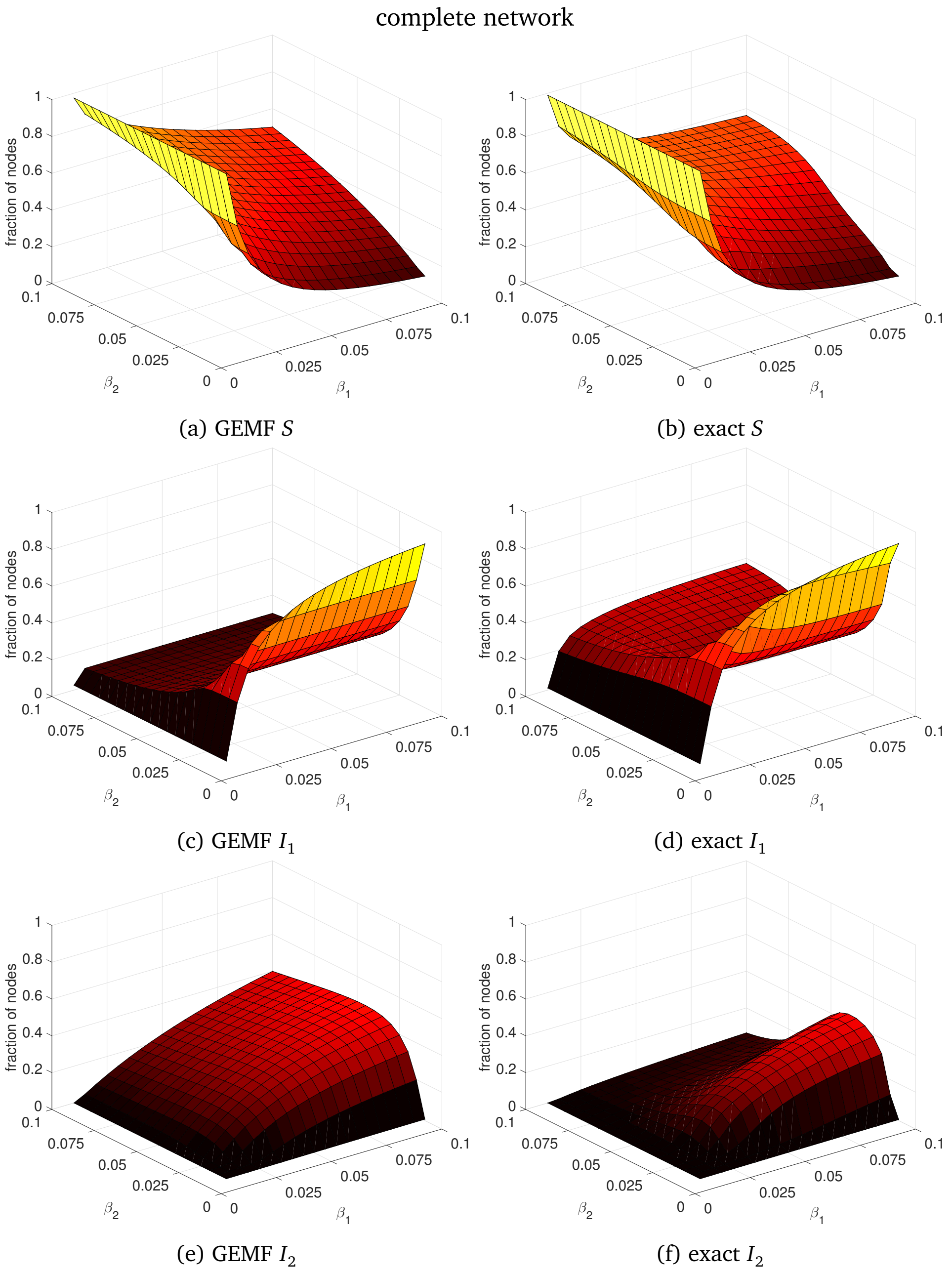
The GEMF-equations [121] are a set of non-linear differential equations that describe the behavior of the epidemic over time for $i = 1,\dots,N$:

$$\frac{dv_i}{dt} = -Q_\delta^T v_i - \sum_{l=1}^{L}\left(\sum_{j=1}^{N}(a_l)_{ij} v_{j,q_l}\right) Q_{\beta_l}^T v_i \tag{5.2}$$

where L is the number of layers and A_l with elements $(a_l)_{ij}$ is the adjacency matrix of the contact network of layer l with q_l being the corresponding influencer compartment. The matrices Q_δ, Q_{β_1} and Q_{β_2} are the Laplacians matrices of the matrices A_δ, A_{β_1} and A_{β_2}, where in general the Laplacian matrix Q of a matrix A is defined as:

$$Q = \mathrm{diag}\left(\sum_{i,j=1}^{N}(a_{ij})\right) - A. \tag{5.3}$$

Solving (5.2) gives a first order approximation for the average fraction of nodes in each compartment for the steady state of the underlying Markov process. For the complete network, the GEMF-equation reduces to the following:



(a) GEMF S

(b) exact S

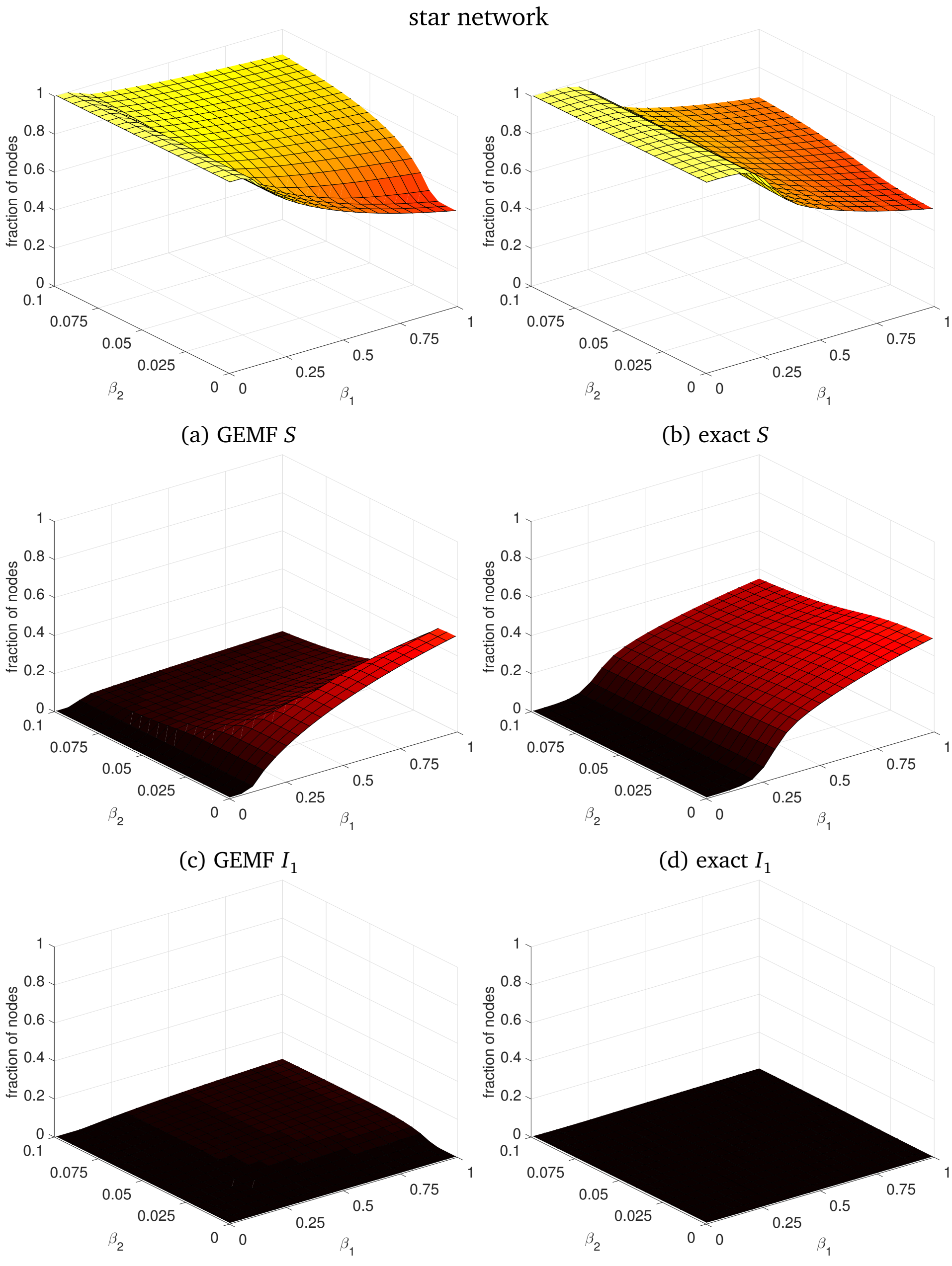
(c) GEMF I_1

(d) exact I_1

(e) GEMF I_2

(f) exact I_2

Figure 5.5: Expected fraction of nodes in corresponding compartments in the steady state obtained by GEMF (left column) and the exact process (right column) in a complete network of $N = 100$ nodes.



(a) GEMF S

(b) exact S

(c) GEMF I_1

(d) exact I_1

(e) GEMF I_2

(f) exact I_2

Figure 5.6: Expected fraction of nodes in corresponding compartments in the steady state obtained by GEMF (left column) and the exact process (right column) in a star network of $N = 100$ nodes.

$$\begin{aligned}\frac{d}{dt}\begin{bmatrix} v_S \\ v_{I_1} \\ v_{I_2} \end{bmatrix} = &-\begin{bmatrix} \mu_1 & -\delta_1 & -\delta_2 \\ -\mu_1 & \delta_1+\mu_2 & 0 \\ 0 & -\mu_2 & \delta_2 \end{bmatrix}\begin{bmatrix} v_S \\ v_{I_1} \\ v_{I_2} \end{bmatrix} \\ &-(N-1)v_{I_1}\begin{bmatrix} \beta_1 & 0 & 0 \\ -\beta_1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}\begin{bmatrix} v_S \\ v_{I_1} \\ v_{I_2} \end{bmatrix} \\ &-(N-1)v_{I_2}\begin{bmatrix} 0 & 0 & 0 \\ 0 & \beta_2 & 0 \\ 0 & -\beta_2 & 0 \end{bmatrix}\begin{bmatrix} v_S \\ v_{I_1} \\ v_{I_2} \end{bmatrix}.\end{aligned} \tag{5.4}$$

The quality of the GEMF-approximation, is assessed by a comparison between the solution of (5.4) and the exact process for the complete network of $N = 100$ nodes. The values of the fractions of the exact process were determined by continuously applying the transition matrix of the Markov process until the rate of change in the fraction of nodes in each compartment was less than $\varepsilon = 10^{-6}$ for the last 10 iterations. We assume that the steady state has been reached if this condition is true. The parameters were set to $\delta_1 = \delta_2 = 1$ and $\mu_1 = \mu_2 = 0.001$, while values ranging from 0.01 to 0.1 for β_1 and β_2 were investigated. The results are shown in Figure 5.5.

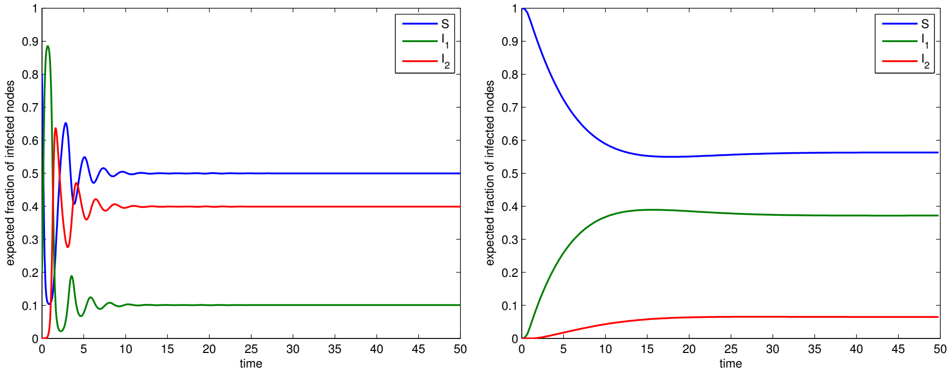
There exists a discrepancy in the number of infected nodes by virus 1, which is most of the time underestimated by GEMF especially for higher superinfection rates. The number of infected nodes by virus 2, however, seems to be overestimated by GEMF, which is most apparent for higher superinfection rates. Figure 5.7 shows the convergence of both, the exact process and the GEMF-approximation for $\beta_1 = \beta_2 = 0.1$. These parameters correspond to one of the 4 corners (the northern one) of the different surfaces plotted in Figure 5.5, where a high discrepancy can be observed.

Apart from giving a rather inaccurate estimation on the fractions for the steady state, GEMF shows a damped oscillation not observable for the exact process for the first 10 time units. These oscillations might arise in cases where the average fraction of virus 1 nodes approaches 0, indicating a near-extinction event from which the viruses recover. A similar artifact is known from differential equations for predator prey models, and is sometimes coined the *atto-fox problem* [147], where a population of less than one individual, which would be practically extinct, is able to resurrect. As a side-effect, the proportion of infected nodes by virus 2 is overestimated, as it would inevitably extinct soon after virus 1 as well.

These observations lead us to the conclusion that the GEMF-approximation is reasonably accurate except for cases in which the infection rate β_1 is above the epidemic threshold together with a relatively high superinfection rate β_2. The next section will elaborate on the interaction of both viruses in general.

5.5. The Course of Superinfection

The course of a superinfection is divided in 3 phases: phase 1 begins with all nodes being susceptible, so only mutations can move a node from compartment S to I_1. The nodal mutations are necessary, but are set intentionally at small rates ($\mu_1 = \mu_2 = 0.001$)



(a) GEMF

(b) exact

Figure 5.7: The exact process (right) and the GEMF approximation (left) over 50 time units with $\beta_1 = \beta_2 = 0.1$. At the beginning of the process, GEMF shows strong oscillations.

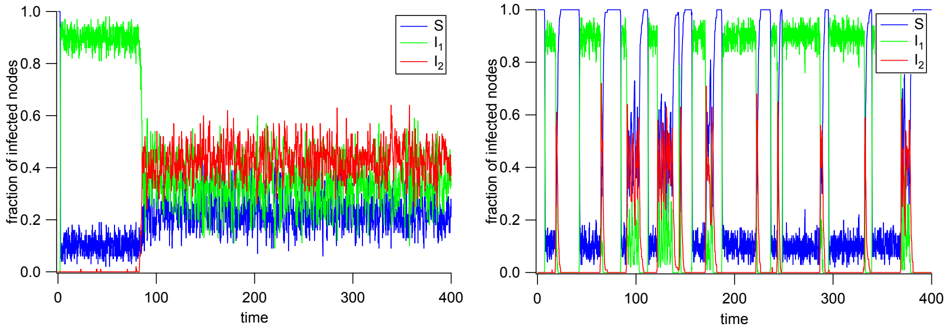
in comparison to the nodal curing $\delta_1 = \delta_2 = 1$. This assures that mutation has only a minor impact on the overall rate of newly infected nodes compared to their spreading over links. We want to outline the impact of the spreading rates β_1 and β_2 on the process, Therefore, we keep $\delta_1, \delta_2, \mu_1$ and μ_2 at those default values for all our analysis.

Once enough nodes are infected by mutations, the spreading rate β_1 determines, whether virus 1 becomes *endemic* (which means that it infects a larger part of the network) or not. If β_1 is too low, the process remains at phase 1 and apart from tenuous mutations, there is only a tiny number of infected nodes by virus 1 observable. Thus, an occurrence of virus 2 is unlikely[2].

If β_1 is higher than the epidemic threshold of the corresponding network, virus 1 becomes endemic, eventually acquiring a larger fraction of the network. This marks a transition into phase 2, which is characterized by the first mutation events that will move single nodes from I_1 to I_2. Similar to virus 1 and β_1, the rate of β_2 determines, whether virus 2 will become endemic as well: If β_2 is too low to infect enough other nodes via spreading, virus 2 is only observed in tiny numbers and the process remains in Phase 2 with virus 1 persistent in the network. If β_2 is above a certain threshold, which we call σ_l, virus 2 becomes endemic and spreads inside the population that was previously infected, effectively reducing the fraction of infected nodes of virus 1.

The outbreak of virus 2 marks the transition to phase 3 in which a *coexistence* of both viruses in the network is established. In coexistence, both viruses infect (on average) a constant fraction of nodes in the network. The duration of the coexistence depends on another threshold, which we call σ_u. If β_2 is above σ_u, the spread of virus 2 is so strong that virus 1 can no longer infect enough new susceptible nodes to survive. If there are no more nodes in compartment I_1 left, the number of nodes in compartment I_2 goes down to zero as well, as the tenuous mutations are a much weaker force than the nodal curing. We call the extinction of both viruses in the network an *extinction event*. After

[2] As the majority of the nodes remains in compartment S, virus 2 would also not be able to spread effectively.



(a) acyclic superinfection ($\beta_2 = 0.03$)

(b) cyclic superinfection ($\beta_2 = 0.08$)

Figure 5.8: A single simulation of the superinfection process in the complete network with $N = 100, \beta_1 = 0.1, \delta_1 = \delta_2 = 1.0$ and $\mu_1 = \mu_2 = 0.001$. All 3 phases of the superinfection repeat rapidly in the cyclic case while they occur only once in the time interval of 400 for the acyclic case.

an extinction event, the superinfection restarts at phase 1 and repeats until the next extinction event. Thus, we call this process *cyclic*.

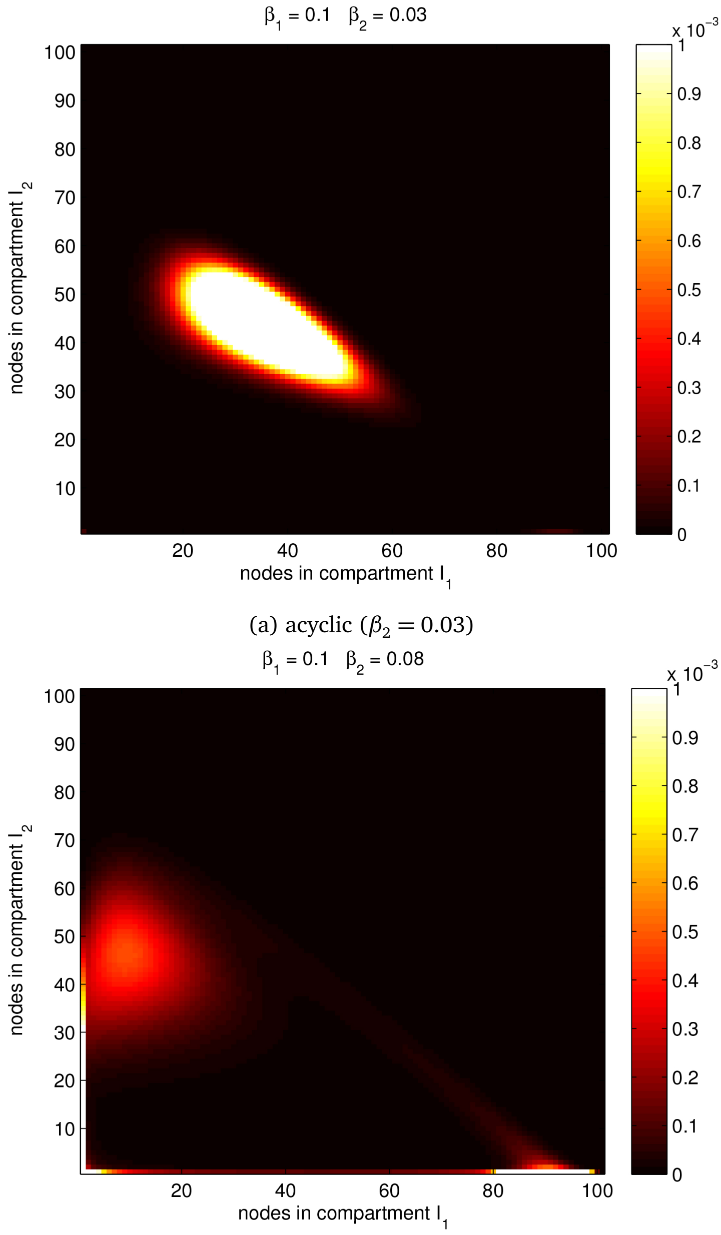
If β_2 is in between σ_l and σ_u, a stable coexistence is maintained. Extinction events are unlikely and the process remains in phase 3. We call this process *acyclic*. Figure 5.8 illustrates the course of a cyclic and an acyclic superinfection in comparison, obtained by simulation.

The probability distribution of the Markov state space reveals the difference between acyclic and cyclic superinfection: In the former case, the probability mass is concentrated at network states in which a mixture of both viruses exist (Figure 5.9a). However, if β_2 approaches σ_u, the probability mass shifts to network states, where only one or even none (extinction) viruses exist (Figure 5.9b).

To further investigate the extinctions, we simulated superinfection in the complete network and counted the number of simulations that resulted in at least one extinction event. More precisely, for each combination of β_1 and β_2 over a certain range, we started 10 simulations for a fixed amount of time $T = 400$, which we assume to be long enough to observe possible extinctions in most cases. We count a simulation as extinct if two conditions are fulfilled: there is a time t_1 at which at least 10% of all nodes are infected with virus 2 and there is a time $t_2 > t_1$ at which all nodes are cured.

We observe that in this experiment the extinctions seem to be related to the second largest eigenvalue ζ_2 of the infinitesimal generator Q of the Markov process. Contrary to the standard SIS-model [148], the eigenvalue ζ_2 can be complex in the superinfection model. Figure 5.10 hints that the number of extinct simulations is higher for pairs of β_1 and β_2 that have a complex ζ_2. Although a real-valued ζ_2 does not necessarily imply an acyclic superinfection, the experiment suggests that a complex ζ_2 results most likely in a cyclic superinfection. Furthermore, discrepancies in Figure 5.5 seem to happen mostly when ζ_2 is complex, suggesting that GEMF has difficulties to capture extinctions.

Although the interactions of the 6 parameters of the superinfection model are complex, a qualitative description on their influence on the average number of nodes for



(a) acyclic ($\beta_2 = 0.03$)

(b) cyclic ($\beta_2 = 0.08$)

Figure 5.9: Distribution of probabilities over the state space of the Markov process of superinfection in a complete network of 100 nodes. The rates are $\mu_1 = \mu_2 = 0.001, \delta_1 = \delta_2 = 1.0$ and $\beta_1 = 0.1$. In an acyclic coexistence (left) the probability mass is divided between states of mixed compartments. In a cyclic coexistence (right), the border states in which only one sort of virus exists have a considerably higher amount of probability mass than the mixed states.

Table 5.1: Qualitative influence of the parameters on the average number of the nodes belonging to the three compartments. A +-sign indicates that a high rate of the corresponding parameters will increase the average number of nodes in that compartment while a −-sign indicates a decrease. The $*$-sign for the superinfection rate β_2 on compartment I_2 is a special case: in general, the value has to be high to infect more nodes, but a too high value results in an extinction event, negatively influencing the average number of superinfected nodes.

	μ_1	μ_2	β_1	β_2	δ_1	δ_2
S	-	+	-	+	+	+
I_1	+	-	+	-	-	+
I_2	+	+	+	*	-	-

each compartment can be devised as shown in Table 5.1. The superinfection rate β_2 is a noteworthy exception as this parameter is particularly sensitive. Since virus 2 spreads inside a network that dynamically changes its size in response to the infection, we assume that σ_l and σ_u might depend on β_1 and (to a lesser extent) on μ_1.

Another strong influence is the topology of the underlying contact network. Figure 5.6 shows the exact process of the Markov chain from Figure 5.4 and its GEMF-approximation in the star network of $N = 100$ nodes. In comparison to the complete network shown in Figure 5.7, there seems to be no combination of β_1 and β_2 that allows for a larger support of compartment I_2. However, β_2 has an influence on the number of nodes infected by virus 1, which diminishes for higher values due to possible extinction events.

5.6. CHAPTER SUMMARY

In summary, we proposed a natural extension of the standard SIS-model to analyze nested epidemic processes in which a dominant virus spreads within the population of a another virus. We presented a succinct representation of the Markov chain for the complete network and the star network that we used to evaluate the quality of the GEMF-approximation. In particular, we observed that for high superinfection rates β_2, the GEMF model shows damped oscillations and overestimates the fraction of infected nodes of the stronger virus.

A closer look at the exact process revealed rich and complex dynamics, ranging from endemic behaviour, stable coexistence to extinction cycles triggered by the occurrence of extinction events. It has been shown that the superinfection processes in 3 different phases that are dependent on the parameters. In particular, β_2 is the most sensitive parameter as it determines whether the process repeats itself in cycles or maintains a state of stable coexistence.

If β_2 is higher than a certain threshold σ_u, an extinction event will eventually eradicate both viruses. We expect σ_u to be strongly dependent on the infection rate β_1 of the weaker virus and the topology of the underlying network. As we focused our analysis on β_1 and β_2, it remains open to which extent the nodal parameters $\delta_1, \delta_2, \mu_1$ and μ_2 influence the process. However, their influence is reflected in the infinitesimal generator

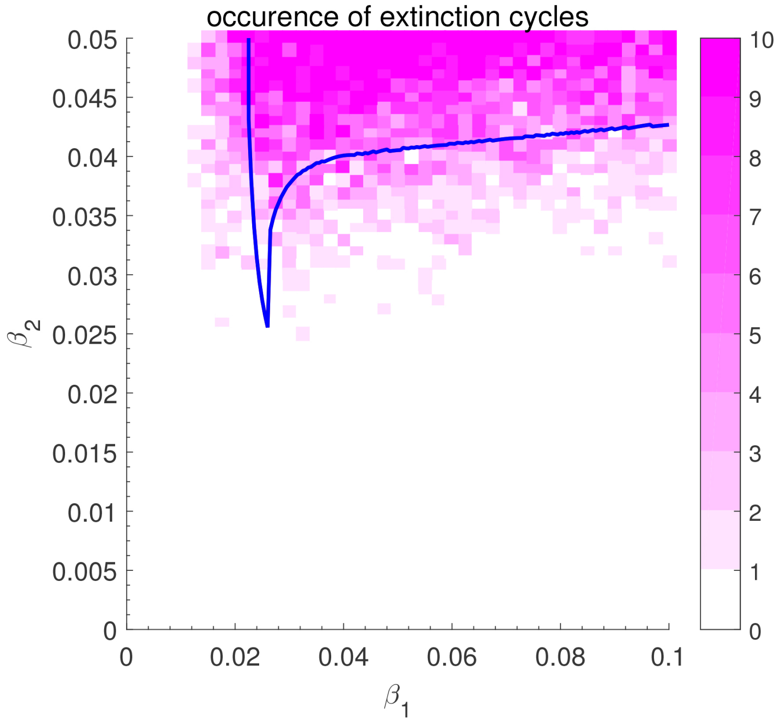


Figure 5.10: Each pink square corresponds to a fixed value of β_1 and β_2. Its saturation is proportional to the number of simulations that resulted in at least one extinction event, with white indicating no extinctions at all. The blue curve interpolates for each β_1 the smallest β_2 for which the second largest eigenvalue of the infinitesimal generator of the Markov process is no longer real. The underlying network is the complete network of $N = 100$ nodes.

Q of the Markov process for which we observe that complex eigenvalues may occur. In particular, a complex second largest eigenvalue ζ_2 seems to be a sufficient condition to observe extinction cycles.

So far, we were unable to find conditions for coexistence or extinctions for the star network, which seems to be more robust against superinfection. We further conjecture that the inaccuracies in the GEMF model are correlated to the occurrence of extinction events. Considering our model, a superinfection that cures from compartment I_2 back to I_1 instead of S might be worth investigating as well, with respect to real-world applications.

6

Symbolic Regression on Network Properties

Networks are continuously growing in complexity, which creates challenges for determining their most important characteristics. While analytical bounds are often too conservative, the computational effort of algorithmic approaches does not scale well with network size. Cartesian Genetic Programming for symbolic regression is used to evolve mathematical equations that relate network properties directly to the eigenvalues of network adjacency and Laplacian matrices. In particular, we show that these eigenvalues are powerful features to evolve approximate equations for the network diameter and the isoperimetric number, which are hard to compute algorithmically. Our experiments indicate a good performance of the evolved equations for several real-world networks and we demonstrate how the generalization power can be influenced by the selection of training networks and feature sets.

This chapter is based on a published paper [149].

6.1. Introduction

One of the first and most important steps for modelling and analyzing complex real-world relationships is to understand their structure. Networks are an effective way to organize our data so that nodes describe certain actors or entities, while relations are expressed as links connecting two nodes with each other. The resulting topological representation (adjacency matrix) provides an abstract model that is amenable for further analysis. For example, algorithms for finding shortest paths, spanning trees or similar structures usually take the topological representation of the network as input. Community detection algorithms can cluster groups of nodes that are more connected within their group than outside. Computing node centrality metrics allows for the identification of important nodes or critical connections. A well-known example is Google's Pagerank algorithm [43], which uses the eigenvector centrality of a node in order to assess the rank of a webpage with respect to Google's search queries.

Eigenvector centrality [150] is interesting from a different perspective as well. It shows that spectral network properties can improve our understanding of such vast aggregations of data like in the world-wide web. Spectral graph theory explicitly seeks to understand the relations between eigenvalues, eigenvectors and characteristic polynomials of various network matrices. Many links to fundamental questions of mathematics and complexity theory arise from spectral graph theory, making this area of research both valuable and intricate. It is possible that many topological network properties are reflected in the spectrum, only waiting to be discovered.

In this chapter we propose symbolic regression as a method to automatically derive insights in the spectral domain and their corresponding topological reflections in the network. Only a minimal number of assumptions are needed, in particular in comparison to the frequently used procedure of curve fitting, which assumes already a pre-knowledge of a certain function like a polynomial, exponential, etc. In contrast, symbolic regression is guided by supervised learning for a regression task that explicitly constructs free-form equations out of numeric features and elementary arithmetic operations.

The topological representation may be a cumbersome feature space for machine learning techniques, if only the binary features of the adjacency matrix are considered. Therefore, we examine the usage of features from the spectral domain of the network. By training the symbolic regression system on a set of carefully constructed networks, we are able to estimate target features. Consequently, symbolic regression may assist researchers to unravel the hidden structures in the spectral domain and to propose first-order approximations for difficult-to-compute properties.

This chapter is structured as follows: Section 6.2 introduces the concept of symbolic regression by giving references to previous work where this technique proved useful. Section 6.3 provides the necessary background in network science by introducing network properties that will be used as features and targets for our experiments. The setup of our experiments is outlined in Section 6.4 and their results are discussed in Section 6.5. We conclude with directions for future research in Section 6.6.

6.2. Related Work

6.2.1. Symbolic Regression

One of the most influential works on symbolic regression is due to Michael Schmidt and Hod Lipson [151], who demonstrated that physical laws can be derived from experimental data (observations of a physical system) by algorithms, rather than physicists. The algorithm is guided by evolutionary principles: a set of (initially random) parameters and constants are used as inputs, which are subsequently combined with arithmetic operators like $\{+,-,\times,\div\}$ to construct building blocks of formulas. Genetic operations like crossover and mutation recombine the building blocks to minimize various error metrics. The algorithm terminates after a certain level of accuracy is reached; the formulas that describe the observed phenomenon best are delivered as output for further analysis.

In the work of Schmidt and Lipson [151], symbolic regression was able to find hidden physical conservation laws, which describe invariants over the observed time of physical systems in motion, like oscillators and pendulums. It is remarkable that symbolic regression was able to evolve the Hamiltonian of the double pendulum, a highly non-linear dynamic system [152] that undergoes complex and chaotic motions. Also, accurate equations of motions were automatically derived for systems of coupled oscillators.

While symbolic regression rarely deduces error-free formulas, the output may deepen our insight in the problem and may help to eventually find exact solutions. One example is the case of solving *iterated functions*, which asks for a function $f(x)$ that fulfills $f(f(x)) = g(x)$ for some given function $g(x)$. Despite the simple description of the problem, there exist difficult cases for which highly non-trivial algebraic techniques seem to be needed to find solutions.

6

One example is the iterated function $f(f(x)) = x^2 - 2$, for which the best known analytic approach to find $f(x)$ requires the substitution of special function forms and recognizing relations between certain Chebyshev polynomials. Again, Schmidt and Lipson [153] were able to evolve a couple of symbolic expressions that were so close at describing a solution, that a simple proof by basic calculus could be inferred.

Most recently, symbolic regression has been explored in the context of generative network models by Menezes and Roth [154]. They present a stochastic model in which each possible link has a weight computed by an evolved symbolic expression. The weight-computation-rules are executed and the resulting networks are compared by a similarity-metric with some target networks (corresponding to the observations of a physical system), which guides evolution to incrementally improve the underlying generative model.

One particular benefit of symbolic regression and automatic generation of equations is reduction of the bias introduced sometimes unknowingly by human preferences and assumptions. Thus, it is possible for symbolic regression to discover relations that would be deemed counter-intuitive by humans. This makes symbolic regression especially attractive for finding non-linear relationships, for which the human mind often lacks insight and intuition.

With the exception of the deterministic FFX-algorithm by McConaghy [155], most symbolic regression algorithms are based on Genetic Programming [156], where an evolutionary process typically uses grammars [157, 158] to evolve expression trees. Our work can be potentially implemented by many of these Genetic Programming variants, but we selected Cartesian Genetic Programming (CGP) for reasons outlined in the following subsection.

6.2.2. Cartesian Genetic Programming (CGP)

CGP was originally developed by Julian Miller [159, 160] to represent electronic circuits on 2d-grids (hence the name *Cartesian*), but it soon became a general purpose tool for genetic programming. It has been used in numerous applications, e.g. to develop Robot Controllers [161], Neural Networks [162], Image Classifiers [163] and Digital Filters [164]. A recent result by Vasicek and Sekanina shows how approximate digital circuits can be efficiently evolved by CGP, giving human-competitive results in the field of approximate computing [165]. Vasicek also shows how CGP can be scaled to deal with a large number of parameters in order to optimize combinatorial circuits [166].

The reason why CGP is so popular (especially for circuit design) is due to its internal representation of the Genetic program. CGP uses a flexible encoding that represents the wiring of a computational network. Each node in this network is an arithmetic operation that needs a certain amount of inputs to produce an output. A simple $1+4$ evolutionary strategy changes the interconnections between those nodes in order to improve a fitness function (minimizing errors). Input parameters and constants are forward-propagated by applying the computational nodes until output nodes are reached. At these output nodes, the chain of mathematical operations on the inputs can be reconstructed as an equation.

A surprising property of CGP is that only a minor fraction of nodes actually contribute to the final computation. Similar to a human genome, only part of it is actively used, while inactive parts are dormant, but subject to genetic drift. This redundancy is often argued to be beneficial for the evolutionary process in CGP (see Miller and Smith [167]). There is also evidence that CGP does not suffer much from bloat [168], a major issue in other genetic programming techniques that tend to produce very large program sizes even for simple tasks.

6.3. Networks

In this section, we formally define some network properties and notation that will be used throughout our experiments.

6.3.1. Network Representations

A network is represented as a graph $G = (\mathcal{N}, \mathcal{L})$, where $\mathcal{N}$ is the set of nodes and $\mathcal{L} \subseteq \mathcal{N} \times \mathcal{N}$ is the set of links. The number of nodes is denoted by $N =| \mathcal{N} |$ and the number of links by $L =| \mathcal{L} |$. The set $\mathcal{L}$ is typically represented by an $N \times N$ adjacency matrix A with elements $a_{ij} = 1$ if node i and j are connected by a link and $a_{ij} = 0$ otherwise. As we restrict ourselves to simple, undirected networks without self-loops in this chapter, A is always symmetric. We call A the *topological representation* of G as each element of A directly refers to a structural element (a link) of the network. The number of all neighbors of a node i is called its degree $d_i = \sum_{j=1}^{N} a_{ij}$.

The adjacency matrix A is not the only possible representation of a network. Of equal importance is the Laplacian matrix $Q = \Delta - A$, where Δ is a diagonal matrix consisting of the degrees d_i for each node $i \in \mathcal{N}$.

A different view on the network can be derived by its eigenstructure. Given the adjacency matrix A, there exists an eigenvalue decomposition [80]

$$A = X \Lambda X^T \tag{6.1}$$

such that the columns of X contain the eigenvectors $x_1, x_2, \ldots, x_N$ belonging to the real eigenvalues $\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_N$, respectively, contained in the diagonal matrix Λ.

While obtaining the spectral representation of the network requires computational effort by itself (usually, the network is given in its topological representation for which the eigenvalues still need to be determined), it provides a different perspective on the network's properties. For example, the largest eigenvalue λ_1 is linked with the vulnerability of a network to epidemic spreading processes [109].

A similar decomposition is possible for the Laplacian matrix, whose eigenvalues are denoted by $\mu_1 \geq \mu_2 \geq \ldots \geq \mu_N$ and whose real eigenvectors are $y_1, y_2, \ldots, y_N$. The second smallest eigenvalue μ_{N-1} is known as the *algebraic connectivity* [82] and its corresponding eigenvector is known as *Fiedler's vector*. Spectral clustering [80] is a possible application of Fiedler's vector.

Both eigensystems constitute the *spectral representation* of G. Our goal is to describe network properties typically computed by algorithms on the *topological representation* of G by simple functions consisting of elements from the *spectral representation* of G. An example is the number of triangles $\blacktriangle_G$ in a network. A way of computing $\blacktriangle_G$ is to enumerate all possible triples of nodes in a graph and checking whether they are connected in A. However, the number of triangles can also be expressed as

$$\blacktriangle_G = \frac{1}{6} \cdot \sum_{k=1}^{N} \lambda_k^3 \tag{6.2}$$

and is thus directly computable from the spectral representation without the need of exhaustive enumeration (see Appendix C for a proof of Equation (6.2)).

6.3.2. NETWORK PROPERTIES

NETWORK DIAMETER

Many applications of networks are concerned with finding and using shortest-path structures in networks. A *path* between two distinct nodes i and j is a collection of links that can be traversed to reach i from j and vice versa. A *shortest path* is a path with minimal number of links. The *diameter* ρ of a network is defined as the length of the *longest* shortest path in the network, i.e. the maximum over all shortest-path lengths between all node-pairs. Algorithms that solve the all-pairs shortest path problem (like the Floyd-Warshall algorithm) are able to compute the diameter in $\mathcal{O}(N^3)$. While more efficient algorithms exist for sparse networks, an exact computation of the diameter is usually too expensive for very large networks.

There exist multiple upper bounds for the diameter [169, 170], but we find the bound of Chung et al. [171] most tight in almost all cases:

$$\rho \leq \left\lfloor \frac{\cosh^{-1}(N-1)}{\cosh^{-1}\left(\frac{\mu_1+\mu_{N-1}}{\mu_1-\mu_{N-1}}\right)} \right\rfloor + 1. \tag{6.3}$$

This bound was independently derived by Van Dam and Haemers [172].

Isoperimetric Number

For each subset of nodes $X \subset \mathcal{N}$ we can define the set ∂X as the set of links that have exactly one endpoint in X and the other endpoint in $\mathcal{N} \backslash X$. The *isoperimetric number* η of a network is defined as

$$\eta = \min_{\substack{X \subseteq \mathcal{N} \\ |X| \leq \frac{1}{2}N}} \frac{|\partial X|}{|X|}. \tag{6.4}$$

Essentially, the isoperimetric number is a measure related to *bottlenecks* in networks. Intuitively, a low isoperimetric number indicates that the network can be separated in two reasonably big parts by only cutting a minimum amount of links. While the isoperimetric number is a good descriptor of network robustness, its computation for general networks is intractable, as the computational effort scales with the amount of possible cuts of the network. More information on the isoperimetric number can be found in [80, 173].

6.4. Experiments

This section describes technical details of the symbolic regression process we deployed to infer equations for the network diameter and the isoperimetric number. As symbolic regression is a supervised learning technique, we describe the sets of networks that were used for training and testing, together with the features we extracted for each case.

6.4.1. Network Diameter

In order to find a suitable formula for the network diameter, we trained CGP on 3 different sets of networks:

- augmented path graphs,
- barbell graphs and
- the union of both.

The augmented path graphs were generated by iteratively adding random links to a simple path graph of N nodes. With each additional link, there is a chance to lower the diameter of the network. Following this procedure, it is possible to generate a set of relatively sparse graphs of constant node-size with uniformly distributed diameters.

A barbell graph $B(a, b)$ is generated by taking two cliques of size a and connecting them with a path graph of size b. The total number of nodes is $N = 2a + b$. The diameter $\rho(B(a, b))$ is always $b + 3$. Adjusting the length of the path graph allows for generating graphs of different diameters. Changing the size of the cliques allows for creating graphs with the same diameter, but different number of nodes. We sample again such that the network diameter is uniformly distributed within the set of all barbell graphs. Compared with augmented path graphs, barbell graphs are (in general) denser networks.

The set of mixed graphs is the union of the set of augmented path graphs and barbell graphs. See Figure 6.1 for examples of these networks and Table 6.1 for a summary of all sets of networks for the experiments.

One reason why we have chosen these sets of networks instead of, for example, Erdős-Rényi (ER) random graphs [174], is to control the distribution of our target feature, the network diameter. Preliminary experiments have shown that too little variance in our target feature will push CGP to converge to a constant function, which does not include any information about the relation between spectral features and the target that we want to extract. For an ER graph of N nodes and with link probability p, Bollobás [174] showed that, for fixed p and N large, ρ can only have one of two possible neighboring values with high probability. Thus, sampling uniform diameters for the random graph model requires careful adjustment of N and p, where we found the usage of augmented paths and barbell graphs more convenient.

For the supervised learning of CGP, each set of networks was separated in a 60% training and a 40% test set. Table 6.2 gives an overview of the various parameters we set for CGP. In preliminary experiments, we changed each of these parameters independently from another and found the settings of Table 6.2 to provide the most useful results in terms of fitness and formula complexity. A more thorough parameter tuning approach is needed for maximum performance, but is outside the scope of this work. For the meaning of these parameters, see Miller [159]. Effective tuning of CGP was researched by Goldman and Punch [175].

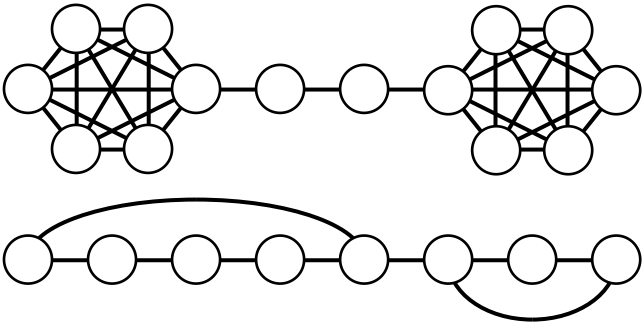


Figure 6.1: Example of the barbell graph $B(6,2)$ with $\rho = 5$ on top and an augmented path graph with $\rho = 4$ at the bottom.

In our experiments, we tried a vast selection of different features to evolve formulas. To keep this section organized, we report only results derived from two of the most generic, but useful, sets of features:

A) $N, L, \lambda_1, \lambda_2, \lambda_3, \lambda_N$

B) $N, L, \mu_1, \mu_{N-1}, \mu_{N-2}, \mu_{N-3}$.

Table 6.1: Properties of network sets.

	aug. path	**barbell**	**mixed**
networks	1672	1675	3347
nodes	$N = 70$	$7 \leq N \leq 667$	$7 \leq N \leq 667$
diameter	$2 \leq \rho \leq 69$	$4 \leq \rho \leq 70$	$2 \leq \rho \leq 70$
avg. link density	0.04845	0.36985	0.20910

Table 6.2: Parameterisation of CGP.

parameter	**value**
fitness function	sum of absolute errors
evolutionary strategy	1+4
mutation type and rate	probabilistic (0.1)
node layout	1 row with 200 columns
levels-back	unrestricted
operators	$+, -, \times, \div, \cdot^2, \cdot^3, \sqrt{\cdot}, \log$
number of generations	$2 \cdot 10^5$

Additionally, the natural numbers $1,\ldots,9$ were provided as network independent constants for CGP to adjust evolved terms appropriately.

The choice of feature sets A and B provides a reasonable trade-off between formula complexity and fitness. While selecting the complete spectrum of eigenvalues as features is possible, we observed that it leads to a high formula complexity without providing considerable improvements in fitness. Additionally, the largest adjacency (smallest Laplacian) eigenvalues are the ones that are suggested to have the strongest influence on network properties [80]. Lastly, since the number of nodes in our network instances is not (in every case) constant, giving the complete spectrum would mean that several features would be missing in networks with low number of nodes. It is unclear, how an appropriate substitution of the missing features should be realized. Thus, some of the discovered formulas could be inapplicable for some networks.

Since the evolutionary procedures of CGP to optimize the fitness of the evolved expressions are stochastic, we deployed multiple runs for each combination of feature and network set. We aggregated those multiple runs into batches, as the test-environment was implemented to run on a computational cluster. Each batch consisted of 20 runs of CGP for a specific set of features. Out of those 20 runs, only the one with the best (lowest) fitness is reported. The fitness is the sum of absolute errors on the test instances of the corresponding set of networks. More formally, if $\hat{\rho_G}$ is the estimate on the diameter ρ_G of network G given by the evolved formula $\hat{\rho}$ and $\mathcal{G}_{test}$ is the set of all networks for testing, the fitness $f(\hat{\rho})$ is defined as:

$$f(\hat{\rho}) = \sum_{G \in \mathcal{G}_{test}} |\rho_G - \hat{\rho_G}|. \tag{6.5}$$

Table 6.3: Experimental results for the network diameter.

networks	feature set	avg. fitness	min. fitness	min. approx. error
aug. path	A	3694.98750	3404.53700	5.08899
	B	842.89691	778.98900	1.16441
barbell	A	1.66654	0.00900	0.00001
	B	50.53473	$< 10^{-5}$	$< 10^{-5}$
mixed	A	5313.91179	4500.68900	3.36123
	B	1462.61943	1134.34100	0.84716

Furthermore, we define the approximation error $e(\hat{\rho})$ as the average deviation from the diameter over the complete test set:

$$e(\hat{\rho}) = \sum_{G \in \mathcal{G}_{test}} \frac{|\rho_G - \hat{\rho_G}|}{|\mathcal{G}_{test}|}. \tag{6.6}$$

We present the results over 100 batches for each combination of feature and network test set in Table 6.3.

6.4.2. Isoperimetric Number

The training set of networks for the isoperimetric number η had to be limited to relatively small networks, since the computation of η becomes intractable even for general medium-sized networks. Thus, we decided to exhaustively enumerate all networks of size $N = 7$, for which the computation was still practical. This set consists of 1046 non-isomorphic networks, which we randomly split into a training set of 627 and a test set of 419 networks. We applied the same parameters to CGP as shown in Table 6.2, with one exception: we created 100 batches for each of the following sets of operators:

1. $+, -, \times, \div, \sqrt{\cdot}, \log$
2. $+, -, \times, \div, \cdot^2, \log$
3. $+, -, \times, \div, \sqrt{\cdot}$
4. $+, -, \times, \div, \cdot^2$.

Since we have only networks of size $N = 7$, we can select the full spectrum as our features, resulting in the following feature sets:

A') $N, L, \lambda_1, \lambda_2, \lambda_3, \lambda_4, \lambda_5, \lambda_6, \lambda_7$

B') $N, L, \mu_1, \mu_2, \mu_3, \mu_4, \mu_5, \mu_6$.

Since the smallest Laplacian eigenvalue μ_N always equals 0, $\mu_7 = 0$ and is thus not included as a feature. Additionally, we provided the natural numbers $1, 2$ and 3 as constants. Each batch consisted of 5 independent runs from which only the best one is reported.

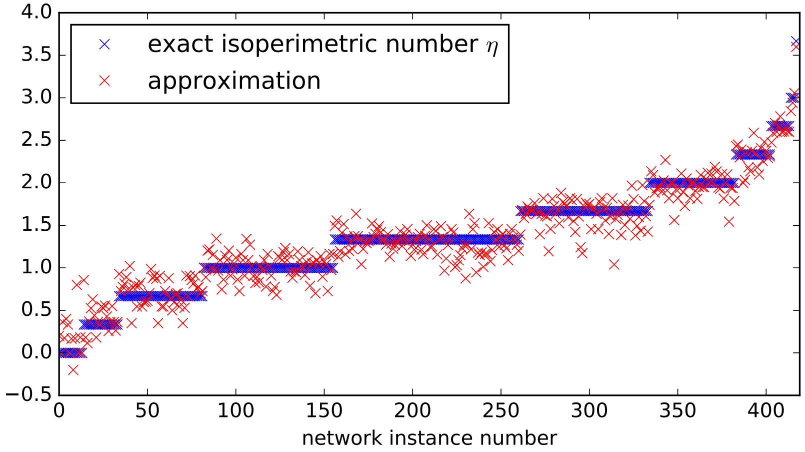


Figure 6.2: All 419 networks from the test set ordered by their actual isoperimetric number ascending from left to right. The red crosses show the approximation given by Equation (6.7).

Feature set A' delivered on average much better results than feature set B'. The best result was found with feature set A' and operator-set $+, -, \times, \div, \cdot^2$, log, although not all of these operators appear:

$$\hat{\eta_1} = \frac{L - \lambda_2^2 - 2}{\frac{\lambda_2}{2} + 5} \tag{6.7}$$

Although Equation (6.7) is short (low complexity), it had still the best fitness (53.215) of all evolved formulas. The approximation of η on the test set is shown in Figure 6.2.

6.5. Discussion

In the previous section, we evolved approximate equations for hard to compute network properties. The approximation errors were largely dependent on the used networks for training and testing. For example, the best equations for the diameter found for barbell networks have almost no error, while noticeable errors exist for augmented path and mixed network. This raises two questions:

- how is the quality of the approximate equations influenced by the pre-selected networks and
- how do the approximate equations generalize to other networks?

To give answers, we compute already established analytical equations from the literature as reference points for quality and provide appropriate selections of networks which were not involved in the generation process of the evolved equations.

6.5.1. Network Diameter

As a measure for the quality of our evolved equations, we compare their estimates of diameter ρ to the upper bound given by Equation (6.3). An upper bound and an approximation are different: while the bound is always above the real diameter, approximations may be above or below without any guarantees. Yet, we believe the bound can mark a reference point for a qualitative comparison in addition to the exact diameter itself.

As additional networks, we selected 12 real-world data sets available at networkrepository.com [176], where more information, interactive visualization and analytics can be found. While these networks should only be viewed as examples, they might give an idea about the applicability of the presented technique. Figure 6.3 shows a visualization of a subset of these networks. To eliminate the selection-bias and gather significant results, one would need to sample the network space in a representative and meaningful way, which is notoriously difficult. For example, simple network generators like Erdős-Rényi random graphs or the Barabási-Albert model are not sufficient, as they will only allow to sample certain degree distributions. Consequently, we restrict ourselves to examples.

We expect that the equations with the lowest fitness give the best results. Because feature set B consistently outperformed feature set A in terms of fitness, we analyzed the approximations given by the best equations of feature set B. The explicit equations are:

$$N-\frac{1-\frac{1}{(L-N)^{\frac{3}{2}}}}{\frac{6-\frac{6}{(L-N)^{\frac{3}{2}}}}{\sqrt{L-N}}+4\sqrt{L-N}}-2\sqrt{L-N}-\frac{1}{\sqrt{L-N}} \tag{6.8}$$

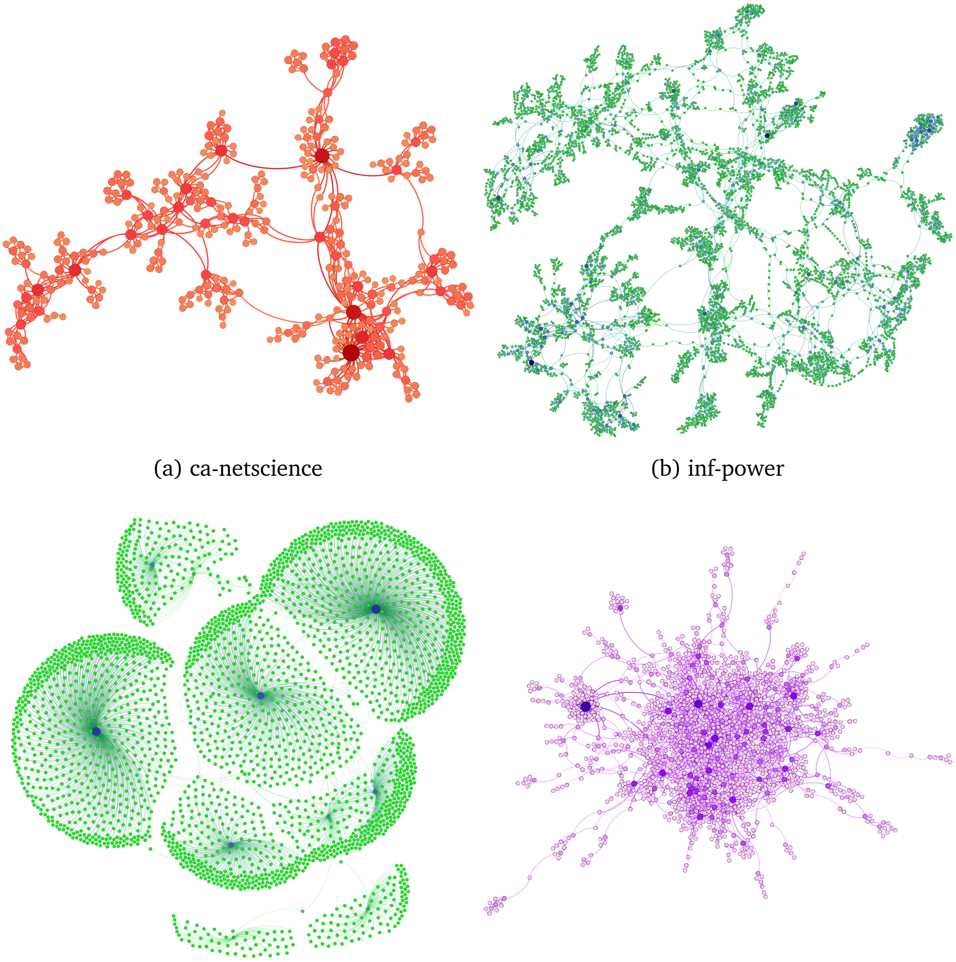
$$\frac{\log\left(2L\mu_{N-3}+6\right)+6}{\log\left(L\mu_{N-3}+\sqrt{5}\sqrt{\frac{1}{\mu_{N-1}}}\right)}+\sqrt{5}\sqrt{\frac{1}{\mu_{N-1}}}+3\sqrt{82}\sqrt{\frac{1}{729L\mu_{N-2}\mu_{N-3}-5}} \tag{6.9}$$

$$\sqrt{\sqrt{N}+\frac{45\mu_{N-3}}{(\mu_{N-1}+\mu_{N-3})^2}+\log\left(\frac{216}{(\mu_{N-1}+\mu_{N-3})^2}\right)-\frac{16}{9\mu_{N-3}}+\frac{8\sqrt[4]{\mu_{N-3}}}{L\mu_{N-1}\mu_{N-2}}} \tag{6.10}$$

The numerical values are all listed together with some basic properties of our validation networks in Table 6.4. First, we observe that Equation (6.8) performs extremely poorly by giving huge overestimations of ρ, despite its fitness of almost 0 for the networks of the original test set. The reason is that Equation (6.8) was evolved on barbell graphs only, which have a fixed and symmetric structure. In particular, the difference $N-L$, which is a frequent subterm of the formula, is higher in the dense barbell graphs compared to the rather sparse networks of our validation set. Thus, Equation (6.8) seems to be *overfitted* to the class of barbell graphs.

Equation (6.9) was evolved only on augmented path graphs and provides a much better approximation of ρ for our validation networks. This might be the case since the validation networks are more similar to the sparse augmented path graphs than to the dense barbell graphs. A visual comparison of the approximation of Equation (6.9) is given by Figure 6.4.

Adding the barbell graphs to the training set, like in the evaluation of Equation (6.10), shows that the accuracy of the approximation of ρ increases by roughly 10%, which must be the effect of the barbell graphs adding a selective force towards accuracy on more



(a) ca-netscience

(b) inf-power

(c) socfb-nips-ego

(d) bio-yeast

Figure 6.3: Visual representation of a subset from the validation networks for the network diameter. Basic properties of these networks are noted in Table 6.1. Bigger nodes in darker colors correspond to higher degree nodes. All networks are particularly different from augmented path and barbell graphs and show some form of modular structure.

Table 6.4: Diameter on validation networks.

name	N	L	ρ	Eq. (6.9)	Eq. (6.8)	Eq. (6.10)	Eq. (6.3)
ca-netscience	379	914	17	21	333	24	160
bio-celegans	453	2025	7	7	374	8	104
rt-twitter-copen	761	1029	14	16	728	18	126
soc-wiki-Vote	889	2914	13	10	799	12	133
ia-email-univ	1133	5451	8	6	1002	8	58
ia-fb-messages	1266	6451	9	7	1122	10	96
web-google	1299	2773	14	29	1222	35	336
bio-yeast	1458	1948	19	19	1414	22	208
tech-routers-rf	2113	6632	12	14	1979	17	237
socfb-nips-ego	2888	2981	9	52	2869	61	2466
web-edu	3031	6474	11	36	2914	40	663
inf-power	4941	6594	46	98	4860	110	749

denser networks. Moreover, it seems that CGP focused on finding a good approximation for the augmented path graphs in the mix rather than considering to find an equation for both classes of networks. In the majority of the cases, ρ is still overestimated a little, but by far not as much as by the upper bound in Equation (6.3).

6.5.2. Isoperimetric Number

The quality of the equations approximating the isoperimetric number will be related to the Cheeger inequality (see Mohar [173]) that gives us bounds in relation to the algebraic connectivity μ_{N-1} and the maximum degree d_{max} of the network G:

$$\frac{\mu_{N-1}}{2} \leq \eta \leq \sqrt{\mu_{N-1}(2d_{max} - \mu_{N-1})} \tag{6.11}$$

Since our equations were evolved by an exhaustive enumeration of all non-isomorphic networks of $N = 7$, we are interested how their quality of fit will differ with N. However, as pointed out before, the computation of the exact value for η is in general only feasible for very small networks. Consequently, we cannot use any of the validation-networks from Table 6.4. Instead we decided to sample random networks of $N = 20$ nodes and links from $22 \leq L \leq 190$. In total, we generated 4984 non-isomorphic connected networks with roughly uniformly distributed link densities by a variant of the ER random graph model. In these networks, the isoperimetric number η ranges from 0.2 to 10.0, while in our training set η was between 0 and 4.

Surprisingly, Equation (6.7) deduced from all networks with $N = 7$ nodes is performing poorly on the set of random networks, as shown by the green dots in Figure 6.5. The estimates are most of the time not even below the bound of the Cheeger inequality, shown in grey. By analyzing the sum of absolute errors on this new set of networks, we found that from all evolved formulas, the following equation for the isoperimetric

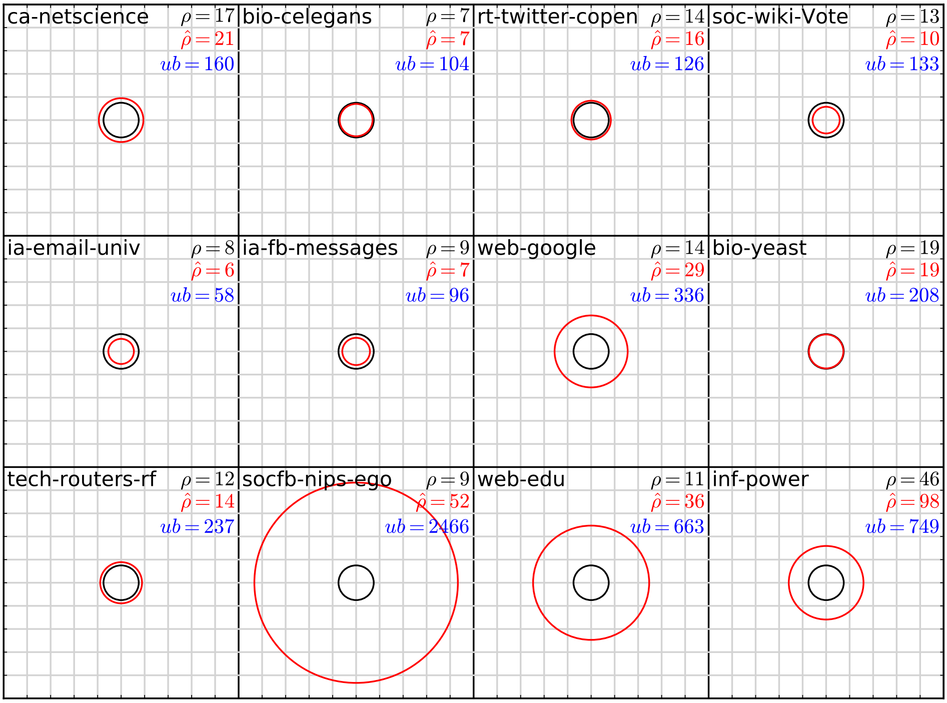


Figure 6.4: Red circles: approximate diameter $\hat{\rho}$ by Equation (6.9) relative to the network diameter ρ as a black circle. All network diameters are scaled in each network to have unit-length in the figure. All values are rounded to the next integer. The upper bound Equation (6.3) values are given as ub in blue (too large to plot).

number gives the best performance:

$$\hat{\eta}_2 = \frac{1}{N^2}\left(L\left(\frac{\mu_1}{\mu_2}+\mu_2\right)-1\right). \tag{6.12}$$

We observe that for over 98% of the random networks, the estimate of Equation (6.12) was within $(1 \pm 0.2)\cdot\eta$. Since this equation incorporates not only spectral features, but also N and L, we believe it generalizes better to networks of different size other than those used in the training set. Additionally, our experiments show that a low fitness value does not necessarily correspond to good generalization. Out of the 800 batches used to find a formula for η, only 259 returned expressions that did not create artifacts (like square roots of negative numbers or divisions by zero, which CGP evaluates to 0 by definition). While Equation (6.7) ranked first with a fitness of 53.215, Equation (6.12) was one of the unranked expressions, since on some of the unconnected networks of the training set, μ_2 was 0, while $\mu_2 = 0$ did never appear for the connected random networks.

It is also noteworthy that Equation (6.12) seems to slightly overestimate η as soon as networks with $\eta > 4$ are encountered. This does not seem to be a coincidence, as 4

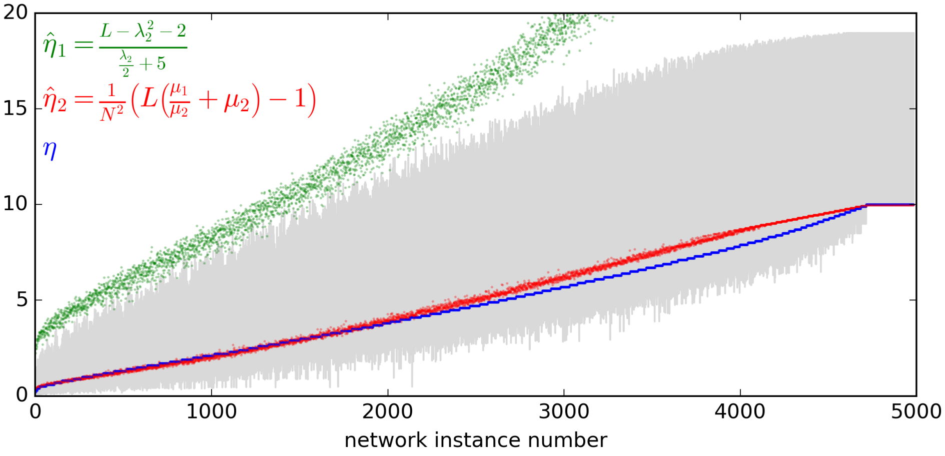


Figure 6.5: A set of 4984 random networks with $N = 20$ ordered by their isoperimetric number (blue). The grey area corresponds to the Cheeger inequalities given by Equation (6.11). The lower bound coincides with η and the red approximation formula for the last 100 networks, as they are all fully connected.

was the maximum value for η in the training set.

6.6. Chapter Summary

Our experiments provide a first demonstration that symbolic regression can be applied to analyze networks. For the first time, to the best of our knowledge, an automated system has inferred approximate equations for network properties, which otherwise would have required a high algorithmic effort to be determined. Although these equations are not rigorously proven and might be cumbersome for humans to comprehend, they are able to exploit the hidden relationships between the topological and the spectral representation of networks, which has been elusive to analytical treatment so far.

We do not expect that symbolic regression at the current level will substitute researchers deriving meaningful equations, but we do believe that symbolic regression can be a meaningful tool for these researchers. As the proposed techniques are not biased by human preconceptions, unexpected results might provide inspiration and stimulating starting points for the development of formal proofs or more accurate formulas. While this lack of bias can be an advantage, the proposed system nevertheless allows for the incorporation of *a priori* expert knowledge. If certain features and operators are suspected to be correlated to an unknown target quantity, their usage can be enforced easily.

Understanding which conditions give rise to equations with a high generalization power for networks will be the main challenge for the future. While a good fitness of an equation does not necessarily imply a high generalization power, our experiments indicate that symbolic regression is clearly able to produce equations that are reasonably accurate for unknown networks. In order to prevent overfitting to the training set and to increase this generalization power, selecting a good set of networks for training seems to be the key. This makes symbolic regression especially appealing when dealing with networks that can be characterized by their structural and degree-related properties, like scale-free or small-world networks. As networks with such properties are ubiquitous, discovering explicit relations between their features will pave the way for a deeper insight into our increasingly connected environments.

7

Conclusion

This thesis described the conditions for the spread of information in networks. The structural aspect considered properties of networks that facilitated the spread of information while the dynamical aspect put a focus on the propagation process. Information is understood as a placeholder for an abstract exchange of data which becomes measurable in some form. Thus, when confronted with applications and challenges of our interconnected society, a major task is to infer the structure and the dynamics from our observations and measurements as best as possible. This final chapter reviews the potential of the developed and deployed techniques of this thesis, but also presents a critical view on their limitations. Based on our findings, new directions for future research are suggested.

7.1. Main Contributions

7.1.1. Analyzed Contagions

Analyzing propagation phenomena in complex systems should begin with the question what exactly is the entity (contagion) that is expected to spread? Throughout the thesis, we have introduced and investigated multiple of such phenomena of different nature:

- Chapter 2 presented a study of the spread of anti-social behavior in the context of multi-player online games. The entity of spread was a negative and offensive attitude (toxicity), whose measurement demanded to look and interpret contextual information.
- Chapter 3 was a study about the human brain and its activity during resting-state. Brain activity was the entity of spread and was measured by magnetoencephalography (MEG) and transformed into phase transfer entropy.
- Chapter 4 described the long-term evolution of a computer worm. While the computer worm was clearly the contagion of this scenario, our measurements were limited to infected IP-addresses obtained from the logs of the worms sinkhole.

Thus, it has to be noted that the entity of spread is rarely ever directly observed, which remains true for many related applied studies. Instead, one has to work with the traces of the spreading process within the data available. Changes in behavior, like in the case of toxicity, are important to detect, as otherwise we cannot be assured that we are following the right contagion. The toxicity detection that was introduced in Chapter 2 is thus an essential part of the study in information propagation. The main contribution of this chapter was the annotation systems which allowed to search for n-grams of offensive language in a short contextual window. Standard techniques from natural language processing were not suitable for the elliptic and erroneous jargon of the players, so the development of new techniques like the letterset method were necessary. While the detected toxicity could not be related to the overall win-rate of the involved players, it could be related to triggers like kill-events. It was also shown that a toxic language becomes more dominant for losing teams at the end of their matches. The application of the annotation system to the game data can be compared conceptually to the preprocessing of sinkhole log files of the computer worm of Section 4.3.2 and the signal processing of the MEG measurements of the brain from Section 3.2.1.

The main purpose of this step is to carve out the contagion as best as possible from presumably noisy data. While it is essential to reduce the disturbing factors for the study, it has to be emphasized that especially over longer periods of time, multiple similar processes in such systems might occur. Thus, there might be multiple contagions that appear, spread, interact, compete and disappear again. The superinfection introduced in Chapter 5 is a possible way of looking into interacting spreading processes. In particular, the evolution of nested epidemic processes which can be described by the SIS-model can be explained in terms of superinfection. The main contribution of this chapter was a study of the interactions between a dominant virus, which spreads inside the population of a less powerful virus, both being the contagions of interest. The advantage of the SIS-model is clear: Markov theory is at our disposal and noise in data can be completely eliminated if synthetic networks are analyzed. Thus, the state of the infection processes is perfectly clear and the contagions are directly observable all the time. However, the simplified nature and the assumptions of the underlying SIS-processes do not correspond with the complex systems that are encountered in application domains. Nevertheless, the superinfection has been observed in biological system (e.g. hyperparasites [177, 178]) and insights from the evolution of the nested SIS-processes might provide first explanations for observable behavior. In particular, we have outlined the circumstances under which a coexistence of both viruses is possible. The superinfection rate was identified as the most influential parameter, which determines whether extinction cycles are observed or coexistence is stable. Thus, if we observe sudden extinction cycles in complex systems, in which a nested propagation process is assumed, a decrease in the spreading power of the dominant contagion could stabilize the system.

7.1.2. Analyzed Systems

After the effects of the contagion have become clearly traceable within the data, the next two questions to answer are:

1. Where does the contagion spread to? (structure)

2. How does the contagion spread? (dynamics)

Typically, the nodes of a network are assumed to host the contagion while the paths of the propagation are given by links. Challenges appear if the network is either implicit and/or changes rapidly over time.

- The network of players in Chapter 2 is determined by the 5vs5 nature of the game. Thus, a direct exchange of information happens only on the level of a single match within the teams (or between the competing teams respectively). Long-term transfers are possible if players get inserted in different teams and play multiple matches over time.

- In Chapter 3, the network is not given but has to be constructed from the observed brain signals. Depending on the thresholding deployed, different motifs arise (see Section 3.4).

- Concerning Chapter 4, the IP address space is dynamic and does not correspond 1:1 with susceptible host machines. Depending on up- and downtimes, nodes representing IP-addresses in this network might appear, disappear or reappear representing different machines.

To overcome these challenges, reasonable assumptions about the networks have to be made. The transfer entropy (in the form of phase transfer entropy) has been shown as a meaningful measure for the mutual influence in brain activity between different regions of interest and allows us to infer a basic (functional) network structure within the brain. The main contribution of Chapter 3 was the discovery of a stable and highly over-expressed motif, the bi-directional two-hop path, which subsequently was shown to be the main building block for facilitating information transfer. A clustering based on this motif revealed a higher order organization of the brain, explaining how posterior and frontal brain regions communicate.

The transfer entropy has been applicable to the toxicity data from Chapter 2 as well, revealing a different structure. While the phase transfer entropy in the brain seems approximately normal distributed (see Figure 3.2), the transfer entropy between players has been found to be exponential (see Figure 2.9). This discovery highlights that the influence (and henceforth the propagation of the contagion) between players has a much higher variance than the information flow within the brain. Thus, the time-dependent SIS model that was deployed in Chapter 4 is most likely not the best choice for explaining propagation of anti-social behavior, as it assumes a homogeneous population, in which the susceptibility of a node only depends on the (time-dependent) spreading and curing rates and the amount of links to infected neighbors. However, the exponential distribution shows that a large number of players are nearly immune to toxic influence, while a small number in the tail of the distribution is highly susceptible. Consequently, a spreading model for toxicity has to consider the temporal structure of the player network and the heterogeneous susceptibility of the players.

Next to the transfer entropy, clustering has been a helpful concept as well. While the motif-based clustering of the brain network provided a hard partition of spatial related brain regions, the topic model applied to the gaming data provided a softer clustering, which allowed for overlaps and mixtures of identified topics. Thus, on a higher level, clustering methods and the closely related topic models are able to reveal hidden structures which are induced and enabled by spreading processes.

7.1.3. Automated Inference of Relations

As discussed, there are multiple techniques at our disposal to analyze and reveal spreading processes in our physical world, but the assumptions of the corresponding application domains are crucial. Conclusions about the spread of computer worms that assume a change in the applied countermeasures do not hold for brain activity in resting state, for which no change in the environment is expected. Vice versa, there is no strong reason to believe that the spread of a computer worm would be supported by motifs, which are nonetheless essential in the neurological domain and regarded as a possible explanation for the principles of integration and segregation in the brain. Thus, if concepts like clustering, transfer entropy or time-dependent SIS are to be transferred, they need careful adaptation and a critical assessment.

Chapter 6 presented a different approach for which domain-specific assumptions are less important. We showed that genetic programming can be a method of high potential for discovering previously unknown functional relations between network properties. One of this network properties, the diameter, puts a lower bound on any hop-based propagation scheme and is thus of utmost importance to know for any network in which hop-based propagation occurs. Symbolic regression based on genetic programming is capable to learn approximate formulas for the diameter from simple classes of networks like barbell graphs and augmented path graphs. While no exact solutions were obtained, the evolved formulas provided significant improvements over previously analytically derived bounds and generalized reasonably well to networks of various application domains. Thus, linking structure with dynamics can be automated by deploying such a technique. The two strongest limiting factors for the performance of symbolic regression are the selection of training networks and their corresponding features. While the procedure is prone to over-fit, generating relatively simple equations which hold up to some degree of error for a subclass of networks might already be extremely helpful in some application domains, as hidden relations become apparent and further research gets sparked.

7.2. FUTURE RESEARCH

Information propagation in complex networks is a wide topic which can be approached from multiple angles. The goal of this thesis was to showcase phenomena from different application domains and focus on techniques that will help understand the relationship between network structure and spreading dynamics. Naturally, future work is suggested in each application domain, as the introduction of the network-related concepts opens up new challenges:

- Research into toxicity is still at the beginning and closely related to studies of antisocial behavior in general, like trolling [179], manipulation of online discusions [180] and vandalism [181]. Although a sensitive topic, there is a clear mutual benefit in the collaboration between game developers and scientists. The collection of rich data about player behavior is absolutely fundamental. With access to longitudinal data, studies of effectiveness of countermeasures against the spread of anti-social behavior could be enabled. In order to corroborate the toxicity detection described in this thesis, a survey about toxic comments and their reception is needed. The current detection of toxicity relies on the interpretation of frequent n-grams and it would be helpful to crowd-source opinions about the toxicity in statements from a larger number of people. This would not only decrease subjective bias, but might open up ways to quantify toxicity by some score, which would be a further improvement over the binary detection presented.

- There have been studies who relate Alzheimer's disease and other neurological disorders to abnormal changes in the topology of brain networks [182, 183]. If synchronization between brain regions is disturbed by such conditions, the information flow within the brain should be impaired. If this hypothesis holds true, the clustering for patients suffering from such disorders would differ from those of healthy controls. Thus, clustering of information flow motifs might provide a way for diagnosis and the assessment of treatment. Also here, large-scale studies require more data and are sensitive in nature. It is necessary to develop effective ways to ensure the privacy of study subjects without hampering future studies.

- Threats for our cyber security by computer worms, bot-nets and malware become increasingly complex and develop at a rapid speed. We foresee the development of a new sub-discipline, which could be described as *cyber-epidemiology* in the near future. Development of sophisticated epidemic models that not only take time-dependence but also heterogeneity of hosts into account would pave the way for a deep understanding of rapid global dynamics of modern computer worms. Botnet takeover [102] is a scenario, in which a process similar to superinfection is at play.

> Further investigation into interactions, co-infection and superinfection dynamics might provide us with predictive models that can simulate worm outbreaks and evaluate the corresponding risk profiles.

On a more general level, some promising theoretical research questions arise as well. Regarding network epidemics, there seems to be a relation between the time-dependent SIS model and the SIR model, with the former being able to mimic the behavior of the latter. As the time-dependent SIS model is still, to some extent, analytically tractable, it could provide a general epidemic theory, encompassing both SIS and SIR as potential corner-cases.

Generally, further research into temporal networks [184] and their interaction with spreading processes is needed, as the assumption that a spreading process happens in a static network that does not undergo changes in nodes or links rarely holds for information propagation on longer time scales in real-world networks.

Lastly, there is a high potential for additional research into symbolic regression. Newly developed differential genetic programming frameworks [185] provide convenient ways to evolve numerical constants, which might significantly improve the quality of the evolved formulas. However, the impact of different training sets of networks is still largely unexplored. Additionally, a study in feature selection for this task, possibly exploiting the spectral representation of networks, could further improve performance. A breakthrough in this area would allow us to deploy a large collection of algorithms that have shown their potential in artificial intelligence research for challenges in complex networks. Potential results would enable us to learn more about information propagation, but might be equally helpful for analyzing other networks properties like robustness or community structures.

Bibliography

[1] J. Travers and S. Milgram, *The small world problem,* Psychology Today **1**, 61 (1967).

[2] D. J. Watts and S. H. Strogatz, *Collective dynamics of 'small-world' networks,* Nature **393**, 440 (1998).

[3] D. Easley and J. Kleinberg, *Networks, crowds, and markets: Reasoning about a highly connected world* (Cambridge University Press, 2010).

[4] J. D. L. Rivas and C. Fontanillo, *Protein–protein interactions essentials: Key concepts to building and analyzing interactome networks,* PLOS Computational Biology **6**, e1000807 (2010).

[5] C. J. Stam, A. Hillebrand, H. Wang, and P. Van Mieghem, *Emergence of modular structure in a large-scale brain network with interactions between dynamics and connectivity,* Frontiers in Computational Neuroscience **4**, 133 (2010).

[6] P. Kaluza, A. Kölzsch, M. T. Gastner, and B. Blasius, *The complex network of global cargo ship movements,* Journal of The Royal Society Interface **7**, 1093 (2010).

[7] V. Colizza, A. Barrat, M. Barthélemy, and A. Vespignani, *The role of the airline transportation network in the prediction and predictability of global epidemics,* Proceedings of the National Academy of Sciences **103**, 2015 (2006).

[8] E. Bompard, M. Masera, R. Napoli, and F. Xue, *Assessment of structural vulnerability for power grids by network performance based on complex networks,* in *Lecture Notes in Computer Science* (Springer Berlin Heidelberg, 2009) pp. 144–154.

[9] H. Wang, E. Van Boven, A. Krishnakumar, M. Hosseini, H. Van Hooff, T. Takema, N. Baken, and P. Van Mieghem, *Multi-weighted monetary transaction network,* Advances in Complex Systems **14**, 691 (2011).

[10] D. Krioukov, M. Kitsak, R. S. Sinkovits, D. Rideout, D. Meyer, and M. Boguñá, *Network cosmology,* Scientific Reports **2** (2012), 10.1038/srep00793.

[11] F. A. Azevedo, L. R. Carvalho, L. T. Grinberg, J. M. Farfel, R. E. Ferretti, R. E. Leite, W. J. Filho, R. Lent, and S. Herculano-Houzel, *Equal numbers of neuronal and nonneuronal cells make the human brain an isometrically scaled-up primate brain,* Journal of Comparative Neurology **513**, 532 (2009).

[12] A.-L. Barabási and R. Albert, *Emergence of scaling in random networks,* Science **286**, 509 (1999).

[13] R. Milo, S. Shen-Orr, S. Itzkovitz, N. Kashtan, D. Chklovskii, and U. Alon, *Network motifs: Simple building blocks of complex networks,* Science **298**, 824 (2002).

[14] J. Cheng, L. Adamic, P. A. Dow, J. M. Kleinberg, and J. Leskovec, *Can cascades be predicted?* in *Proceedings of the 23rd International Conference on World Wide Web* (ACM, New York, NY, USA, 2014) pp. 925–936.

[15] M. Märtens, S. Shen, A. Iosup, and F. A. Kuipers, *Toxicity detection in multiplayer online games,* in *International Workshop on Network and Systems Support for Games, NetGames 2015, Zagreb, Croatia, December 3-4, 2015* (IEEE, 2015) pp. 1–6.

[16] R. Games, *Our games,* `https://www.riotgames.com/our-games` (2017), accessed: August 2017.

[17] E. Zhao, *EEDAR Predicts the MOBA Landscape in 2016,* `http://goo.gl/bBKggU` (gamasutra.com) (2015), accessed: August 2017.

[18] e-Sports Earnings, `https://www.esportsearnings.com/tournaments` (2017), accessed: August 2017.

[19] F. Kuipers, R. Kooij, D. De Vleeschauwer, and K. Brunnström, *Techniques for measuring quality of experience,* in *Lecture Notes in Computer Science* (Springer, 2010) pp. 216–227.

[20] J. Lin, *More science behind shaping player behavior in online games,* `http://gdcvault.com/play/1022160/More-ScienceBehind-Shaping-Player` (2015), accessed: August 2017.

[21] D. Team, *Welcoming new players,* `http://blog.dota2.com/2017/07/welcoming-new-players/` (2017), accessed: August 2017.

[22] C. D. Manning, P. Raghavan, and H. Schütze, *Introduction to Information Retrieval* (Cambridge University Press, 2008).

[23] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, *Scikit-learn: Machine learning in Python,* Journal of Machine Learning Research **12**, 2825 (2011).

[24] J. Lafferty and D. Blei, *Topic models,* in *Text Mining* (Chapman and Hall/CRC, 2009) pp. 71–93.

[25] D. M. Blei, A. Y. Ng, and M. I. Jordan, *Latent dirichlet allocation,* Journal of Machine Learning Research **3**, 993 (2003).

[26] T. Schreiber, *Measuring information transfer,* Physical Review Letters **85**, 461 (2000).

[27] M. Wibral, B. Rahm, M. Rieder, M. Lindner, R. Vicente, and J. Kaiser, *Transfer entropy in magnetoencephalographic data: Quantifying information flow in cortical and cerebellar networks,* Progress in Biophysics and Molecular Biology **105**, 80 (2011).

[28] R. Vicente, M. Wibral, M. Lindner, and G. Pipa, *Transfer entropy—a model-free measure of effective connectivity for the neurosciences,* Journal of Computational Neuroscience **30**, 45 (2011).

[29] G. Ver Steeg and A. Galstyan, *Information transfer in social media,* in *Proceedings of the 21st international conference on World Wide Web* (ACM, 2012) pp. 509–518.

[30] P. K. Smith, J. Mahdavi, M. Carvalho, S. Fisher, S. Russell, and N. Tippett, *Cyberbullying: Its nature and impact in secondary school pupils,* Journal of Child Psychology and Psychiatry **49**, 376 (2008).

[31] A. H. Ivory and C. E. Kaestle, *The effects of profanity in violent video games on players' hostile expectations, aggressive thoughts and feelings, and other responses,* Journal of Broadcasting & Electronic Media **57**, 224 (2013).

[32] S. Sood, J. Antin, and E. Churchill, *Profanity use in online communities,* in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (ACM, 2012) pp. 1481–1490.

[33] J. Suler, *The online disinhibition effect,* International Journal of Applied Psychoanalytic Studies **2**, 184 (2005).

[34] T. Chesney, I. Coyne, B. Logan, and N. Madden, *Griefing in virtual worlds: causes, casualties and coping strategies,* Information Systems Journal **19**, 525 (2009).

[35] C. Y. Foo and E. M. I. Koivisto, *Defining grief play in MMORPGs: player and developer perceptions,* in *Proceedings of the 2004 ACM SIGCHI International Conference on Advances in computer entertainment technology*, ACM (ACM Press, 2004) pp. 245–250.

[36] H. Esmaeili and P. C. Woods, *Calm down buddy! it's just a game: Behavioral patterns observed among teamwork MMO participants in WARGAMING's world of tanks,* in *2016 22nd International Conference on Virtual System & Multimedia (VSMM)* (IEEE, 2016).

[37] M. V. Birk, B. Buttlar, J. T. Bowey, S. Poeller, S. C. Thomson, N. Baumann, and R. L. Mandryk, *The effects of social exclusion on play experience and hostile cognitions in digital games,* in *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems* (ACM Press, 2016).

[38] A. Tyack, P. Wyeth, and D. Johnson, *The appeal of MOBA games,* in *Proceedings of the 2016 Annual Symposium on Computer-Human Interaction in Play - CHI PLAY'16* (ACM Press, 2016).

[39] K. B. Shores, Y. He, K. L. Swanenburg, R. Kraut, and J. Riedl, *The identification of deviance and its impact on retention in a multiplayer game,* in *Proceedings of the 17th ACM conference on Computer supported cooperative work & social computing - CSCW'14* (ACM Press, 2014).

[40] J. Blackburn and H. Kwak, *STFU NOOB!: predicting crowdsourced decisions on toxic behavior in online games,* in *Proceedings of the 23rd international conference on World Wide Web* (ACM, 2014) pp. 877–888.

[41] H. Kwak and J. Blackburn, *Linguistic analysis of toxic behavior in an online video game,* in *Lecture Notes in Computer Science* (Springer International Publishing, 2015) pp. 209–217.

[42] J. Y. Shim, T. H. Kim, and S. W. Kim, *Decision support of bad player identification in MOBA games using PageRank based evidence accumulation and normal distribution based confidence interval,* International Journal of Multimedia and Ubiquitous Engineering **9**, 13 (2014).

[43] L. Page, S. Brin, R. Motwani, and T. Winograd, *The PageRank citation ranking: bringing order to the web.*, Tech. Rep. (Stanford University, Stanford, CA, 1998).

[44] Y. Ki, J. Woo, and H. K. Kim, *Identifying spreaders of malicious behaviors in online games,* in *Proceedings of the 23rd International Conference on World Wide Web* (ACM Press, 2014).

[45] J. Woo, A. R. Kang, and H. K. Kim, *The contagion of malicious behaviors in online games,* ACM SIGCOMM Computer Communication Review **43**, 543 (2013).

[46] M. Märtens, J. Meier, A. Hillebrand, P. Tewarie, and P. Van Mieghem, *Brain network clustering with information flow motifs,* Applied Network Science **2**, 25 (2017).

[47] J. Meier, M. Märtens, A. Hillebrand, P. Tewarie, and P. Van Mieghem, *Motif-based analysis of effective connectivity in brain networks,* in *Studies in Computational Intelligence* (Springer International Publishing, 2016) pp. 685–696.

[48] C. J. Stam and E. Van Straaten, *The organization of physiological brain networks,* Clinical Neurophysiology **123**, 1067 (2012).

[49] E. Bullmore and O. Sporns, *Complex brain networks: graph theoretical analysis of structural and functional systems,* Nature Reviews Neuroscience **10**, 186 (2009).

[50] A. Hillebrand, P. Tewarie, E. van Dellen, M. Yu, E. W. S. Carbo, L. Douw, A. A. Gouw, E. C. W. van Straaten, and C. J. Stam, *Direction of information flow in large-scale resting-state networks is frequency-dependent,* Proceedings of the National Academy of Sciences **113**, 3867 (2016).

[51] M. Rosenblum, A. Pikovsky, J. Kurths, C. Schäfer, and P. A. Tass, *Phase synchronization: from theory to data analysis,* Handbook of Biological Physics **4**, 279 (2001).

[52] M. Lobier, F. Siebenhühner, S. Palva, and J. M. Palva, *Phase transfer entropy: a novel phase-based measure for directed connectivity in networks coupled by oscillatory interactions,* Neuroimage **85**, 853 (2014).

[53] M. Paluš and A. Stefanovska, *Direction of coupling from phases of interacting oscillators: An information-theoretic approach,* Physical Review E **67**, 055201 (2003).

[54] M. Dauwan, E. van Dellen, L. van Boxtel, E. C. van Straaten, H. de Waal, A. W. Lemstra, A. A. Gouw, W. M. van der Flier, P. Scheltens, I. E. Sommer, and C. J. Stam, *EEG-directed connectivity from posterior brain regions is decreased in dementia with lewy bodies: a comparison with Alzheimer's disease and controls,* Neurobiology of Aging **41**, 122 (2016).

[55] J.-Y. Moon, U. Lee, S. Blain-Moraes, and G. A. Mashour, *General relationship of global topology, local dynamics, and directionality in large-scale brain networks,* PLOS Computational Biology **11**, e1004225 (2015).

[56] P. Tewarie, A. Hillebrand, E. van Dellen, M. M. Schoonheim, F. Barkhof, C. Polman, C. Beaulieu, G. Gong, B. W. van Dijk, and C. J. Stam, *Structural degree predicts functional network connectivity: A multimodal resting-state fMRI and MEG study,* Neuroimage **97**, 296 (2014).

[57] R. L. Buckner, J. R. Andrews-Hanna, and D. L. Schacter, *The brain's default network,* Annals of the New York Academy of Sciences **1124**, 1 (2008).

[58] G. M. Edelman and J. A. Gally, *Reentry: a key mechanism for integration of brain function,* Frontiers in Integrative Neuroscience **7**, 63 (2013).

[59] T. Van Kerkoerle, M. W. Self, B. Dagnino, M.-A. Gariel-Mathis, J. Poort, C. Van Der Togt, and P. R. Roelfsema, *Alpha and gamma oscillations characterize feedback and feedforward processing in monkey visual cortex,* Proceedings of the National Academy of Sciences **111**, 14332 (2014).

[60] A. M. Bastos, J. Vezoli, C. A. Bosman, J.-M. Schoffelen, R. Oostenveld, J. R. Dowdall, P. De Weerd, H. Kennedy, and P. Fries, *Visual areas exert feedforward and feedback influences through distinct frequency channels,* Neuron **85**, 390 (2015).

[61] A. Sirota, S. Montgomery, S. Fujisawa, Y. Isomura, M. Zugaro, and G. Buzsáki, *Entrainment of neocortical neurons and gamma oscillations by the hippocampal theta rhythm,* Neuron **60**, 683 (2008).

[62] O. Sporns and R. Kötter, *Motifs in brain networks,* PLOS Biology **2**, e369 (2004).

[63] C. J. Honey, R. Kötter, M. Breakspear, and O. Sporns, *Network structure of cerebral cortex shapes functional connectivity on multiple time scales,* Proceedings of the National Academy of Sciences **104**, 10240 (2007).

[64] O. Sporns, C. J. Honey, and R. Kötter, *Identification and classification of hubs in brain networks,* PLOS ONE **2**, e1049 (2007).

[65] D. Battaglia, A. Witt, F. Wolf, and T. Geisel, *Dynamic effective connectivity of inter-areal brain circuits,* PLOS Computational Biology **8**, e1002438 (2012).

[66] F. Battiston, V. Nicosia, M. Chavez, and V. Latora, *Multilayer motif analysis of brain networks,* Chaos: An Interdisciplinary Journal of Nonlinear Science **27**, 047404 (2017).

[67] E. J. Friedman, K. Young, G. Tremper, J. Liang, A. S. Landsberg, and N. S. and, *Directed network motifs in Alzheimer's disease and mild cognitive impairment,* PLOS ONE **10**, e0124453 (2015).

[68] S. Mangan and U. Alon, *Structure and function of the feed-forward loop network motif,* Proceedings of the National Academy of Sciences **100**, 11980 (2003).

[69] M. Kondoh, *Building trophic modules into a persistent food web,* Proceedings of the National Academy of Sciences **105**, 16631 (2008).

[70] C. M. Schneider, V. Belik, T. Couronne, Z. Smoreda, and M. C. González, *Unravelling daily human mobility motifs,* Journal of the Royal Society Interface **10**, 20130246 (2013).

[71] A. R. Benson, D. F. Gleich, and J. Leskovec, *Higher-order organization of complex networks,* Science **353**, 163 (2016).

[72] M. Hämäläinen, R. Hari, R. J. Ilmoniemi, J. Knuutila, and O. V. Lounasmaa, *Magnetoencephalography—theory, instrumentation, and applications to noninvasive studies of the working human brain,* Reviews of Modern Physics **65**, 413 (1993).

[73] A. Hillebrand, G. R. Barnes, J. L. Bosboom, H. W. Berendse, and C. J. Stam, *Frequency-dependent functional connectivity within resting-state networks: an atlas-based MEG beamformer solution,* Neuroimage **59**, 3909 (2012).

[74] G. Gong, Y. He, L. Concha, C. Lebel, D. W. Gross, A. C. Evans, and C. Beaulieu, *Mapping anatomical connectivity patterns of human cerebral cortex using in vivo diffusion tensor imaging tractography,* Cerebral Cortex **19**, 524 (2009).

[75] N. Tzourio-Mazoyer, B. Landeau, D. Papathanassiou, F. Crivello, O. Etard, N. Delcroix, B. Mazoyer, and M. Joliot, *Automated anatomical labeling of activations in SPM using a macroscopic anatomical parcellation of the MNI MRI single-subject brain,* Neuroimage **15**, 273 (2002).

[76] N. Kashtan, S. Itzkovitz, R. Milo, and U. Alon, *mfinder tool guide,* Department of Molecular Cell Biology and Computer Science and Applied Mathematics, Weizmann Institute of Science, Rehovot Israel, Tech Rep (2002).

[77] S. Maslov and K. Sneppen, *Specificity and stability in topology of protein networks,* Science **296**, 910 (2002).

[78] S. Emmons, S. Kobourov, M. Gallant, and K. Börner, *Analysis of network clustering algorithms and cluster quality metrics at scale,* PLOS ONE **11**, e0159161 (2016).

[79] M. R. Garey and D. S. Johnson, *Computers and intractability*, Vol. 29 (W.H. Freeman and Company, New York, 2002).

[80] P. Van Mieghem, *Graph spectra for complex networks* (Cambridge University Press, 2011).

[81] U. Von Luxburg, *A tutorial on spectral clustering,* Statistics and Computing **17**, 395 (2007).

[82] M. Fiedler, *Algebraic connectivity of graphs,* Czechoslovak mathematical journal **23**, 298 (1973).

[83] J. Leskovec and R. Sosič, *SNAP: A general-purpose network analysis and graph-mining library,* ACM Transactions on Intelligent Systems and Technology (TIST) **8**, 1 (2016).

[84] B. C. van Wijk, C. J. Stam, and A. Daffertshofer, *Comparing brain networks of different size and connectivity density using graph theory,* PLOS ONE **5**, e13701 (2010).

[85] R. Noldus and P. Van Mieghem, *Assortativity in complex networks,* Journal of Complex Networks **3**, 507 (2015).

[86] L. Harriger, M. P. van den Heuvel, and O. Sporns, *Rich club organization of macaque cerebral cortex and its role in network communication,* PLOS ONE **7**, e46497 (2012).

[87] R. Vicente, L. L. Gollo, C. R. Mirasso, I. Fischer, and G. Pipa, *Dynamical relaying can yield zero time lag neuronal synchrony despite long conduction delays,* Proceedings of the National Academy of Sciences **105**, 17157 (2008).

[88] L. L. Gollo and M. Breakspear, *The frustrated brain: from dynamics on motifs to communities and networks,* Philosophical Transactions of the Royal Society B: Biological Sciences **369**, 20130532 (2014).

[89] L. L. Gollo, C. Mirasso, O. Sporns, and M. Breakspear, *Mechanisms of zero-lag synchronization in cortical motifs,* PLOS Computational Biology **10**, e1003548 (2014).

[90] L. L. Gollo, A. Zalesky, R. M. Hutchison, M. van den Heuvel, and M. Breakspear, *Dwelling quietly in the rich club: brain network determinants of slow cortical fluctuations,* Philosophical Transactions of the Royal Society B: Biological Sciences **370**, 20140165 (2015).

[91] Y. Wei, X. Liao, C. Yan, Y. He, and M. Xia, *Identifying topological motif patterns of human brain functional networks,* Human Brain Mapping **38**, 2734 (2017).

[92] O. Sporns, D. R. Chialvo, M. Kaiser, and C. C. Hilgetag, *Organization, development and function of complex brain networks,* Trends in Cognitive Sciences **8**, 418 (2004).

[93] G. Tononi, G. M. Edelman, and O. Sporns, *Complexity and coherency: integrating information in the brain,* Trends in Cognitive Sciences **2**, 474 (1998).

[94] V. P. Zhigulin, *Dynamical motifs: Building blocks of complex dynamics in sparsely connected random networks,* Physical Review Letters **92**, 238701 (2004).

[95] M. P. van den Heuvel, R. S. Kahn, J. Goñi, and O. Sporns, *High-cost, high-capacity backbone for global brain communication,* Proceedings of the National Academy of Sciences **109**, 11372 (2012).

[96] J. Meier, X. Zhou, A. Hillebrand, P. Tewarie, C. J. Stam, and P. Van Mieghem, *The epidemic spreading model and the direction of information flow in brain networks,* Neuroimage **152**, 639 (2017).

[97] M. Märtens, H. Asghari, M. van Eeten, and P. Van Mieghem, *A time-dependent SIS-model for long-term computer worm evolution,* in *2016 IEEE Conference on Communications and Network Security (CNS)* (IEEE, 2016).

[98] S. Qing and W. Wen, *A survey and trends on Internet worms,* Computers & Security **24**, 334 (2005).

[99] Y. Wang, S. Wen, Y. Xiang, and W. Zhou, *Modeling the propagation of worms in networks: A survey,* IEEE Communications Surveys & Tutorials **16**, 942 (2014).

[100] J. Shearer, *W32.downadup,* (2008).

[101] D. Goodin, *Superworm seizes 9m PCs, 'stunned' researchers say,* `http://www.theregister.co.uk/2009/01/16/9m_downadup_infections` (2009), accessed: September 2017.

[102] B. Stone-Gross, M. Cova, L. Cavallaro, B. Gilbert, M. Szydlowski, R. Kemmerer, C. Kruegel, and G. Vigna, *Your botnet is my botnet: analysis of a botnet takeover,* in *Proceedings of the 16th ACM conference on Computer and communications security* (ACM, 2009) pp. 635–647.

[103] W. O. Kermack and A. G. McKendrick, *A contribution to the mathematical theory of epidemics,* Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences **115**, 700 (1927).

[104] M. E. J. Newman, *Spread of epidemic disease on networks,* Physical Review E **66**, 016128 (2002).

[105] R. M. Anderson and R. M. May, *Infectious diseases of humans*, Vol. 1 (Oxford University Press, Oxford, 1991).

[106] R. Pastor-Satorras, C. Castellano, P. Van Mieghem, and A. Vespignani, *Epidemic processes in complex networks,* Reviews of Modern Physics **87**, 925 (2015).

[107] P. Van Mieghem, *Performance Analysis of Complex Networks and Systems* (Cambridge University Press, 2014).

[108] P. Van Mieghem, *Epidemic phase transition of the SIS type in networks,* EPL (Europhysics Letters) **97**, 48004 (2012).

[109] P. Van Mieghem, J. Omic, and R. Kooij, *Virus spread in networks,* IEEE/ACM Transactions on Networking (TON) **17**, 1 (2009).

[110] J. O. Kephart and S. R. White, *Directed-graph epidemiological models of computer viruses,* in *Proceedings of the IEEE Computer Society Symposium on Research in Security and Privacy*, IEEE (WORLD SCIENTIFIC, 1991) pp. 343–359.

[111] P. Van Mieghem, *SIS epidemics with time-dependent rates describing ageing of information spread and mutation of pathogens*, Tech. Rep. Report20140615 (Delft University of Technology, 2014).

[112] Maxmind, *GeoIP2 country database,* `http://maxmind.com/app/geoip_country` (2017), accessed: September 2017.

[113] H. Asghari, *pyasn,* `https://github.com/hadiasghari/pyasn` (2017), accessed: September 2017.

[114] M. Abu Rajab, J. Zarfoss, F. Monrose, and A. Terzis, *My botnet is bigger than yours (maybe, better than yours): why size estimates remain challenging,* in *Proceedings of the 1st USENIX Workshop on Hot Topics in Understanding Botnets, Cambridge, USA* (2007).

[115] G. C. M. Moura, C. Ganan, Q. Lone, P. Poursaied, H. Asghari, and M. van Eeten, *How dynamic is the ISPs address space? towards internet-wide DHCP churn estimation,* in *2015 IFIP Networking Conference (IFIP Networking)*, IFIP (IEEE, 2015).

[116] P. Porras, H. Saidi, and V. Yegneswaran, *An analysis of Conficker's logic and rendezvous points,* Computer Science Laboratory, SRI International, Tech. Rep (2009).

[117] H. Asghari, M. Ciere, and M. van Eeten, *Post-mortem of a zombie: Conficker cleanup after six years,* in *24th USENIX Security Symposium (USENIX Security 15)* (USENIX Association, Washington, D.C., 2015) pp. 1–16.

[118] G. Serazzi and S. Zanero, *Computer virus propagation models,* in *Performance Tools and Applications to Networked Systems* (Springer, 2004) pp. 26–50.

[119] T. M. Chen and J.-M. Robert, *Worm epidemics in high-speed networks,* Computer **37**, 48 (2004).

[120] S. Fei, L. Zhaowen, and M. Yan, *A survey of internet worm propagation models,* in *2nd IEEE International Conference on Broadband Network & Multimedia Technology (IC-BNMT09)*, IEEE (IEEE, 2009) pp. 453–457.

[121] F. D. Sahneh, C. Scoglio, and P. Van Mieghem, *Generalized epidemic mean-field model for spreading processes over multilayer complex networks,* IEEE/ACM Transactions on Networking (TON) **21**, 1609 (2013).

[122] S. Peng, S. Yu, and A. Yang, *Smartphone malware and its propagation modeling: A survey,* IEEE Communications Surveys & Tutorials **16**, 925 (2014).

[123] S. Peng, G. Wang, and S. Yu, *Modeling the dynamics of worm propagation using two-dimensional cellular automata in smartphones,* Journal of Computer and System Sciences **79**, 586 (2013).

[124] C. C. Zou, W. Gong, and D. Towsley, *Code red worm propagation modeling and analysis,* in *Proceedings of the 9th ACM conference on Computer and communications security* (ACM, 2002) pp. 138–147.

[125] C. Zhang, S. Zhou, and B. M. Chain, *Hybrid Epidemics — A Case Study on Computer Worm Conficker,* PLOS ONE **10** (2015), 10.1371/journal.pone.0127478.

[126] M. Märtens, R. van de Bovenkamp, and P. Van Mieghem, *Superinfection in networks,* in *Proceedings of the 11th International Conference on Signal-Image Technology & Internet-Based Systems (SITIS)*, IEEE (IEEE, 2015) pp. 413–420.

[127] P. Van de Perre, M. Segondy, V. Foulongne, A. Ouedraogo, I. Konate, J.-M. Huraux, P. Mayaud, and N. Nagot, *Herpes simplex virus and HIV-1: deciphering viral synergy,* The Lancet Infectious Diseases **8**, 490 (2008).

[128] M. Bailey, E. Cooke, F. Jahanian, and D. Watson, *The blaster worm: Then and now,* IEEE Security and Privacy Magazine **3**, 26 (2005).

[129] G. Bransfield, *The Welchia Worm,* Global Information Assurance Certification , 27 (2004).

[130] M. A. Nowak and R. M. May, *Superinfection and the evolution of parasite virulence,* Proceedings of the Royal Society of London. Series B: Biological Sciences **255**, 81 (1994).

[131] R. M. May and M. A. Nowak, *Coinfection and the evolution of parasite virulence,* Proceedings of the Royal Society of London. Series B: Biological Sciences **261**, 209 (1995).

[132] J. Mosquera and F. R. Adler, *Evolution of virulence: a unified framework for coinfection and superinfection,* Journal of Theoretical Biology **195**, 293 (1998).

[133] F. D. Sahneh and C. Scoglio, *Competitive epidemic spreading over arbitrary multilayer networks,* Physical Review E **89**, 062817 (2014).

[134] B. Karrer and M. E. J. Newman, *Competing epidemics on complex networks,* Physical Review E **84**, 036106 (2011).

[135] B. A. Prakash, A. Beutel, R. Rosenfeld, and C. Faloutsos, *Winner takes all: competing viruses or ideas on fair-play networks,* in *Proceedings of the 21st international conference on World Wide Web*, ACM (ACM Press, 2012) pp. 1037–1046.

[136] R. van de Bovenkamp, F. Kuipers, and P. Van Mieghem, *Domination-time dynamics in susceptible-infected-susceptible virus competition on networks,* Physical Review E **89**, 042818 (2014).

[137] M. E. J. Newman and C. R. Ferrario, *Interacting epidemics and coinfection on contact networks,* PLOS ONE **8**, e71321 (2013).

[138] Q. Wu, M. Small, and H. Liu, *Superinfection behaviors on scale-free networks with competing strains,* Journal of Nonlinear Science **23**, 113 (2013).

[139] F. Prado, A. Sheih, J. D. West, and B. Kerr, *Coevolutionary cycling of host sociality and pathogen virulence in contact networks,* Journal of Theoretical Biology **261**, 561 (2009).

[140] G. E. Leventhal, A. L. Hill, M. A. Nowak, and S. Bonhoeffer, *Evolution and emergence of infectious diseases in theoretical and real-world networks,* Nature Communications **6**, 6101 (2015).

[141] S. A. Myers and J. Leskovec, *Clash of the contagions: Cooperation and competition in information diffusion,* in *IEEE 12th International Conference on Data Mining (ICDM)*, IEEE (IEEE, 2012) pp. 539–548.

[142] J. Leskovec, L. Backstrom, and J. Kleinberg, *Meme-tracking and the dynamics of the news cycle,* in *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, ACM (ACM Press, 2009) pp. 497–506.

[143] A. Mirshahvalad, A. V. Esquivel, L. Lizana, and M. Rosvall, *Dynamics of interacting information waves in networks,* Physical Review E **89**, 012809 (2014).

[144] P. L. Simon, M. Taylor, and I. Z. Kiss, *Exact epidemic models on graphs using graph-automorphism driven lumping,* Journal of Mathematical Biology **62**, 479 (2011).

[145] P. Van Mieghem and E. Cator, *Epidemics in networks with nodal self-infection and the epidemic threshold,* Physical Review E **86**, 016116 (2012).

[146] R. Pastor-Satorras and A. Vespignani, *Epidemic dynamics and endemic states in complex networks,* Physical Review E **63**, 066117 (2001).

[147] F. Campillo and C. Lobry, *Effect of population size in a prey-predator model,* arXiv preprint arXiv:1111.6460 (2011).

[148] P. Van Mieghem, *Decay towards the overall-healthy state in SIS epidemics on networks,* arXiv preprint arXiv:1310.3980 (2013).

[149] M. Märtens, F. Kuipers, and P. Van Mieghem, *Symbolic regression on network properties,* in *Lecture Notes in Computer Science* (Springer International Publishing, 2017) pp. 131–146.

[150] P. Van Mieghem, *Graph eigenvectors, fundamental weights and centrality metrics for nodes in networks,* arXiv preprint arXiv:1401.4580 (2014).

[151] M. D. Schmidt and H. Lipson, *Distilling free-form natural laws from experimental data,* Science **324**, 81 (2009).

[152] S. H. Strogatz, *Nonlinear Dynamics and Chaos: With Applications to Physics, Biology, Chemistry, and Engineering*, Studies in Nonlinearity (Westview Press, 2014).

[153] M. D. Schmidt and H. Lipson, *Solving iterated functions using genetic programming,* in *Proceedings of the 11th Annual Conference Companion on Genetic and Evolutionary Computation Conference: Late Breaking Papers*, ACM (ACM Press, 2009) pp. 2149–2154.

[154] T. Menezes and C. Roth, *Symbolic regression of generative network models,* Scientific Reports **4** (2014), 10.1038/srep06284.

[155] T. McConaghy, *FFX: Fast, scalable, deterministic symbolic regression technology,* in *Genetic and Evolutionary Computation* (Springer, 2011) pp. 235–260.

[156] J. R. Koza, *Genetic programming: on the programming of computers by means of natural selection*, Vol. 1 (MIT press, 1992).

[157] T. McConaghy, T. Eeckelaert, and G. Gielen, *CAFFEINE: Template-free symbolic model generation of analog circuits via canonical form functions and genetic programming,* in *Proceedings of the conference on Design, Automation and Test in Europe*, Vol. 2 (IEEE Computer Society, 2005) pp. 1082–1087.

[158] D. A. Augusto and H. J. Barbosa, *Symbolic regression via genetic programming,* in *Proceedings. Vol.1. Sixth Brazilian Symposium on Neural Networks*, IEEE (IEEE Comput. Soc, 2000) pp. 173–178.

[159] J. F. Miller, *Cartesian genetic programming* (Springer, 2011).

[160] J. F. Miller, *Cartesian genetic programming (webpage),* `http://www.cartesiangp.co.uk` (2017), accessed: August 2017.

[161] S. Harding and J. F. Miller, *Evolution of robot controller using cartesian genetic programming,* in *Lecture Notes in Computer Science* (Springer, 2005) pp. 62–73.

[162] M. M. Khan and G. M. Khan, *A novel neuroevolutionary algorithm: Cartesian genetic programming evolved artificial neural network (cgpann),* in *Proceedings of the 8th International Conference on Frontiers of Information Technology* (ACM, 2010) p. 48.

[163] S. Harding, V. Graziano, J. Leitner, and J. Schmidhuber, *MT-CGP: Mixed type cartesian genetic programming,* in *Proceedings of the 14th annual conference on Genetic and evolutionary computation*, ACM (ACM Press, 2012) pp. 751–758.

[164] J. F. Miller, *Evolution of digital filters using a gate array model,* in *Evolutionary Image Analysis, Signal Processing and Telecommunications* (Springer, 1999) pp. 17–30.

[165] Z. Vasicek and L. Sekanina, *Evolutionary approach to approximate digital circuits design,* IEEE Transactions on Evolutionary Computation **19**, 432 (2015).

[166] Z. Vasicek, *Cartesian GP in optimization of combinational circuits with hundreds of inputs and thousands of gates, Lecture Notes in Computer Science*, Genetic Programming , 139 (2015).

[167] J. F. Miller and S. L. Smith, *Redundancy and computational efficiency in cartesian genetic programming,* IEEE Transactions on Evolutionary Computation **10**, 167 (2006).

[168] A. J. Turner and J. F. Miller, *Cartesian genetic programming: Why no bloat?* in *Lecture Notes in Computer Science* (Springer, 2014) pp. 222–233.

[169] N. Alon and V. D. Milman, *$\lambda 1$, isoperimetric inequalities for graphs, and superconcentrators,* Journal of Combinatorial Theory, Series B **38**, 73 (1985).

[170] B. Mohar, *Eigenvalues, diameter, and mean distance in graphs,* Graphs and Combinatorics **7**, 53 (1991).

[171] F. R. K. Chung, V. Faber, and T. A. Manteuffel, *An upper bound on the diameter of a graph from eigenvalues associated with its Laplacian,* SIAM Journal on Discrete Mathematics **7**, 443 (1994).

[172] E. R. Van Dam and W. H. Haemers, *Eigenvalues and the diameter of graphs,* Linear and Multilinear Algebra **39**, 33 (1995).

[173] B. Mohar, *Isoperimetric numbers of graphs,* Journal of Combinatorial Theory, Series B **47**, 274 (1989).

[174] B. Bollobás, *Random graphs* (Springer, 1998).

[175] B. W. Goldman and W. F. Punch, *Analysis of cartesian genetic programming's evolutionary mechanisms,* IEEE Transactions on Evolutionary Computation **19**, 359 (2015).

[176] R. A. Rossi and N. K. Ahmed, *An interactive data repository with visual analytics,* ACM SIGKDD Explorations Newsletter **17**, 37 (2016).

[177] E. H. Poelman, M. Bruinsma, F. Zhu, B. T. Weldegergis, A. E. Boursault, Y. Jongema, J. J. A. van Loon, L. E. M. Vet, J. A. Harvey, and M. Dicke, *Hyperparasitoids use herbivore-induced plant volatiles to locate their parasitoid host,* PLOS Biology **10**, e1001435 (2012).

[178] B. S. Toguebaye, Y. Quilichini, P. M. Diagne, and B. Marchand, *Ultrastructure and development of nosema podocotyloidis n. sp. (microsporidia), a hyperparasite of podocotyloides magnatestis (trematoda), a parasite of parapristipoma octolineatum (teleostei)*. Parasite (Paris, France) **21**, 44 (2014).

[179] J. Cheng, M. Bernstein, C. Danescu-Niculescu-Mizil, and J. Leskovec, *Anyone can become a troll: Causes of trolling behavior in online discussions,* in *Proceedings of the 2017 ACM Conference on Computer Supported Cooperative Work and Social Computing*, CSCW '17 (ACM, New York, NY, USA, 2017) pp. 1217–1230.

[180] T. Solorio, R. Hasan, and M. Mizan, *A case study of sockpuppet detection in wikipedia,* in *Workshop on Language Analysis in Social Media (LASM) at NAACL HLT* (2013) pp. 59–68.

[181] S. Kumar, F. Spezzano, and V. Subrahmanian, *VEWS: A wikipedia vandal early warning system,* in *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '15 (ACM, New York, NY, USA, 2015) pp. 607–616.

[182] C. J. Stam, W. de Haan, A. Daffertshofer, B. F. Jones, I. Manshanden, A. M. van Cappellen van Walsum, T. Montez, J. P. A. Verbunt, J. C. de Munck, B. W. van Dijk, H. W. Berendse, and P. Scheltens, *Graph theoretical analysis of magnetoencephalographic functional connectivity in Alzheimer's disease,* Brain **132**, 213 (2008).

[183] C. J. Stam, *Use of magnetoencephalography (MEG) to study functional brain networks in neurodegenerative disorders,* Journal of the Neurological Sciences **289**, 128 (2010).

[184] N. Masuda and R. Lambiotte, *A Guide to Temporal Networks* (World Scientific, 2016).

[185] D. Izzo, F. Biscani, and A. Mereta, *Differentiable genetic programming,* in *Lecture Notes in Computer Science* (Springer International Publishing, 2017) pp. 35–51.

A

Word Annotations

This appendix contains details and word lists for the annotation system introduced in Section 2.3.

The reader is advised to examine the following lists at his own discretion.

A.1. Non-latin

Special characters, typically from a non-latin language family. All words which included a symbol that was not part of the extended ASCII encoding were marked as non-latin.

A.2. Praise

Acts of courtesy, kindness, sport spirit or gratitude. list:

’gl’,’hf’,’glhf’,’gg’,’wp’,’thx’,’gj’,’nice’,’ty’

A.3. Bad

Profanity, swear words, insults and other inappropriate language. We used the list available at: `http://www.cs.cmu.edu/~biglou/resources/bad-words.txt`. As it contains 1384 bad words, we refrain from listing them here completely. Additionally to this list, we used the letterset method to mark all words bad constructible out of

’noob’,’noobs’,’retard’,’retards’,’tard’,’tards’,’idiot’,’idiots’,’stupid’,’fu’,’nap’,’moron’, ’morons’

A.4. Laughter

Acronyms expressing laughter and excitement. All words constructible by the letterset

’hahahaha’,’lol’,’rofl’,’lmao’

A.5. Smiley

Emoticon, symbols resembling faces or emotions. Matching pattern: the word begins with either ’:’ or ’;’ and is no longer than 6 characters. Additionally, the following list was applied:

’=)’, ’=))’, ’=(’, ’=D’, ’=/’, ’=; ’=)))’, ’=?’, ’=.=’, ’=P’, ’=))))’, ’=]’, ’=O’, ’=)))))’, ’-.-’, ’-_-’, ’-.-”, ’-,-’, ’-.-'”, ’-..-’, ’-_-”, ’-.-*’, ’-__-’, ’>.<’, ’>_>’, ’>.>’, ’>_<’, ’>:D’, ’<.<’, ’^ ^’, ’^_^’, ’^ ^ ^’, ’^.^’, ’^ ^”, ’^-^’, ’._.’, ’TT’, ’T_T’, ’xd’, ’XD’, ’xD’, ’oO’, ’Oo’, ’O_O’, ’o_o’, ’o_O’, ’O_o’

A.6. Symbol

Symbols or numbers. Matches if the word is either numeric or consists out of symbols which are neither a letter nor a digit (non-alphanumeric).

A.7. Slang

DotA-specific game-technical terms, mostly used to coordinate with the team. All words from the following 4 lists were selected for this category:

a) Terms based on the Dota 2 glossary (http://dota2.gamepedia.com/Glossary):

’bottle crow’, ’ck’, ’ss’, ’cm’, ’lothars’ ’lothars edge’, ’rat’, ’safe lane’, ’tp’, ’gg’, ’mid’, ’squishy’, ’feeder’, ’fade time’, ’kotl’, ’dot’, ’fortification’, ’ks’, ’micro’, ’bb’, ’long lane’, ’et’, ’suicide lane’, ’ggwp’, ’pl’, ’jungling’, ’qop’, ’summoner’, ’solo’, ’aoe’, ’ff’, ’mkb’, ’skill shot’, ’ww’, ’support’, ’proc’, ’bd’, ’potm’, ’tb’, ’ro3’, ’pull or pulling’, ’pet’, ’sf’, ’farm or farming’, ’sb’, ’cc’, ’fog of war’, ’wd’, ’deny’, ’dp’, ’cd’, ’ricer’, ’sk’, ’aa’, ’aggro’, ’dd’, ’debuff’, ’observer ward’ ’wards’, ’hh’, ’buildings’ ’barracks’, ’flash farming’, ’ulti’, ’carry’, ’caster’, ’wr’, ’bs’, ’ac’, ’cw’, ’throne’, ’wipe’, ’miss or mia’, ’nuke’, ’silence’, ’ms’, ’bot or bottom(btm)’, ’es’, ’roamer’, ’sheepstick’, ’wp’, ’tank’, ’fow’, ’buff’, ’pa’, ’cooldown (cd)’, ’hard carry’, ’cancer’, ’jungler’, ’necro’, ’true sight’, ’flash farming skill/ability’, ’stacking’, ’throw’, ’offlane’, ’am’, ’furion’, ’glhf’, ’hard lane’, ’short lane’, ’orb walk’, ’reuse’, ’cs’, ’first blood|fb’, ’recrow’, ’easy lane’, ’rax’, ’kite or kiting’, ’oom’, ’bkb’, ’brb’, ’roshan or rosh’, ’ta’, ’dps’, ’b’, ’disable’, ’top’, ’attackspeed’, ’pp’, ’burst or burst damage’, ’p’, ’sny’, ’chicken or chick’, ’crow’, ’rp’, ’ganking’ ’gank’, ’bot’ ’bots’, ’exp’, ’static farming’

b) Names and nicknames of the heroes in the game:

’admiral’, ’alchemist’, ’rexxar’, ’bm’, ’cent’, ’es’, ’omni’, ’panda’, ’sven’, ’tiny’, ’tc’, ’tp’, ’io’, ’wisp’, ’alch’, ’clock’, ’rattle’, ’dk’, ’huskar’, ’bristle’, ’bb’, ’phoenix’, ’tusk’, ’tuskar’, ’tres’, ’tresdin’, ’rizzrak’, ’es’, ’am’, ’sniper’, ’troll’, ’jugg’, ’juggernaut’, ’luna’, ’morph’, ’morphling’, ’meepo’, ’naga’, ’phantom’, ’pl’, ’potm’, ’riki’, ’rikimaru’, ’gyro’, ’drow’, ’traxex’, ’ta’, ’lanaya’, ’ursa’, ’venge’, ’vengeful’, ’gondar’, ’bh’, ’xin’, ’ember’, ’cm’, ’crystal’, ’rylai’, ’ench’, ’enchantress’, ’puck’, ’chen’, ’kotl’, ’zeus’, ’furi’, ’furion’, ’malfurion’, ’silencer’, ’nortrom’, ’lina’, ’slayer’, ’storm’, ’wind’, ’wr’, ’windrunner’, ’disruptor’, ’thrall’, ’ogre’, ’techies’, ’jakiro’, ’tinker’, ’rasta’, ’rhasta’, ’rubick’, ’rubik’, ’sky’, ’skywrath’, ’oracle’, ’axe’,

’chaos’, ’doom’, ’naix’, ’abaddon’, ’abba’, ’bane’, ’lycan’, ’night’, ’azgalor’, ’pit’, ’pudge’, ’sk’, ’slardar’, ’dirge’, ’undying’, ’tide’, ’tidehunter’, ’magnus’, ’bara’, ’spiritbreaker’, ’sand’, ’sand king’, ’crixalis’, ’blood’, ’clinkz’, ’brood’, ’nyx’, ’weaver’, ’mort’, ’pa’, ’mortred’, ’sf’, ’shadow’, ’shadow fiend’, ’terror’, ’terrorblade’, ’spectre’, ’mercurial’, ’veno’, ’venomancer’, ’viper’, ’razor’, ’slark’, ’void’, ’faceless’, ’medusa’, ’dusa’, ’zet’, ’arc’, ’arc warden’, ’bane’, ’dark’, ’dp’, ’death’, ’lion’, ’enigma’, ’lich’, ’necro’, ’necrolyte’, ’pugna’, ’oblivion’, ’desi’, ’od’, ’qop’, ’warlock’, ’shadow’, ’sd’, ’bat’, ’batrider’, ’dazzle’, ’kael’, ’invoker’, ’invo’, ’visage’, ’lesh’, ’leshrac’, ’wd’, ’witch doctor’, ’aa’, ’ww’, ’winter’, ’mirana’, ’mira’, ’trax’, ’husk’, ’legion’, ’jugger’, ’akasha’, ’crix’, ’kunka’, ’kunkka’, ’mother’, ’nerub’, ’rasta’, ’maiden’, ’shaker’, ’balanar’, ’dirge’

c) Names and nicknames of the items in the game:

’ac’, ’heart’, ’bkb’, ’aegis’, ’shiva’, ’shivas’, ’bloodstone’, ’linken’, ’linkens’, ’van’, ’vanguard’, ’bm’, ’blademail’, ’blade’, ’hood’, ’manta’, ’scythe’, ’sheep’, ’sheepstick’, ’orchid’, ’eul’, ’euls’, ’force’, ’force staff’, ’dagon’, ’necro’, ’necrobook’, ’agh’, ’aghs’, ’aghanims’, ’refresh’, ’refresher’, ’veil’, ’rod’, ’atos’, ’mek’, ’mekansm’, ’vlad’, ’vlads’, ’boots’, ’arcane’, ’manaboots’, ’aquila’, ’buckler’, ’branch’, ’ring’, ’basi’, ’basil’, ’basilius’, ’pipe’, ’urn’, ’head’, ’headress’, ’medallion’, ’drums’, ’janggo’, ’tranquil’, ’tranquils’, ’divine’, ’rapier’, ’mkb’, ’radiance’, ’radi’, ’butterfly’, ’buri’, ’basher’, ’battlefury’, ’abyss’, ’crystalys’, ’crit’, ’armlet’, ’lothar’, ’lothars’, ’etheral’, ’eblade’, ’sange’, ’yasha’, ’satanic’, ’mjollnir’, ’skadi’, ’helm’, ’mael’, ’maelstrom’, ’deso’, ’desolator’, ’mom’, ’diffusal’, ’hh’, ’halberd’, ’bot’, ’bots’, ’phase’, ’treads’, ’soul’, ’midas’, ’perseverance’, ’poorman’, ’bracer’, ’wraithband’, ’null’, ’wand’, ’stick’, ’gloves’, ’mask’, ’kelen’, ’kelens’, ’blink’, ’sobi’, ’boots’, ’gem’, ’cloak’, ’evasion’, ’ghost’, ’shadow’, ’clarity’, ’salve’, ’healing’, ’tango’, ’bottle’, ’ward’, ’obs’, ’sentries’, ’sentry’, ’dust’, ’smoke’, ’chicken’, ’bird’, ’donkey’, ’tp’, ’flying’, ’gauntlet’, ’gauntlets’, ’slippers’, ’mantle’, ’ironwood’, ’belt’, ’robe’, ’circ’, ’circlet’, ’orb’, ’claws’, ’claymore’, ’stout’, ’shield’, ’javelin’, ’mithril’, ’chain’, ’helm’, ’plate’, ’quelling’, ’blade’, ’relic’, ’void’, ’venom’, ’wards’, ’dagger’, ’courier’, ’meka’, ’agha’, ’maelstorm’

d) General terms to describe game mechanics:

’crow’, ’cs’, ’deny’, ’dive’, ’farm’, ’gang’, ’gank’, ’push’, ’juke’, ’pull’, ’juked’, ’juking’, ’pushed’, ’woods’, ’wood’, ’jungle’, ’lane’, ’off’, ’safe’, ’quas’, ’wex’, ’exort’, ’care’, ’rosh’, ’ult’, ’ulti’, ’missing’, ’miss’, ’bot’, ’top’, ’mid’, ’help’, ’stun’, ’sup’, ’support’, ’carry’, ’rune’, ’haste’, ’dd’, ’invis’, ’roam’, ’creep’, ’ancient’, ’ancients’, ’camp’, ’block’, ’disable’, ’miss’, ’report’, ’tower’, ’gg’, ’wp’, ’ez’, ’back’, ’bb’, ’bbb’, ’regen’, ’fountain’, ’base’, ’gj’, ’take’, ’mana’, ’oom’, ’omw’, ’back’, ’stay’, ’heal’, ’hex’, ’afk’, ’creeps’, ’farming’, ’roshan’, ’warded’, ’cares’, ’bottom’, ’ulty’, ’go’, ’rax’

A.8. COMMAND

In-game commands and control words that trigger certain effects.
pattern: word starts with either ’!’ or ’-’.

A.9. STOP

English stop word from the following list:
’a’, ’an’, ’another’, ’any’, ’certain’, ’each’, ’every’, ’her’, ’his’, ’its’, ’its’, ’my’, ’no’, ’our’, ’some’, ’that’, ’the’, ’their’, ’this’, ’and’, ’but’, ’or’, ’yet’, ’for’, ’nor’, ’so’, ’as’, ’aboard’, ’about’, ’above’, ’across’, ’after’, ’against’, ’along’, ’around, ’at’, ’before’, ’behind’, ’below’, ’beneath’, ’beside’, ’between’, ’beyond’, ’but’, ’by’, ’down’, ’during’, ’except’, ’following’, ’for’, ’from’, ’in’, ’inside’, ’into’, ’like’, ’minus’, ’near’, ’next’, ’of’, ’off’, ’on’, ’onto’, ’opposite’, ’out’, ’outside’, ’over’, ’past’, ’plus’, ’round’, ’since’, ’than’, ’through’, ’to’, ’toward’, ’under’, ’underneath’, ’unlike’, ’until’, ’up’, ’upon’, ’with’, ’without’, ’i’, ’u’, ’me’, ’you’, ’we’, ’is’, ’dont’, ’all’, ’y’, ’can’, ’it’, ’get’, ’have’, ’why’, ’need’, ’he’, ’come’, ’just’, ’not’, ’are’, ’they’, ’now’, ’got’, ’ok’, ’what’, ’was’, ’will’, ’do’, ’if’

A.10. TIMEMARK

Automatically generated time-stamp, attached to the chat in pause-mode.
pattern: matches ’[dd:dd]’ where d is a digit {0123456789}.

B

Toxic n-grams

This appendix lists the n-grams which were used to define toxic statements in Chapter 2. The n-grams were selected out of the 100 most frequent n-grams for $n = 1, 2, 3, 4$ by the following criteria:

1. if $n = 1$, the n-gram is toxic if it is a word that could - without context - be interpreted as an insult.

2. if $n > 2$, the n-gram is toxic, if it can be interpreted as an insult directed to another person.

The reader is advised to examine the following lists at his own discretion.

B.1. Toxic n-grams (n = 1)

’noob’, ’idiot’, ’fu’, ’retard’, ’noobs’, ’tard’, ’idiots’, ’stupid’, ’moron’, ’retards’, ’bitch’, ’tards’, ’nap’, ’fucker’, ’fucktard’, ’nooob’, ’motherfucker’, ’morons’, ’retarded’, ’noob?’, ’blind?’, ’gay’, ’blind’, ’kid’, ’asshole’, ’pussy’, ’sucker’, ’fuckers’, ’dumb’, ’fag’, ’idiot?’, ’retard?’, ’faggot’, ’loser’, ’nob’, ’nooobs’, ’bitches’, ’nooob’, ’dumbass’, ’fuuu’, ’fuu’, ’stupid?’, ’ass’, ’stupids’, ’cunt’

B.2. Toxic n-grams (n = 2)

(’mother’, ’fker’) (’noob’, ’shittttttttttttttt’) (’fker’, ’noob’) (’kkkkkkkkkkkk??????????????’, ’u’) (’fucking’, ’noob’) (’fuck’, ’u’) (’u’, ’suck’) (’fuck’, ’you’) (’u’, ’fucking’) (’fucking’, ’idiot’) (’fuck’, ’off’) (’a’, ’noob’) (’fucking’, ’retard’) (’you’, ’fucking’) (’fucking’, ’noobs’) (’noob’, ’team’) (’you’, ’suck’) (’u’, ’noob’) (’suck’, ’my’) (’piece’, ’of’) (’fucking’, ’tard’)

B.3. Toxic n-grams (n = 3)

(’mother’, ’fker’, ’noob’) (’fker’, ’noob’, ’shittttttttttttttt’) (’u’, ’such’, ’mother’) (’such’, ’mother’, ’fker’) (’piece’, ’of’, ’shit’) (’suck’, ’my’, ’dick’) (’what’, ’a’, ’noob’) (’shut’, ’the’,

’fuck’) (’u’, ’are’, ’noob’) (’the’, ’fuck’, ’u’) (’u’, ’fucking’, ’noob’) (’son’, ’of’, ’a’) (’fu’, ’fu’, ’fu’) (’are’, ’u’, ’fucking’) (’u’, ’are’, ’fucking’) (’he’, ’is’, ’noob’) (’are’, ’you’, ’fucking’) (’die’, ’in’, ’hell’) (’a’, ’fucking’, ’noob’) (’u’, ’suck’, ’so’) (’such’, ’a’, ’noob’) (’you’, ’are’, ’noob’) (’you’, ’are’, ’fucking’) (’u’, ’fucking’, ’idiot’) (’fucking’, ’noob’, ’team’) (’go’, ’fuck’, ’yourself’) (’so’, ’fucking’, ’bad’) (’u’, ’fucking’, ’retard’) (’fuck’, ’your’, ’mother’) (’noob’, ’noob’, ’noob’) (’you’, ’suck’, ’so’) (’u’, ’are’, ’idiot’)

B.4. Toxic n-grams (n = 4)

(’mother’, ’fker’, ’noob’, ’shitttttttttttttttt’) (’u’, ’such’, ’mother’, ’fker’) (’such’, ’mother’, ’fker’, ’noob’) (’shut’, ’the’, ’fuck’, ’up’) (’son’, ’of’, ’a’, ’bitch’) (’why’, ’the’, ’fuck’, ’u’) (’fu’, ’fu’, ’fu’, ’fu’) (’the’, ’fuck’, ’are’, ’you’) (’u’, ’suck’, ’so’, ’hard’) (’of’, ’shit’, ’you’, ’are’) (’the’, ’fuck’, ’are’, ’u’) (’piece’, ’of’, ’shit’, ’you’) (’fucking’, ’piece’, ’of’, ’shit’) (’u’, ’piece’, ’of’, ’shit’) (’what’, ’the’, ’fuck’, ’are’) (’noob’, ’noob’, ’noob’, ’noob’) (’you’, ’suck’, ’so’, ’hard’) (’saying’, ’piece’, ’of’, ’shit’) (’the’, ’fuck’, ’did’, ’u’) (’u’, ’are’, ’fucking’, ’noob’) (’you’, ’are’, ’a’, ’noob’) (’get’, ’the’, ’fuck’, ’out’) (’you’, ’are’, ’fucking’, ’noob’) (’u’, ’are’, ’a’, ’fucking’) (’you’, ’are’, ’so’, ’fucking’) (’u’, ’suck’, ’so’, ’much’) (’you’, ’piece’, ’of’, ’shit’) (’u’, ’cant’, ’do’, ’shit’) (’u’, ’are’, ’a’, ’noob’) (’u’, ’son’, ’of’, ’a’) (’retarded’, ’piece’, ’of’, ’shit’) (’what’, ’a’, ’fucking’, ’noob’) (’of’, ’bitch’, ’son’, ’of’) (’i’, ’hope’, ’u’, ’die’) (’a’, ’piece’, ’of’, ’shit’) (’u’, ’suck’, ’like’, ’hell’)

C

Number of Triangles in a Network

Theorem. *The number of closed triangles in an undirected network of N nodes is*

$$\blacktriangle_G = \frac{1}{6}\sum_{k=1}^{N}\lambda_k^3$$

where $\lambda_1, \ldots, \lambda_N$ are the eigenvalues of the adjacency matrix A of the network.

To prove the theorem, we first show how the powers of an adjacency matrix A^k are related to closed walks and connect this result to the spectral decomposition of A^k.

Lemma 1. *Let A be an adjacency matrix of a network and k be an integer. Then the entry $(A^k)_{ij}$ of A^k describes the number of walks of length k between nodes i and j.*

Proof. Induction on k: for $k = 1$ there can only be 1 or 0 possible walks between any pair of nodes i and j, depending whether there exists a link between i and j or not. Thus, $(A)_{ij}$ is the number of 1-walks between any nodes i and j.

Let us assume the theorem holds for all integers up to k. For any node ℓ, $(A^k)_{i\ell}$ is the number of k-walks from i to ℓ by the induction hypothesis. If node ℓ is adjacent to j, there are $(A^k)_{i\ell}$ walks of length $k+1$ from i to j with ℓ being the second-last node. If ℓ is not adjacent to j, there are zero walks of length $k+1$ from i to j with ℓ being the second-last node. Thus, the total number of $k+1$-walks between i and j is given by the summation of the $k+1$ walks over all possible nodes ℓ. This is the same as the definition of the matrix product:

$$(A^{k+1})_{ij} = \sum_{\ell=1}^{N}(A^k)_{i\ell}A_{\ell j}.$$

□

From this lemma follows that $(A^k)_{ii}$ describes the number of closed walks of length k from one node i back to itself. Each triangle in a network is a connected triplet of three different nodes a, b, c. There are 6 permutations of the ordering for these nodes, each corresponding to a different 3-walk. It follows that

$$\blacktriangle_G = \frac{1}{6}\sum_{i=1}^{N}(A^3)_{ii} = \frac{1}{6}\,\mathrm{trace}(A^3).$$

Note that the trace is the sum of all eigenvalues of a matrix:

$$\mathrm{trace}(A) = \sum_{i=1}^{N}\lambda_i.$$

Let $\Lambda = \mathrm{diag}(\lambda_i)$. By an eigenvalue decomposition we can write $A = X\Lambda X^{-1}$ where X is a matrix containing the corresponding normalized eigenvectors. It follows that

$$A^2 = (X\Lambda X^{-1})(X\Lambda X^{-1}) = X\Lambda(X^{-1}X)\Lambda X^{-1} = X\Lambda^2 X^{-1}.$$

With induction, one can show that

$$A^k = X\Lambda^k X^{-1}.$$

Consequently, it follows that

$$\mathrm{trace}(A^k) = \sum_{i=1}^{N}\lambda_i^k.$$

Thus, for $k = 3$ we conclude that

$$\blacktriangle_G = \frac{1}{6}\,\mathrm{trace}(A^3) = \frac{1}{6}\sum_{i=1}^{N}\lambda_i^3.$$

□

An immediate consequence of this theorem is that any two networks with the same spectrum must have the same number of triangles.

Acknowledgements

During my four years of research at the TU Delft I made contact with some remarkable people. Each of them had an impact on me and my thesis and I want to express my gratitude to them.

At first, my two supervisors and (co)-promotors are to be named. I am grateful that Piet provided me the encouragement and freedom to pursue my own, sometimes odd ideas. His rigor, working speed and compassion for science have been a great inspiration. Fernando was the one who originally hired me to the NAS group and has been guiding me from the beginning. His ever reassuring and optimistic attitude filled me with confidence and his kind advises lead me through some of the most difficult phases. It has been my honor to have been working with both of you.

Next I have to mention my direct office mates, with whom I shared a lot of my time. The first colleague I connected personally to was Ruud, who helped to kickstart my research and provided for some of my most memorable moments. The dutch parts of this thesis are owed to him. Hale has the unique talent to cheer me up even when I am stressed or gloomy. That and the fact that she frequently shared her food with me saved her a special place in my heart. Together with Xiangrong, Cong and Zhidong we made our office truly the best office of the floor.

The multiple members and associates of the NAS group have been great colleagues to me. Working together with Jil on the brain was one of the most effective and fruitful collaborations I had. Niels has always been there to support me with a smile every time I struggled with some technical challenges of my work. Norberts zenlike attitude towards life impressed me deeply and might have saved my mind from panic mode countless times. More thanks go to Song, Christian, Edgar, Rogier, Qiang, Chuan, Luxing, Stojan, Farabi, Annalisa, Evangelos, Joyce, Wendy, Marloes, Rani, Remco, Negar, Shruti, Aleksander, Huijuan, Karel, Yingli, Rob and Bastian. Drinks on Friday with you will never be forgotten.

Also outside NAS, there have been many people who gave me the strength to face the challenges of the PhD. I loved the Magic nights I had together with Luís, Dario, Francesco, Sante, Krzysztof and others, which not only provided fun but a lot of scientific inspiration. Special thanks go to Sante, who always welcomed me and raised my confidence during countless online tea sessions. Nuno, Nídia, Ingvild and John have accompanied me a long time by now and I will never forget the epic board game fun we had together. In general, cheers to all the former and current people from the ACT and ESA!

Caroline and Matze have been there for me when needed and we enjoyed some great ramen and sushi together! Viktoria has been a fantastic shrine to me that helped me to relax at stressful times.

Jeroen, Liesbeth and Wendy are among my Dutch friends who made my stay in this country enjoyable. Dank u wel!

Anja, Steffi and Amok provided me the breaks I needed from work and cheered me up in my darkest hours of doubt. Marcel is without question the best laning partner in Dota one could hope for and an amazing travel companion. I never received any toxicity from him! They are all special to me and the most loyal friends one could hope for.

In fact, I often wonder whether I really deserve such good friends, but despite my doubts and flaws, they still stick to me, regardless of what I do. I am truly grateful to have you all in my life! My sincerest apologies go to my friends whose names I have not listed explicitly at this point. You have not been forgotten and deserve my gratitude as well.

Finally, my parents have given me the care, love and support to enable me to develop my talents, visit universities and finally obtain this title. I thank my closest family, my brother Kai and my nephew Paul. You will always be with me no matter where I am. Thank you all for loving me.

Marcus Märtens
Delft, September 2017

DANKSAGUNGEN

Während meiner vier Jahre der Forschung an der TU Delft habe ich Kontakt mit ein paar besonderen Personen schließen können. Jede hatte einen Einfluss auf mich und diese Arbeit und ich möchte ihnen meinen Dank aussprechen.

Zunächst sind meine beiden Betreuer und (Co)-promotoren zu nennen. Ich bin dankbar für die Ermutigungen von Piet und die Freiheiten, welche er mir einberäumte, um meine eigenen oft seltsamen Ideen zu verfolgen. Seine Genauigkeit, Geschwindigkeit und Leidenschaft für die Wissenschaft waren stets eine große Inspiration für mich. Fernando war es, der mich zur NAS Gruppe gebracht und vom Anfang bis zum Ende begleitet hat. Seine stets zuversichtliche und optimistische Einstellung hat mir viel Sicherheit gegeben und seine freundlichen Ratschlägen führten mich durch manch schwierige Phasen. Es war mir eine Ehre, mit euch beiden zusammen zu arbeiten.

Als nächstes möchte ich meine Bürokollegen nennen, mit denen ich viel meiner Zeit verbrachte. Der erste Kollege, zu dem ich persönlich Kontakt aufbauen konnte, war Ruud. Er hat mir geholfen, meine Forschung in Gang zu setzen und war für viele erinnerungsvolle Momente zuständig. Die holländischen Teile dieser Arbeit verdanke ich ihm. Hale hat das einzigartige Talent, mich selbst dann noch aufzuheitern, wenn ich völlig gestresst oder niedergeschlagen bin. Dies, und die Tatsache, dass sie häufig ihr Essen mit mir geteilt hat, hat ihr einen besonderen Platz in meinen Herzen reserviert. Zusammen mit Xiangrong, Cong und Zhidong haben wir unser Büro wahrlich zum besten Büro des Flures gemacht.

Die vielen Mitglieder und Partner der NAS Gruppe waren großartige Kollegen für mich. Die Zusammenarbeit mit Jil am Gehirn war eine der effektivsten und fruchtbarsten Kollaborationen die ich hier hatte. Niels hat mich immer unterstützt und mit einem Lächeln empfangen, wenn ich mit technischen Problemen zu ihm kam. Norberts zen-artige Einstellung zum Leben hat mich zutiefst beeindrucked und meinen Verstand unzählige Male davor bewahrt, in Panik zu verfallen. Ich danke ebenso Song, Christian, Edgar, Rogier, Qiang, Chuan, Luxing, Stojan, Farabi, Annalisa, Evangelos, Joyce, Wendy, Marloes, Rani, Remco, Negar, Shruti, Aleksander, Huijuan, Karel, Yingli, Rob und Bastian. Die Freitagsdrinks mit euch werde ich nicht vergessen.

Auch außerhalb von NAS gab es viele Menschen, die mir die Stärke gegeben haben, es mit den Herausforderungen eines Doktorandenstudiums aufzunehmen. Ich liebte die Magic Nächte, die ich zusammen mit Luís, Dario, Francesco, Sante, Krzysztof und anderen verbrachte. Sie waren nicht nur sehr spaßig, sondern gaben eine Menge wissenschaftliche Inspiration. Ein besonderer Dank geht an Sante, der mich stets willkommen geheißen und mein Selbstvertrauen in zahllosen Online Tee Treffs aufgebaut hat. Nuno, Nídia, Ingvild und John haben mich jetzt bereits lange Zeit begleitet und niemals werde ich die legendären Brettspielabende mit euch vergessen. Allgemein, möchte ich sämtliche ehemaligen und aktuellen Mitglieder des ACT und der ESA grüßen!

Caroline und Matze waren für mich da, als ich sie brauchte und wir haben gemeinsam großartigen Ramen und Sushi genossen. Viktoria ist ein fantastischer Schrein, der mir in stressigen Zeiten half zu entspannen.

Jeroen, Liesbeth und Wendy sind meine holländischen Freunde, die meinen Aufenthalt in diesem Land erfreulich gestaltet haben. Dank u wel!

Anja, Steffi und Amok haben mir die nötigen Pausen von der Arbeit gegeben und mich in meinen dunkelsten Stunden voller Zweifel aufgeheitert und gestärkt. Marcel ist ohne Frage der beste Laning Partner in Dota den man sich wünschen kann und zudem ein fantastischer Reisebegleiter. Ich habe niemals Toxicity von ihm erfahren! Ihr seid alle besonders für mich und die treuesten Freunde, die man sich wünschen kann.

Tatsächlich frage ich mich oft, ob ich solch gute Freunde wirklich verdiene, aber trotz meiner Fehler und Zweifel, bleiben sie doch irgendwie bei mir, egal was ich mache. Ich bin wirklich dankbar, euch alle in meinem Leben zu wissen! Ich möchte mich aufrichtig bei meinen Freunden entschuldigen, deren Namen ich hier nicht explizit auflisten konnte. Ich habe euch nicht vergessen und danke euch ebenso.

Zu guter letzt, möchte ich meinen Eltern danken, welche mir die Sorgfalt, Liebe und Unterstützung gaben, um meine Talente zu entwickeln, Universitäten zu besuchen und schlußendlich diesen Doktortitel zu erringen. Ich danke meiner engsten Familie, meinem Bruder Kai und meinem Neffen Paul. Ihr seid immer bei mir, egal wo ich mich aufhalte. Danke, dass ihr mich liebt.

Marcus Märtens
Delft, September 2017

Curriculum Vitæ

Marcus Märtens

19-12-1984	Born in Aschersleben, Germany.

Education

1994–2005	Gymnasium Ascaneum, Aschersleben, Germany
2005–2008	Bachelor of Science, Computer Science University of Paderborn, Germany *Thesis:* Clustering von Genexpressionsdaten mit Hilfe von Kernmengen *Supervisor:* Prof. dr. Christian Sohler
2008–2011	Master of Science, Computer Science University of Paderborn, Germany *Thesis:* The Monotone Complexity of Computing k-Clique on Random Graphs *Supervisor:* Prof. dr. Johannes Blömer
2013–2017	PhD. Candidate Delft University of Technology, Netherlands Network Architectures and Services Group (NAS) *Thesis:* Information Propagation in Complex Networks *Promotor:* Prof. dr. ir. P. F. A. Van Mieghem *Supervisor:* Dr. ir. F. A. Kuipers

Professional Work Experience

2011	Summer Intern, Opera Software ASA, Norway
2012	Consultant, CGI Germany GmbH, Düsseldorf, Germany
2012–2013	Young Graduate Trainee, Advanced Concepts Team, European Space Agency, The Netherlands

List of Publications

7. **M. Märtens**, J. Meier, A. Hillebrand, P. Tewarie, P. Van Mieghem, *Brain network clustering with information flow motifs*, Apllied Network Science, 2:25, Springer Open, 2017.

6. F. Kuipers, **M. Märtens**, E. van der Hoeven, A. Iosup, *The power of social features in online gaming*, bookchapter to appear in Social Interaction in Virtual Worlds, edited by K. Lakkaraju, GṠukthankar, R. T. Wigang, Cambridge University Press, 2017.

5. **M. Märtens**, F. Kuipers, P. Van Mieghem, *Symbolic regression on network properties*, Lecture Notes in Computer Science, 131–146, Springer International Publishing, 2017. *

4. **M. Märtens**, H. Asghari, M. van Eeten, P. Van Mieghem, *A time-dependent SIS-model for long-term computer worm evolution*, Conference on Communications and Network Security (CNS), IEEE, 2016.

3. J. Meier, **M. Märtens**, A. Hillebrand, P. Tewarie, P. Van Mieghem, *Motif-based analysis of effective connectivity in brain networks*, Studies in Computational Intelligence, 685–696, Springer International Publishing, 2016.

2. **M. Märtens**, S. Shen, A. Iosup, F. A. Kuipers, *Toxicity detection in multiplayer online games*, International Workshop on Network and Systems Support for Games (NetGames), 1–6, IEEE, 2015. †

1. **M. Märtens**, R. van de Bovenkamp, P. Van Mieghem, *Superinfection in networks*, Proceedings of the 11th International Conference on Signal-Image Technology & Internet-Based Systems (SITIS), 413–420, IEEE, 2015.

† best paper award
∗ best paper nomination