Dynamic social graphs

Dynamic Social Graphs:

Mining and Modeling

DISSERTATION

to obtain the degree of Doctor at
Maastricht University,
on the authority of the Rector Magnificus,
Prof. dr. Rianne M. Letschert,
in accordance with the decision
of the Board of Deans,
to be defended in public
on Friday October 28, 2016 at 14:00 hours

by

Bijan Ranjbar-Sahraei

Supervisors:

Prof. dr. G. Weiss

Prof. dr. K. P. Tuyls (University of Liverpool)

Assessment Committee:

Prof. dr. ir. R. L. M. Peeters (*chairman*)

Prof. F. Coenen (University of Liverpool)

Dr. ir. K. Driessens

Prof. dr. ir. J. C. Scholtes

Prof. dr. ing. H. Voos (University of Luxembourg)

The research reported in this thesis has been carried out under the support of NWO, the Netherlands Organisation for Scientific Research (project no. 640.005.003).

ISBN 978-94-6233-402-1

Contents

1

Introduction

We are surrounded by vast amounts of information. The size and variety of data corpora all around the world are growing continuously. This growth has gained a large momentum due to recent inventions of technologies for ubiquitous sensing, big data mining, and complex network analysis.

The abundance of data has changed the way we conduct our personal and social life. In fact, in recent decades, we have faced an explosion of *social data*, in which unprecedented variety of personal information has become accessible to the public. Social data consists of two different data categories: First is the social data on individuals. This category captures the history of their actions, their habits, and their future plans. The second category is the social data on interactions among individuals, such as friendship interactions, professional connections and co-location observations. Once we integrate these two categories of social data, *social graphs* emerge. The importance of taking control over integration, modeling and manipulation of the social graphs has attracted interest from various research communities. As a result, much progress has been made in the analysis of social graphs. However, a faster progress has been made in data collection technologies resulting in a higher volume of data, more variety in forms of data and less veracity in the available sources.

A fundamental step to overcome the lack of veracity in data in the process of social graph generation is answering the question of *Who is Who?* This is an important requirement, as individuals referred to in different sources have different profiles in different contexts. In addition, different individuals can have very similar and in some cases, identical profile information. The task of *Identity Resolution* is undergoing intense study in the information retrieval community.

Having the Identity Resolution phase conducted successfully, the analysis of resulting social graph becomes a big challenge. A social graph represents a social network consisting of social agents and the ties among them. Depending on the richness of input social data, the social graph might include various profile attributes for each social agent and the characteristics of ties between agents. In fact, a social graph

turns out to be a very complex entity. Therefore, the traditional methods that were capable of analyzing static graphs, the time series, or relational databases cannot digest such complex evolving social graphs.

This thesis contributes to the analysis of dynamic social graphs, focusing in particular on Identity Resolution techniques required to generate such graphs from multiple sources of real-world data and analytical models capable of understanding the underlying dynamics of evolving social graphs.

1.1 Datafication

In our modern world, new technologies are turning different aspects of our life into digital data. This digitization trend is called *Datafication*. For instance, LinkedIn datafies our professional connections, Twitter datafies our thoughts, Facebook datafies our Likes, ResearchGate datafies our academic activity, Trivago datafies our traveling preferences, Uber datafies our transportations, Pinterest datafies our interests, Youtube datafies our media preferences, and the list goes on with Google Search, Instagram, Publons, Viber, Flickr, Goodreads, Tripadvisor, Expedia, Netflix, Reddit, etc.

While various companies are datafying our daily activities, many efforts focus on the digitization of historical archives from previous centuries. A good example is the CATCH program (for Continuous Access to Cultural Heritage), carried out in the Netherlands in the course of 12 years (2004-2016). In CATCH, IT researchers and heritage managers work together to make heritage available digitally. CATCH aims at making the collections of museums, archives, and historical associations digitally accessible. CATCH program includes, but is not limited to the following projects. The FACT project explores Dutch folktales based on available annotations; COGITCH, which focuses on multimedia archives and designs games to collect information from users about properties of available images and musics; the Webart project, which archives the web (.nl domains) to support the research of future heritage, and LINKS and MiSS, which explore different aspects of genealogical data and aim at reconstructing the population of previous centuries. This thesis is linked to the MiSS project.

Datafication has brought many new challenges to the scientific and engineering communities. Privacy of information is one of these challenges. Companies that own social data are heavily struggling to respect the privacy of their clients while allowing researchers to explore the data and develop data products. The huge amount of data, which has to be stored and processed, has pushed the research communities toward new methodologies such as *cloud storage* and *cloud computing*. Due to the abundance and variety of data, comparing and linking information across different sources of data has become a big challenge, thus tremendous research is being carried out on *Identity Resolution*. Furthermore, digestible visualization of data and developing scalable

data-products are among the challenges.

As a consequence of immense, widespread and multi-aspect datafication that is an ongoing process, structure and dynamics of available data has become much more complex than what the existing mathematical models can represent. Therefore, traditional analytic methods are incapable of analyzing the available data and cannot extract the necessary insights. Many research communities, such as the *Statistics, Data Mining* and *Machine Learning* communities, persistently invent new techniques for a better exploitation of available data. Their parallel efforts to deal with complexity and abundance of data have given birth to a new research area called *Data Science*. Data Science aims at providing a systematic methodology to handle data, analyze it, and extraction of insights from data in various forms. Data Science is evolving so quickly such that defining its exact boundaries is an ongoing process. Schutt and O'Neil (2013) define Data Science as the intersection of a) Mathematics and Statistics, b) hacking skills, and c) substantive expertise.

1.2 Social Graphs

A *social graph* is a mathematical representation of a social network that uses vertices or nodes for social agents and links or ties for the interactions among them. Each social agent can represent an individual person or an organization; we also refer to social animals such as kangaroos, wolves or monkeys as social agents (this will be discussed in detail in Chapter 5). The interactions can be anything ranging from friendship or business collaborations to co-observation of social agents.

In this thesis, we assume that a social graph can consist of heterogeneous nodes, while the ties can represent heterogeneous interaction types. In addition, we assume that nodes and links of a social graph have various attributes that can change over time. Lastly, the topology of social graph itself can also change over time; nodes and links can emerge or disappear at a certain time.

As discussed in the previous section, to generate a social graph from raw data we face many challenges including matching the identities and overcoming uncertainties. However, to conduct research on analysis of social graphs, one can use available datasets provided by research communities including the Koblenz Network Collection[1] and the Stanford Large Network Dataset Collection[2], each being a usefull collection of social graphs. Actor collaborations, Amazon co-purchasing, Air traffic networks, Citation networks, and Dominance networks of animals are just a few examples of the graphs provided by these communities.

[1] http://konect.uni-koblenz.de
[2] https://snap.stanford.edu/data/

Despite the obvious differences between existing social graphs, often social graphs share important common properties. Small diameter and heavy-tailed distribution are among the most well-known common properties in social graphs. According to the former property, also known as small world effect, most of the nodes are not neighbors but can be reached from each other by a small number of intermediate hops. According to the latter property, also described under terms such as scale-free networks, power law, or Pareto distribution, the distribution of some quantities in social graphs have heavier tails than the exponential distribution. For instance, in a friendship network, we expect the number of persons having n friends to be proportional to $n^{-\gamma}$, for a constant γ larger than one. The existence of heavy-tailed distributions in social networks facilitates the flow of information, increases the robustness to random disturbances, and has other benefits that are elaborated by Newman (2003) and Albert and Barabási (2002).

Lastly, social graphs provide a platform to study the *dynamics* of networks. In other words, to study how the status of each social agent is influenced by the status of other social agents, we can use the structure of social graph. The analysis of dynamics over networks has attracted interest among many social scientists, economists, and engineers, and has led to the analysis of Consensus Dynamics studied by Estrada and Vargas-Estrada (2013); Ren et al. (2005b); Olfati-Saber and Murray (2004)), the analysis of evolution of cooperation over networks studied by Pinheiro et al. (2012); Nowak (2006), and the description of Opinions and Epidemics dynamics studied by Boccaletti et al. (2006); Amblard and Deffuant (2004); Hegselmann and Krause (2002).

1.3 Problem Statement and Research Questions

A direct consequence of the Datafication process is the abundance of heterogeneous social data. In contrast to uncomplicated traditional social data, which can be represented by a flat table or a simple graph, the social data that is being collected nowadays consists of information from various sources. The nodes of the corresponding social graphs are augmented by rich time-varying personal information, and the connections are augmented by strengths and linking type. Thus, there is a great need for identity resolution techniques capable of integrating data from multiple sources of heterogeneous data. In addition, in such sophisticated social graphs, instead of analyzing binary states (e.g., rich/poor or friend/unfriend), we need to analyze the *infinite states* (e.g., a full spectrum representing the richness or friendship strength of agents). Considering these requirements we can formulate the following problem statement.

Problem Statement:

With the rise of online social networks and widely accessible historical archives, the complexity and dynamics of social graphs have led to the following two basic research challenges. First, how to extract the structure and dynamics of social graphs form heterogeneous social data, that is, how to turn social data into social graphs. Second, how to model and analyze the properties of dynamic social graphs, that is, how to analyze evolving social graphs.

This thesis addresses these two key challenges and focuses on (i) the development of scalable data mining techniques for integration of heterogeneous sources of social data, and (ii) the development of analytical models to understand the dynamics of social graphs.

In the following, we first formulate two research questions that target the data-driven and practical aspects of social data integration. These two questions seek the requirements for a feasible and scalable knowledge discovery system for real-world settings. This system can turn the raw social data into a rich dynamic social graph. Next, we formulate three research questions with a focus on modeling the evolving social graphs. These questions are in quest of new analytical models that can capture the continuous nature of behaviors and interactions in dynamic social graphs.

Question 1: *To what extent cross-matching of existing sources of structured and unstructured data can eliminate the need for human inputs during an information extraction process?*

A large amount of information in social graphs is provided in form of unstructured data. Examples range from a notarial act of 16th century in Dutch that reports the act of transferring a property between two couples, to a Facebook post about a memorable trip of a group of friends. In general, the field of Information Extraction deals with the task of automatically extracting structured information from unstructured documents. However, the existing techniques are limited due to the need of labeled data from human experts, being specific to domains or languages, and limited in application when the data evolves in time (Ferrara et al., 2014). Existence of structured data that shares common entities and relationships with the unstructured source of data can act as a valuable source of distant supervision that eliminates the need of human input and allows for adaptation to evolving data. **Chapter 3** explores the use of an extra source of data in information extraction.

Question 2: *How to develop a simple, interpretable and yet scalable identity resolution tool for heterogeneous social data?*

Scalability is the major limitation of existing identity resolution approaches. Despite using various blocking and hashing techniques, often, the available approaches cannot scale up to large amounts of streaming data (Christen and Gayler, 2008). Furthermore, in real-world settings, the developed tool for an identity resolution task should be simple and interpretable such that it can be integrated in different domains and experts can understand its matching policy. This is very important, specifically in the tasks where the identity resolution tool is meant to assist a group of domain experts in their knowledge discovery task. In **Chapter 4**, the development of a practical tool is discussed.

Question 3: *What are the origins of the heavy-tailed distributions in social graphs?*

Heavy-tailed distributions are ubiquitous among social graphs. Despite the vast study of the characteristics of these distributions, understanding their origins has remained a challenge (Muchnik et al., 2013; Wilk and Włodarczyk, 2013). In recent years, there has been an immense improvement in data acquisition technologies and data integration techniques (e.g., the techniques that are explored in answering **Questions 1** and **2**). Thus, the available datasets on social graphs are turned into networks of social agents who are tied with various types of interactions. The strength of ties can be inferred from data (e.g., from the frequency of mutual interactions), and the behavior of individuals can be tracked (Barrat et al., 2004a). This recent transition in social data from binary states to infinite continuous states allows for a deeper analysis of heavy-tailed distributions. **Chapter 5** proposes an analytical model that takes into account the continuous tie strengths of social networks. Using this model we backtrack the origins of heavy-tailed distributions to the hierarchical structures in social graphs.

Question 4: *To what extent can the theory of dynamical systems contribute to modeling the feedback loop between behaviors and interactions in social graphs?*

Social networks around us are dynamic in many respects. The topology of these networks change in time and space; the behaviors of social agents also change and thus the strength of ties among them changes. Evolutionary Game Theory provides a framework to study the evolution of behaviors in a population. Such studies often assume binary state spaces - whereas in many real settings, behaviors can be better thought of as being a continuous trait (Killingback and Doebeli, 2002). Moreover,

this simplification may hide many interesting dynamics of the evolution process; this prevents a full analysis of such settings. A promising alternative for studying the behaviors in continuous state spaces is the theory of dynamical systems; this field studies the behaviors and properties of evolving dynamical systems. **Chapter 6** represents the feedback loop between behaviors and interactions in a continuous state space model and studies the dependence between network evolution and its initial states. Specifically, this chapter focuses on the dilemmatic problem of *Evolution of Cooperation* in social graphs.

Question 5: *What are the mutual effects of evolving behaviors and emerging topologies?*

Studying the evolution of behaviors and interactions — as an answer to the previous research question — provides insights into underlying mechanisms in social graphs. However, a more fundamental phenomenon (which happens in social graphs) is the simultaneous evolution of the behaviors and graph topology. The changes in behaviors of social agents can be both cause and consequence of changes in the network topology. Thus, ignoring the effect of one on another leads to unrealistic analysis of social networks. **Chapter 7** studies the co-evolution of behaviors and topology, and provides new findings on the mutual effects of behaviors and topology. In particular, this chapter studies the influence of evolving behaviors on diversity of emerging graph structures. Moreover, it studies the role of initial founders of a social network in imposing bias upon the evolution of behaviors.

1.4 How to Read this Thesis

The analysis of dynamic social graphs is a common theme in this thesis. **Chapter 2** provides the background knowledge on information retrieval, social graphs, and the theory of dynamical systems that is required for a better understanding of this thesis. It also introduces the datasets that are used throughout the thesis for experimental analysis and numerical verifications. The rest of this thesis can be divided into two parts with respect to the proposed methods.

In the first part of this thesis, which focuses on practical data mining tools, **Chapters 3** and **4** propose some tools for matching identities across various sources of social data. In the second part, which focuses more on proof-oriented methods, **Chapters 5** to **7** propose analytical models for understanding and influencing the dynamics of social graphs.

Chapter 3 introduces a technique called REDS (for Relation Extraction based on Distant Supervision), which extracts information from unstructured data. With

REDS, we take the first step toward the integration of data from multiple sources of heterogeneous data. REDS has three main contributions. First, it introduces the concept of relationship fingerprints and uses it to build an inverted index for the available structured data, namely the knowledge repository. Second, REDS uses this knowledge repository for extracting relations from unrestricted text documents. Third, REDS turns the relations extracted in the previous step into a training set that is used for training an interpretable decision tree. The evaluation results of testing REDS with different real-world datasets with the assistance of genealogical experts are reported at the end of this chapter.

Chapter 4 introduces an end-to-end identity resolution tool called HiDER (for Historical Data Entity Resolution) with specific application to heterogeneous genealogical data. The backend of this tool uses an inverted index of the so-called fingerprints introduced in Chapter 3, which allows for the use of the context information in a very scalable way. This tool is simple and interpretable, demonstrating reliable results for genealogists. The frontend of this tool provides various web-based interfaces for visualizing the entity network and navigating through it. This chapter concludes the first part of this thesis by demonstrating a successful transformation of raw data into a social graph.

Chapter 5 introduces an analytical model for evolution of tie strengths in a social graph. The proposed model assumes simple hierarchical order between social agents, and it derives, in closed form, the heavy-tailed distribution of individual strengths in the network. The proposed models, namely PBEM (for Prestige-Based Evolution Model) and DBEM (for Dominance-Based Evolution Model) help in better understanding of the evolutionary foundations of hierarchies in societies, and provide new insights into the origins of heavy-tailed distributions in social networks. In this chapter, real-world networks are used to validate the proposed evolution models.

While the previous chapter focuses on explaining a macroscopic characteristic of social graphs, **Chapter 6** focuses on the evolution of microscopic behaviors of social agents. This chapter proposes a mathematical model based on the theory of dynamical systems as an alternative to the framework of evolutionary game theory. The proposed model is called CAIPD (for Continuous Action Iterated Prisoners' Dilemma) and is designed to study the Evolution of Cooperation in social graphs, by using the theory of dynamical systems. We discuss the stability of CAIPD and possible indirect influence on the behavior of social agents.

Chapter 7 builds on top of CAIPD and describes a new model called SEE (for Simultaneous Evolution and Emergence) that analyzes the co-evolution of behaviors and topology in a social graph. We consider an emerging network in which simultaneously the behavior of recently added agents evolves based on the network topology, and the topology of the network evolves based on the behaviors. This co-evolutionary

process shows that this process results in a variety of new topologies in the network, and it reveals the importance of initiators of a social graph in steady behaviors of the overall network.

We conclude in **Chapter 8** by summarizing the contributions of this thesis and answering the research questions of this thesis. We discuss open problems related to each of the topics covered and provide directions for future research.

2

Preliminaries

This chapter provides the main background information which forms the foundation of this thesis. The Preliminaries section is divided into four sections. The first section introduces the process of Data Mining. Moreover, it describes different data manipulation and data analysis techniques that will be used throughout this thesis. The Identity Resolution techniques and Relationship Extraction algorithms are among these methods which are the requirements to understand **Chapters 3-4**. In the second section, we describe various modeling techniques used in analysis of social networks. These models are related to the network topology and evolution of behaviors. The third section of this chapter introduces the theory of dynamical systems in which differential equations are used to study the behavior of systems. The second and third sections are the main requirements to understand **Chapters 5-7**. Finally, in the forth section we introduce different datasets that will be used for validation purposes throughout this thesis.

2.1 Data Mining

Data Mining is a systematic process to collect and process large amount of data in quest of consistent patterns and new insights. Eventually, data mining techniques transform the data into an interpretable structure (e.g., grammar, timelines, visualization). Next we briefly introduce some components of a data mining process that will be used later in this thesis.

Data Collection - The process of gathering information on a target variable is called data collection. While *Questionnaires* and *Personal Interviews* have been among the popular traditional tools to gather information, the *web-based technologies* have become the popular modern ways to collect information. To illustrate, consider that at every minute, over 4,000,000 search queries are received by Google, 277,000 text messages are shared on Twitter and 72 hours of new video are uploaded to Youtube (DOMO, 2014). Such data serves as input to various data analysis projects.

Database Management System (DBMS) - In order to store the collected data, and query it efficiently a database management system is required. The most well-known classical DBMS is MySQL which is a relational database. In MySQL the database schema is pre-defined based on the requirements of the project and data is stored in the form of rows (tuples) which are stored in tables (relations). In recent decade, a new class of document-oriented DBMS has emerged which is known as NoSQL database. HBase, MangoDB, Neo4j are among the famous NoSQL DBMSs. What is common in such databases is the fast query access and dynamic schemas with which there is no need to predefine any structure. Also, in many cases, denormalization (e.g., existence of redundant data) is common in such DBMSs.

Data Cleaning - The process of improving the quality of data by eliminating the errors, inconsistencies and redundancies is known as *data cleaning*, *data cleansing* or *data scrubbing*. Data cleaning starts by processing the available records in search of anomalies. As explained by Müller and Freytag (2005), anomalies can be found by looking at minimum and maximum lengths, density of values, number of Null values and specific patterns. Once the anomalies are found, the *cleaning workflow* is specified and is executed which takes the dirty data as input and applies the refinement in order to eliminate the anomalies. Very often, the results of cleaning workflow should be carefully checked and the possible flaws should be iteratively resolved.

Information Extraction - Information extraction refers to transforming the unstructured data (e.g., free text of an email or a wikipedia page) to a structured document. While field of Natural Language Processing focuses on extracting information out of human language text, other fields focus on extracting the content out of audio, video and images. Recognition of known entity names (i.e., Named Entity Extraction), extraction of relation between entities and co-reference resolution (i.e., detection of different expressions which refer to the same real entity) are among the main tasks of Information Extraction. **Chapter 3** specifically focuses on the relation extraction from text documents. Therefore, later in this chapter we discuss, in more detail, the Relation Extraction process and the existing work in this field.

Identity Resolution - Identity Resolution also known as Co-reference Resolution, Entity Resolution or Record Linkage plays an important role in extracting new insights out of multiple sources of structured data, where different records refer to the same entity across various sources. The Identity Resolution techniques bring different pieces of information together and extract new insights out of the existing data sources.

Relation Extraction

A vast amount of information available to us comes in form of the unstructured data, which can not be efficiently linked to the other sources of information in its original form (Winkler, 2006; McCallum, 2005). Therefore, most of the identity resolution approaches are not applicable in multi source applications which have both structured and unstructured data coexisting. It is important to develop techniques to convert the unstructured data to structured form by extracting the named entities and existing relations. For this, the unstructured data, which is in form of text, needs to be segmented into meaningful chunks of strings (Choi, 2000), and in turn the chunks which correspond to named entities should be identified (i.e., Named Entity Recognition (NER) studied by Nadeau and Sekine (2007); Tjong Kim Sang and De Meulder (2003); Ratinov and Roth (2009)). In the second step, relations between these entities should be extracted (i.e., the topic of relation extraction studied by Zelenko et al. (2003); GuoDong et al. (2005)).

Using supervised learning for extracting information from a text imposes many restrictions such as the human effort needed to prepare the training set and being domain specific. In order to solve these restrictions for text segmentation, Agichtein and Ganti (2004) proposed the use of existing structured data in the data warehouse to learn an automatic segmentation system called CRAM. A very similar approach was used by Borkar et al. (2001) in developing a tool called DATAMOLD which was able to automatically segment text. Unsupervised NER has also been well researched as by Etzioni et al. (2005); Nadeau et al. (2006); Cucchiarelli and Velardi (2001), where important information of the text can be extracted in a recursive manner.

However, solving the relation extraction in the unstructured data, which is the main focus of this thesis, is more challenging than the text segmentation and NER. We divide the existing work on relation extraction, which is proposed to overcome the restrictions of supervised relation extraction, into three categories: **C1**) iterative relation extraction based on frequent named entities; **C2**) relation extraction based on frequent patterns, and **C3**) relation extraction based on distant supervision (the proposed approach in this thesis is in this category).

In **C1**, starting from a small seed of entity pairs, with known relations, all occurrences of the pairs in data are found, which allows for extracting the patterns for those relations. These patterns help in finding more entity pairs and in turn the entity pairs increase the number of retrieved patterns. Iteratively, a large set of entity pairs and patterns to detect them can be obtained. The work of Brin (1999) and the Snowball system proposed byAgichtein and Gravano (2000) are examples of this approach.

In **C2**, the frequent patterns seen in the database is used to generate a relational dataset. This dataset is then used by clustering algorithms to extract the informative

patterns. For instance, Shinyama and Sekine (2006) proposed a *Preemptive* information extraction approach, in which first, the basic patterns which frequently appear in a text were extracted. Then, a clustering technique was used to cluster the sets of similar patterns. In another attempt, Banko et al. (2007) and Yates et al. (2007) introduced the OpenIE (Open Information Extraction) approach and the TextRunner as a scalable implementation of OpenIE, in which first a learner automatically labels a sample corpus and then the labeled data is used to train a Naive Bayes classifier. By applying the classifier on large amounts of data many relational tuples can be extracted.

While both **C1** and **C2** have very useful real-world applications, they rely on abundance of data and frequent occurrences of entities and relation patterns, respectively. Therefore, such approaches have limitations in certain databases where the entities are not reported frequently and the relation patterns have high variety. In order to resolve this limitation, **C3** suggests to use a second source of information for extracting the relations. This second source which contains a large collection of entity tuples with known relations helps in finding large sets of patterns in the unstructured data corresponding to known relations. Work of Mintz et al. (2009), Nguyen and Moschitti (2011) and Riedel et al. (2010) are in **C3**.

The work proposed in **Chapter 3** contributes to **C3** by introducing a relation extraction method based on distant supervision called REDS; it uses a knowledge repository of structured form for predicting the relations in the original unstructured data. In order to increase the precision and recall of the relation extraction technique it uses the concept of identity resolution to disambiguate the entities at the time of linking. Compared to the existing work in **C3**, REDS takes into account the possible spelling errors and name alternatives. The use of fingerprints for building an inverted index for the knowledge repository makes the searching process very fast and scalable. REDS also measures the confidence level of each predicted relation based on data statistics, and provides a training set for supervised learning of a classifier. This classifier can extract relations from the unstructured data for which no knowledge repository exists. It should be mentioned that REDS parses the unstructured data and queries the knowledge repository for getting the distant supervision, this is in contrast with the typical approach in relation extraction, e.g., work of Mintz et al. (2009), which runs queries on the index of unstructured data. As the unstructured data usually contains more uncertainty than the structured knowledge repositories, REDS can apply preprocessing on the text documents, thus gaining a high accuracy.

Identity Resolution

Dealing with the *identity resolution* problem is an important step in mining and analysis of various social networks. Due to appearance of multiple Online Social Networks (OSNs) in recent two decades, integrating these networks has become an interesting but challenging research topic. Applications of the identity resolution in OSNs include targeted advertisements discussed by Boal (2013); Stibel and Stibel (2014), building expertise networks discussed by Purohit et al. (2012, 2013) and refining the individuals' contact lists discussed by Bartunov et al. (2012). In addition to the OSNs which provide networked data, the genealogical datasets also provide information about a large group of social agents and their interactions in the course of the centuries (Schraagen and Hoogeboom, 2011). Prior to analysis of historical archives, the implementation of identity resolution is a mandatory task, as social agents are mentioned in different documents in absence of any unique identification number. The name variations, errors, and missing data are among the challenges in dealing with such datasets.

The first intuitive approach in identifying a group of so-called *references* which refer to the same entity is to compare the reference attributes. In a social network setting, such attributes might include the *given name* and *family name*, *gender*, *dates* and *locations*. Vosecky et al. (2009), and Motoyama and Varghese (2009) studied such biographical attributes for searching and matching individuals across multiple OSNs. Köpcke et al. (2010) gave an extensive comparison of some of the non-learning and learning based approaches which mainly use the biographical attributes. Efremova et al. (2014b) studied a baseline approach based on the biographical attributes to show that in presence of name variations, missing data and errors, the precision of matchings is very low. In their work, they witnessed different entities having very similar and in many cases identical bibliographic attributes.

To improve the accuracy of identity resolution, researchers have exploited the information on social relationships in the network. Considering the neighbor nodes in a social network can significantly improve the precision of the matchings. For instance, Bartunov et al. (2012) combined the usage of profile attributes and social relationships to identify all references to an entity. Buccafurri et al. (2015) also used the common neighbors of two references to predict location of the *me* links that connect two references to the same entity. Bhattacharya and Getoor (2007) proposed a relational clustering algorithm to use both the reference attributes and the connections among them.

Using the network relationships increases precision of the identity resolution algorithms; however can cause computational issues. For instance, Bartunov et al. (2012) explained that they can apply their identity resolution algorithm in small subgraphs

but face a major challenge to scale it up to large social graphs. This type of limitation is studied in detail by Christen and Gayler (2008). They propose the use of inverted indexing techniques, as a common technique in information retrieval, for real-time, fast and scalable identification of entities. They show that inverted index approaches are up to one hundred times faster than some of the traditional entity matching techniques.

In **Chapter 4**, similar to the work by Christen and Gayler (2008) we address the identity resolution problem in the information retrieval framework. However, instead of matching the references we aim at matching the relationships between them. By choosing the relationships as the targets of the identity matching problem, we achieve a low computational complexity and still preserve a high accuracy.

2.2 Social Network Analysis

Networks describe collections of social agents (nodes) and the relation between them (edges). Formally, a network can be represented by a graph $\mathbb{G} = (\mathcal{V}, \mathcal{W})$ consisting of a non-empty set of nodes (or vertices) $\mathcal{V} = \{v_1, \ldots, v_N\}$ and an $N \times N$ adjacency matrix $\mathcal{W} = [w_{ij}]$ where non-zero entries w_{ij} indicate the (possibly weighted) connection from v_i to v_j. If $\mathcal{W}$ is symmetrical, such that $w_{ij} = w_{ji}$ for all i, j, the graph is said to be undirected, meaning that the connection from node v_i to v_j is equal to the connection from node v_j to v_i. In social networks, for example, one might argue that friendship is usually mutual and hence undirected. This is the approach followed in this work. In general however this need not be the case, in which case the graph is said to be directed, and $\mathcal{W}$ asymmetrical. The neighborhood, $\mathbb{N}$, of a node v_i is defined as the set of nodes it is directly connected to, i.e. $\mathbb{N}(v_i) = \cup_j v_j : w_{ij} > 0$. The node's degree $\deg[v_i]$ is given by the cardinality of its neighborhood.

Structure Modeling

Several types of network models have been proposed that capture the structural properties found in large social, technological or biological networks, two well-known examples being the *Watts-Strogatz* and *Barabàsi-Albert* models. The Watts-Strogatz model exhibits short average path lengths between nodes and high clustering, two features often found in real-world networks, also known as small-world phenomenon (Watts and Strogatz, 1998). Alternatively, the Barabàsi-Albert model is used to generate random scale-free networks, characterized by their heavy-tailed degree distribution following a power law (Barabási and Albert, 1999). In scale-free networks the majority of nodes have a small degree while simultaneously there are some nodes with very large degree, the latter being the hubs or connectors of the network. Due

to importance of the scale-free networks in **Chapters 5**, **6** and **7** of this thesis, next we elaborate on the Barabàsi-Albert model.

The Barabàsi-Albert model is based on the assumption that, in many social settings, the chance of making new connections grows proportionally with the number of connections that an agent already has. This is known as the *preferential attachment*, *the rich get richer* or *Yule process*. The Barabási-Albert model simulates this process by growing the network over time, adding one new node at a time, and linking it to a fixed number of existing nodes, these being chosen proportionally to their current degree. Specifically, starting from an initial network of m_0 nodes, at every time step one new node is added to the network. The new node forms $m < |m_0|$ connections to existing nodes, where the probability p_i that the new node connects to existing node v_i is proportional to its degree:

$$p_i = \frac{\deg[v_i]}{\sum_j \deg[v_j]} \tag{2.1}$$

Preferential attachment generates a heavy-tailed degree distribution following a *power-law*:

$$P(k) \sim k^{-\alpha}$$

with k denoting the degree of the nodes. The power-law exponent for the Barabási-Albert model is $\alpha = 3$; in comparison, many real-world complex networks have been shown to lie in the range $2 \leq \alpha \leq 4$ (Barabási and Albert, 1999; Newman, 2005).

For a detailed description of various social network models and their properties, the interested reader is referred to (Jackson, 2008).

Behavior Modeling

In the previous subsection, we introduced two mathematical/algorithmic models that can be used for generating graphs with a structure similar to real-world social networks. Additionally, models exist for describing the behavior of social agents that are interacting within the social network structure. Assuming that social agents act rationally and influence each other based on their social interactions, *Game Theory* is one of the most important frameworks for modeling social behaviors. We will use Game Theoretical models in **Chapters 6** and **7** for this purpose.

Game Theory models strategic interactions in the form of games (Gibbons, 1992), where each player has a set of actions, and a preference over the joint action space that is captured by the received payoffs. The goal for each player is to come up with a strategy (a probability distribution over its actions) that maximizes his expected payoff in the game. A strategy that maximizes the payoff given fixed strategies for all

opponents is called a best response to those strategies. The players are thought of as individually rational, in the sense that each player purely tries to maximize his own payoff, and assumes the others are doing likewise. However, this reasoning might not always lead to a beneficial outcome for everyone, and might even be detrimental to all players in the game. Often, there is tension between individual rationality on the one hand, and social welfare on the other.

This archetypal dilemma is aptly captured by the Prisoner's Dilemma (Axelrod and Hamilton, 1981). In this one-shot interaction, players simultaneously choose between either cooperation or defection, after which payoffs are distributed based on their joint action. Cooperation is costly, however cooperators distribute benefits among the other players. Defectors do not pay a cost, but do receive benefits from cooperators as well. In this game, defection (free-riding) is a best response against any opponent strategy, and therefore individually rational players can be expected to defect. However, if all players would cooperate their distributed benefits would outweigh the cost of cooperation, and hence all players would be strictly better off. Herein lies the dilemma.

Modeling the evolution of cooperation in social networks has recently attracted much attention, aiming to understand how individuals work together and influence each other, and how society as a whole evolves over time (Nowak and May, 1992; Santos and Pacheco, 2005; Ohtsuki et al., 2006; Lazer et al., 2009; Hofmann et al., 2011). Progress made towards understanding how this evolution comes about has been mostly empirical in nature. Though compelling, deeper insights are better gained from an analytical analysis of the problem.

2.3 Theory of Dynamical Systems

Generally speaking, the evolution of any set of measurable quantities over time can be represented in form of a *dynamical system*. Once we find a fixed rule to describe how these quantities evolve in response to their own values in previous time, we can use the *theory of dynamical systems* to mathematically describe the evolution of values, study characteristics of this evolution and make predictions. At any given time such a dynamical system has a *state* given by a set of variables (i.e., a vector), and a set of differential equations (difference equations) explain how the future states at any real time (integer time) follow from the current state. Next, we first explain how a dynamical system can be modeled using the so-called *state space* representation which will be used in **Chapters 5-7**. Then we briefly introduce the field of Control Theory which will be used later in **Chapter 6**.

System Modeling

A model can be regarded as an accurate mathematical representation of the (nonlinear) dynamics of a system. Essentially, the goal is the discovery of (nonlinear) differential equations describing the transient behavior of some state variables in a system. Typically, state representations are collected in a state vector $\mathbf{x} = [x_1, x_2, \ldots, x_N]^{\mathrm{T}}$ and control variables (i.e., actions applied to affect the state vector) are collected in a vector $\mathbf{u} = [u_1, u_2, \ldots, u_q]^{\mathrm{T}}$ where x_i and u_i denote the i^{th} state and input respectively. A linear and time invariant system (LTI) can thus be represented by

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u}$$

where $\mathbf{A}$ and $\mathbf{B}$ correspond to the dynamic and control matrices, respectively.

When the system dynamics are nonlinear and/or time varying, as is the case in this thesis, the state space model has to be extended to a more general form

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \vdots \\ \dot{x}_N \end{bmatrix} = \begin{bmatrix} f_1(t; x_1, \ldots, x_N, \mathbf{u}) \\ f_2(t; x_1, \ldots, x_N, \mathbf{u}) \\ \vdots \\ f_N(t; x_1, \ldots, x_N, \mathbf{u}) \end{bmatrix}$$

where the change in the state variables is a nonlinear mapping of the state variables and the control action. Moreover, each state variable is governed by its own dynamics. Compactly this can be written in matrix form as $\dot{\mathbf{x}} = \mathbf{f}(t; \mathbf{x}, \mathbf{u})$.

Control Theory

One of the main goals in control theory is the manipulation of the system's inputs to follow a reference over time. In other words, this manipulation feeds back the difference between the state variable $\mathbf{x}$ and the reference point $\mathbf{x}_{\mathrm{ref}}$ at any instance in time. Such a rule, where $\mathbf{u} = l(\mathbf{x}, \mathbf{x}_{\mathrm{ref}})$, is called a feedback controller. Controller design is a wide-spread field and its discussion is beyond the scope of this chapter. Interested readers are referred to (Levine, 1996).

Here, the main interest is in stability and convergence analysis of dynamical systems. The type of stability we are referring to in this thesis is the Lyapunov stability that is introduced below.

The Lyapunov stability is studied in the vicinity of equilibrium points (i.e., points where $\dot{\mathbf{x}} = 0$). In the following, $\mathcal{B}(\bar{\mathbf{x}}, \epsilon)$ denotes an open ball centered at $\bar{\mathbf{x}}$ with a radius $\epsilon > 0$, that is the set $\{\mathbf{x} \in \mathbb{R}^d : ||\mathbf{x} - \bar{\mathbf{x}}|| < \epsilon\}$, where $||\cdot||$ represents the L_2-norm. The following definition can be stated:

Definition 1 (Lyapunov Stability)**.**

Let $\boldsymbol{\psi}(t; 0, \bar{\boldsymbol{x}})$ denote the solution $\boldsymbol{x}(t)$ to $\dot{\boldsymbol{x}} = \boldsymbol{f}(t; \boldsymbol{x})$ that corresponds to $\boldsymbol{x}(0) = \bar{\boldsymbol{x}}$. Then, an equilibrium point $\boldsymbol{x}_e$ of this nonlinear system is said to be stable, if for all $\epsilon > 0$ there exists a $\delta > 0$ such that:

$$\bar{\boldsymbol{x}} \in \mathcal{B}(\boldsymbol{x}_e, \delta) \implies \boldsymbol{\psi}(t; 0, \bar{\boldsymbol{x}}) \in \mathcal{B}(\boldsymbol{x}_e, \epsilon) \quad \textit{for all } t \geq 0.$$

To analyze the convergence of a system we first define the notion of Invariant sets and then introduce the Global Invariant Set Theorem (Slotine et al., 1991).

Definition 2 (Invariant Sets)**.**

Let $\dot{\boldsymbol{x}} = \boldsymbol{f}(t; \boldsymbol{x})$. A set of points in $\mathcal{M}$ in state space is invariant if

$$\boldsymbol{x}(t_0) \in \mathcal{M} \implies \boldsymbol{x}(t) \in \mathcal{M} \textit{ for all } t > t_0.$$

Definition 3 (Global Invariant Set Theorem)**.**

If there exists a continuously differentiable function V such that

$$\begin{aligned} &V \textit{ is positive definite} \\ &\dot{V}(\boldsymbol{x}) \leq 0 \\ &V(\boldsymbol{x}) \to \infty \textit{ as } ||\boldsymbol{x}|| \to 0 \end{aligned}$$

then

(i) $\dot{V}(\boldsymbol{x}) \to 0$ at $t \to \infty$

(ii) $\boldsymbol{x}(t) \to \mathcal{M}$ = the largest invariant set contained in $\mathcal{R}$ where

$$\mathcal{R} = \{\boldsymbol{x} : \dot{V}(\boldsymbol{x}) = 0\}$$

In the above definition, if $\dot{V}$ is negative definite then $\mathcal{M} = 0$. This case is known as Lyapunov's direct method.

Some other theorems required to prove the convergence of dynamical systems will be introduced later in this thesis, depending on the specific system properties in each chapter.

2.4 Data Setup

In the rest of this thesis, we use some real-world datasets to assess the proposed methods/models. The work in **Chapters 3 and 4** is evaluated using a heterogeneous genealogical dataset containing Civil Registers and Notarial Acts from 18^{th} and 19^{th} centuries and before that, which has been provided by Brabant Historical Information

Centre (BHIC)[1]. In **Chapter 5** we use four real-world datasets received from the Koblenz Network Collection (KONECT)[2] to validate the proposed model. Next, we introduce each of these datasets, and specifically we provide details on the first dataset.

MiSS Dataset

The genealogical dataset used in this thesis consists of two different corpora: 1) the corpus of civil registers in form of structured data, and 2) the corpus of notarial acts in form of unstructured data.

Corpus of Civil Registers - this corpus contains three main types of certificates, namely "Birth", "Death" and "Marriage" certificates. Table 2.1 lists the content features for each certificate type. As shown in Table 2.1, Birth certificates include three individual references (i.e., child, father and mother). The Death certificates include four individual references (i.e., deceased, father, mother and relative of deceased). Finally, the Marriage certificates include six references (i.e., groom, bride and parents of each).

Table 2.1: Features of each certificate type in corpus of Civil registers.

Birth Certificate	FirstName, LastName, Gender, BirthDate, BirthPlace, FatherFirstName, FatherLastName, MotherFirstName, MotherLastName
Death Certificate	FirstName, LastName, Gender, BirthDate, BirthPlace, DeathDate, DeathPlace, FatherFirstName, FatherLastName, MotherFirstName, MotherLastName
Marriage Certificate	GroomFirstName, GroomLastName, GroomAge, BrideFirstName, BrideLastName, BrideAge, GroomFatherFirstName, GroomFatherLastName, GroomMotherFirstName, GroomMotherLastName, BrideFatherFirstName, BrideFatherLastName, BrideMotherFirstName, BrideMotherLastName

This corpus has been very dynamic in the sense that BHIC is continuously digitizing the handwritten archives by using the help of many volunteers. For instance the number of digitized documents has increased from 1,600,000 documents in 2012 to more than 2,000,000 documents in 2015. This amount is increasing everyday.

Figure 2.1 shows the number of digitized certificates per year for the most recent version of data in 2015. As can be seen in this figure, the amount of birth certificates has been less than number of marriage and death certificates. One of the reasons is the policies of BHIC and the fact that the regions for which marriage and death certificates

[1]http://www.bhic.nl
[2]http://konect.uni-koblenz.de

are turned into structured data spans a larger area than the ones corresponding to birth certificates.

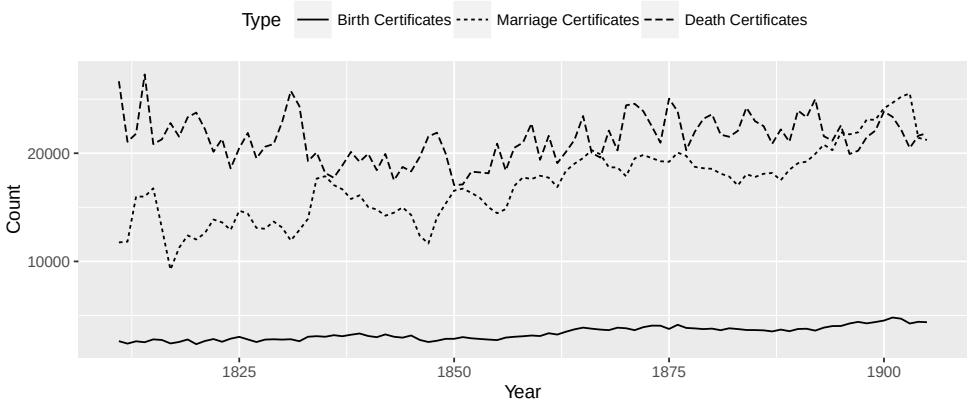


Figure 2.1: Number of civil registers per year of issue.

We use a Relational database model (Codd, 1990) to integrate and persist the three discussed certificate types considering Entity, Referential and Domain integrity constraints (Elmasri and Navathe, 1999). We choose the Relational database model since this model is widely used, easy to apply and is highly maintainable. Elmasri and Navathe (1999) discuss in details the Relational database model and its advantages over other databases models.

Corpus of Notarial Acts - This corpus contains free text documents from various categories such as property transfers, inheritance reports, and loan declarations.

This dataset contains 226,751 records related the the period 1458 to 1900[3]. Each record contains written narration of facts drawn up by civil-law notaries, notary public or an alderman (Schepen in Dutch) authenticated by signature and official seal. Each record contains the main text of narration, place and date of issue and some other details.

Figure 2.2 shows the number of notarial acts per year. 86,000 notarial acts (i.e., 37% of all notarial acts) are issued after 1800 and overlap with the year of issue in civil registers.

Howler Monkey Groups (Linton C. Freeman, 2015a)

This dataset represents the social network among mantled howler monkeys, Alouatta palliata, which is collected by Froehlich and Thorington (1981) and Sailer and Gaulin

[3]Since analyzing this dataset on March of 2014, its size has increased significantly.

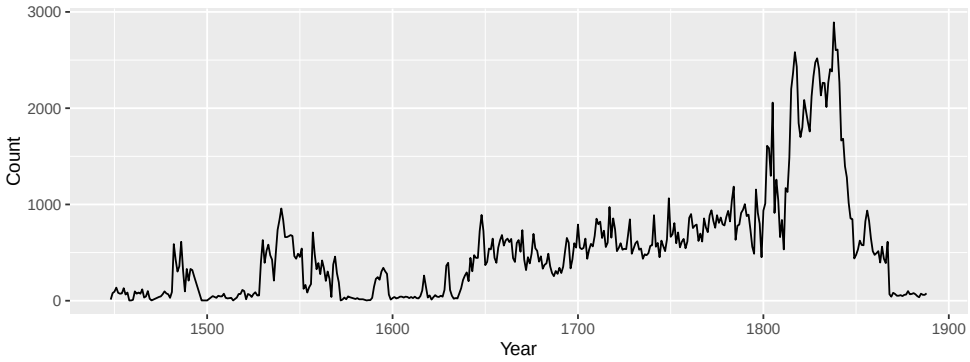


Figure 2.2: Number of notarial acts per year of issue.

(1981). The dataset represents the co-observations in a group of 17 monkeys, where the co-observations of every two monkeys is reported in form of a weighted adjacency matrix.

Kangaroos (KONECT, 2015a; Grant, 1973)

This dataset represents the social network among free-ranging grey kangaroos. A weighted adjacency matrix shows the number of observed physical proximities among a group of 17 kangaroos. Observations were made in the Nadgee Nature Reserve in New South Wales.

Wolf Dominance (van Hooff and Wensing, 1987; Linton C. Freeman, 2015b)

This dataset represents the social network among a captive family of wolves in Arnheim, Germany. A weighted adjacency matrix shows the number of occasions on which the row wolf was seen to exhibit a "low posture" display directed toward the column wolf. This behavior is a sign of fear and being subordinate.

US Airports (KONECT, 2015b; Opsahl, 2011)

This dataset presents the flights between 1574 US airports in 2010. The elements of the weighted adjacency matrix show the number of flights from the row airport to the column airport in 2010. In this thesis, we consider the first 200 airports with highest overall number of flights. Besides, we set the number of flights between two airports equal to the average of each flight from one to the other one. This way, the

adjacency matrix becomes symmetric, thus compatible with the experimental method introduced in **Chapter 5**.

3

Distant Supervision of Information Extraction

This chapter is based on:
B. Ranjbar-Sahraei, H. Rahmani, G. Weiss, K. Tuyls, "Distant Supervision of Relation Extraction in Sparse Data", Submitted to Knowledge and Information Systems.

An important step in analysis of social data is to extract information from available sources of unstructured data. Named Entity Recognition (NER) and relation extraction are among the main sub-problems of information extraction (Bunescu and Mooney, 2005). NER locates a word or a phrase that refers to a particular named entity within a text (Nadeau and Sekine, 2007), and relation extraction focuses on recognizing relations among the named entities in unstructured text (Bach and Badaskar, 2007). Both sub-problems have been studied extensively in literature (Nadeau and Sekine, 2007; Bach and Badaskar, 2007; Sarawagi, 2008) where supervised and unsupervised techniques are used to tackle these problems. Among these techniques, supervised learning is capable of extracting both the named entities and relations from unstructured data (Kambhatla, 2004; Zhao and Grishman, 2005; Surdeanu and Ciaramita, 2007). However the need for manually labeled data renders its application inefficient in real-world settings. An alternative approach is to benefit from the abundance of data on the web. Data redundancy in the web allows for accurate statistical estimates (Agichtein and Gravano, 2000; Ravichandran and Hovy, 2002); for instance, the state-of-the-art open information extraction tools such as KnowItAll (Etzioni et al., 2004) and Open IE (Banko et al., 2007; Etzioni et al., 2008) are designed to collect huge amounts of data from the web. They use bootstrapping techniques and/or data statistics to precisely extract information.

Ravichandran and Hovy (2002) described how by searching the web for "Mozort 1756" or "Newton 1642" a large amount of patterns indicating the year of birth of

a person can be extracted without the need to a training set. Similarly, the KnowItAll (Etzioni et al., 2004) engine searches the web for specific phrases like "... such as ..." or "... including ..." for extracting new classes of objects.

In contrast to data sources such as Wikipedia and Freebase which contain vast amount of data with high redundancy, sparse datasets have limited occurrences of entities and textual patterns. In absence of *manually labeled training sets* and lack of *data redundancy*, distant supervision can be very beneficial as it uses existing knowledge repositories to perform information extraction. This approach is very promising in dealing with sparse data (Mintz et al., 2009), though there exists challenges in dealing with uncertainties. For instance, a single entity might be referred to with different names due to spelling errors and name variations, or multiple entities might be referred with identical names. Therefore, to guarantee a successful distant supervision, it is mandatory to make use of entity resolution techniques (Christen, 2012; Christen and Gayler, 2008) and in particular *identity resolution* approaches (Watts et al., 2002; Bilge et al., 2009; Bartunov et al., 2012) prior to extraction of relations.

In this chapter, we use the notion of *relation fingerprint* or in short a *fingerprint* that is a noise-tolerant condensed representation of a relation between two or more individuals. This condensed representation allows for introducing the Identity Resolution method into the distant Supervision Approach. We propose REDS (for Relation Extraction based on Distant Supervision) which uses a combination of shape features and lexical features to extract named entities from the unstructured data. It constructs the candidate relations from the tuples of named entities existing in each single unstructured document. Fingerprints corresponding to the candidate relations are generated, and the existing knowledge repository is queried for each fingerprint. If the knowledge repository contains any *Levenshtein neighbor* of the fingerprint the relation between the named entities is revealed. Besides, the statistics of data are used to assess the confidence level of the discovered relation and predict its type. As an additional step, by using the extracted relations as a training set, we learn a classifier to extract relations from the portions of the unstructured data for which no knowledge repository exists.

This chapter makes the following contributions. a) it proposes the REDS approach which extends the available work on relation extraction by considering the sparsity and uncertainties of the data; b) it introduces the notion of *fingerprint* for the relations, which enhances the existing work on the identity resolution by emphasizing on importance of using the framework of information retrieval, and c) it presents an empirical analysis over a genealogical dataset to validate REDS, and demonstrate its high accuracy in real-world settings.

3.1 Motivating Example

Consider the following intuitive example. Querying "Ian" and "Eibe" separately on the web shows that "Ian" is mentioned in more than 300M webpages, and "Eibe" is mentioned in about 500K webpages. These webpages are about musicians, actors, philosophers, fruits, etc. However, as soon as we search the web for the combination "Ian + Eibe" the search results are suddenly narrowed down to a small subset consisting of the joint work of two computer scientists "Ian H. Witten" and "Eibe Frank" on the topic of Data Mining. This suggests that the string "Ian + Eibe" acts indeed like a fingerprint for finding the specific academic relationship between these two researchers. Apparently, by adding the family names to this fingerprint "Ian Witten + Eibe Frank" the search results will be more precise and also more robust to name variations and spelling errors (e.g., after removing, adding or altering one or two characters still the same precise results are attained). The same characteristic can be seen in online social networks and genealogical historical archives. While a person's full name can be very common in a large data corpus, its combination with the full name of his/her friend or spouse immediately narrows the search results down to a relatively precise subset. Generating fingerprints by combining information from multiple records is the main motivation behind the REDS method introduced in this chapter and the HiDER tool described in the next chapter.

3.2 Problem Definition

Consider an unstructured dataset $\mathcal{U}$. Each document $d \in \mathcal{U}$ is represented by a text d.text, where a set of references d.refs are mentioned in d.text. Each reference $r \in d$.refs is a string representing a person name. Also a set of relations[1] d.rels exist in d.text. Each relation $\mathcal{R} \in d$.rels is defined over a set of references such that $\mathcal{R}$.refs $\subset d$.refs, it has a type $\mathcal{R}$.type and a pointer $\mathcal{R}$.cid to the certificate (i.e., the unstructured text document) it is extracted from.

Additionally, there exists a knowledge repository $\mathcal{K}$ that consists of a set of relations. Each relation $\mathcal{R} \in \mathcal{K}$ is defined over a set of references $\mathcal{R}$.refs, has a type $\mathcal{R}$.type and a pointer $\mathcal{R}$.cid to the certificate it is extracted from.

We assume that $\mathcal{U}$ and $\mathcal{K}$ are sparse datasets (i.e., each real entity is referred in $\mathcal{U}$ and $\mathcal{K}$ for a limited number of times). Besides, we assume name variations and spelling errors exists in both datasets. We define our problem as following.

[1] In this chapter the term *relation* refers to the way in which two or more references are connected (e.g., marriage relation, or co-occurrence relation) and is different from the concept of relation in relational databases.

Problem - *Given a knowledge repository $\mathcal{K}$, a given document $d \in \mathcal{U}$ and a set of references $S_r = \{r_1, r_2, \ldots, r_m\}$ that are mentioned in consecutive order in d.text, find type of relation $\mathcal{R} \in d$ with $\mathcal{R}.refs = S_r$, and assess its confidence level.*

In order to solve this problem, we have to first use an appropriate method for interconnecting the two datasets $\mathcal{U}$ and $\mathcal{K}$. To this end, we define the relation fingerprints that make it possible to directly compare the candidate relations of $\mathcal{U}$ with the explicit relations in $\mathcal{K}$.

3.3 The REDS Method

Figure 3.1 illustrates the inputs, outputs and internal components, C1 to C8, of the proposed relation extraction approach. As its input, REDS has access to a corpus of unstructured data, on the left side of Figure 3.1 and a knowledge repository, on the right side of Figure 3.1. For an arbitrary text document chosen form the unstructured data corpus, references are extracted in C1 and potential relations are constructed in C2, followed by fingerprint generation in C3. Then, in C4, the knowledge repository is queried for each fingerprint. The result of this query can be used to detect the relations in text after being assessed by the assessor component C5, and also in combination with surface features of the unstructured data that are extracted in C6, it can be fed into a classifier for training purposes handled in C7. In C8 the classifier can be used to extract relations by using surface features of the text.

Next, we formally define each of these components C1 to C8 and use examples from the MiSS dataset for better understanding of each component.

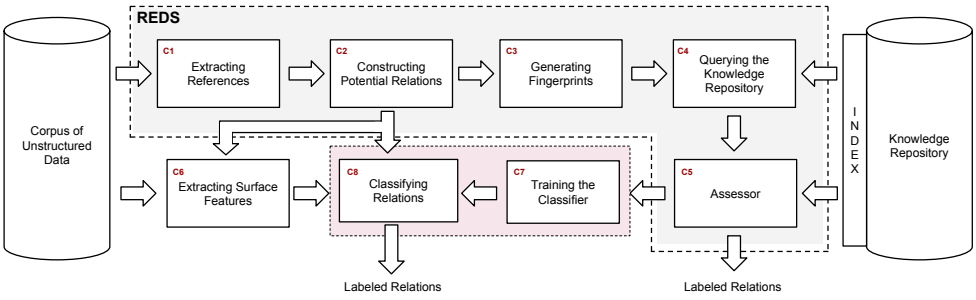


Figure 3.1: The REDS components used for extracting relations from the unstructured data.

Indexing the Knowledge Repository

We, first, describe how to build an inverted index for the knowledge repository $\mathcal{K}$. Please note that the knowledge repository by itself acts as a *forward* index. For each relation in $\mathcal{K}$ we know which references appear in the relation. However, REDS needs an *inverted* index $\mathcal{I}$ that provides a list of references and for each reference the relations it appears in. For creating an inverted index, we parse the knowledge repository and while parsing we use the condensed representation of references, described below, as the index terms.

A person full name might contain prefixes before the given name or family name, middle names and other additional components. Each of these components are subject to change in presence of uncertainties and name variations. Therefore we use the notion of condensed representation of a reference that eliminates the unnecessary information of a full name.

Definition 4. *Let r be a reference. The condensed representation c_r is a single string that is the concatenation of the given name and family name of r after lowercasing every word and filtering out the prefixes.*

$$c_r = \texttt{GetCondensedName}(r)$$

Depending on the quality of data and the domain of use, the condensed representation of a reference in Definition 4 can be further customized. For instance, each given name or family name can be standardized using a list of standard names. However, such customizations are beyond the scope of this chapter; the interested reader can refer to the work of Bloothooft (1994) and Efremova et al. (2014a).

For each condensed representation, we store all the relation ids that the corresponding references appear in. Also, as we need to keep one additional information for each relation, the type of relation is stored too; this will help us with extracting the type of relations at the query time.

Therefore, to build the inverted index, for each relation from the knowledge repository we do the following. For each reference in the relation, we compute the condensed name of the reference. Then, we build the inverted index for the relation by adding the condensed name to the index $\mathcal{I}$ as a key and the tuple of relation id and relation type as the value. Apparently, if a key with the same name exists the tuple of relation id and relation type will be appended to the existing value.

Example - Table 3.1 shows an example of three civil registers. The knowledge repository and inverted index corresponding to these civil registers are shown in Table 3.2, in which the "husband-wife" relations are listed and indexed in $\mathcal{I}$.

Table 3.1: Three civil certificates each containing "husband-wife" and "parent-child" relations. Two family names "Cornelesen" and "Cornelessen" are variations of the same family name. Some extra information such as place and date are not shown.

Certificate 1	Certificate 2	Certificate 3
cid: 1	**cid**: 2	**cid**: 3
type: Marriage Certificate	**type**: Marriage Certificate	**type**: Death Certificate
Groom: Gerardus Willems	**Groom**: Hendrikus Cornelesen	**Deceased**: Maria Cornelesen
Bride: Geertuij Cornelesen	**Bride**: Maria Meerwijck	**Father of deceased**: Cornelis Cornelesen
Father of Groom: Willem Willems	**Father of Groom**: Cornelis Cornelesen	**Mother of deceased**: Johanna Gijsbers
Mother of Groom: Geertruij Hagen	**Mother of Groom**: Johanna Gijsbers	**Spouse of deceased**: -
Father of Bride: Cornelis Cornelessen	**Father of Bride**: Wilhelmus Meerwijck	
Mother of Bride: Johanna Gijsbers	**Mother of Bride**: Anna Camphoven	

Table 3.2: A knowledge repository consisting of 7 "husband-wife" relations (in short 'HW') shown in dictionary format. In total 11 keys are extracted for the inverted index $\mathcal{I}$. By using the inverted index the references and the relations they appear in can be quickly retrieved.

Relations in $\mathcal{K}$	Inverted index $\mathcal{I}$
$\mathcal{R}_1$={refs:{"Gerardus Willems","Geertuij Cornelesen"}, type:'HW', cid:1}	"Gerardus Willems"→ {[$\mathcal{R}_1$,'HW']}
$\mathcal{R}_2$={refs:{"Willem Willems","Geertruij Hagen"}, type:'HW', cid:1}	"Geertuij Cornelesen"→ {[$\mathcal{R}_1$,'HW']}
$\mathcal{R}_3$={refs:{"Cornelis Cornelessen","Johanna Gijsbers"}, type:'HW', cid:1}	"Willem Willems"→ {[$\mathcal{R}_2$,'HW']}
$\mathcal{R}_4$={refs:{"Hendrikus Cornelesen","Maria Meerwijck"}, type:'HW', cid:2}	"Geertruij Hagen"→ {[$\mathcal{R}_2$,'HW']}
$\mathcal{R}_5$={refs:{"Cornelis Cornelesen","Johanna Gijsbers"}, type:'HW', cid:2}	"Cornelis Cornelessen"→ {[$\mathcal{R}_3$,'HW']}
$\mathcal{R}_6$={refs:{"Wilhelmus Meerwijck","Anna Camphoven"}, type:'HW', cid:2}	"Johanna Gijsbers"→ {[$\mathcal{R}_3$,'HW'],[$\mathcal{R}_5$,'HW'],
$\mathcal{R}_7$={refs:{"Cornelis Cornelesen","Johanna Gijsbers"}, type:'HW', cid:3}	[$\mathcal{R}_7$,'HW']}
	"Hendrikus Cornelesen"→ {[$\mathcal{R}_4$,'HW']}
	"Maria Meerwijck"→ {[$\mathcal{R}_4$,'HW']}
	"Cornelis Cornelesen"→ {[$\mathcal{R}_5$,'HW'],
	[$\mathcal{R}_7$,'HW']}
	"Wilhelmus Meerwijck"→ {[$\mathcal{R}_6$,'HW']}
	"Anna Camphoven"→ {[$\mathcal{R}_6$,'HW']}

Extracting References (C1)

Consider a text document $d \in \mathcal{U}$. REDS takes the raw text in d.text and splits it into words using a tokenizer. Next, it tags each token by its part-of-speech tag. As we're interested in named entities that refer to individuals' names, a combination of shape features (i.e., e.g., uppercase, titlecase, or lowercase) and lexical features (e.g., whole word, prefix/suffix) are used to extract the given names and family names, name prefixes and locations. Then it detects multi-token chunks that refer to a person full name. These chunks are added to the set of references d.refs.

Example - Let d be a document with d.text given as following[2].

> "Johana Gijsbers weduwe Cornelius Cornelesen wonende te Erp heeft verkocht aan Hendrik Geert van der Steen land te Erp in de Melvert sectie A. 89 en 90."

This text can be broken up into words by using punctuations and whitespaces into 27 simple tokens, as shown below.

[2]The translation of this Dutch text in English would be: *Johana Gijsbers widow of Cornelius Cornelesen living in Erp has sold to Hendrik Geert van der Steen a land in Erp in the Melvert sections A. 89 and 90.*

Johana_(t1) Gijsbers_(t2) weduwe_(t3) Cornelius_(t4) Cornelesen_(t5) wonende_(t6) te_(t7) Erp_(t8) heeft_(t9) verkocht_(t10) aan_(t11) Hendrik_(t12) Geert_(t13) van_(t14) der_(t15) Steen_(t16) land_(t17) te_(t18) Erp_(t19) in_(t20) de_(t21) Melvert_(t22) sectie_(t23) A._(t24) 89_(t25) en_(t26) 90_(t27)

We introduce the part of sentence tags (GFN) for Given and Family Names, (FNP) for Family Name Prefixes, (LOC) for Locations, (LOP) for Location Prefixes and (UNK) for words with Unknown part of sentence. A Person Reference chucker (PR-Chunker) can detect the person references in the text by searching for the chunks that correspond to a person references, by mainly following the sequence of (GFN) and (FNP) tags. As a full name contains at least two words as the given and family names, no singular (GFN) tag can be accepted as a PR-chunk. Therefore, a full name is considered as a sequence of (GFN) tags or two of such sequences connected via one or more (FNP) tags.

Therefore d.text can be tokenized as following.

Johana_(GFN) Gijsbers_(GFN) weduwe_(UNK) Cornelius_(GFN) Cornelesen_(GFN) wonende_(UNK) te_(LOP) Erp_(LOC) heeft_(UNK) verkocht_(UNK) aan_(UNK) Hendrik_(GFN) Geert_(GFN) van_(FNP) der_(FNP) Steen_(GFN) land_(LOP) te_(LOP) Erp_(LOC) in_(UNK) de_(FNP) Melvert_(UNK) sectie_(UNK) A._(UNK) 89_(UNK) en_(UNK) 90_(UNK)

Constructing Potential References (C2)

For each subset of references that are in consecutive order of each other a potential relation $\mathcal{R}$ can be considered. Depending on the application at hand, relations can consist of two, three or more references in consecutive order.

Example - In analysis of MiSS dataset the focus is very often on "husband-wife" relations that consist of two references. We assume that every two consecutive person references, in the unstructured data, potentially have relations with each other. Therefore, for each unstructured document $d \in \mathcal{U}$ with m references r_1, r_2, ..., r_m, we generate $m-1$ pairs of references as $\mathcal{R}_1 = (r_1, r_2)$, $\mathcal{R}_2 = (r_2, r_3)$, ..., $\mathcal{R}_{m-1} = (r_{m-1}, r_m)$.

The PR-Chunks in this example are r_1 = "Johana Gijsbers", r_2 = "Cornelius Cornelesen", and r_3 = "Hendrik Geert van der Steen". Therefore, the reference pairs lead to two candidate relations $\mathcal{R}_1$.refs={"Johana Gijsbers", "Cornelius Cornelesen"} and $\mathcal{R}_2$.refs={"Cornelius Cornelesen", "Hendrik Geert van der Steen"}.

Generating Fingerprints (C3)

Using Definition 4, we define the fingerprint of a relation $\mathcal{R}$ as following.

Definition 5. *Let $\mathcal{R}$ be a relation over a set of references $\mathcal{R}.refs$. The fingerprint*

$$\begin{aligned} \mathcal{F} &= \texttt{GenerateFingerprint}(\mathcal{R}) \\ &= \big\{\texttt{GetCondensedName}(r) \big| r \in \mathcal{R}.refs\big\} \end{aligned} \tag{3.1}$$

is defined as the set of condensed representation of each reference $r \in \mathcal{R}.refs$.

Example - Using Definitions 4 and 5, REDS generates the following two fingerprints $\mathcal{F}_1$= {"Johana Gijsbers", "Cornelius Cornelesen"} and
$\mathcal{F}_2$={"Cornelius Cornelesen", "Hendrik Steen"}.

Querying the Knowledge Repository (C4)

REDS needs to answer two query types: 1) the exact matching queries: These queries contain some strings, and we want to find every relation that contains references with condensed name equal to one of those strings. 2) the Levenshtein matching queries. These queries contain some strings and an integer a distance threshold $L_d \geq 0$. To answer these queries we want to find every relation that contains a reference for which the Levenshtein distance between the condensed name of the reference and one of the strings doesn't exceed L_d. The former type of query is a simpler case of the latter type when $L_d = 0$. Therefore, from now on we focus on the latter type of query. In the Levenshtein matching queries we have to search the index list $\mathcal{I}$ for the *lexical Levenshtein neighbors* of the query string. To this end, we use the scalable and fast *Levenshtein automata* proposed by Schulz and Mihov (2002). If the Levenshtein automata finds the Levenshtein neighbors of the query term, then the tuples of relation ids and relation types will be retrieved otherwise an empty set will be returned. We define a function for finding the Levenshtein neighbors of a string as following.

Definition 6. *Let s be a string and $\mathcal{I}$ be the index list of $\mathcal{K}$. By using a Levenshtein automata we search for every $i \in \mathcal{I}$ where the Levenshtein distance between s and i doesn't exceed L_d. We define the following function.*

$$\mathcal{N} = \texttt{GetLevenshteinNeighbors}(s, \mathcal{I}, L_d) \tag{3.2}$$

where $\mathcal{N}$ is a list of tuples including the relation ids and relation types.

Assessor (C5)

The Assessor component uses the statistics from data to measure the confidence level of the relation and also predicts its type by using the indexed knowledge repository $\mathcal{K}$. Following comes some definitions we need to assess the relation $\mathcal{R}$ using its fingerprint $\mathcal{F}$.

Definition 7. *Let $\mathcal{F} = \{s_1, s_2, \ldots, s_n\}$ be a relation fingerprint and $L_D \geq 0$ be the maximum acceptable Levenshtein distance for Levenshtein neighbors. Let $\mathcal{N}_i$ be the set of relations that contain the Levenshtein neighbors of s_i*

$$\mathcal{N}_i = \texttt{GetLevenshteinNeighbors}(s_i, \mathcal{I}, L_d)$$

then we define $\texttt{Support}(\{s_i\})$ *to be the number of occurrences of Levenshtein neighbors of s_i in $\mathcal{I}$ as*

$$\texttt{Support}(\{s_i\}) = |\mathcal{N}_i|.$$

Let

$$\mathbf{N}_c = \bigcap_{i=1,2,\ldots,n} \mathcal{N}_i$$

be the common tuples among all $\mathcal{N}_i$s, then we define $\texttt{Support}(s_1, s_2, \ldots, s_n)$ *which is the number of those relations that every $s_1, s_2, \ldots, s_n$ has a levenshtein neighbor in it as*

$$\texttt{Support}(\{s_1, s_2, \ldots, s_n\}) = \left|\mathbf{N}_c\right|.$$

We use the support of fingerprint elements and the set of common levenshtein neighbors in Definition 7 to compute the confidence level and predict the type of a relation. These two will be defined as following.

Definition 8. *Let $\mathcal{F} = \{s_1, s_2, \ldots, s_n\}$ be a fingerprint relation. Then confidence level of $\mathcal{F}$ is defined as*

$$\texttt{GetConfidenceLevel}(\mathcal{F}) = \frac{\texttt{Support}(\{s_1, s_2, \ldots, s_n\})}{\texttt{Support}(\{s_1\}) + \texttt{Support}(\{s_2\}) + \cdots + \texttt{Support}(\{s_n\})}$$

Let $\mathbf{N}_c$ be the set of tuples of relation id and relation type according to Definition 7. Let $\mathbf{N}_c.type$ be a list of all relation types seen in this $\mathbf{N}_c$. Then we use $\mathcal{F}$ to define the predicted type as

$$\texttt{GetPredictedType}(\mathcal{F}) = \texttt{MostFreq}(\mathbf{N}_c.type)$$

where the $\texttt{MostFreq}(list)$ *function, returns one of the elements that is most frequent in the list.*

We define the threshold level θ such that the relation $\mathcal{R}$ is considered to be true if $\texttt{GetConfidenceLevel}(\mathcal{F}) > \theta$.

Algorithm 1 describes how REDS approach can be implemented to extract relations in an unstructured dataset $\mathcal{U}$.

input : The unstructured dataset $\mathcal{U}$
input : The knowledge repository $\mathcal{K}$
output: Set of valid relations $\mathcal{R}_{+}^{\mathcal{U}}$ extracted in $\mathcal{U}$
output: Set of invalid relations $\mathcal{R}_{-}^{\mathcal{U}}$ extracted in $\mathcal{U}$

Build the inverted index $\mathcal{I}$ for $\mathcal{K}$
Initialize the threshold of the confidence level θ
Initialize the Levenshtein distance threshold L_d
$\mathcal{R}_{+}^{\mathcal{U}} := \emptyset$
$\mathcal{R}_{-}^{\mathcal{U}} := \emptyset$
foreach d **in** $\mathcal{U}$ **do**
  Extract the set of references d.refs
  Construct the set of potential relations d.rels
  foreach $\mathcal{R} \in d.rels$ **do**
    $\mathcal{F} := \texttt{GetFingerprint}(\mathcal{R})$
    $\mathcal{R}$.confidence := $\texttt{GetConfidenceLevel}(\mathcal{F})$
    if $\mathcal{R}.confidence > \theta$ **then**
      $\mathcal{R}$.type := $\texttt{GetPredictedType}(\mathcal{F})$
      insert tuple $\mathcal{R}$ to $\mathcal{R}_{+}^{\mathcal{U}}$
    else
      $\mathcal{R}$.type := *null*
      insert tuple $\mathcal{R}$ to $\mathcal{R}_{-}^{\mathcal{U}}$
    end
  end
end
return $\mathcal{R}_{+}^{\mathcal{U}}$,$\mathcal{R}_{-}^{\mathcal{U}}$

Algorithm 1: Extraction of relations by using distant supervision. REDS generates the fingerprint of a candidate relation and queries the index $\mathcal{I}$ of the knowledge base $\mathcal{K}$ for this fingerprint. Based on the query answer if the relation is valid REDS measures the confidence level and predicts the type of the candidate relation. REDS returns sets of valid and invalid relations.

Example - Having an inverted index built for the knowledge repository, we can resolve the stream of queries that are generated to predict the relations in the unstructured data. For each fingerprint $\mathcal{F}$ generated from a relation in a notarial act we query the civil registers for the elements in the fingerprint allowing for a Levenshtein distance of maximum $L_d = 2$. We compute the confidence score of the relation by using Definition 8. For example, consider the three references extracted from the text document r_1 = "Johana Gijsbers", r_2 = "Cornelius Cornelesen", and r_3 = "Hendrik

Geert van der Steen". Their condensed names are s_1 = "Johana Gijsbers", s_2 = "Cornelius Cornelesen", and s_3 = "Hendrik Steen" and their Levenshtein neighbors appear 18, 8 and 9 times in $\mathcal{I}$, respectively. Besides, the fingerprint $\mathcal{F}_1$ ={"Johana Gijsbers", "Cornelius Cornelesen"} appears 7 times (including the 3 times seen in the index of Table 3.2 in $\mathcal{R}_3, \mathcal{R}_5, \mathcal{R}_7$) in $\mathcal{I}$. The fingerprint $\mathcal{F}_2$ ={"Cornelius Cornelesen", "Hendrik Steen"} doesn't appear in $\mathcal{I}$. Therefore based on Definition 8, we have

$\texttt{Support}(s_1) = 18$ $\texttt{Support}(s_2) = 8$ $\texttt{Support}(s_3) = 9$

$\texttt{Support}(s_1, s_2) = 7$ $\texttt{Support}(s_2, s_3) = 0$

$\texttt{GetConfidenceLevel}(\mathcal{F}_1) = 0.33$ $\texttt{GetConfidenceLevel}(\mathcal{F}_2) = 0$

Extracting Surface Features (C6)

For generating the features for the training set we use the surface features of each document $d \in \mathcal{U}$ as described in following.

Definition 9. *Let $\mathcal{R}$ be a candidate relation defined over the m references $r_1, r_2, \ldots, r_m$. The feature vector*

$$V_F = \texttt{ExtractSurfaceFeatures}(\mathcal{R})$$

returns the t words before r_1, the words between every two consecutive references and t words after the last reference r_m, where t is an arbitrary number. Additionally, V_F includes the length of each set of words.

Example - we use the extracted relations by REDS to build a training set for classification of new potential relations as either positive (i.e., a "husband-wife" relation) or negative (i.e., no relation extracted). First, every relation fingerprint $\mathcal{F}$ with $\texttt{GetConfidenceLevel}(\mathcal{F}) > 0$ (i.e., $\theta = 0$) is labeled as a Positive instance and every relation fingerprint $\mathcal{F}$ with $\texttt{GetConfidenceLevel}(\mathcal{F}) = 0$ is labeled as a Negative instance. Second, we extract the surface features V_F: Using Definition 9 for each relation consisting of two references, V_F includes W_l which is the set of $t = 5$ words before the first reference, W_c which is the set of the words in between the two references, and W_r which is the set of $t = 5$ word after the second reference and the length of each one. The list of the words before and after each reference are cut when reaching another reference.

Training the Classifier (C7) and Classifying Relations (C8)

REDS assesses the validity of potential relations in the unstructured data $\mathcal{U}$, and exports two sets of valid $\mathcal{R}^{\mathcal{U}}_{+}$ and invalid relations $\mathcal{R}^{\mathcal{U}}_{-}$. These two sets provide a training set which we can use to find the discriminative patterns of text capable of

detecting the relations in absence of distant supervision. To this end, we label the set $\mathcal{R}^{\mathcal{U}}_{+}$ as Positive instance and the remaining relations in $\mathcal{R}^{\mathcal{U}}_{-}$ as the Negative instances. We assume the number of the False Positives and False negatives to be limited and not affect the discovery of discriminative patterns.

Using Definition 9 and the outputs of Algorithm 1 we can apply a proper machine learning method on the training instances to train classifier DT. DT can then predict the type of a relation based on the surface features of text. Once DT is trained, it can be applied on the text documents to extract relation with no need to distant supervision.

Example - Among the off-the-shelf machine learning methods, we apply the Decision Tree classifier since the patterns discovered by this method are easily interpretable by the domain experts (Ratanamahatana and Gunopulos, 2003). The decision tree is built top-down from the very top node by using ID3 algorithm (Quinlan, 1986, 2014)[3]. In each iteration, the information gain of each attribute of the feature set is computed and the attribute with the largest information gain is chosen to split the dataset. Once the decision tree is constructed it can be used to extract informative patterns for predicting the class label of a new feature set.

We use all the 10,000 Positive instances and then we select, randomly, 10,000 of the Negative instances to build a training set. We extract the surface features for each relation instance, and train a decision tree classier for predicting new relations. Figure 3.2 shows the decision tree with termination criterion of 12 leaves (12 leaves are chosen to keep the decision tree visualizable and interpretable. In practice a much larger decision tree can be built).

Algorithm 2 shows how DT can be used to find the relations for which due to absence of evidence in $\mathcal{K}$, Algorithm 1 is incapable of detecting.

3.4 Empirical Results

In this section, we report the results of implementing REDS on the MiSS dataset.

Effectiveness of Fingerprints

First, we study the effectiveness of using proposed fingerprints by looking at data statistics. Consider a real person as entity e. In the civil registers, we expect references to e for his/her own birth, marriage and death certificates (3 times) and also birth, marriage and death of his/her children (3 times for each children). The death certificate of his/her spouse might refer to e as well (1 time for each marriage).

[3]The C4.5 algorithm, an extension of the ID3 algorithm, can also be used for this purpose. However, for this application we don't expect any significant change in the outcome.

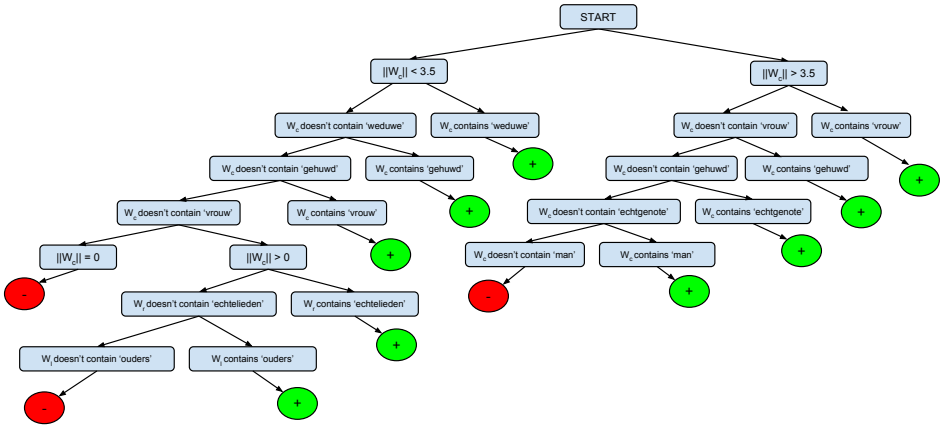


Figure 3.2: A Decision tree built based on the ID3 algorithm to minimize the entropy. This tree generates 12 patterns for labeling a relation as Positive for a "husband-wife" relation or Negative otherwise. The discovered patterns show that in addition to the contents of W_l, W_c and W_r, the length of the text between two references $||W_c||$ plays an important role in extracting relations.

However, as no ground truth exists for this dataset, we can just make an estimation of number of times e is referred to. For an entity who has married once in his/her life and has three children we expect to find 13 references in the civil registers. Let r_i be one of the references referring to e with the condensed name s_i. We expect $\texttt{support}(\{s_i\}) \approx 13$. Also, considering the spouse of e called e', we expect e' to be referred in every civil register where e is referred to except for the birth of e. Let r'_i be a reference to e' with the condensed name s'_i, thus we expect that $\texttt{support}(\{s_i, s'_i\}) \approx 12$. Considering the uncertainties in the dataset and missing data we expect to see less values for the support of single and paired condensed names in most of the cases. To study the two types of support functions, we look at the distribution of supports for single condensed names and compare it with the support distributions for pair of condensed names. First, we define the Complementary Cumulative Distribution Function (CCDF).

$$P_c(k) = \big|\{\alpha | \texttt{support}(\alpha) \geq k\}\big|.$$

The CCDF, $P_c(k)$, shows how often $\texttt{support}(\alpha)$ is above k.

We choose 1000 "husband-wife" relations randomly from the knowledge repository at hand (i.e., the relations in the civil registers). For each relation the fingerprint consisting of a pair of condensed names is extracted. Then, for each condensed name which appears in the fingerprint, we compute its support. We compute the supports for four different Levenshtein distance thresholds $L_d = 0, 1, 2, 3$. Subfigure 3.3a shows

input : The unstructured dataset $\mathcal{U}$
input : The trained classifier DT
output: Set of valid relations $\mathcal{R}_{+}^{\mathcal{U}}$ extracted in $\mathcal{U}$
output: Set of invalid relations $\mathcal{R}_{-}^{\mathcal{U}}$ extracted in $\mathcal{U}$

$\mathcal{R}_{+}^{\mathcal{U}} := \emptyset$
$\mathcal{R}_{-}^{\mathcal{U}} := \emptyset$
foreach d **in** $\mathcal{U}$ **do**
  extract the set of references d.refs
  use d.refs to construct the set of potential relations d.rels
  foreach $\mathcal{R} \in d.rels$ **do**
    $V_F :=$ `ExtractSurfaceFeatures`$(\mathcal{R})$
    $classLabel :=$ DT(V_F)
    if *classLabel* is valid **then**
      $\mathcal{R}$.type = classLabel
      insert tuple $\mathcal{R}$ to $\mathcal{R}_{+}^{\mathcal{U}}$
    else
      $\mathcal{R}$.type = *null*
      insert tuple $\mathcal{R}$ to $\mathcal{R}_{-}^{\mathcal{U}}$
    end
  end
end
return $\mathcal{R}_{+}^{\mathcal{U}}$,$\mathcal{R}_{-}^{\mathcal{U}}$

Algorithm 2: Using the DT classifier to extract relations in the unstructured data $\mathcal{U}$. DT which is trained based on outputs of Algorithm 1, uses the discriminative patterns of text to extract new relations between references.

the CCDF for each threshold in logarithmic scale. We can see that by increasing the distance threshold the maximum support significantly increases. While 36 identical condensed names exist for $L_d = 0$, more than 100 condensed names are Levenshtein neighbors of each other for $L_d = 3$. This is in contrast to our expectations of having about 13 references to an entity. Subfigure 3.3b shows the CCDF for the support of fingerprints (i.e., pair of condensed names). By increasing the distance threshold the average support is increased however the maximum support is fixed around 12.

To further study the changes in maximum support value, for the Levenshtein distance threshold $L_d = 2$ we consider knowledge repositories with different sizes: 500, 800, 1000, 2000, 6000, 8000 and 10,000 relations. Figure 3.4 gives a comparison between the maximum support of single condensed names and paired condensed names for each of these knowledge repositories. In this figure, it is intuitively clear that by increasing the size of the knowledge repository the maximum number of Levenshtein neighbors of single condensed names continuously increases. However, for the paired condensed names (i.e., fingerprints) this maximum value is very constant. Therefore,

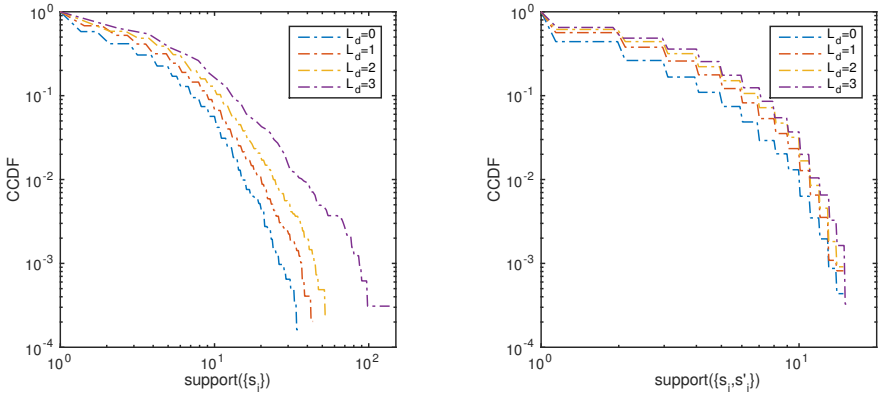


Figure 3.3: The CCDF for the support of single and paired condensed names. In (a) by increasing the Levenshtein distance threshold L_d, the average support and maximum support significantly increase. However, in (b) while the average support increases, the maximum support is constant which shows the effectiveness of using fingerprints in identifying the real entities.

in a large knowledge repository each condensed name and its close Levenshtein neighbors might refer to various entities, while the pair of these condensed names clearly refer to a pair of entities and are very robust to the size of knowledge repository. This result will be confirmed by showing a high precision of REDS in the next subsections.

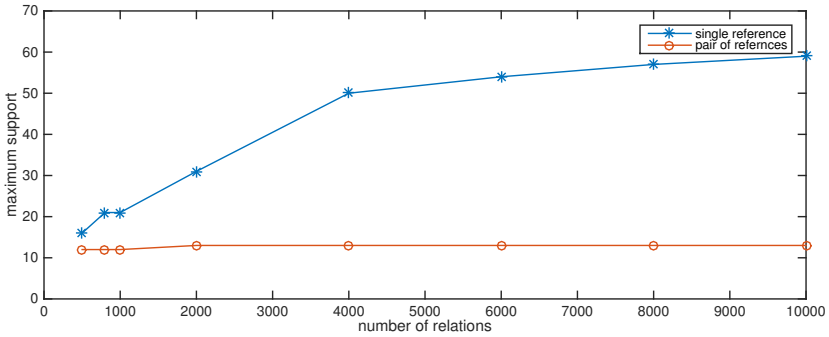


Figure 3.4: The changes of maximum support of single and paired condensed names for knowledge repositories of different sizes. By increasing the size of knowledge repository the number of Levenshtein neighbors of single condensed names increases while the maximum number of paired condensed names in the same Levenshtein neighborhood are very robust to changes in size of the knowledge repository.

Generating and Assessing the Potential Relations

Following the NER approach discussed previously, we extract, on average 4.3 PR-chunks, from 20,000 notarial acts. For more than 14,000 of these notarial acts at least one candidate relation is generated. In total 83,000 relations are extracted, among which about 12,000 receive positive confidence level and are labeled as Positive instances and 71,000 are labeled as Negative. Figure 3.5 shows the distribution of scores for these 83,000 extracted reference pairs.

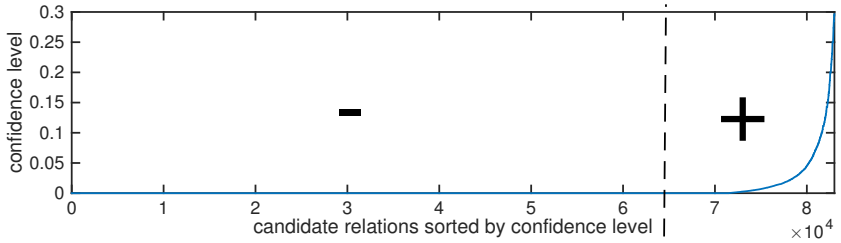


Figure 3.5: Distribution of confidence scores described based on Definition 8. The reference pairs with zero score are labeled as Negative (i.e., left side of the dashed line) and the reference pairs with non-zero score are labeled as Positive (i.e., right side of the dashed line). The number of Negative instances is almost 6 times larger than the number of Positive instances.

Evaluation of Extracted Relations

In order to evaluate the discovered relations by REDS, experts in historical information are asked to check the extracted household relations from notarial acts and decide whether the pair of references have a "husband-wife" relation with each other or not. Manual evaluation of the extracted relations is a very time-consuming process for the domain experts. Clearly it is not possible for the domain experts to evaluate all the 80,000 discovered relations, in an acceptable time. Thus we randomly select 1,000 positive instances, while the confidence level threshold $\theta = 0$ is chosen. Among these instances, experts find 896 true positives and 104 false positives; the precision is then 0.90. Due to absence of any ground truth for this dataset, the measurement of recall is a very difficult task (for discussion on challenges in measuring recall we refer to work of Efremova et al. (2015)). We repeated this evaluation for other confidence level thresholds $0 < \theta \leq 1$ and the changes of precision are shown in Figure 3.6. Figure 3.6 also illustrates the distribution of extracted relations with respect to the confidence level threshold. According to Figure 3.6 although by increasing the confidence level the precision increases, large portion of the extracted relations have very

low but positive confidence level. This confirms that choosing $\theta = 0$ is a right choice for this application as it contains the most amount of true positive relations with an acceptable precision.

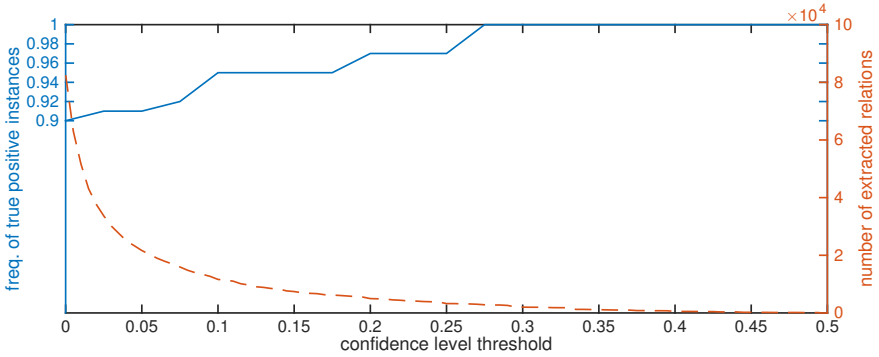


Figure 3.6: Effect of confidence level threshold on precision of extracted relations and overall number of extracted relations. By increasing the threshold of confidence level θ, the number of extracted relations decreases drastically and the number of true positive instances compared to the false positives slightly increases. With threshold $\theta = 0$ REDS extracts 80,000 relations with precision 0.90, while with $\theta = 0.5$ REDS extracts only 100 relations with precision 1.0.

Evaluation of Classifier Relations

Next, we evaluate the decision tree DT. First, we apply DT to the **Test-set 1** which is the same dataset of notarial acts between 1800 to 1912 which REDS was applied on. Second, more importantly, we apply DT to **Test-set 2** consisting the notarial acts between 1700 to 1800, which do not have any time overlap with the available knowledge repository, and can be assumed as a real test set (See Figure 2.2). We select randomly 700 relations in each case including the negative instances. Table 3.3 gives the evaluation results, precision, recall and accuracy of REDS and DT. In this table, as the references in civil registers don't have overlaps with the notarial acts in Test-set 2 the REDS can not be applied on this test set.

Table 3.3: Measuring the accuracy of REDS and DT for extracted relations from the unstructured text. #Ins., #TP, #FP, #TN and #FN are the number of evaluated, true positive, false positive, true negative and false negative instances, respectively.

	#Ins.	#TP	#FP	#TN	#FN	Precision	Recall	F-Measure
REDS	1000	896	104	-	-	0.90	-	-
DT on Test-set 1	700	205	45	415	35	0.82	0.85	0.89
DT on Test-set 2	700	158	52	458	32	0.75	0.79	0.88

3.5 Discussion

The analysis provided in the previous section, showed that REDS is a precise approach which extracts 90% of the relations correctly and has the potential to generate a training set for supervised learning of a classifier. The classifier can extract new relations with precision of 0.75 and recall of 0.79. In Section 3.4, we showed that in our application the confidence level acts as a binary assessor, such that for non-negative levels we can accept the validity of the relation. We consider this as a result of data sparsity; the entities that are mentioned in the text documents are referred to for a handful of times. Therefore, co-occurrence of two references in at least one relation of the knowledge repository is an indication of a high probability for the validity of the potential relation which includes the same two references. We see this behavior of REDS as an advantage that allows Algorithm 1 to provide a fine-grained filtering of results such that there is no need to the ranking of results.

In Section 3.4, we also saw about 25% False Positives as the result of applying the trained decision tree on a test set. Here we mention two main reasons for prediction of invalid relations: First, NER module can make mistakes in forming the PR-Chunks, and in turn the queries on fingerprints return invalid results. Second and more importantly, due to the dynamics of the data in course of centuries, the text grammar, word collections and topics of the documents change; the patterns that the decision tree is trained to detect do not appear in the text documents in the test set, thus the classifier is not capable of predicting correct relations. This is another proof for the importance of using distant supervision in detecting relations.

The proposed REDS approach doesn't make any assumption on quality of the knowledge repository. Although, a deduplicated and cleaned knowledge repository might improve the accuracy of REDS, the use of relation fingerprints provides a fast dedupliation of the knowledge repository on the fly. Thus, building an inverted index for the knowledge repository is the only preprocessing requirement of REDS.

The integration of newly extracted relations were beyond the scope of this chapter. However, one can imagine that as the valid relations are predicted in the unstructured data, relations can be added to the knowledge repository.

4

Toward Scalable Identity Resolution

This chapter is based on:
B. Ranjbar-Sahraei, J. Efremova, H. Rahmani, T. Calders, K. Tuyls and G. Weiss, "HiDER: Query-Driven Entity Resolution for Historical Data", In Proceedings of the European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML PKDD), Porto, Portugal, 2015.

J. Efremova, B. Ranjbar-Sahraei, H. Rahmani, F. A. Oliehoek, T. Calders, K. Tuyls and G. Weiss, "Multi-Source Entity Resolution for Genealogical Data", In G. Bloothooft, P. Christen, K. Mandemakers, and M. Schraagen, editors, *Population Reconstruction*, pp. 129-154, Springer International Publishing, 2015.

In the previous chapter, we described how to use an indexing data structure to extract information from unstructured data. To extend the proposed indexing idea further, in this chapter we combine it with a few data processing steps in a new setting, that to the best of the author's knowledge didn't exist before, to perform scalable identity resolution in heterogenous data. This new setting implements a query-driven approach that retrieves a small subset of data corpora based on the user query, analyses this small subset on the fly and integrates the results in real-time.

We apply this new setting as a tool called HiDER (for Historical Data Entity Resolution), and specifically test it with the MiSS dataset introduced in Section 2.4. The outputs of HiDER are family networks and event timelines visualized in an integrated way. HiDER is being tested at BHIC center, and despite the uncertainties existing at MiSS dataset, the extracted entities have high certainty and are enriched by extra information.

4.1 Query-Driven Identity Resolution

Designing a naive Identity Resolution algorithm for a *small static* dataset is straightforward: assuming each record corresponds to a reference in the dataset at hand, the Content-matchers and Context-matchers can be used to compare every two record in order to reveal whether they refer to the same entity or not. A naive method, however, has two main drawbacks: **d1**) it's a quadratic time algorithm (i.e., its running time is $\mathcal{O}(n^2)$ where n is the number of records); thus, it is not feasible to be applied on large datasets, and **d2**) in case of dynamic time varying data in which records are being removed, added or modified, after each data manipulation the identify resolution algorithm should be rerun on the whole dataset; as the new/modified records can potentially change the contextual information of all existing records. **d1** is immensely studied in the literature, techniques ranging from Blocking methods (Baxter et al., 2003; Michelson and Knoblock, 2006; Bilenko et al., 2006) to collective entity resolution (Bhattacharya and Getoor, 2007) are proposed and shown to be effective in dealing with large datasets. However, **d2** is relatively less addressed by the research community and most of the entity resolution algorithms are not flexible in dealing with dynamic time-varying data.

Inspired by the work of Altwaijry et al. (2013), we propose a query-driven identity resolution method which works as follows. First a user gives a query which consists of one or more entity names. Then, based on the query and using an inverted-index data structure we retrieve relevant information from the existing data corpora. An identity resolution method is implemented on the retrieved information, on the fly, and the final results are presented to the user, in form of list of entities, information about each entity and linkage information between them. As such, the proposed method is flexible in the sense that it adapts with minimal effort to changes in the corpora. Figure 4.1 shows different modules of HiDER.

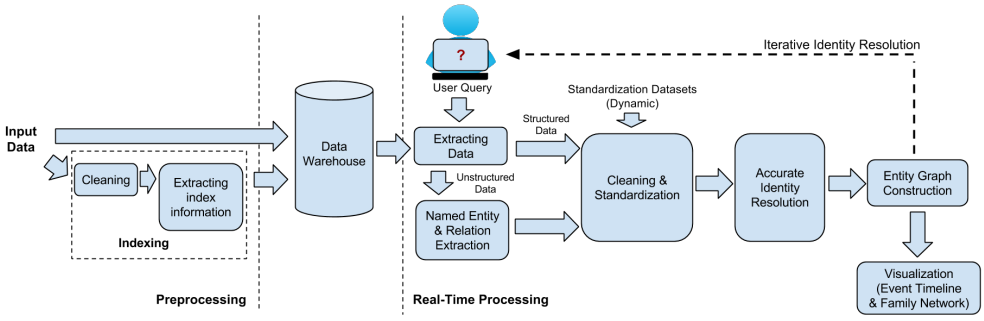


Figure 4.1: The HiDER query-driven identity resolution process.

4.2 The HiDER Backend

The proposed HiDER system is developed on an Apache web server, equipped with Solr search platform. HiDER works as follows.

The **input data** of HiDER is MiSS data which consists of historical documents of the 18^{th} and 19^{th} centuries in the form of structured civil registers and unstructured notarial acts (Further described in Section 2.4). We refer to each civil register or notarial act as a *record* and each person mentioned in a record as a *reference*. Next, we divide the process conducted by HiDER system in two phases, namely *Preprocessing* and *Real-time processing*.

Preprocessing

Upon arrival of a new record or when an existing record is updated, the important information of the record is first **cleaned**. This cleaning process consists of removing stubborn spaces, improper lower/upper cases and nonprinting characters. Also, in this phase HiDER standardizes the date formats and removes redundant information.

After cleaning the records, HiDER *extracts index information* from the records. For the structured civil registers, the condensed name (see Definition 4 in Section 3.3) of each reference will be a key which points to the record address. For the unstructured documents, the superficial features of text as explained in Section 3.3 are used to extract the references. The condensed name of these references are the keys which point to the record document. Additionally, other index information for location, date and type of records are used to build an inverted-index.

The extracted information is added to the **inverted index** data structure and the cleaned record is stored in the data warehouse.

Real-Time Processing

Each time a user query arrives, the Real-Time Processing is carried out by HiDER on the incoming query. The query can contain some keys such as reference name, location, date and place. HiDER uses these keys to retrieve information from the inverted-index data structure. User is also allowed to choose between a *strict* or a *fuzzy* search. The latter allows users to expand their search space and cover possible name alternatives. Also, a *faceting feature* is provided to the user which guides her to drill into her target data (Illustrated as part of the HiDER interface in the next section).

In this stage of real-time processing, we expect to have a small subset of data (e.g., the retrieved amount of data is less than a 1000 records for a user query which contains one or two reference names). The retrieved *unstructured* data is then further processed

for extraction of **Named Entities** and **Relation Extraction** (e.g., by using the DT classifier explained in Section 3.3). Additional **Cleaning** and **Standardization** is applied to the outputs of previous modules. For instance, names with spelling variations are standardized using an external given/family name alternative list[1]. The list is continuously updated based on the user feedback and experts knowledge, and the updates are incorporated in answering future queries.

Once a cleaned and standardized version of retrieved information is provided, we conduct an accurate reference comparison for every reference pair. Please note that in this stage a small subset of original data is retrieved and a pair-wise comparison method with running time $\mathcal{O}(n^2)$ can be easily performed in real-time. Therefore, we use a combination of content-matchers and context-matchers to compare every two reference. For instance, two references are considered to refer to the same entity if their names have edit distance less than or equal to 2 and they have at least one similar household member. Also, the year of document issue and other details are used to avoid any mismatch (for more details please see the work of Rahmani et al. (2016)).

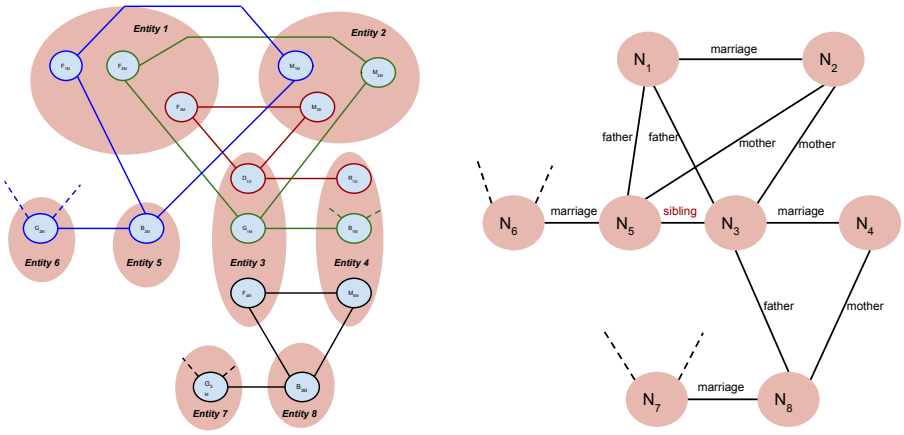
Entity Graph Construction - To use the revealed entities for further visualization, we need to integrate references based on revealed entities and produce an Entity graph. Let $G_R = (V_1, E_1)$ be a graph of references where each node $r_i \in V_1$ is a reference and each edge $e_i \in E_1$ be a household relationship between two references. Each reference r_i contains 9 features $\langle M_1, M_2, \ldots, M_9 \rangle$, where these features are *given name*, *family name prefix*, *family name*, *date* and *place of birth*, *date* and *place of marriage*, and *date* and *place of death*. Each edge has a feature T which determines its type. Depending on the document type, some of these features have null values. Subfigure 4.2a shows a subset of the Reference graph, generated from three Marriage certificates and one Death certificate.

Using revealed entities, we introduce a set of matched references in form of $\mathsf{MR}(N_i) = \{r_{i1}, r_{i2}, \ldots, r_{im}\}$ where every two references r_{ij} and r_{ik} for $j, k = 1, 2, \ldots, m$ are predicated to be a match. In order to convert the Reference graph to Entity graph, every set of references $\mathsf{MR}(N_i)$ are replaced with the node N_i which represents the i^{th} entity. Two entities N_i and N_j are connected with an edge either if there exists $r_{ix} \in \mathsf{MR}(N_i)$ and $r_{jy} \in \mathsf{MR}(N_j)$ where $(r_{ix}, r_{jy}) \in E_1$ or if a sibling relation is detected between N_i and N_j.

For instance, the three matched references shown by F_{1M}, F_{2M} and F_{3D} in Subfigure 4.2a with the role of Father from two Marriage and one Death certificates belong to the set $\mathsf{MR}(N_1)$ and collapse into a single entity node N_1 as shown in Subfigure 4.2b.

Furthermore, the features $\langle M_1, M_2, \ldots, M_9 \rangle$ are updated based on following two

[1]List of alternatives are provided by the Meertens Institute `http://www.meertens.knaw.nl/cms/en/collections/databases`



(a) Reference Graph (b) Entity Graph

Figure 4.2: Converting a Reference graph to an Entity graph, using revealed entities. In (a) F_M, M_M, G_M and B_M denote father, mother, groom and bride in a marriage certificate. F_D, M_D, D_D and R_D denote father, mother, deceased and his/her relative in a death certificate. In (b), $N_1, N_2, \ldots, N_8$ represent the entity nodes.

rules.

R1: In case of conflict among the first three features (i.e., *given name*, *family name prefix* and *family name*) the value with longest length is chosen as the final value.

R2: In case of conflict among the last six features (i.e., *date* and *place* of *birth*, *marriage* and *death*) one of the existing values is chosen randomly.

In practice, **R1** compensates for the typos in form of missing letters and words in given and family names. In most of the cases there is no need to use **R2**, as the references that should be merged have complementary information in form of date and place of birth, marriage and death. In specific cases, for instance merging two born children from two different birth certificates, one of the birth dates and places is chosen randomly and a notification is generated for experts to manually check the conflict.

Let the Entity graph be $G_E = (V_2, E_2)$ where each node $N_i \in V_2$ is an entity, integrating the information of multiple references $r_{i1}, r_{i2}, \ldots, r_{im}$.

Subfigure 4.2b shows the outcome of the entity graph construction procedure applied on all matched references in Subfigure 4.2a.

As, prior to this stage, notarial acts are converted to structured documents, the relationships extracted from notarial acts can also be integrated with the civil registers in the same manner mentioned above.

Finally, it should be mentioned that HiDER allows for **iterative identity resolution**. User of the tool can use entity graph constructed in one round to extend the current query and as such to iteratively construct new entity graphs. Therefore, the user can retrieve the family network of farther relatives of specific entities, and also to manually compensate some of the missing links.

4.3 The HiDER Frontend

Visualization serves as an indispensable tool to evaluate the entity graph manually, and is also a way to deliver the results to the user.

Prototyping - by prototyping a user interface we can efficiently deal with many things that are hard to predict. Throughout this work, to understand both knowledge and expectation of experts in BHIC center, who are the end-users of this tool, we use paper prototyping. Figure 4.3 shows a sample prototyping session. A background poster and a collection of portable paper cards are used to find the best visualization method.

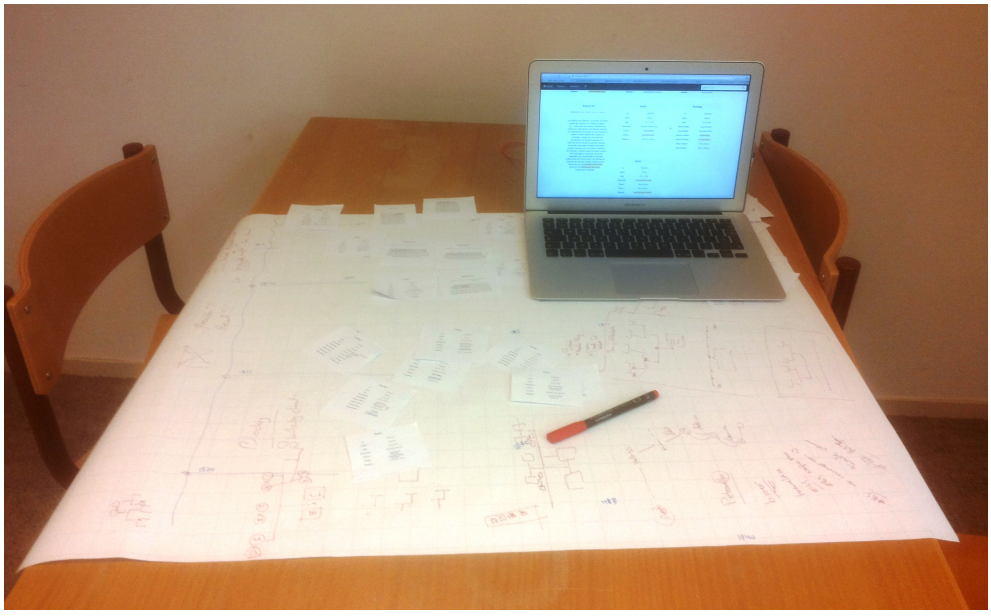


Figure 4.3: Paper prototyping of the HiDER interface in BHIC center with end users.

Timeline Visualization - Using HiDER, the entity graphs can be visualized using *event timelines*. In this visualization method, each record is shown in form of a floating card in which the important named entities are highlighted, and the cards are sorted based on event date (see Figure. 4.4). Using this visualization, a user can easily follow the sequence of events, including birth, marriage, death and property transfers, that have happend throughout the life of an individual or a couple.

Family Network Visualization - To illustrate *family networks*, the main challenge was to visualize complex networks that don't follow the traditional tree structure of a pedigree. Usually, the entity graphs include multiple closed loops and can grow in different directions (e.g., in Subfigure 4.2b the relatives of N_1, N_2, N_4, N_6, N_7 and N_8 can be added that easily make the network structure very complicated). To cope with this complexity, we propose a new visualization scheme *by merging every pair of individual nodes connected with a marriage relationship into single couple nodes*; this decreases the number of nodes in the network and eliminates the network edges drastically. To structure layout of the network with merged nodes, we use graph traversing algorithms to distinguish different generations of the population. Each generation is shown in one single column, that makes it easy to study for genealogists. An example of such a visualization is shown in Figure 4.5.

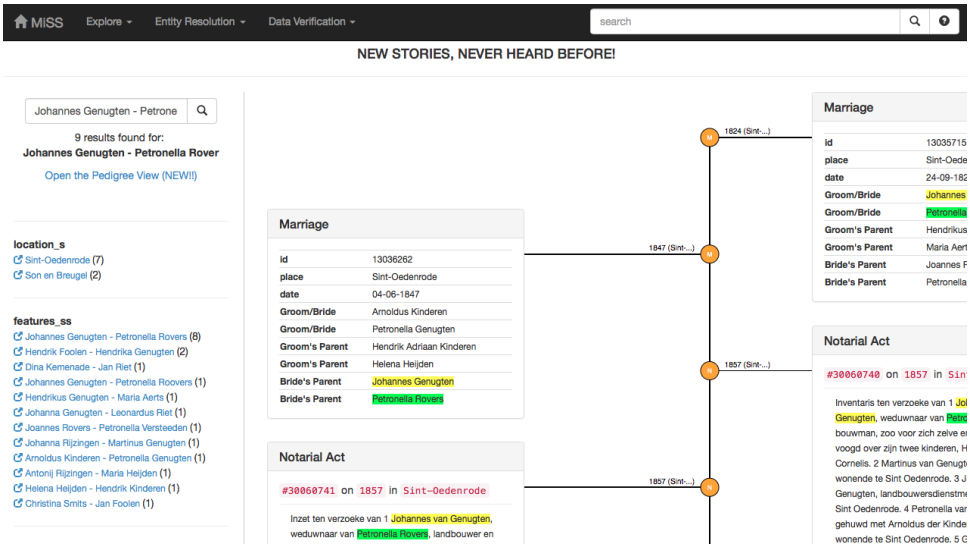


Figure 4.4: Part of the HiDER interface upon arrival of a query: searching tool and faceting are shown on the left, and the event timeline is shown on the right.

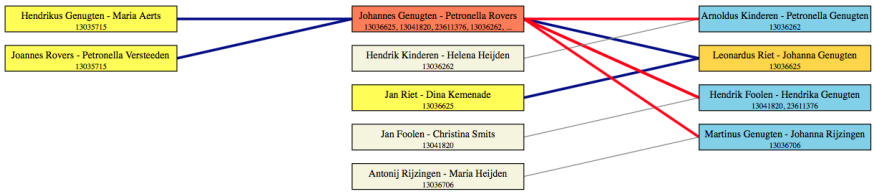


Figure 4.5: The HiDER visualization of a family network: each link connects parents, on the left, to a child and his/her spouse, on the right. Users can interactively focus on the nodes and expand them. Coloring of the nodes depends on the mouse events.

4.4 The Labeling Tool

One of the main difficulties in developing automatic algorithms for information retrieval and entity resolution is lack of training and testing datasets. In both relation extraction and identity resolution (the focus of the previous chapter) we were facing this difficulty, and in order to overcome it, we developed a manual labeling tool which helps us in preparing some training data and also let us verify the final results[2]. In this section, we describe one of the developed labeling tools.

The goal of this labeling tool is to let domain experts to a) extract the references from the text; b) search the civil registers for matches, and c) report the certainty level of the matches. We have designed a search engine, based on the Solr[3] enterprise search platform, which allows experts to use fuzzy search and quickly find the best matches.

Figure 4.6 shows the labeling tool. Features of this tool include automatic name recognition from text documents, suggestion of potential matches and detailed comparison between references.

In practice, the time required to report a correct match between two name-references varies from a few seconds to probably hours of time. This depends on how similar two references are (e.g., whether places, dates, ages, and relatives match or not), and how easy it is to compare those two references. Consequently, the level of confidence in reporting a match varies. We use three qualitative levels: *Absolutely Certain*, *Reasonable Confident* and *Probably*, which give some clues on how historians analyze the available data to find a match. Therefore, the actions that historians take (e.g., which keywords they search for and how fast they can recognize a match), and their level of confidence in reporting the match are all stored in the database. As a result, a rich benchmark is generated that includes the list of matches, the level of

[2]Please note that by using distant supervision and various exploratory data analytics the need to labeled data is minimized in this project

[3]http://lucene.apache.org/solr/

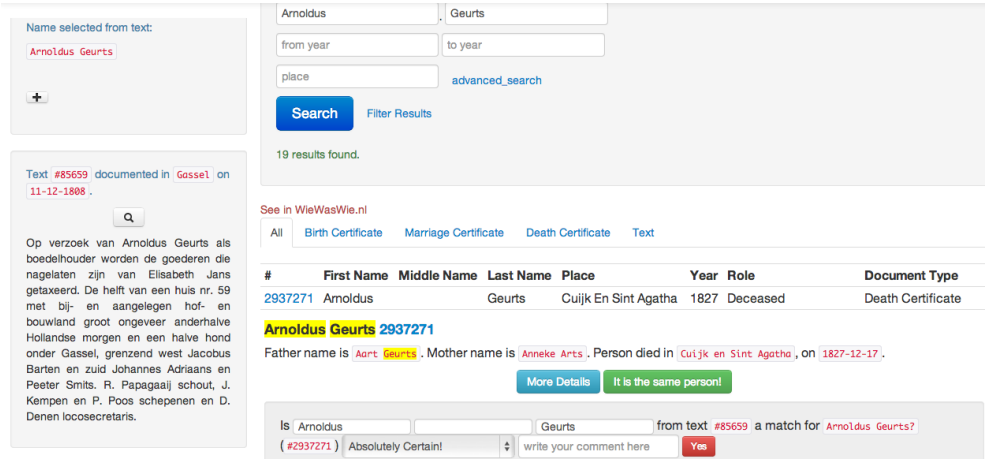


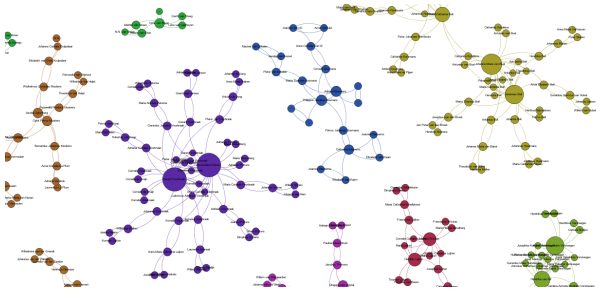
Figure 4.6: Labeling tool to generate training/testing set. A domain expert can confirm a match between the name in notarial act (on the left side) and a reference from a Death certificate (on the right side).

confidence and list of the actions that historians take before reporting the match. This benchmark data has been used by Efremova et al. (2014b) to evaluate the accuracy of entity/identity resolution methods for their genealogical data.

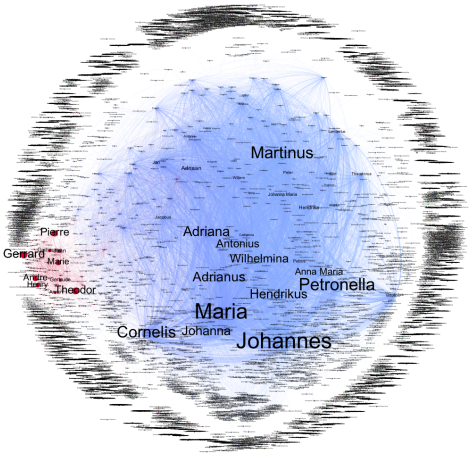
4.5 Discussion

In this chapter, we introduced HiDER which was developed based on the method proposed in **Chapter 3**. This tool has been provided to historians of the BHIC center and has shown to be effective and practical in real life. This tool has the following advantages. (a) HiDER allows for identity resolution across different data sources; (b) the changes in input data and identity resolution algorithm can be incorporated in real time; (c) by using inverted indexing, both structured and unstructured data are handled, and fuzzy search allows for compensating missing data and spelling variations and (d) it visualizes complex family networks in an interpretable way which can not be visualized with traditional methods. According to evaluations by experts of the BHIC center, using HiDER for conducting identity resolution over *MiSS data* generates precise results (e.g., precision above 90% for identity resolution in civil registers as reported by Rahmani et al. (2016)). Also the extraction of named entities and relation extraction from unstructured data has a high precision above 70% as discussed in the previous chapter. However, the evaluation of timelines and family network visualizations is a part of future work due to current limitations in validation

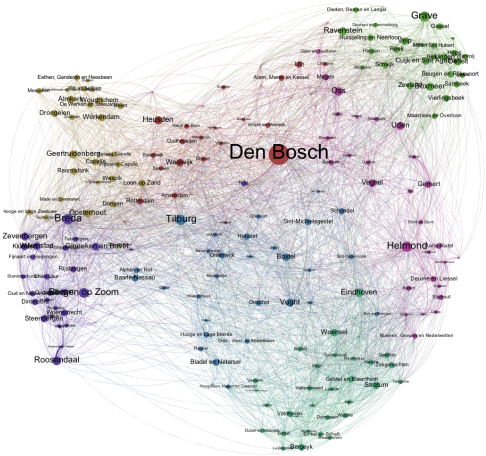
sets and methods.



(a) Extended Family Networks



(b) Name Co-observations



(c) Migration Patterns

Figure 4.7: Various networks extracted from MiSS data by using the HiDER system. (a) By connecting different individual references that refer to the same entity, household relationships can be aggregated that lead to extended family networks. (b) MiSS data provides information on naming conventions at each family. By extracting the co-occurrence information of parent-child names, a network of names is constructed that allows for studying different name communities. (c) By linking civil registers, the location changes for major events in individuals' life are revealed that allow for generation of migration patterns.

Figure 4.7 presents three networks which are generated based on the MiSS data and by using the provided tools of HiDER. Subfigure 4.7a illustrates a snapshot of some extended families which are extracted from the HiDER outputs after searching

for random references in MiSS data. While, the original civil registers contain at most 6 individuals forming an extended family, in this subfigure we see larger extended families. Subfigure 4.7b shows a network in which nodes represent the given names mentioned in civil registers and two given names are connected if they are seen in the same *parent-child* relation. The larger the size of nodes the more often they are seen in the records. The weight of edges correspond to frequency of co-observation of each two names. For this subfigure the community detection algorithm proposed by Blondel et al. (2008) detects two major Name communities, distinguished by different colors. Roughly speaking, the left small community corresponds to French names, while the other large community contains the British and Dutch names. Subfigure 4.7c shows a network in which nodes are the cities mentioned as birth-, marriage- or death-location of individuals and the weight of a link between two nodes shows how often two major events of a person's life are reported in those two cities, thus it gives an indication of migration between those two cities.

The three network snapshots illustrated in Figure 4.7 show the huge potentials in performing identity resolution across various data sources. Please note that each of these networks are dynamic in terms of time and location and the nodes and edges are augmented with various types of attributes. Next chapters will focus on analysis of such networks from an analytical perspective.

5

Social Hierarchies and Heavy-Tailed Distributions

This chapter is based on:
B. Ranjbar-Sahraei, H. Bou-Ammar, K. Tuyls, G. Weiss, "On the Prevalence of Hierarchies in Social Networks", Social Network Analysis and Mining. 2016, 6(58).

B. Ranjbar-Sahraei, H. Bou-Ammar, K. Tuyls, G. Weiss, "On the Skewed Degree Distribution of Hierarchical Networks", In Proceedings of the IEEE/ACM international conference on Advances in Social Network Analysis and Mining (ASONAM), pp. 298-301, Paris, France, 2015.

To analyze the emergence of social networks, a variety of mathematical models have been proposed. The earliest dates back to the 1900's, where Yule (1925) studied the biological evolution of species based on age and population data. Others, e.g., Lotka (1926) provided rules required for describing and analyzing scientific publications. Resulting from these studies, was the identification of the power-law degree distribution by Cancho and Fernández (2008) as a shared common characteristic for a wide-range of networks including the world wide web, protein-protein interaction, airlines and social networks.

Given such a widely-shared characteristic, Barabási and Albert suggested a preferential attachment model for the generation of scale-free graphs exhibiting a power-law degree distribution (Barabási and Albert, 1999). As noted by Durrett (2006), the definition of their process was rather informal. Since then, different precise forms of the Barabási-Albert model have been studied in literature (Bollobás and Riordan, 2003). Though successful at recovering the power-law degree distribution, these studies impose several restricting assumptions on the underlying graph generating process. For instance, such techniques typically adopt a binary attachment model, in which two nodes are either connected or not (Barabási and Albert, 1999; Watts and Strogatz,

1998). Apart from this modeling restriction, another problem inherent to existing binary models lies in their explanatory capabilities. For instance, they fail to manifest connection strengths between individuals; a property being at the core of behavioral emergence in real networks (Barrat et al., 2004a; Granovetter, 1973; Newman, 2001; Barrat et al., 2004b; Garlaschelli et al., 2005; Ranjbar-Sahraei et al., 2014a).

On the other hand, the existence of hierarchical relationships is another shared common characteristic for a wide-range of networks (Clauset et al., 2008; Mones et al., 2012). Research has shown that human physique and body hormones play a crucial role in enabling dominance in the society. While most of the animal societies base their hierarchies on dominance, human societies replace dominance by "prestige" to construct reciprocal relationships between leaders and followers (Price and van Vugt, 2014). Thus, evolutionary considerations of real-world networks suggests the emergence of scale-free behavior (i.e., networks exhibiting a power-law degree distribution) in networks as a result of hierarchal attachment processes that are not reflected through current preferential attachment models.

To provide more realistic modeling outcomes, in this chapter, we contribute by proposing *deterministic* hierarchal evolution processes for dominance-based and prestige-based societies. Contrary to preferential attachment models, our approach only assumes hierarchal connections between individuals, thus bridging the modeling gap to real-world evolutionary networks. Among many advantages, our deterministic setting enables the derivation of the strength distribution in closed-form. Performing this derivation recovers, surprisingly, the exponential and power-law degree distribution as the main property of the resultant hierarchal networks, which explains the prevalence of such hierarchies in societies.

In short, our contributions can be summarized as *(a)* providing a deterministic modeling of linear hierarchal networks[1]; *(b)* validating the proposed model by four real-world datasets, and *(c)* measuring the time complexity and assortativity of the proposed models. Moreover, for the specific case of hierarchical networks with all-to-all connections among individuals we *(d)* derive, for the first time, a closed form of the skewed distribution among individuals in networks having hierarchical interactions; *(e)* explain the prevalence of hierarchies in societies as a result of the characteristics of derived skewed distribution (e.g., high robustness and small average distance (Albert and Barabási, 2002)), and *(f)* compute the Geodesic distance and closeness centrality of the networks in closed form.

[1]linear in the sense that if node A is superior to node B, and node B is superior to node C, then node A is also superior to node C

5.1 Preliminaries

In this section, we present the basic notations and definitions that will be used throughout this chapter.

Notation

General Notation

We define a network as a weighted-graph, $\mathbb{G} = (\mathbb{V}, \mathbb{W})$, consisting of a set of N nodes (or vertices) $\mathbb{V} = \{\boldsymbol{v}_1, \ldots, \boldsymbol{v}_N\}$ and an $N \times N$ adjacency matrix $\mathbb{A}$ as:

$$[\mathbb{A}]_{ij} = \begin{cases} 1 & : \text{if } i \neq j \\ 0 & : \text{otherwise.} \end{cases}$$

Note that we handle the symmetric setting, where if node $\boldsymbol{v}_i$ exhibits a tie with $\boldsymbol{v}_j$, then $[\mathbb{A}]_{ij} = \boldsymbol{a}_{ij} = \boldsymbol{a}_{ji} = 1$. The $N \times N$ weight matrix $\mathbb{W}$ is used to depict the strength of a tie between two vertices $\boldsymbol{v}_i$ and $\boldsymbol{v}_j$, i.e., if $\boldsymbol{a}_{ij} = 1$, $[\mathbb{W}]_{ij} = \boldsymbol{w}_{ij} = \boldsymbol{w}_{ji} \neq 0$ else $\boldsymbol{w}_{ij} = \boldsymbol{w}_{ji} = 0$.

Finally, the neighborhood of a node $\boldsymbol{v}_i$, $\mathbb{N}(\boldsymbol{v}_i)$, is defined as the set containing its adjacent vertices, i.e., $\mathbb{N}(\boldsymbol{v}_i) = \{\boldsymbol{v}_j \mid a_{ij} = 1\}$. Consequently, the degree of a node $\boldsymbol{v}_i$, $\deg(\boldsymbol{v}_i)$, is given by the cardinality of $\mathbb{N}(\boldsymbol{v}_i)$.

Network Hierarchy Notation

Consider a hierarchical constitution for $\mathbb{G}$ such that each individual i observes the tie strengths between every two individuals j and k if $k < i$, $j < i$ and $\boldsymbol{a}_{kj} = 1$. An individual j is called superior to i if $j < i$ and $\boldsymbol{a}_{ij} = 1$. Therefore, we define $\mathbb{H}$ as the set of all tuples (j, i) such that j is superior to i.

The strength of a node is of major importance in the analysis of hierarchical networks. Next, we define three concepts needed in the remainder of the chapter being relative strength, strength observation, and absolute strength.

The relative node strength is defined relative to two nodes i and j. Thus, the relative strength of the j^{th} node with respect to i^{th} node with $j < i$, denoted by $\boldsymbol{\Psi}_i(\boldsymbol{v}_j)$, represents the sum over all edge weights between the j^{th} node and every k^{th} node with $k < i$:

$$\boldsymbol{\Psi}_i(\boldsymbol{v}_j) = \sum_{k=1}^{i-1} \boldsymbol{w}_{jk}. \tag{5.1}$$

In words, when node i is observing node j with $j < i$, it just observes these connections from the other nodes k to j which satisfy $k < i$. The importance of this

concept will be shown in Section 5.3.

The strength observation of the i^{th} node is denoted by the vector

$$\vec{\boldsymbol{\Psi}}_i = [\boldsymbol{\Psi}_i(\boldsymbol{v}_j)|j < i, (i,j) \in \mathbb{H}].$$

This vector contains the observations of the i^{th} node from every other superior j^{th} node (i.e., $(i,j) \in \mathbb{H}$). As we will show in next section, the strength of each tie that the i^{th} individual establishes with superiors depends on the values of such an observation vector. Finally, the absolute strength (i.e., the strength recorded by an external observer) of node i is defined as:

$$\boldsymbol{\Psi}(\boldsymbol{v}_i) = \sum_{j=1}^{N} \boldsymbol{w}_{ij}. \tag{5.2}$$

Mathematical Series

The harmonic series and fraction product are two ingredients which are needed in our analysis for determining closed forms of the strength distributions. Here, we provide two lemmas presenting upper and lower bounds on the values of such summations.

Lemma 1 (Harmonic Series)**.** *Let the harmonic series L_H defined as*

$$L_H(i,N) = \sum_{k=i}^{N-1} \frac{1}{k},$$

then

$$\ln\left(\frac{N}{i}\right) < L_H(i,N) < \ln\left(\frac{N-1}{i-1}\right).$$

Proof. The relatively simple proof of the above lemma is based on the integration results of harmonic series, where $L_H(i,N)$ is lower bounded by $\int_{x=i-1}^{N} \frac{1}{x}\mathrm{d}x$ and upper bounded by $\int_{x=i}^{N+1} \frac{1}{x}\mathrm{d}x$. □

Lemma 2 (Fraction Product Series)**.** *Consider the following product of fractions*

$$L_F(i,N) = \prod_{k=i+2}^{N+1} \frac{2k-4}{2k-5},$$

then

$$\gamma i^{-\frac{1}{2}} < L_F(i,N) < \gamma(i-1)^{-\frac{1}{2}},$$

with $\gamma = \sqrt{N-1}$.

Proof. We use the comparison test to compute the lower and upper bounds of $L_F(i,N)$. Firstly, consider

$$Q(i,N) = \prod_{k=i+2}^{N+1} \frac{2k-5}{2k-6}. \tag{5.3}$$

Clearly, $L_F(i,N) < Q(i,N)$ and $L_F(i,N)Q(i,N) = \frac{2N-2}{2i-2}$. Therefore, $L_F(i,N) < \sqrt{\frac{2N-2}{2i-2}} \leq \gamma(i-1)^{-\frac{1}{2}}$ concluding the upper-bound. To determine the lower bound, define

$$Q'(i,N) = \prod_{k=i+2}^{N+1} \frac{2k-3}{2k-4}. \tag{5.4}$$

It can be shown that $L_F(i,N) > Q'(i,N)$ and $L_F(i,N)Q'(i,N) = \frac{2N-1}{2i-1}$. Therefore,

$$L_F(i,N) > \sqrt{\frac{2N-1}{2i-1}} > \sqrt{\frac{2N-2}{2i}} \geq \gamma i^{-\frac{1}{2}}. \tag{5.5}$$

□

Power-Law & Exponential Degree Distributions

In the analysis of weighted networks, typically the Distribution Function (DF) is introduced:

$$P(k) = \left| \left\{ \boldsymbol{v}_i | \forall i, k \leq \boldsymbol{\Psi}(\boldsymbol{v}_i) < k+1 \right\} \right|, \tag{5.6}$$

where $\boldsymbol{\Psi}(\boldsymbol{v}_i)$ defined in (5.2) denotes the strength of node $\boldsymbol{v}_i$ and $|\cdot|$ being the cardinality of the corresponding set.

To ease the analysis, in this work we make use of the Complementary Cumulative Distribution Function (CCDF) defined as:

$$P_c(k) = \left| \left\{ \boldsymbol{v}_i | \forall i, \boldsymbol{\Psi}(\boldsymbol{v}_i) \geq k \right\} \right|. \tag{5.7}$$

The following two lemmas signify the relation between DFs and CCDFs for networks with power-law and exponential distributions:

Lemma 3 (Exponential Distribution)**.** *Consider an exponential distribution of the form $P(k) = ce^{-\alpha k}$. The CCDF can be written as $P_c(k) = \frac{c}{\alpha} e^{-\alpha k}$.*

Proof. Can be easily seen by simple integration. □

Lemma 4 (Power-law Distribution)**.** *Consider a power-law distribution in form of* $P(k) = ck^{-\alpha}$*, where* α *is the power-law exponent. The CCDF* $P_c(k)$ *also follows a power-law but with an exponent* $\alpha - 1$*.*

Proof. Can be easily seen by simple integration. □

Having laid out our notation and providing the required background knowledge, next we present and analyze two dynamical models that reflect networks constructed by dominance and prestige-based evolutionary models. Not only we provide iterative constructing algorithms, but also present a set of theorems studying their stationary points, which interestingly relate to the exponential and power-law distributions.

5.2 Network Dynamics in Hierarchical Networks

We propose, for the first time, a dynamical process which captures the edge dynamics of hierarchical networks. Let $\boldsymbol{w} = \{w_{ij} | \forall (i,j) \in \mathbb{H}\}$ denote the state vector of the process. Each state variable w_{ij} corresponds to the weight of the link between the j^{th} and i^{th} node. To determine the dynamics of the change in the state-variable, one typically considers the rate of change in w_{ij} as a function of all state variables:

$$\dot{w}_{ij} = f(\boldsymbol{w}). \tag{5.8}$$

Due to the nature of hierarchal networks and to simplify the analysis, however, we make use of the following assumption:

Assumption 1. *The tie between* i *and* j*, where* i *is superior to* j*, depends on all connections between* i *and* k *where* k *is also superior to* j*.*

This leads us to study the edge dynamics of a node i as a function of its own weight state as well as its strength observation:

$$\dot{w}_{ij} = f_\Psi(w_{ij}, \vec{\boldsymbol{\Psi}}_i), \qquad j < i. \tag{5.9}$$

In other words, we assume that the dynamics of the linking strength between i and j is independent of any other node l which is higher than i or j in the hierarchy.

Using f_Ψ from Equation 5.9, and sorting the state variables w_{ij} increasingly (based on $Ni + j$), the overall dynamic process can be written as

$$\begin{aligned} \dot{\boldsymbol{w}} &= \frac{\mathrm{d}}{\mathrm{d}t}\left[w_{21}, \ldots, w_{N(N-1)}\right]^{\mathsf{T}} \\ &= \left[f_\Psi(w_{21}, \vec{\boldsymbol{\Psi}}_2), \ldots, f_\Psi(w_{N(N-1)}, \vec{\boldsymbol{\Psi}}_N)\right]^{\mathsf{T}}. \end{aligned} \tag{5.10}$$

To finalize the dynamical model, $f_\Psi(\cdot)$ has to be defined. Considering real-world hierarchal networks, next, we introduce two such models, $f_\Psi^{(\mathbb{D})}(\cdot)$ and $f_\Psi^{(\mathbb{P})}(\cdot)$ corresponding to dominance and prestige based dynamics.

5.3 Dominance-Based Evolution Model (DBEM)

In the dominance-based evolution model (DBEM), the strength of ties between the i^{th} and every other j^{th} individual, with $(i,j) \in \mathbb{H}$ and $i > 1$, follows a simple dynamical rule:

$$\dot{w}_{ij} = f_\Psi^{(\mathbb{D})}\left(w_{ij}, \left|\vec{\mathbf{\Psi}}_i\right|\right), \tag{5.11}$$

where $|\cdot|$ denotes the cardinality of the vector and

$$f_\Psi^{(\mathbb{D})}(w_{ij}, \left|\vec{\mathbf{\Psi}}_i\right|) \quad = \quad \frac{1}{\left|\vec{\mathbf{\Psi}}_i\right|} - w_{ij}.$$

In the above, $\left|\vec{\mathbf{\Psi}}_i\right|$ is a fixed integer denoting the number of superiors to the i^{th} individual. The difference between $\frac{1}{\left|\vec{\mathbf{\Psi}}_i\right|}$ and w_{ij} determines the direction of changes of w_{ij} (i.e., $\dot{w}_{ij}$).

For computing the equilibrium point of the above system, consider an energy function for w_{ij} of the form:

$$V_{ij} = \left(\frac{1}{\left|\vec{\mathbf{\Psi}}_i\right|} - w_{ij}\right)^2. \tag{5.12}$$

By taking derivative of V_{ij} and using the update rule in (5.12), we can write for a fixed i:

$$\dot{V}_{ij} \quad = \quad -2\left(\frac{1}{\left|\vec{\mathbf{\Psi}}_i\right|} - w_{ij}\right)\dot{w}_{ij} = -2\left(\frac{1}{\left|\vec{\mathbf{\Psi}}_i\right|} - w_{ij}\right)^2. \tag{5.13}$$

Using the Invariant Set Theorem (introduced in Section 2.3), we can show that the overall dynamical process has a stable equilibrium point, in which the link between the i^{th} and j^{th} node, $j < i$, converges to $w_{i\star}^{(\mathbb{D})}$:

$$w_{i\star}^{(\mathbb{D})} = \frac{1}{\left|\vec{\mathbf{\Psi}}_i\right|}. \tag{5.14}$$

The equilibrium point in (5.14) explains that the links of node i to all nodes with

lower order (i.e., $j < i$) depends on i. Further, it clarifies that the higher the order is the lower the strength of links are.

Example: To illustrate, consider N agents in a complete graph. Continuously each agent shares its available resources to superior agents. The strength of the connection between nodes i and j reflect the amount of resources transmitted from i to j. According to (5.14) the second individual shares all resources with the 1^{st} (i.e., $w_{21} = \frac{1}{2-1} = 1$). The third however, shares half of the resources with the second and the other half with the first (i.e., $w_{32} = w_{31} = \frac{1}{3-1} = \frac{1}{2}$). Similarly, any agent i shares $\frac{1}{i-1}$ units of the resources with each of the j individuals as long as $j < i$. Therefore, one can see that this model directly captures the dominance of individuals in a linear hierarchical network, where every individual is sharing resources among dominated individuals.

Next, we study the amount of resources each individual receives in such dominance-based network (captured by node's strengths), and compute the distribution of node strengths.

In the following subsections, we focus on complete networks (allowing us to derive numerous characteristics in closed-form) where every j^{th} individual is superior to the i^{th} individual if $j < i$. Thus, $\left|\vec{\boldsymbol{\Psi}}_i\right| = i - 1 \; \forall i > 1$, and

$$w_{i\star}^{(\mathbb{D})} = \frac{1}{i-1}. \tag{5.15}$$

Analysis of Node's Strength

Building on $w_{i\star}^{(\mathbb{D})}$'s definition in Equation 5.15, one can calculate the absolute strength of the i^{th} node, $\boldsymbol{\Psi}(\boldsymbol{v}_i)$ as:

$$\begin{aligned} \boldsymbol{\Psi}(\boldsymbol{v}_i) &= \sum_{j=1}^{N} \boldsymbol{w}_{ij}^{(\mathbb{D})} = \sum_{j=1}^{i-1} \boldsymbol{w}_{ij}^{(\mathbb{D})} + \sum_{j=i+1}^{N} \boldsymbol{w}_{ij}^{(\mathbb{D})} = (i-1)\boldsymbol{w}_{i\star}^{(\mathbb{D})} + \sum_{j=i+1}^{N} \boldsymbol{w}_{j\star}^{(\mathbb{D})} \\ &= 1 + \sum_{j=i+1}^{N} \frac{1}{j-1} = 1 + \sum_{j=i}^{N-1} \frac{1}{j}. \end{aligned} \tag{5.16}$$

Using Lemma 1, it is straightforward to show that:

$$1 + \ln\left(\frac{N}{i}\right) < \boldsymbol{\Psi}(\boldsymbol{v}_i) < 1 + \ln\left(\frac{N-1}{i-1}\right). \tag{5.17}$$

Analysis of Node's Strength Distribution

The distribution of strengths in the DBEM model can be directly computed from the bounds provided in Equation 5.17. The following theorem shows how the CCDF, and consequently the DF of strengths in this model follow an exponential distribution:

Theorem 1 (Strength Distribution in DBEM Model)**.** *For the complete weighted network $\mathbb{G}$, generated using the DBEM model, the DF of the global strength k follows an exponential distribution of the form*

$$P(k) \propto e^{-k}.$$

Proof. Using Equation 5.17 we have:

$$\boldsymbol{\Psi}(\boldsymbol{v}_i) \geq k, \text{ for } \; i \in \left\{1, 2, 3, \ldots, \left\lfloor \frac{N}{e^{k-1}} \right\rfloor \right\}.$$

Hence:

$$P_c(k) = \left| \left\{1, 2, 3, \ldots, \left\lfloor \frac{N}{e^{k-1}} \right\rfloor \right\} \right| \simeq Ne \cdot e^{-k}, \tag{5.18}$$

and consequently:

$$P_c(k) \propto e^{-k}. \tag{5.19}$$

Using Lemma 3, it's straightforward to see that the DF corresponding to (5.19) is exponential, i.e., $P(k) \propto e^{-k}$. □

5.4 Prestige-Based Evolution Model (PBEM)

Having introduced the above model, next we present a prestige-based model, taking our framework a step closer to the formation of hierarchies in real social networks. Consider an arbitrary undirected network with $\mathbb{A}$ as its adjacency matrix and $\mathbb{H}$ as its hierarchical structure. The overall strength of node i in establishing connections with every other j^{th} node with $(i, j) \in \mathbb{H}$ and $i > 1$ is assumed to be limited and sums to 1. Let

$$\dot{w}_{ij} = f_{\Psi}^{(\mathbb{P})}(w_{ij}, \boldsymbol{\Psi}_i(\boldsymbol{v}_j), \left|\vec{\boldsymbol{\Psi}}_i\right|), \tag{5.20}$$

and

$$f_{\Psi}^{(\mathbb{P})}(w_{ij}, \boldsymbol{\Psi}_i(\boldsymbol{v}_j), \left|\vec{\boldsymbol{\Psi}}_i\right|) = \frac{\boldsymbol{\Psi}_i(\boldsymbol{v}_j)}{\left|\vec{\boldsymbol{\Psi}}_i\right|} - w_{ij}, \quad (i, j) \in \mathbb{H}. \tag{5.21}$$

By studying the dynamic process proposed in Equation 5.21, it can be easily seen that $\dot{w}_{ij}, \; i > j$ is a function of w_{kl} for all $k, l < i$. Without loss of generality, we

assume $\boldsymbol{w}_{11}^{(\mathbb{P})} = 1$, such that:

$$\boldsymbol{\Psi}_2(\boldsymbol{v}_1) = 1. \tag{5.22}$$

We also assume that $\boldsymbol{w}_{ii}^{(\mathbb{P})} = 0$ for every $i > 1$. It is again straightforward to compute the equilibrium point of such system as:

$$\boldsymbol{w}_{ij}^{(\mathbb{P})} = \frac{\boldsymbol{\Psi}_i(\boldsymbol{v}_j)}{\left|\vec{\boldsymbol{\Psi}}_i\right|}. \tag{5.23}$$

It is clear that the equilibrium point in (5.23) explains that the connection strength between node i and node j depends on the strength of the ties between nodes i or j and every other k^{th} node with $k < \max\{i, j\}$.

Example: To illustrate, imagine N agents in a complete graph. Continuously the agents with higher order share their available resources with agents exhibiting lower order. The strength of the link between i and j shows the amount of resources which are transmitted. According to (5.23) the second agent shares all resources with the first individual (i.e., $w_{21} = \frac{1}{1} = 1$). The third agent shares *one third* of the resources with the second and *two thirds* with the first (i.e., $w_{32} = \frac{1}{1+2} = \frac{1}{3}$ and $w_{31} = \frac{2}{1+2} = \frac{2}{3}$). Similarly, the i^{th} agent shares portions of the resources with each of the j agents with $j < i$. Those with a lower order, however, receive higher resources compared to the ones with a higher order. This also explains our naming referring to the model as a prestige-based one, where lower orders reflect a "prestige" in the group receiving more resources compared to others.

An immediate result of (5.23) is that:

$$\sum_{j=1}^{i} \boldsymbol{w}_{ij}^{(\mathbb{P})} = \sum_{j=1}^{i-1} \boldsymbol{w}_{ij}^{(\mathbb{P})} = \sum_{j=1}^{i-1} \frac{\boldsymbol{\Psi}_i(\boldsymbol{v}_j)}{\left|\vec{\boldsymbol{\Psi}}_i\right|} = \frac{\left|\vec{\boldsymbol{\Psi}}_i\right|}{\left|\vec{\boldsymbol{\Psi}}_i\right|} = 1. \tag{5.24}$$

Next, we focus on complete networks where every j^{th} individual is superior to the i^{th} individual if $j < i$. For such networks, we will compute the amount of resources each individual receives and also the distribution of node strengths.

Analysis of Node's Strength

Given a complete network, here we determine a closed form solution for the sum over the strength of every j^{th} node from the perspective of the i^{th} node, as long as $j < i$.

Lemma 5. *In the prestige-based evolution model, the sum of the relative node strengths of every j^{th} node from perspective of the i^{th} node, with $j < i$ is:*

$$\boldsymbol{K}(i) : |\vec{\boldsymbol{\Psi}}_i| = 2i - 3.$$

Proof. The above lemma can be proved by using induction:

Initial Step: According to Equation (5.22) we have $|\vec{\boldsymbol{\Psi}}_2| = \boldsymbol{\Psi}_2(\boldsymbol{v}_1) = 1$. Therefore, $\boldsymbol{K}(i)$ holds for $i = 2$.

Inductive Step: Let

$$\boldsymbol{K}(i-1) : |\vec{\boldsymbol{\Psi}}_{i-1}| = 2i - 5,$$

and also note that $\boldsymbol{\Psi}_i(\boldsymbol{v}_j) = \boldsymbol{\Psi}_{i-1}(\boldsymbol{v}_j) + \boldsymbol{w}^{(\mathbb{P})}_{(i-1)j}$. Therefore we can write:

$$\begin{aligned}
|\vec{\boldsymbol{\Psi}}_i| =& \sum_{j=1}^{i-1} \boldsymbol{\Psi}_i(\boldsymbol{v}_j) \\
=& \boldsymbol{\Psi}_i(\boldsymbol{v}_{i-1}) + \sum_{j=1}^{i-2} \left(\boldsymbol{\Psi}_{i-1}(\boldsymbol{v}_j) + \boldsymbol{w}^{(\mathbb{P})}_{(i-1)j} \right) \\
=& \sum_{j=1}^{i-1} \boldsymbol{w}^{(\mathbb{P})}_{(i-1)j} + |\vec{\boldsymbol{\Psi}}_{i-1}| + \sum_{j=1}^{i-2} \boldsymbol{w}^{(\mathbb{P})}_{(i-1)j}.
\end{aligned}$$

By using $\boldsymbol{K}(i-1)$ and Equation 5.24, we arrive at:

$$|\vec{\boldsymbol{\Psi}}_i| = \sum_{j=1}^{i-1} \boldsymbol{\Psi}_i(\boldsymbol{v}_j) = 1 + 2i - 5 + 1 = 2i - 3. \tag{5.25}$$

Therefore, $\boldsymbol{K}(i)$ holds for every i, concluding the proof. □

Analysis of Edge Weights

We can compute the edge weight between the i^{th} and j^{th} node as follows:

Lemma 6 (Edge Weight)**.** *For the weighted graph $\mathbb{G}$, evolved with PBEM, the i^{th} node is connected to the j^{th} node with an edge of weight:*

$$\boldsymbol{K}(i) : \boldsymbol{w}^{(\mathbb{P})}_{ij} = \frac{1}{2i-2} \prod_{k=1}^{i-j} \frac{2i-2k}{2i-2k-1}, \forall j < i. \tag{5.26}$$

Proof. The validity of Equation 5.26 can be proved for each i and for every $j < i$ using induction.

Initial Step: The second node is connected to the first node with $\boldsymbol{w}^{(\mathbb{P})}_{21} = 1$, meaning that $\boldsymbol{K}(2)$ holds.

Inductive Step: Now assume that

$$\boldsymbol{K}(i-1): \boldsymbol{w}^{(\mathbb{P})}_{(i-1)j} = \frac{1}{2i-4}\prod_{k=1}^{i-j-1}\frac{2i-2k-2}{2i-2k-3},$$

holds for every $j < i-1$. For computing the edge weight between the i^{th} and the j^{th} node, recall that $\boldsymbol{\Psi}_i(\boldsymbol{v}_j) = \boldsymbol{\Psi}_{i-1}(\boldsymbol{v}_j) + \boldsymbol{w}^{(\mathbb{P})}_{(i-1)j}$. By using (5.23) and Lemma 5, it can be seen that:

$$\begin{aligned}\boldsymbol{\Psi}_i(\boldsymbol{v}_j) &= \boldsymbol{\Psi}_{i-1}(\boldsymbol{v}_j) + \boldsymbol{w}^{(\mathbb{P})}_{(i-1)j} \\ &= (2i-5)w_{(i-1)j} + \boldsymbol{w}^{(\mathbb{P})}_{(i-1)j} \\ &= (2i-4)w_{(i-1)j}. \end{aligned} \tag{5.27}$$

Using Equations 5.23, 5.27 and Lemma 5, the edge weight between the i^{th} and the j^{th} node can be written as:

$$\boldsymbol{w}^{(\mathbb{P})}_{ij} = \frac{\boldsymbol{\Psi}_i(\boldsymbol{v}_j)}{\sum_{k=1}^{i-1}\boldsymbol{\Psi}_i(\boldsymbol{v}_k)} = \frac{1}{2i-2}\prod_{k=1}^{i-j}\frac{2i-2k}{2i-2k-1}.$$

for $j < i-1$. Therefore, $\boldsymbol{K}(i)$ holds for every i, concluding the proof. □

Before, computing the distribution of strengths for PBEM, we present the following proposition providing the relative strength of the j^{th} node from the perspective of the i^{th} for every $i > j$ in closed form:

Proposition 1 (Relative Node Strength)**.** *For the weighted graph $\mathbb{G}$, evolved according to PBEM, the strength of the j^{th} node from perspective of the i^{th} node is given by:*

$$\boldsymbol{K}(i): \begin{cases} \boldsymbol{\Psi}_i(\boldsymbol{v}_j) = \prod_{k=j+2}^{i}\frac{2k-4}{2k-5} & \text{for } j < i-1 \\ \boldsymbol{\Psi}_i(\boldsymbol{v}_j) = 1 & \text{for } j = i-1. \end{cases} \tag{5.28}$$

Proof. Again, induction can be used to prove the validity of Equation 5.28. Starting with the initial step we get:

Initial Step: From Equation 5.22, the strength of the first node from the perspective of the second node is $\boldsymbol{\Psi}_2(\boldsymbol{v}_1) = 1$. Besides, using Lemma 6 we can deduce that:

$$\boldsymbol{\Psi}_3(\boldsymbol{v}_1) = \frac{\boldsymbol{w}^{(\mathbb{P})}_{11} + \boldsymbol{w}^{(\mathbb{P})}_{21}}{3} = \frac{2}{3}.$$

Therefore, $\boldsymbol{K}(2)$ holds. For the inductive step we proceed as follows:

Inductive Step: Assume that following holds.

$$\boldsymbol{K}(i-1): \begin{cases} \boldsymbol{\Psi}_{i-1}(\boldsymbol{v}_j) = \prod_{k=j+2}^{i-1} \frac{2k-4}{2k-5} & \text{for } j < i-2 \\ \boldsymbol{\Psi}_{i-1}(\boldsymbol{v}_j) = 1 & \text{for } j = i-2. \end{cases}$$

For computing $\boldsymbol{\Psi}_i(\boldsymbol{v}_j)$, consider $\boldsymbol{\Psi}_i(\boldsymbol{v}_j) = \boldsymbol{\Psi}_{i-1}(\boldsymbol{v}_j) + \boldsymbol{w}^{(\mathbb{P})}_{(i-1)j}$. Using Equation 5.23 and Lemma 5, we can show that for every $j < i-1$:

$$\boldsymbol{\Psi}_i(\boldsymbol{v}_j) \quad = \quad \boldsymbol{\Psi}_{i-1}(\boldsymbol{v}_j) + \boldsymbol{w}^{(\mathbb{P})}_{(i-1)j} = \prod_{k=j+2}^{i} \frac{2k-4}{2k-5}.$$

Besides using Equation (5.24), $\boldsymbol{\Psi}_i(\boldsymbol{v}_j) = 1$ for $j = i-1$. Therefore, $\boldsymbol{K}(i)$ holds for every i and the proof is concluded. □

Lemma 7 (Global Strength)**.** *For the weighted graph $\mathbb{G}$, evolved with PBEM, the global strength of the i^{th} node is:*

$$\begin{cases} \boldsymbol{\Psi}(\boldsymbol{v}_i) = \prod_{k=i+2}^{N+1} \frac{2k-4}{2k-5} & \textit{for } i < N \\ \boldsymbol{\Psi}(\boldsymbol{v}_i) = 1 & \textit{for } i = N. \end{cases} \tag{5.29}$$

Proof. We know that $\boldsymbol{\Psi}(\boldsymbol{v}_i) = \boldsymbol{\Psi}_N(\boldsymbol{v}_i) + \boldsymbol{w}^{(\mathbb{P})}_{iN}$ for every $i < N$. Using Equation 5.23 and Proposition 1, we have:

$$\begin{aligned} \boldsymbol{\Psi}(\boldsymbol{v}_i) &= \boldsymbol{\Psi}_N(\boldsymbol{v}_i) + \frac{\boldsymbol{\Psi}_N(\boldsymbol{v}_i)}{2N-3} \\ &= \frac{2N-2}{2N-3} \prod_{k=i+2}^{N} \frac{2k-4}{2k-5} \\ &= \prod_{k=i+2}^{N+1} \frac{2k-4}{2k-5}, \end{aligned}$$

for every $i < N$. Based on Equation (5.24), we have:

$$\boldsymbol{\Psi}(\boldsymbol{v}_N) = \sum_{i=1}^{N-1} \boldsymbol{w}^{(\mathbb{P})}_{Ni} = 1.$$

This concludes the proof. □

Finally, we can compute the strength distribution in a closed form. The following theorem provides the strength distribution of a PBEM model:

Theorem 2 (Strength Distribution)**.** *For the complete weighted graph* $\mathbb{G}$ *evolved with PBEM, the distribution of the global strength* k *follows a power-law with exponent* -3*:*

$$P(k) \propto k^{-3}.$$

For proving Theorem 2, we use Lemmas 2 and 4 to analyze the results of Lemma 7.

Proof. From Lemma 2, the following lower and upper bounds can be computed for the strength of the i^{th} node

$$\gamma i^{-\frac{1}{2}} < \mathbf{\Psi}(\boldsymbol{v}_i) < \gamma(i-1)^{-\frac{1}{2}}, \tag{5.30}$$

where $\gamma = \sqrt{N-1}$. From Equation (5.30), we have

$$\mathbf{\Psi}(\boldsymbol{v}_i) \geq k, \text{ for } \; i \in \{1, 2, 3, \ldots, \left\lfloor \frac{\gamma^2}{k^2} \right\rfloor\}, \tag{5.31}$$

$$P_c(k) = \left| \left\{ 1, 2, 3, \ldots, \frac{\gamma^2}{k^2} \right\} \right| \simeq \gamma^2 k^{-2}. \tag{5.32}$$

Therefore,

$$P_c(k) \propto k^{-2} \tag{5.33}$$

Using Lemma 4, we have

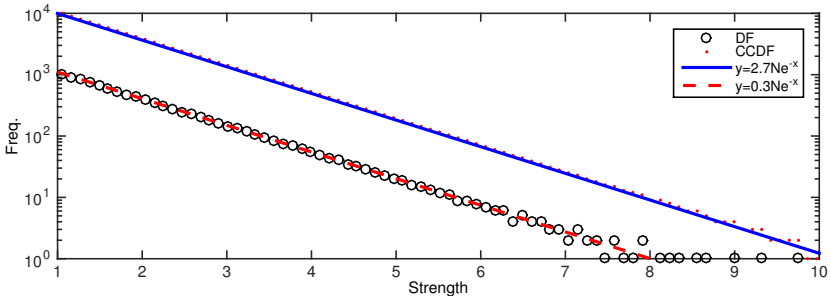
$$P(k) \propto k^{-3}, \tag{5.34}$$

thus proof is concluded. □

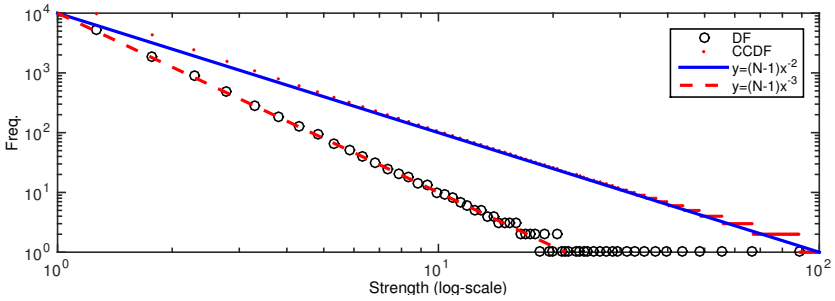
Simulation Validation: Next, we provide a simulation to validate the analytical results on the strength distribution of both DBEM and PBEM models. We initiate a complete graph with 10^4 nodes and random weight adjacency. This network is then evolved under the dynamical processes of both DBEM and PBEM models. The strengths of nodes in the equilibrium point of the evolved networks are extracted and their distribution illustrated in Figure 5.1. As can be seen, the DBEM model is generating an exponential strength distribution (i.e., a straight line in semilogarithmic plot) while PBEM model produces a power-law strength distribution (i.e., a straight line in log-log plot).

5.5 Network Properties

In this section, we introduce and analyze two important properties of weighted networks for each of the DBEM and PBEM models. First, we introduce the distance



(a) Exponential strength DF and CCDF of DBEM Model in Semi-Log Scale



(b) Power-law strength DF and CCDF of PBEM Model in Log-Log Scale

Figure 5.1: The DF and CCDF of strengths in *DBEM* model and *PBEM* models. It is clear that DBEM generates an exponential strength distribution, while PBEM produces a power-law strength distribution.

between individuals and study the distribution of Geodesic distance in networks, and second we analyze closeness centrality in networks evolving according to the proposed models.

Geodesic Distance

Geodesic distance is an important property in social networks (Freeman, 1978; Kretschmer, 2004; Leskovec et al., 2008). To measure the Geodesic distance, we first need to introduce a measure of distance between two connected individuals. This is defined as the inverse of link weights:

$$d_{ij} = \frac{1}{w_{ij}},$$

if $i \neq j$, $a_{ij} = 1$ and $d_{ii} = 0$ for every i. To illustrate, let w_{ij} denote the number of times individual i is co-observed with individual j. Then, the more these two

individuals are seen together the closer they are in the network (i.e., d_{ij} is smaller).

While, d_{ij} represents the distance between two individuals that are directly connected in the network, we can also define the Geodesic path between two individuals as the path with the minimum sum of distances. The length of a Geodesic path is called the Geodesic distance. In large scale networks, the average Geodesic distance is expected to be short compared to the number of nodes and the direct distances between individuals. To better understand this phenomenon, next, we calculate the Geodesic distance between two arbitrary individuals in a complete hierarchical network that is evolved under either DBEM or PBEM models.

Geodesic Distance in DBEM

Let d_{ij}^G be the Geodesic distance between individuals i and j. The following theorem states that in a complete hierarchical network evolved based on DBEM, the Geodesic path between individuals i and j is their direct connection and the Geodesic distance $d_{ij}^G = d_{ij}$.

Theorem 3. *In a complete hierarchical network evolved based on DBEM, the geodesic distance d_{ij}^s between the i^{th} and j^{th} individuals is equal to the distance associated with the connection between them:*

$$d_{ij}^G = d_{ij} = \frac{1}{w_{ij}}.$$

Proof. The proof of the above theorem can be attained by contradiction. Without loss of generality, assume $i > j$, and thus $d_{ij} = i - 1$ (see Equation 5.15). Suppose that the Geodesic path starts from the i^{th} individual and crosses a third individual k with $k \neq i, j$. The distance d_{ik} can be determined as:

$$d_{ik} = \begin{cases} i-1 & i > k \\ k-1 & k > i \end{cases} \tag{5.35}$$

We know that the Geodesic distance is equal to the sum of distances on the Geodesic paths. Therefore, $d_{ij}^G > d_{ik}$. Using Equation 5.35, it can be easily seen that $d_{ij}^G > i - 1 > j - 1$. Hence, the direct connection between two individuals has a shorter distance that the Geodesic distance. Thus, the supposition is false and the shortest path can not pass any third individual. This completes the proof of the above theorem. □

Geodesic Distance in PBEM

In contrast to DBEM, in which the Geodesic path between two individuals is the direct link connecting them, the following theorem shows that in PBEM, the Geodesic path always passes through the first individual in a complete hierarchical network:

Theorem 4. *In a complete hierarchical network evolved based on DBEM, the Geodesic distance d^s_{ij} between the i^{th} and j^{th} individuals, for $i \neq j$ is*

$$d^G_{ij} = d_{i1} + d_{j1}. \tag{5.36}$$

Before providing the proof of this theorem, we use Equation 5.26 to derive the distance between node i and j

$$d_{ij} = (2i-2)\prod_{k=1}^{i-j}\frac{2i-2k-1}{2i-2k}. \tag{5.37}$$

Proof. The proof follows again by contradiction. Without loss of generality we assume that $i > j$. Suppose there exists individuals i and j for which $d^G_{ij} = d_{ij}$, then:

$$d^G_{ij} = d_{ij} = (2i-2)\prod_{k=1}^{i-j}\frac{2i-2k-1}{2i-2k}, \tag{5.38}$$

thus,

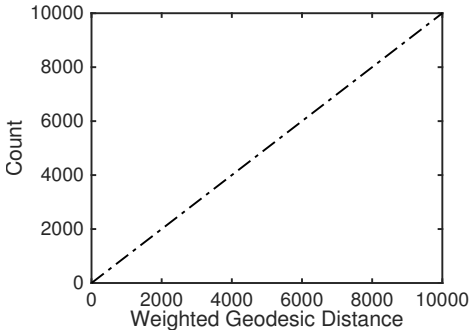
$$\begin{aligned} d^G_{ij} &= \prod_{k=i-j+1}^{i-1}\frac{2i-2k}{2i-2k-1}\cdot\prod_{k=1}^{i-1}\frac{2i-2k-1}{2i-2k} \\ &= \prod_{k=i-j+1}^{i-1}\frac{2i-2k}{2i-2k-1}\cdot d_{i1} \\ &= 2\prod_{k=i-j+1}^{i-2}\frac{2i-2k}{2i-2k-1}\cdot d_{i1} \\ &\geq 2d_{i1}. \end{aligned} \tag{5.39}$$

Hence:

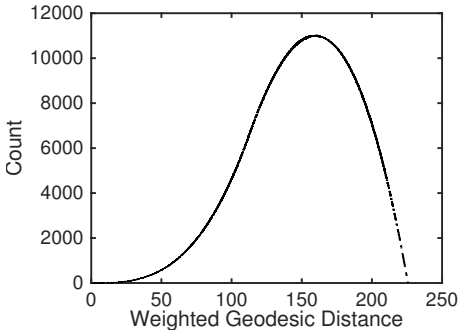
$$d^G_{ij} > d_{i1} + d_{j1}. \tag{5.40}$$

Therefore, every direct link between two individuals can be replaced via a path that passes through the first individual. Hence, the supposition is false completing the proof. □

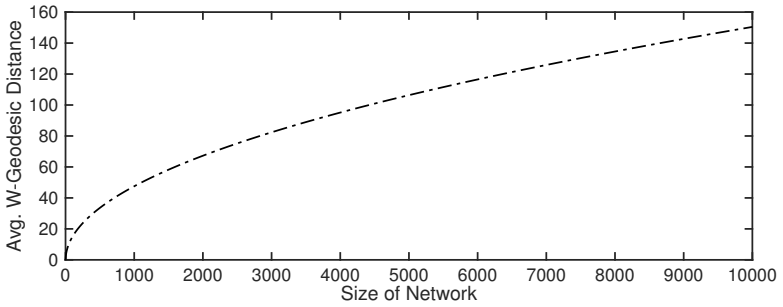
The distribution of Geodesic distances for individuals in DBEM and PBEM for a network of 10^4 nodes (as studied in Figure 5.1) is illustrated in Subfigures 5.3a and 5.3b. Subfigure 5.3c illustrates the changes in average of weighted Geodesic distances in networks of different sizes.



(a) DBEM Model



(b) PBEM Model



(c) PBEM Model

Figure 5.2: Weighted Geodesic Distance in DBEM- and PBEM-based complete networks. (a) in a DBEM-based evolved network, the Geodesic distance between individuals at the bottom of the hierarchy quickly increases; (b) in the PBEM-based evolved network the Geodesic distance has a distribution close to normal; (c) in the PBEM-based evolved networks the number of hops in the Geodesic path between individuals is always equal to 1 or 2, but the length of this path gradually increases by the increase in size of network.

Closeness Centrality

In this subsection, we study the closeness centrality of individuals in complete hierarchical networks. The closeness centrality of the i^{th} individual, c_i, is defined as the inverse of the sum of its Geodesic distance to other individuals:

$$c_i = \left[\sum_{j=1}^{N} d_{ij}^{G} \right]^{-1} . \tag{5.41}$$

Thus, the lower the total Geodesic distance of one individual from other nodes is, the more central the individual. Given the different distribution of Geodesic distances

produced by DBEM and PBEM models, we also expect to see different profiles in the centrality of nodes. Next, a detailed study of this measure for each of these networks is presented.

Closeness Centrality in DBEM

Using Theorem 3, the closeness centrality for individual i in a complete hierarchal network, evolved based on DBEM, is given as below.

$$\begin{aligned} c_i^{(\mathbb{D})} &= \left(\sum_{j=1}^{N} d_{ij}^{G}\right)^{-1} = \left(\sum_{j=1}^{N} d_{ij}\right)^{-1} \\ &= \left(\sum_{j=1}^{i-1} d_{ij} + \sum_{j=i+1}^{N} d_{ij}\right)^{-1} \\ &= \frac{2}{i^2-3i+(N^2-N+2)}. \end{aligned} \tag{5.42}$$

The above equation allows us to measure centrality of each individual in a DBEM network, in closed form.

Closeness Centrality in PBEM

In contrast to DBEM, in which the Geodesic path between two individuals is the direct connection between them, in Theorem 4, we saw that the Geodesic path in PBEM-based networks always passes through the first individual who is at the top of the hierarchy. Therefore, the Geodesic distance d_{ij}^{s} between two individuals is given by Equation 5.36. The closeness centrality for the i^{th} individual is then:

$$c_i^{(\mathbb{P})} = \left(\sum_{\substack{j=1 \\ j\neq i}}^{N} d_{ij}^{G}\right)^{-1} = \left(\sum_{\substack{j=1 \\ j\neq i}}^{N} \left(d_{i1} + d_{j1}\right)\right)^{-1} = \left((N-1)d_{i1} + \sum_{\substack{j=1 \\ \jmath\neq i}}^{N} d_{j1}\right)^{-1}. \tag{5.43}$$

By replacing d_{i1} and d_{j1} from Equation 5.37 into Equation 5.43, the closeness centrality for PBEM can be attained in closed form.

The closeness centrality of individuals in DBEM and PBEM for a network of 10^4 nodes is illustrated in Figure 5.3. This centrality measure is normalized in a way such that the maximum closeness becomes 1. As can be seen, in DBEM the individuals centrality decreases much slower compared to that in PBEM.

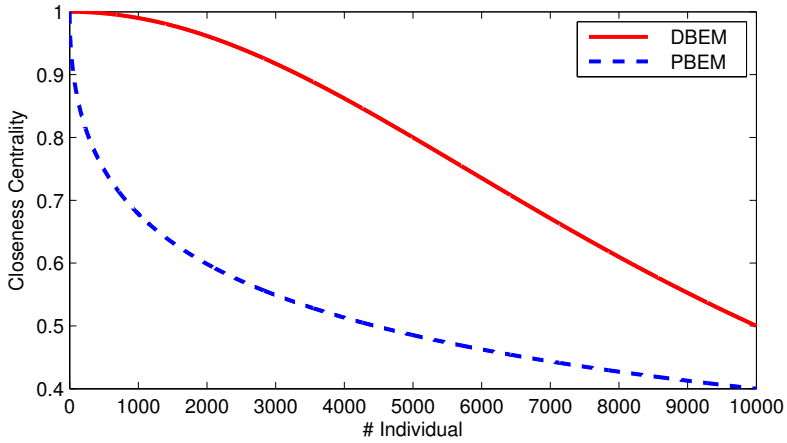


Figure 5.3: The normalized closeness centrality of *DBEM* and *PBEM* networks.

5.6 Real-World Verification

In this section, we use real-world interaction networks to validate the proposed DBEM and PBEM models. The datasets used for next experiments have been introduced in Section 2.4 of this thesis. In each of the introduced datasets, the adjacency and weight matrices are denoted by $\mathbb{A}^d = [a^d_{ij}]$ and $\mathbb{W}^d = [w^d_{ij}]$.

Experiment Methodology

Although, in all four interaction networks *Howler Monkey Groups*, *Kangaroo*, *Wolf Dominance* and the *US Airports* the interactions between every two nodes are available, except for the *Wolf Dominance* network, no hierarchy is explicitly given for the other three networks. The methodology used to compare the interaction networks with PBEM and DBEM are given as:

Extraction of hierarchy - In many real-world networks the interaction frequency/strength between individuals are reported, while the hierarchy (i.e., details of who initiates or dominates in the interaction) is not revealed. However, as shown in this chapter, extraction of hierarchies in the network play a crucial role in understanding the underlying mechanism of interactions.

The linear hierarchies and dominance orders in social networks are studied by many researchers e.g., by Kendall (1948); Appleby (1983); de Vries (1995, 1998); Shizuka and McDonald (2012); Sales-Pardo et al. (2007). In most of these studies, authors assume existence of data in form of frequencies of wins and loses of the same dyad member for each pair of individuals. Unfortunately, this is not the case for the

three networks under study in this chapter, *Howler Monkey Groups*, *Kangaroo* and the *US Airports*, and many other real-world networks.

Therefore, we rather use a simple yet efficient technique to extract the hierarchy of the network. Namely, we assume the nodes with more interactions are higher in the hierarchy. Therefore, we rank the nodes based on the sum of interaction frequencies exhibited by each node. Then, for every pair of nodes i and j that $a^d_{ij} = 1$ and rank of i is higher than j the tuple (j, i) is added to hierarchy set $\mathbb{H}$.

Evolving the models based on hierarchy set - Once the hierarchy set $\mathbb{H}$ is extracted from an interaction network, both proposed models, DBEM and PBEM, can be easily evolved using the dynamical system in (5.11) and (5.20), respectively. Each model results in a set of interaction weights and consequently node strengths.

Normalization of the interaction matrix - The dynamical models of DBEM and PBEM generate normalized weight matrices $\mathbb{W}$ where the sum of interaction weights between i and all its superordinate j is equal to 1. Therefore, we use the following rule to acquire a normalized weight matrix $\mathbb{W}^{d(n)} = [w^{d(n)}_{ij}]$:

$$\forall i, j : w^{d(n)}_{ij} = \frac{w^d_{ij}}{\sum_{j \in \{j|(i,j)\in\mathbb{H}\}} w^d_{ij}}. \tag{5.44}$$

Next subsection, presents the comparison of generated models by DBEM and PBEM to normalized real-world networks.

Results

To compare the estimations of DBEM and PBEM with data from real-world networks, we first compute the absolute strengths in each real-world network by using the normalized Weight matrix $\mathbb{W}^{d(n)}$. The absolute strength of each node is then estimated using the DBEM and PBEM models, based on the corresponding hierarchical structure of each real-world network.

We use Kolmogorov-Smirnov to test the equality of the distribution of the node strengths in real-world networks and the estimations of these strengths computed by the proposed models in this chapter. Table 5.1 provides the p-value of the Kolmogorov-Smirnov test for the real-world datasets. As can be seen, for the *Howler Monkey Groups* dataset, the p-value of DBEM estimation has a larger value compared to the PBEM estimations. For the other datasets, the p-values of PBEM estimations have larger values. Therefore, we assume that the network interactions in the first network are evolved based on only Dominance of individuals, while the other three networks follow a Prestige-based evolution model. In all four datasets, the Kolmogorov-Smirnov test accepts the null hypothesis that both sets are drawn from the same distribution at the 5% significance level.

Table 5.1: p-value of the Kolmogorov-Smirnov test for predictions made by DBEM and PBEM models. A large value of p supports the hypothesis that the distribution of estimated values is similar to the distribution of real-world values.

	Howler Monkey Groups	Kangaroos	Wolf Dominance	US Airports
DBEM	**0.93**	0.67	0.63	0.06
PBEM	0.67	**0.73**	**0.99**	**0.23**

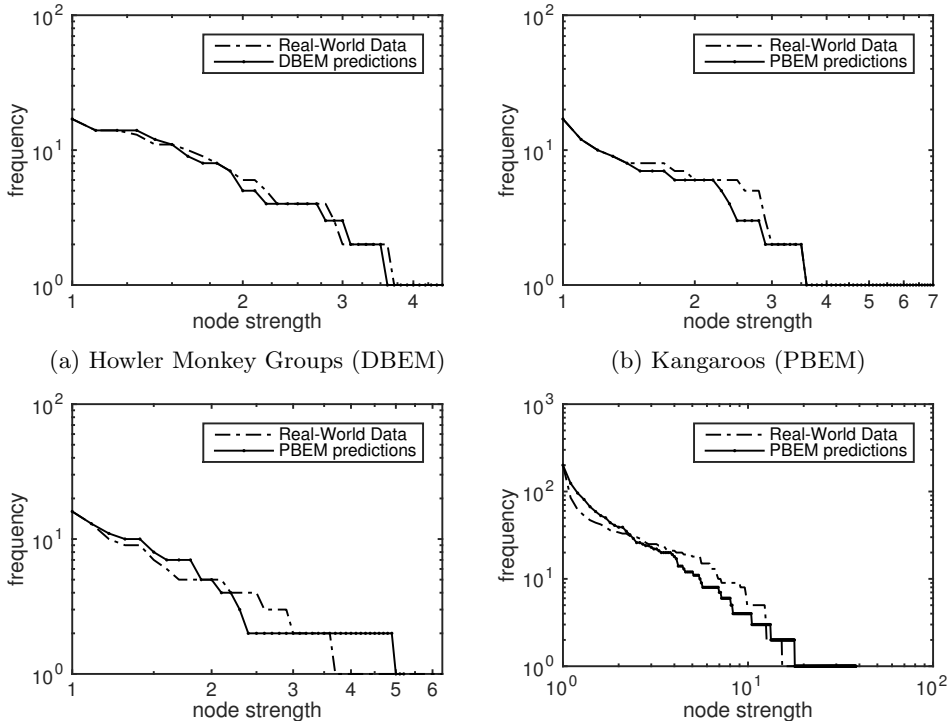
Figure 5.4 illustrates the CCDF of strengths in the real-world datasets. Estimations by DBEM, for the *Howler Monkey Groups* are shown in Figure 5.4a, and estimations by PBEM, for the other datasets are shown in Subfigures 5.4b-5.4d.

To measure the accuracy of estimations of DBEM and PBEM models, illustrated in Figure 5.4, we perform a statistical analysis of the absolute difference between estimated intensity of edges and their real intensity for each of the four real-world networks. The average estimation errors are 0.081 for *Howler Monkey Groups* estimated by DBEM (and 0.120 for its estimation with PBEM), 0.070 for *Kangaroos* estimated by PBEM (and 0.086 for its estimation with DBEM), 0.080 for *Wolf Dominance* estimated by PBEM (and 0.096 for its estimation with DBEM), and 0.027 for the *US Airports* estimated by PBEM (and 0.030 for its estimation with DBEM). The distribution of errors based on their minimum, first quartile, median, third quartile, and maximum are shown in Figure 5.5. The pairwise statistical comparisons between these four distributions show significant differences (p-value is less than 10^{-5} for all four comparisons, using the Kolmogorov-Smirnov test). Such significant difference can be explained by the difference in hierarchical structures of each real-world network.

Time Complexity

In this subsection, we study the time complexity of the proposed evolutionary models. Firstly, it should be considered that for the complete hierarchical networks that were studied in Sections 4-6 the properties of each network can be calculated in closed form. For instance, the expected global strength or closeness centrality of an individual i in a PBEM-based evolved network can be directly calculated by Equations 5.29 and 5.43. Such closed form expressions can be efficiently computed for any network with any size. For the general case of incomplete networks, however, the equilibrium of each model should be computed by evolving the dynamical model, introduced in (5.10), based on the underlying rules of either Equation (5.11), or Equation (5.11).

To perform a study reflecting the running times of the proposed models, we ran a variety of simulations. All simulations were run on an iOS with a 2.9 GHz Intel Core i7 processor and 8GB RAM, with MATLAB R2014b. The time steps used for running



(a) Howler Monkey Groups (DBEM)

(b) Kangaroos (PBEM)

(c) Wolf Dominance (PBEM)

(d) US Airports (PBEM)

Figure 5.4: Estimation of the strength CCDF in real-world networks by DBEM in (a) and PBEM in (b)-(d). While the prestige doesn't play an important role in the social network of monkeys (a), the co-observation of Kangaroos (b), the mocking battle among Wolves (c) and the traffic between top 200 US airports (d) is highly influenced by the prestige of each member in the network.

the discretized version of (5.10) were chosen to $\Delta t = 0.1$. The dynamical model was considered to be at equilibrium when the error condition $e(t) = ||\boldsymbol{w}||_2 < 0.01$ was satisfied.

In practice, it turns out that both DBEM and PBEM models have very close convergence rates. Therefore, we study the amount of time required for the PBEM model on a set of randomly generated small-world and scale-free networks. The size of networks vary from 20 nodes to 10,000 nodes and every network has an average degree of 4. For each network size we generate 50 networks, where the scale-free networks follow the preferential attachment model provided by Barabási and Albert (1999) and the small world networks follow the algorithm given by Watts and Strogatz (1998) with rewiring probability 0.1. The results are illustrated in Figure 5.6.

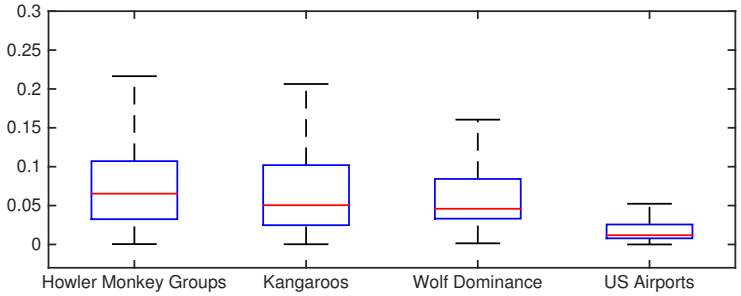


Figure 5.5: Statistical comparison of the absolute difference between estimated weight of edges and their real weight for four real-world networks. The median of errors are 0.065, 0.050, 0.046 and 0.012 for *Howler Monkey Groups*, *Kangaroos*, *Wolf Dominance* and the *US Airports* networks, respectively.

According to the results provided in Figure 5.6, the time complexity of PBEM model for a small-world network can be represented as $T(n) = \mathcal{O}(n^2)$. Although, the running time of this model for very large networks (e.g., 1,000,000 nodes) can be relatively high, in contrast to the stochastic models, this model requires just one run of the simulation to get to the final equilibrium and computations of all characteristics of the network. Also, the use of parallel processing can be beneficial in decreasing the running time for very large networks.

Network Assortativity

The assortativity property of networks measures the preference of network nodes to attach to other nodes that are similar in terms of degree or strength where the latter is applicable for weighted networks (Newman, 2002; Leung and Chau, 2007; Xie et al., 2007). As the models proposed in this work generate weighted networks, we use the *average nearest neighbor strength* measure for this purpose. Let $\Psi_{nn}(v_i)$ be the average strength of nearest neighbors of i^{th} node as

$$\Psi_{nn}(v_i) = \frac{1}{\boldsymbol{\Psi}(\boldsymbol{v}_i)} \sum_{1}^{n} w_{ij} \boldsymbol{\Psi}(\boldsymbol{v}_j).$$

This value can be averaged over classes of nodes with strength $\boldsymbol{\Psi}$ and be represented as $\Psi_{nn}(\boldsymbol{\Psi})$ that can provide a probe of correlation between strength of neighboring nodes. If $\Psi_{nn}(\boldsymbol{\Psi})$ is an increasing function of $\boldsymbol{\Psi}$, then nodes with similar strengths tend to establish ties with high intensity and otherwise nodes with dissimilar strengths tend to establish strong ties.

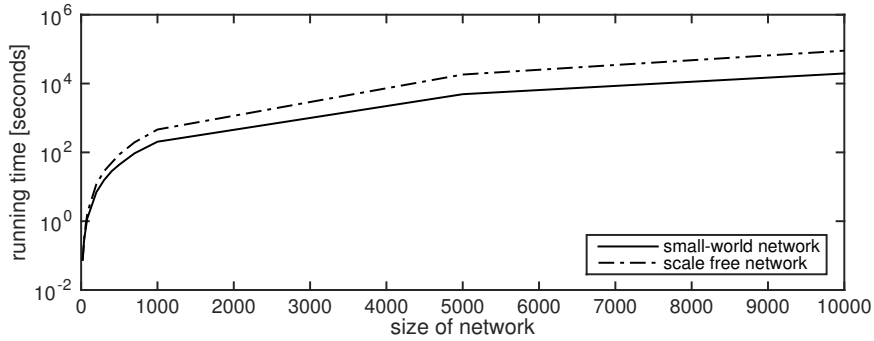


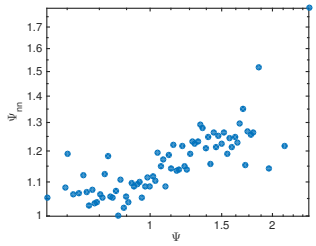
Figure 5.6: Time complexity of the PBEM model for networks of size 20 to 10,000 nodes in logarithmic scale for y-axis. The simulation of a network with 1000 nodes converges to equilibrium in 3.4min for a small-world network of average degree 4 and in 7.6min for a scale free network with average degree of 4.

Subfigures 5.7a-5.7d illustrate the average nearest neighbor strength for four different networks all with 1000 nodes and average degree 4. Subfigures 5.7a and 5.7b correspond to two networks evolved via DBEM model over hierarchical networks with small-world and scale free structures, respectively. As can be seen, in the small-world subfigure, DBEM shows an assortative behavior in which the nodes with high intensity have a higher average strength of nearest neighbor compared to the nodes with lower strength. In the scale free network, the Ψ_{nn} is a decreasing function for low degree nodes and an increasing function for high degree nodes. The assortativity of networks evolved based on PBEM model are shown in Subfigures 5.7a and 5.7b; the PBEM model shows assortative behavior (i.e., increasing Ψ_{nn}) in the small-world network and shows disassortative behavior (i.e., decreasing Ψ_{nn}) in the scale-free network.

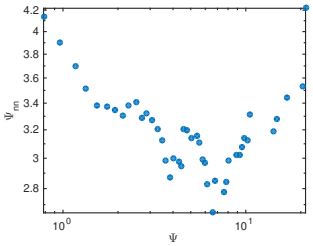
As can be seen in Subfigures 5.7a-5.7d, the assortativity of the networks evolved based on DBEM and PBEM models highly depends on the underlying structure of these networks.

5.7 Discussion

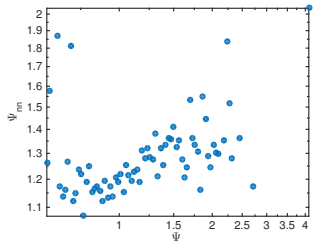
Distinguishing the role of *dominance* and *prestige* in evolution of social networks is a difficult task. To illustrate, consider the perspective of van Vugt and Tybur (2016) who believe that dominance is very common among non-human primates where members of the social group achieve priority through threat and intimidation. In contrast, they believe Prestige is more specific to humans and is granted to individuals because they help other individuals achieve their goals. In a different context, Ridely (Ri-



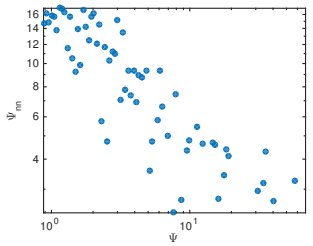
(a) Small-world network (DBEM)



(b) Scale free network (DBEM)



(c) Small-world network (PBEM)



(d) Scale free network (PBEM)

Figure 5.7: Assortativity in networks evolved based on DBEM and PBEM models. All subfigures are plotted in log-log scale for networks of 1000 nodes and average degree 4. While both models generate assortative behavior in an small-world network, the PBEM model generates disassortative behavior in scale free network and DBEM model generates a mixed behavior (i.e., first decreasing and then increasing Ψ_{nn}) in the scale free network.

dley, 1994, Chapter 5), refers to the behavior of hens in Lek and explains that "it hardly matters whether the male chosen is the best male; what counts is that he is the most fashionable". In other words, Ridely sees the high status (i.e., being fashionable) of some birds a more important criterion than their dominance in reaching more popularity.

The proposed two analytical models in this chapter allow us to mathematically distinguish the behaviors of Dominance-based and Prestige-based evolving networks. Although the models are simple, they illustrate how a minor change in evolution of the network can result in fundamental differences in the network's behavior. Theorems 1 and 2 illustrate a major difference in distribution of individuals' interaction intensities. Additionally, Theorems 3 and 4 analyze the Geodesic distance of individuals and reveal how in prestige-based evolving networks a central hub is formed where all shortest paths in the network pass this hub. In contrast such hubs are not seen in dominance-based evolving networks. By considering the beneficial role of hubs in complex networks (Newman, 2008; Guimera et al., 2005; van den Heuvel and Sporns,

2013), Theorems 3 and 4 can shed some light on evolutionary foundations in adoption of prestige-based strategies in some species.

Finally, the real-world validations not only verify the correct estimation of DBEM and PBEM models, but also introduce a new method to distinguish between dominance-based and prestige-based evolving networks. As shown in Subsection 5.6 the co-observation of monkeys in a group highly depends on the dominance of each individual, while interactions among a group of kangaroos and a group of wolves and traffic between US airports follows the prestige-based dynamic rules.

6

Evolution in Dynamic Social Networks

This chapter is based on:
B. Ranjbar-Sahraei, H. Bou-Ammar, D. Bloembergen, K. Tuyls, G. Weiss, "Evolution of Cooperation in Arbitrary Complex Network", In Proceedings of the 13th International Conference on Autonomous Agents and Multiagent Systems (AAMAS), Paris, France, 2014.

B. Ranjbar-Sahraei, H. Bou-Ammar, D. Bloembergen, K. Tuyls, G. Weiss, "Theory of Cooperation in Complex Social Networks", In Proceedings of the 28th AAAI Conference on Artificial Intelligence (AAAI), Qubec City, Canada, 2014.

This chapter presents a theoretical study on evolution of behaviors in complex social networks. First, we introduce the Continuous Action Iterated Prisoner's Dilemma (CAIPD) model which allows for representing the evolution of social behaviors in state space. Then, using CAIPD, we study the convergence to network-wide agreement and provide proofs for both evolutionary networks with fixed interaction dynamics, as well as, for evolutionary networks with a feedback from interactions to behaviors. Moreover, an extension to the CAIPD model is proposed that allows to model influence on the evolution of behaviors in social networks. As such, this chapter contributes to a better understanding of behavioral change on social networks, and provides a first step towards their active control.

6.1 Preliminaries

This work makes use of lemmas that are next briefly discussed. Firstly, Stochastic Indecomposable and Aperiodic (SIA) matrices, and $\lambda(\cdot)$ functions are introduced.

Definition 10 (SIA Matrices)**.** *A matrix $\boldsymbol{P}$ with all positive elements is stochastic if all its row sums are +1. $\boldsymbol{P}$ is called SIA if $\lim_{n\to\infty} \boldsymbol{P}^n = \mathbf{1}\nu^{\mathsf{T}}$, where ν is a column vector.*

Definition 11 ($\lambda(\cdot)$ Function)**.** *For a square matrix $\boldsymbol{S}$, λ is defined as*

$$\lambda(\boldsymbol{S}) = 1 - \min_{i_1,i_2} \left\{ \sum_j \min\left(\boldsymbol{S}_{i_1 j}, \boldsymbol{S}_{i_2 j}\right) \right\}.$$

Having introduced the above, three Lemmas needed for the proofs provided in this chapter are presented:

Lemma 8. *(Wolfowitz, 1963) Let $\boldsymbol{M} = \{\boldsymbol{M}_1, \boldsymbol{M}_2, \dots\}$ be an infinite set of SIA matrices, where for each finite positive length sequence of $\boldsymbol{M}_{i_1}, \boldsymbol{M}_{i_2}, \dots, \boldsymbol{M}_{i_j}$, the matrix product $\boldsymbol{M}_{i_j}\boldsymbol{M}_{i_{j-1}}\dots\boldsymbol{M}_{i_1}$ is SIA. If for every $\boldsymbol{W} = \boldsymbol{M}_{k_1}\boldsymbol{M}_{k_2}\dots\boldsymbol{M}_{N_t+1}$, where N_t is the number of different types of all SIA matrices of appropriate sizes, there exists a constant $0 \leq d < 1$ satisfying $\lambda(\boldsymbol{W}) \leq d$, then for each infinite sequence of $\boldsymbol{M}_{i_1}, \boldsymbol{M}_{i_2}, \dots, \boldsymbol{M}_{i_j}, j \to \infty$ there exists a column vector $\boldsymbol{\nu}$ such that $\lim_{j\to\infty} \boldsymbol{M}_{i_j}\boldsymbol{M}_{i_{j-1}}\dots\boldsymbol{M}_{i_1} = \mathbf{1}\boldsymbol{\nu}^{\mathsf{T}}$.*

Lemma 9. *(Ren et al., 2005b) For $\boldsymbol{\mathcal{L}}$ as a constant Laplacian matrix associated with a strongly connected network, the matrix $e^{-\boldsymbol{\mathcal{L}}t}, \forall t > 0$ is a stochastic matrix with positive diagonal values that $\lim_{t\to\infty} e^{-\boldsymbol{\mathcal{L}}t} = \mathbf{1}\nu^{\mathsf{T}}$ where $\boldsymbol{\nu} = [\nu_1, \nu_2, \dots, \nu_n]^{\mathsf{T}} \geq 0$ and $\sum_{i=1}^{N} \nu_i = 1$.*

Lemma 10. *(Jadbabaie et al., 2003) Let $\boldsymbol{M}_1, \boldsymbol{M}_2, \dots, \boldsymbol{M}_m$ be a finite set of non-negative matrices. Then $\boldsymbol{M}_1\boldsymbol{M}_2\dots\boldsymbol{M}_m \geq \delta(\boldsymbol{M}_1 + \boldsymbol{M}_2 + \dots + \boldsymbol{M}_m)$ for a δ which can be specified from matrices $\boldsymbol{M}_i, i \in \{1, 2, \dots, m\}$.*

6.2 Dynamical Modeling

This section details the continuous dynamical model proposed in this chapter. Firstly, the model of a 2-player Continuous Action Iterated Prisoner's Dilemma (CAIPD) is derived. This model is then generalized to the N-player case.

2-Player CAIPD

In the 2-player continuous action iterated prisoner's dilemma (CAIPD) each player can choose a level of cooperation from a continuous set of strategies.

Let $x_i \in [0, 1]$ denote the strategy of the i^{th} player with $i \in \{1, 2\}$ representing each player. Here, $x_i = 0$ corresponds to full defection, while $x_i = 1$ represents full

cooperation. A player pays a cost cx_i while the opponent receives a benefit bx_i, with $b > c$. It is clear that a defector (i.e., $x_i = 0$) pays no cost and distributes no benefits. The fitness of player i, $F(x_i)$, can be thus defined as:

$$F(x_i) = -cx_i + bx_j \tag{6.1}$$

Using (6.1), the difference between the fitnesses of two players can be derived as

$$\begin{aligned} \Delta F_{ji} &= F(x_j) - F(x_i) \\ &= -c(x_j - x_i) - b(x_j - x_i) \\ &= (-c - b)(x_j - x_i) \end{aligned}$$

Following the imitation dynamics described by Hauert and Szabó (2005), where each player switches to a neighboring strategy with a certain probability, the following strategy evolution law is introduced:

$$x_i(k+1) = (1 - p_{ij})x_i(k) + p_{ij}x_j(k), \tag{6.2}$$

where k represents the iteration number and $p_{ij} = \text{sig}(\beta \Delta F_{ji})$, with $\text{sig}(\beta \Delta F_{ji}) = 1/(1 + \exp(-\beta \Delta F_{ji}))$ and $\beta > 0$. In words, Equation 6.2 states that in iteration k a player switches to a neighboring strategy with a probability p_{ij}.

The change $\Delta x_i(k) = x_i(k+1) - x_i(k)$ in strategies between two iterations $k+1$ and k can be rewritten as:

$$\begin{aligned} \Delta x_i(k) &= \Big(x_i(k+1) - x_i(k)\Big)\Delta t \\ &= p_{ij}\Big(x_j(k) - x_i(k)\Big)\Delta t \end{aligned}$$

for step size $\Delta t = 1$. The strategy adaptation law of player i can also be written in continuous form as

$$\dot{x}_i(t) = p_{ij}(x_j(t) - x_i(t)), \tag{6.3}$$

In essence, the adaptation law of Equation 6.3 shows that for high values of p_{ij}, the i^{th} player switches its strategy to the opponent's strategy, while for low values of p_{ij} it keeps its own strategy.

N-Player CAIPD

Having introduced the 2-player CAIPD, this subsection details the more general N-player case. The N-player CAIPD is defined for a group of N players on a weighted

graph $\mathbb{G} = (\mathcal{V}, \mathcal{W})$ where $\mathcal{V} = \{v_1, v_2, \ldots, v_N\}$ represents the set of nodes (i.e., each player is represented by a node), and $\mathcal{W} = [w_{ij}]$ denotes the symmetric weighted adjacency matrix, where $w_{ij} \in \{0, 1\}$ is a binary variable describing the connection between players i and j, $\forall (i, j) \in \{1, 2, \ldots, N\} \times \{1, 2, \ldots, N\}$. Further, w_{ii} is assumed to be zero for all $i \in \{1, 2, \ldots, N\}$.

Let $x_i \in [0, 1]$ denote the cooperation level of node v_i. A network with value $\mathbf{x}$ and topology $\mathbb{G}$ is defined as $\mathbb{G}_{\mathbf{x}} = (\mathbb{G}, \mathbf{x})$ with $\mathbf{x} = [x_1, x_2, \ldots, x_N]^T$. Supposing that each node of the network $\mathbb{G}_{\mathbf{x}}$ is a dynamic player with

$$\dot{x}_i \quad = \quad h_i(\mathbf{x}), \tag{6.4}$$

the network $\mathbb{G}_{\mathbf{x}}$ can be regarded as a dynamical system in which the value $\mathbf{x}$ evolves according to the network dynamics $\dot{\mathbf{x}} = \mathbf{H}(\mathbf{x})$.

Having a general form for the network dynamics, the next step is to determine $h_i(\cdot)$ in Equation 6.4 for all $i \in \{1, 2, \ldots, N\}$. To determine $h_i(\cdot)$ in its full form, firstly, the i^{th} node/player fitness needs to be derived. Generalizing the 2-player CAIPD, the following can be computed:

$$F(x_i) \quad = \quad -\deg(v_i) c x_i + b \sum_{j=1}^{N} w_{ij} x_j \tag{6.5}$$

where $\deg(v_i)$ is the degree of node v_i. In Equation 6.5, the i^{th} player pays a cost of cx_i for each of its neighbors j (i.e., $-\deg(v_i)cx_i$) and receives a benefit of bx_j for all its neighbors j (i.e., $b\sum_{j=1}^{N} w_{ij}x_j$, with $w_{ij} \in 0, 1$ indicating whether i and j are connected). Therefore, the difference between the i^{th} and j^{th} player fitnesses can be written as:

$$\begin{aligned} \Delta F_{ji} &= F(x_j) - F(x_i) \\ &= c\left(\deg(v_i) x_i - \deg(v_j) x_j \right) + b\left(\sum_{k=1}^{N} (w_{jk} - w_{ik}) x_k \right) \end{aligned}$$

Given that the probability of strategy adaptation $p_{ij} = \mathrm{sig}(\beta \Delta F_{ji})$, similar to (6.2) the evolution law for i^{th} player in the network can be derived as

$$x_i(k+1) = \frac{1}{\deg(v_i)} \left[\sum_{j=1}^{N} (1 - p_{ij} w_{ij}) x_i(k) + p_{ij} w_{ij} x_j(k) \right] \tag{6.6}$$

The difference equations in (6.6) can again be converted to differential equations,

which leads to

$$\dot{x}_i(t) \quad = \quad \frac{1}{\deg(v_i)}\left[\sum_{j=1}^{N} p_{ij} w_{ij}\Big(x_j(t) - x_i(t)\Big)\right]$$

This dynamical system can be re-written in a standard form by introducing the Laplacian of $\mathbb{G}$, $\mathcal{L}(\cdot)$ as

$$\dot{\boldsymbol{x}}(t) = -\mathcal{L}\left[\boldsymbol{x}(t)\right]\boldsymbol{x}(t), \tag{6.7}$$

where

$$\boldsymbol{\mathcal{L}}_{ij} = \begin{cases} -\,p_{ij}/\deg[v_i] & \text{if } i \neq j \\ \sum_{j=1}^{N} p_{ij}/\deg[v_i] & \text{if } i = j \end{cases}$$

Next section elaborates on complexities in analysis of Equation 6.7.

6.3 Why are Social Networks Complex?

Three sources of complexity arise in the mathematical analysis of CAIPD. Firstly, $\boldsymbol{\mathcal{L}}(.)$ is time varying due to its nonlinear dependence on the state-variable $\boldsymbol{x}(t)$. Secondly, the system is highly state coupled, where many off-diagonal entries $\boldsymbol{\mathcal{L}}_{ij}, i \neq j$ can take on arbitrary non-zero values. Finally, the analysis of Equation 6.7 resides in high dimensional spaces, rendering intuitive predictions difficult.

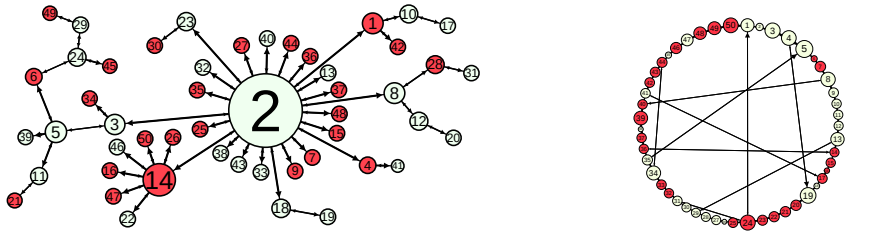
These complexities are relaxed by considering the structure of $\boldsymbol{\mathcal{L}}$ as a key for analysis, while relaxing its changes with respect to time. The main goal is then to determine for which network topologies convergence to an agreement, $\boldsymbol{x} \to x^{\star}\mathbf{1}$ as $t \to \infty$ with $\mathbf{1} = [1, 1, \ldots, 1] \in \mathbb{R}^N$, occurs.

To ensure the existence of an agreement, this chapter uses the strong connectivity of social networks in which the graph $\mathbb{G}$ associated with $\mathbb{G}_{\boldsymbol{x}}$ has directed paths from any $v_i \in \mathcal{V}$ to $v_j \in \mathcal{V}$.

The analysis performed in this chapter deals with two distinct scenarios with respect to the time varying nature of $\boldsymbol{\mathcal{L}}$ in the dynamical model of Equation 6.7. Firstly, $\boldsymbol{\mathcal{L}}$ matrix is assumed to be fixed which refers to the situations that individuals' don't update their beliefs on each other's fitness. We refer to this as an *evolutionary network* where just the strategies evolve in time.

Although such model pose a simplification of the original problem, its analysis can shed light on the dynamical behavior of the original system, where some of the attained results can be directly extended to the original problem.

Secondly, the theoretical analysis is extended to the more general case of time varying Laplacian matrix, where both strategies and fitnesses evolve in time and a



(a) Scale free network (b) Small world network

Figure 6.1: Sample networks; nodes with green and red colors represent cooperators and defectors respectively. The size of the nodes represents their initial fitness value, while the size of the arrow heads are computed based on the corresponding elements of $\boldsymbol{\mathcal{L}}$ matrix at the initial step, indicating the strength of influence between each set of nodes.

feedback from interaction to fitnesses exists. We refer to this as an *evolutionary network with feedback loop*. Empirical results confirming the provided theorems are also presented.

In what comes next, experiments are performed on two sample networks, shown in Figure 6.1, each consisting of $N = 50$ individuals. The first network is scale-free and follows Barabási-Albert model (Barabási and Albert, 1999) with an average degree of two. The other network has small world property and follows Watts-Strogatz model (Watts and Strogatz, 1998) and has average degree 4 and rewiring probability 0.1. Both networks are initialized with 25 pure cooperators and 25 pure defectors. The benefits that cooperators share and the costs of cooperation are $b = 4$ and $c = 1$, respectively. The sigmoid function used to calculate the strategy adoption strength uses $\beta = 1$.

Evaluation of the proposed CAIPD model on larger networks with various sizes up to 1000 nodes, and also comparison of CAIPD with an existing model based on Game Theory are provided by Ranjbar-Sahraei et al. (2014b).

6.4 Analysis of Evolutionary Networks

This section is dedicated to the analysis of evolutionary networks. In such types, the Laplacian matrix $\mathcal{L}$ is time-invariant. Therefore, CAIPD, in Equation 6.7, can be written in the general closed-loop control system form:

$$\begin{cases} \dot{\boldsymbol{x}} &= \boldsymbol{A}\boldsymbol{x} + \boldsymbol{B}\boldsymbol{u} \\ \boldsymbol{y} &= \boldsymbol{C}\boldsymbol{x} + \boldsymbol{D}\boldsymbol{u}, \end{cases} \tag{6.8}$$

by setting $\boldsymbol{A} = \boldsymbol{0}$, $\boldsymbol{B} = -\boldsymbol{I}_N$, $\boldsymbol{C} = \boldsymbol{I}_N$ and $\boldsymbol{D} = \boldsymbol{0}$ with a feedback law of: $\boldsymbol{u} = \boldsymbol{\mathcal{L}}\boldsymbol{y}$.

Agreement in Evolutionary Networks

Convergence to the same value (i.e., $x_i(t) \to x_j(t),\ \forall(i,j) = 1,\ldots,N$ as $t \to \infty$) is first proved for evolutionary networks of form (6.8). An interpretation of the exact value of this agreement as a weighted average over all initial states is also derived. Further, it is shown that this value can be determined using the left eigenvector of the zero eigenvalue of the Laplacian matrix.

Theorem 5 (Evolutionary Agreement)**.** *For system* (6.8) *with* $\boldsymbol{A} = \boldsymbol{0}$, $\boldsymbol{B} = -\boldsymbol{I}_N$, $\boldsymbol{C} = \boldsymbol{I}_N$, $\boldsymbol{D} = \boldsymbol{0}$ *and* $\boldsymbol{u} = \boldsymbol{\mathcal{L}y}$, *every individual's state* x_i, $i = 1,2,\ldots,N$ *converges to an agreement of the form* $x_i(t) \to \boldsymbol{r}^T\boldsymbol{x}(0)$, $t \to \infty$, *where* $\boldsymbol{r}$ *is the trivial left eigenvector of* $\boldsymbol{\mathcal{L}}$ *(i.e., the eigenvector associated with the zero eigenvalue).*

Proof. As shown in (Olfati-Saber and Murray, 2004, Theorems 1 and 2), a Laplacian matrix of a strongly connected digraph with N nodes, has $N-1$ eigenvalues with *positive real parts* and a *singular trivial eigenvalue* $\lambda_0 = 0$. The trivial right eigenvector of the Laplacian matrix is $\boldsymbol{1} = [1,1,\ldots,1]^T \in \mathbb{R}^N$ and the trivial left eigenvector $\boldsymbol{r} = [r_1, r_2, \ldots, r_n]^\mathsf{T} \in \mathbb{R}^N$, where $\boldsymbol{r}^\mathsf{T}\boldsymbol{1} = 1$. First, $\boldsymbol{\mathcal{L}}$ is mapped to the Jordan normal form as: $\mathcal{J} = \boldsymbol{T}_l\boldsymbol{\mathcal{L}}\boldsymbol{T}_r$, with $\mathcal{J}$ being an upper triangular matrix having $\mathcal{J}_{11} = 0$, and $\mathcal{J}_{ii} = \lambda_j$ for $j = 1,2,\ldots,N-1$ and $i = 2,3,\ldots,N$. Further, $\boldsymbol{T}_l \in \mathbb{R}^{N\times N}$ contains the transpose of left eigenvectors of $\boldsymbol{\mathcal{L}}$ with $\boldsymbol{r}^\mathsf{T}$ in the first row, and $\boldsymbol{T}_r \in \mathbb{R}^{N\times N}$ incorporates all right eigenvectors of $\boldsymbol{\mathcal{L}}$ with $\boldsymbol{1}$ in the first column. Moreover, $\boldsymbol{T}_l\boldsymbol{T}_r = \boldsymbol{T}_r\boldsymbol{T}_l = \boldsymbol{I}_N$. Next, consider the state transformation $\tilde{\boldsymbol{x}} = \boldsymbol{T}_l\boldsymbol{x}$, with $\tilde{\boldsymbol{x}} = [\tilde{x}_1, \tilde{x}_2, \ldots, \tilde{x}_N]^\mathsf{T}$. The system in Equation 6.8 can be represented in terms of $\tilde{\boldsymbol{x}}$ as:

$$\dot{\tilde{\boldsymbol{x}}} = \boldsymbol{T}_l\boldsymbol{B}\boldsymbol{u} = -\boldsymbol{T}_l\boldsymbol{\mathcal{L}}\boldsymbol{T}_r\boldsymbol{T}_l\boldsymbol{x} = -\mathcal{J}\tilde{\boldsymbol{x}} \tag{6.9}$$

The solution of the above system can be computed using $\tilde{\boldsymbol{x}}(t) = e^{-\mathcal{J}t}\tilde{\boldsymbol{x}}(0)$. It can be easily shown that $\tilde{\boldsymbol{x}}(t) \to [\tilde{x}_1(0), 0, 0\ldots, 0]^\mathsf{T}$ as $t \to \infty$, with $\tilde{x}_1 = \boldsymbol{r}^\mathsf{T}\boldsymbol{x}(0)$. Using the state transformation $\boldsymbol{x} = \boldsymbol{T}_r\tilde{\boldsymbol{x}}$ it can be seen that $\boldsymbol{x}(t) \to \boldsymbol{T}_r\,[\tilde{x}_1(0)\ 0\ 0\ldots\ 0]^\mathsf{T} = \boldsymbol{r}^\mathsf{T}\boldsymbol{x}(0)$, thus concluding the proof. □

Theorem 5 shows that all individuals in an evolutionary network eventually agree on the scalar state variable $\boldsymbol{r}^\mathsf{T}\boldsymbol{x}(0)$. Therefore, the final agreement is a weighted average of the initial states:

$$x_i(t) \to r_1x_1(0) + r_2x_2(0) + \ldots r_Nx_N(0) \tag{6.10}$$

for every i, with $\sum_{i=1}^{N} r_i = 1$.

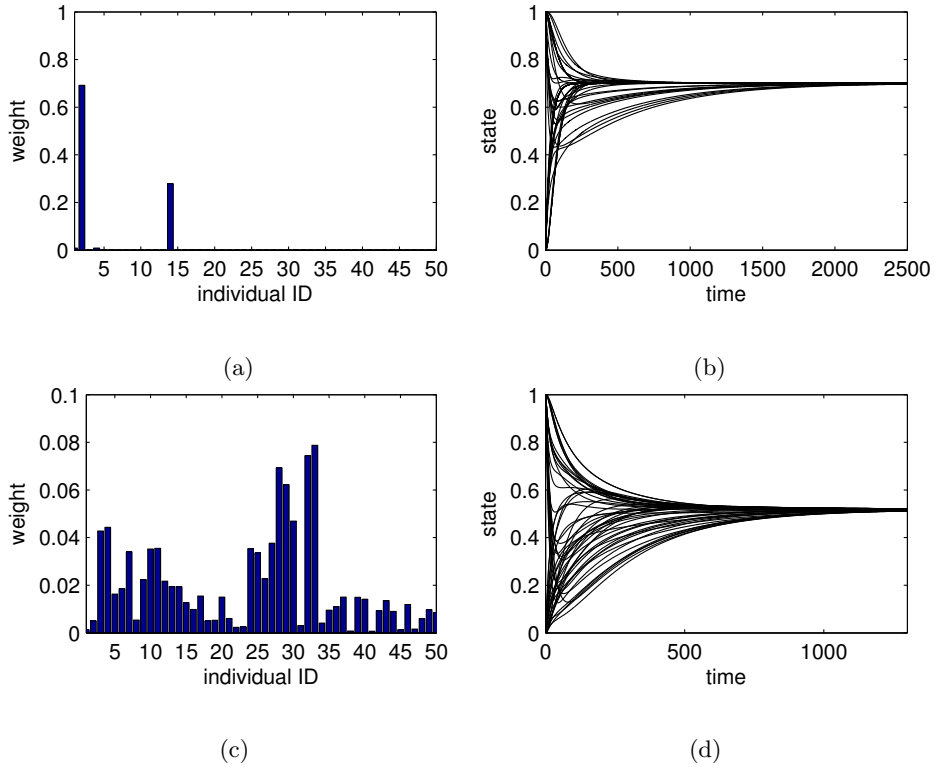


Figure 6.2: Agreement analysis for evolutionary networks (a)-(b) scale free network (c)-(d) small world network.

Experimental Validation:

According to the above theorem, pinpointing the individual(s) with the highest weights in (6.10) can help in approximating the final agreement value of the network. For instance, considering the network in Subfigure 6.1a, the elements of the trivial left eigenvector associated with this network can be computed as $r_1 = 0.01, r_2 = 0.68, r_4 = 0.01, r_{14} = 0.30$, and $r_i \approx 0$ for every $i \neq 1, 2, 4, 14$. These are illustrated in Subfigure 6.2a. Using the derived result of Equation 6.10, the final agreement is expected to be reached at $x^\star = 0.68$. Clearly, this is verified by Subfigure 6.2b presenting the evolution of state trajectories. Subfigures 6.2c and 6.2d again demonstrate these results in small world networks. The differences between the weight distributions are due to the fact that scale free networks possess a power-law degree distribution.

Based on the above results, next CAIPD is extended to allow for influencing the behavior of an evolutionary social network by incorporating modification to its action matrix model. Theoretical analysis of the newly acquired model as well as empirical validation are then provided.

Agreement with Multi-rate State Updates

To allow weighted averaged manipulations, the action matrix $\boldsymbol{B}$ in Equation 6.8 is modified to $\boldsymbol{B}_m = \text{diag}(b_{m,i})\ \forall i \in \{1, 2, \ldots, N\}$, where $-1 \leq b_{m,i} < 0,\ \forall i \in 1, 2, \ldots, N$.

At this stage, $\boldsymbol{B}_m$ can be regarded as a multi-rate input matrix since as the absolute values of $b_{m,i}$ increase so does the variational speed of x_i, and vice versa. The following theorem, studies the agreement behavior of system (6.8) while using the multi-rate input matrix $\boldsymbol{B}_m$. It shows that all individuals eventually agree, even in the presence of such a multi-rate state update. It further demonstrates that individuals with lower update rates contribute more to the final agreement compared to those with a high update rate.

Theorem 6 (Evolutionary Multirate Agreement)**.** *For system* (6.8) *with* $\boldsymbol{A} = \boldsymbol{0}$*, input matrix* $\boldsymbol{B}_m = diag(b_{m,i})$*, where* $-1 \leq b_{m,i} < 0,\ \forall i \in 1, 2, \ldots, N$*,* $\boldsymbol{C} = \boldsymbol{I}_N$*,* $\boldsymbol{D} = \boldsymbol{0}$ *and input vector* $\boldsymbol{u} = \boldsymbol{\mathcal{L}}\boldsymbol{y}$*, each individual's state* x_i*, for* $i = 1, 2, \ldots, N$ *converges to an agreement:*

$$x_i(t) \to \frac{1}{||\boldsymbol{r}^\top \boldsymbol{B}_m^{-1}||_1} \boldsymbol{r}^\top \boldsymbol{B}_m^{-1} \boldsymbol{x}(0),\ \ as\ t \to \infty \tag{6.11}$$

where $\boldsymbol{r}$ *is the trivial left eigenvector of* $\boldsymbol{\mathcal{L}}$ *and* $||\cdot||_1$ *denotes the* L_1 *norm.*

Proof. It can be easily shown that for any negative matrix $\boldsymbol{B}_m$, the $\boldsymbol{\mathcal{L}}_m = \boldsymbol{B}_m \boldsymbol{\mathcal{L}}$ still has one singular trivial eigenvalue associated, and its trivial right eigenvector will be $\boldsymbol{1}$. Furthermore, after some manipulations it can be seen that the normalized trivial left eigenvector of $\boldsymbol{\mathcal{L}}_m$ is $\boldsymbol{r}_m = \frac{1}{||\boldsymbol{r}^\top \boldsymbol{B}_m^{-1}||_1} \boldsymbol{r}^\top \boldsymbol{B}_m^{-1}$. Please note that the non-trivial eigenvalues and eigenvectors might have changed from the original system. Following a similar procedure to that of Theorem 5, it becomes clear that as $t \to \infty$, the state of the i^{th} individual converges to: $x_i \to \frac{1}{||\boldsymbol{r}^\top \boldsymbol{B}_m^{-1}||_1} \boldsymbol{r}^\top \boldsymbol{B}_m^{-1} \boldsymbol{x}(0)$, thus concluding the proof. □

According to the above theorem, the final agreement of evolutionary networks with multi-rate action matrices can still be seen as a weighted average of the initial states: $x^\star = \left(\frac{r_1}{\alpha b_{m,1}} x_1(0) + \frac{r_2}{\alpha b_{m,2}} x_2(0) + \cdots + \frac{r_N}{\alpha b_{m,N}} x_N(0) \right)$, where $\alpha = ||\boldsymbol{r}^\top \boldsymbol{B}_m^{-1}||_1$.

Such a result, represents an important characteristic of the evolution of cooperation in social networks. Namely, the smaller the update rate $b_{m,i}$ of an arbitrary individual, the more its state will contribute to the final agreement value. Such a conclusion can be used to *control* the evolutionary behavior of these networks. For instance, if internally or externally one can decrease the update rate of an individual (or a group of individuals), consequently the state value of that individual (or group)

will play a more prominent role in the overall group's agreement. Next, we make use of the above results to study the extreme case in which a network has to be driven to a stationary stable reference state. Firstly, the following action matrix: $\boldsymbol{B}_r = \text{diag}(b_{r,i})$ for $i = 1, 2, \ldots, N$ is introduced, with:

$$b_{r,i} = \begin{cases} -1 & \text{if } i \neq \text{ref.} \\ 0 & \text{if } i = \text{ref.} \end{cases}, \; i = 1, 2, \ldots, N \tag{6.12}$$

with "ref." being the number of the reference individual. The following theorem, shows that eventually all individual states will converge to $x_{\text{ref.}}$.

Theorem 7 (Evolutionary State-Reference Agreement)**.** *For system* (6.8) *with* $\boldsymbol{A} = \boldsymbol{0}$, $\boldsymbol{B} = \boldsymbol{B}_r$ *as in* (6.12), $\boldsymbol{C} = \boldsymbol{I}_N$, $\boldsymbol{D} = \boldsymbol{0}$ *and* $\boldsymbol{u} = \boldsymbol{\mathcal{L}y}$ *every individual's state* $x_i, i = 1, 2, \ldots, N$ *converges to an agreement* $x_i(t) \to x_{ref.}$ *as* $t \to \infty$.

Proof. Here, the network is not strongly connected since the associated digraph to $\boldsymbol{\mathcal{L}}_r = \boldsymbol{B}_r\boldsymbol{\mathcal{L}}$ is not strongly connected. However, one directed spanning tree containing all the nodes of the graph with $v_{\text{ref.}}$ as its root, exists. In (Ren and Beard, 2005, Corollary 1), it is shown that presence of such a spanning tree is enough for the Laplacian matrix to have a singular trivial eigenvalue and positive real parts for the other eigenvalues. Furthermore, it can be easily seen that the trivial right eigenvector of $\boldsymbol{\mathcal{L}}_r$ is $\boldsymbol{1}$ and that the trivial left eigenvector of $\boldsymbol{\mathcal{L}}_r$ is $\boldsymbol{r}_r = [r_{r,1}, r_{r,2}, \ldots, r_{r,N}]^\mathsf{T}$ with:

$$r_{r,i} = \begin{cases} 1 & \text{if } i = \text{ref.} \\ 0 & \text{if } i \neq \text{ref.} \end{cases}, \; i = 1, 2, \ldots, N$$

Following a procedure similar to the proof of Theorem 5, it can be shown that as $t \to \infty$, the i^{th} individual state $x_i(t) \to x_{\text{ref.}}(t)$. □

Intuitively, the above results show that as an individual, i, insists on retaining its state and refuses to switch its initial value, eventually all others will arrive at an agreement with that i^{th} individual.

Experimental Validation:

Theorems 6 and 7 are empirically demonstrated in two control experiments performed on the sample networks of Figure 6.1. In the first, the update rate of a cooperator is decreased. According to Theorem 6, as the cooperator's update rate decreases, its contribution to the final agreement increases. This fact is illustrated in Subfigures 6.3a-6.3d. In the second experiment, Theorem 7 was tested, where one defector was chosen as the reference individual. Results illustrated in Subfigures 6.3e and 6.3f

confirm the conclusions of Theorem 7 by showing that all individuals eventually converge to pure defection.

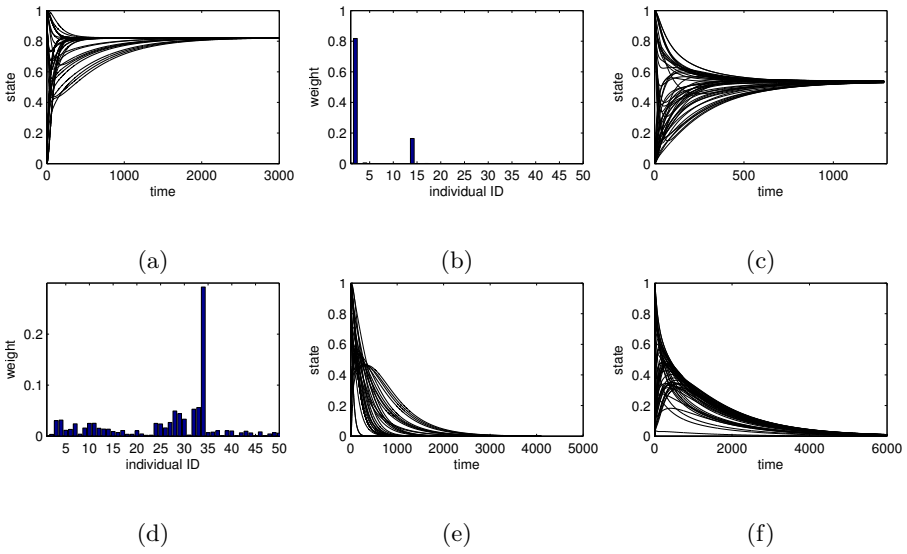


Figure 6.3: (a)-(b) scale-free network with multirate update $b_{m,2} = -0.5$ and $b_{m,i} = -1$ for every $i \neq 2$ (c)-(d) small world network with multirate update $b_{m,34} = -0.1$ and $b_{m,i} = -1$ for every $i \neq 34$ (e) scale-free network with state-reference agreement ref. $= 14$ (f) small world network with state-reference agreement ref. $= 50$.

6.5 Agreement in Networks with Feedback Loop

In what comes next, a generalization of the previous analysis is considered. Precisely, the evolutionary case with a varying Laplacian is studied.

Firstly, the concept of a dwell time τ Jadbabaie et al. (2003) is used to re-write the dynamics of CAIPD as:

$$\dot{\boldsymbol{x}}(t) = -\mathcal{L}_k \boldsymbol{x}(t), \tag{6.13}$$

where $\mathcal{L}_k = \mathcal{L}\left[\boldsymbol{x}(k\tau)\right]$ for $k\tau < t < (k+1)\tau$ and $k = 1, 2, \ldots$. Clearly, as $\tau \to 0$, the system in Equation 6.13 collapses to (6.7). On the other hand, as $\tau \to \infty$, evolutionary networks, discussed in the previous section, can be derived as special cases. For any other τ, a network with feedback loop, studied here, is attained.

Using the theory of matrix differential equations, the solution of (6.13) has the general form of:

$$\boldsymbol{x}(t) = \lim_{j \to \infty} e^{\mathcal{L}_j \tau} e^{\mathcal{L}_{j-1} \tau} \ldots e^{\mathcal{L}_0 \tau} \boldsymbol{x}(0) \tag{6.14}$$

where $\boldsymbol{x}(0)$ represents the initial network's configuration.

Before studying the stability of (6.14), however, the following proposition reflects that strong node connectivity in $\boldsymbol{\mathcal{L}}_k$, for any k, remains intact under $e^{-\boldsymbol{\mathcal{L}}_k\tau}$.

Proposition 2. *If $\boldsymbol{\mathcal{L}}_k$ is associated with a strongly connected network (i.e. there is a directional path between any two nodes), then $e^{-\boldsymbol{\mathcal{L}}_k\tau}$ is strongly connected for every $\tau \in \mathbb{R}$.*

Proof. The matrix $\boldsymbol{\mathcal{L}}_k$ is associated with a strongly connected network. Accordingly, it can be written as $\boldsymbol{\mathcal{L}}_k = \boldsymbol{M} - d\boldsymbol{I}_n$, where $d = \max\{|\boldsymbol{\mathcal{L}}_{k_{ii}}|\}$, with $\boldsymbol{\mathcal{L}}_{k_{ii}}$ being the diagonal entries of $\boldsymbol{\mathcal{L}}_k$. Therefore, $e^{-\boldsymbol{\mathcal{L}}_k\tau} = e^{(\boldsymbol{M}-d\boldsymbol{I}_n)\tau} = e^{-d\boldsymbol{I}\tau}e^{\boldsymbol{M}\tau} \geq \delta\boldsymbol{M}$ for some $\delta > 0$. This shows that any two nodes with a direct link in $\boldsymbol{M}$ and $\boldsymbol{\mathcal{L}}_k$ (i.e., $-\boldsymbol{\mathcal{L}}_{k_{ij}} = \boldsymbol{M}_{ij} > 0$ and $-\boldsymbol{\mathcal{L}}_{k_{ji}} = \boldsymbol{M}_{ji} > 0$) have a direct link in $e^{-\boldsymbol{\mathcal{L}}_k\tau}$. Therefore, $e^{-\boldsymbol{\mathcal{L}}_k\tau}$ is associated with a strongly connected network, thus concluding the proof. □

Following the previous proposition and making use of Lemmas 9 and 10, a theorem showing that $\boldsymbol{x}(t)$ asymptotically converges to an agreement (i.e. $x_i(t) \to x_j(t), \forall i, j = 1, 2, \ldots, N$) is next presented and proven.

Theorem 8 (Evolutionary Agreement with Feedback Loop)**.** *For system* (6.8) *with $\boldsymbol{A} = \boldsymbol{0}$, $\boldsymbol{B} = -\boldsymbol{I}_N, \boldsymbol{C} = \boldsymbol{I}_N$, $\boldsymbol{D} = \boldsymbol{0}$ and $\boldsymbol{u} = \boldsymbol{\mathcal{L}}_k\boldsymbol{y}$, where $k\tau < t < (k+1)\tau$ and τ is the dwell time, every individual's state $x_i, i = 1, 2, \ldots, N$ converges to an agreement as $x_i(t) \to x_j(t), t \to \infty$ for every i, j.*

Proof. According to Lemma 9, $e^{\boldsymbol{\mathcal{L}}_k\tau}$ converges to $\boldsymbol{1}\nu^\mathsf{T}$ for all k, with $\sum_{i=1}^{N}\nu_i = 1$. Further, it can be verified that $\lim_{n\to\infty}\left(e^{\boldsymbol{\mathcal{L}}_k\tau}\right)^n = \boldsymbol{1}\nu^\mathsf{T}$. Therefore, the matrix $e^{\boldsymbol{\mathcal{L}}_k\tau}$ is an SIA.

Using Lemma 10 and Proposition 2, it is clear that the state-transition matrix $\boldsymbol{\Psi} = e^{\boldsymbol{\mathcal{L}}_m\tau}e^{\boldsymbol{\mathcal{L}}_{m-1}\tau}\ldots e^{\boldsymbol{\mathcal{L}}_0\tau}$ represents a strongly connected network. Furthermore, $\boldsymbol{\Psi}$ is stochastic, therefore, according to Lemma 9, such a matrix is SIA.

Note that according to (Ren et al., 2005a, Proof of Theorem 3.2) and the fact that the Laplacian matrices $\boldsymbol{\mathcal{L}}_k$ for every k share the same spanning trees through the evolution, the condition required for convergence of a sequence of an infinite number of SIA matrices in Lemma 8 (i.e., $\lambda(.) \leq d, 0 \leq d < 1$) holds. Therefore, it can be proven that $\lim_{j\to\infty} e^{\boldsymbol{\mathcal{L}}_j\tau}e^{\boldsymbol{\mathcal{L}}_{j-1}\tau}\ldots e^{\boldsymbol{\mathcal{L}}_0\tau} = \boldsymbol{1}\nu^\mathsf{T}$, with ν being a column vector summing to one. Hence

$$\boldsymbol{x}(t) = \lim_{j\to\infty} e^{\boldsymbol{\mathcal{L}}_j\tau}e^{\boldsymbol{\mathcal{L}}_{j-1}\tau}\ldots e^{\boldsymbol{\mathcal{L}}_0\tau}\boldsymbol{x}(0) = \boldsymbol{1}\nu^\mathsf{T}\boldsymbol{x}(0) = x^\star\boldsymbol{1}, \tag{6.15}$$

where $x^\star \in \mathbb{R}$ denotes the agreement point, thus concluding the proof. □

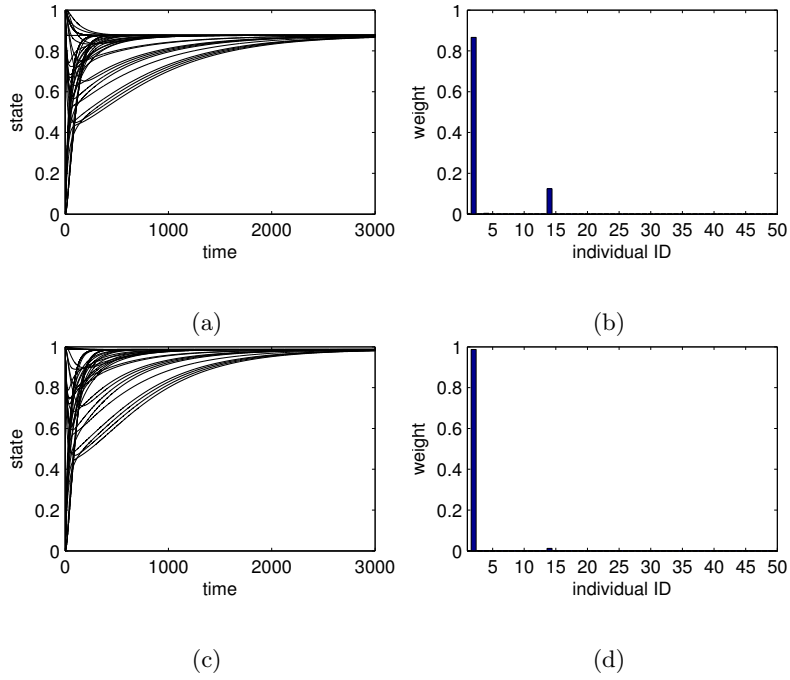


Figure 6.4: Agreement in evolutionary scale free network with feedback loop under different dwell times (a)-(b) $\tau = 20$ (c)-(d) $\tau = 0.01$.

Theorem 8 shows that the time varying Laplacian matrix, represented by the dynamical system of Equation 6.7 converges to an agreement: $x_i(t) \to x^\star$ as $t \to \infty$ for every i, with $x^\star$ being a weighted average of $x_i, i = 1, 2, \ldots, N$.

Experimental Validation:

Figures 6.4 and 6.5 illustrate the evolution of the sample networks of Figure 6.1 for two different dwell times. Furthermore, the elements of ν in Equation 6.15 are also shown. Clearly, Theorem 8 is validated since an agreement can be asymptotically reached. Moreover, it can be seen that systems of exactly the same initial configurations but different dwell times can have unequal final agreements.

Although no closed form solution can be derived for the multi-rate agreement of *evolutionary* networks with feedback loop, next a theorem for state-reference agreement showing that an individual with a fixed state inevitably determines the final agreement value is presented and proved.

Theorem 9 (Evolutionary State-Reference Agreement with Feedback Loop)**.** *For system* (6.8) *with* $\boldsymbol{A} = \boldsymbol{0}$, $\boldsymbol{B} = \boldsymbol{B}_r$ *as in* (6.12), $\boldsymbol{C} = \boldsymbol{I}_N$, $\boldsymbol{D} = \boldsymbol{0}$ *and* $\boldsymbol{u} = \boldsymbol{\mathcal{L}}_k \boldsymbol{y}$, *where*

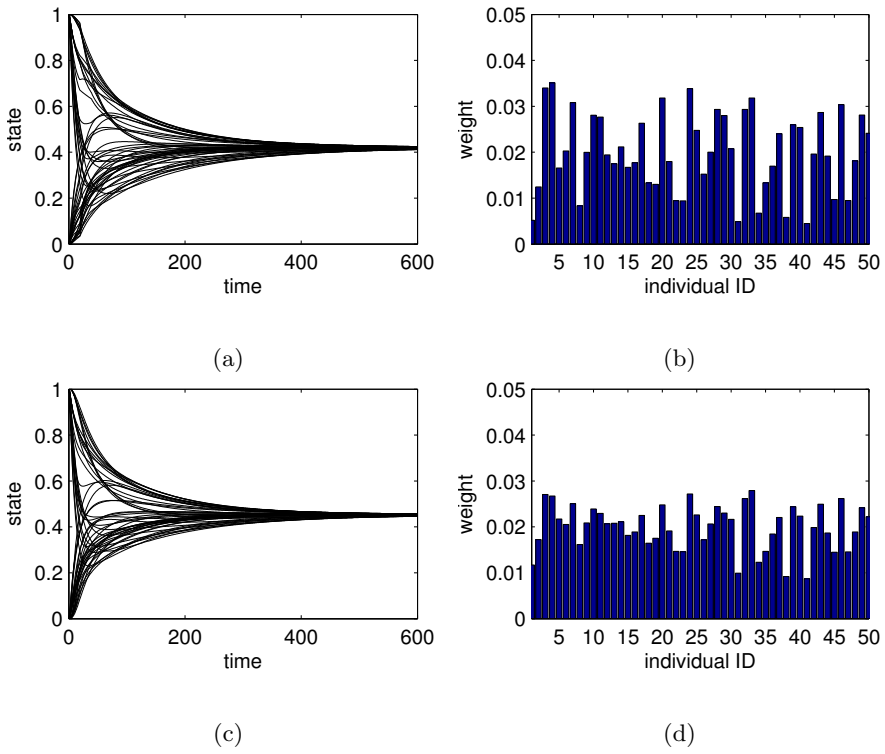


Figure 6.5: Agreement in evolutionary small world network with feedback loop under different dwell times (a)-(b) $\tau = 20$ (c)-(d) $\tau = 0.01$.

$k\tau < t < (k+1)\tau$ and τ being the dwell time, every individual's state $x_i, i = 1, 2, \ldots, N$ converges to an agreement as $x_i(t) \to x_{ref.}$ as $t \to \infty$.

Proof. Equation 6.8 is rewritten as

$$\dot{\boldsymbol{x}} = \boldsymbol{B}_r \mathcal{L}_k \boldsymbol{x} \tag{6.16}$$

The solution of (6.16) can be expressed as

$$\boldsymbol{x}(t) = \lim_{j \to \infty} e^{\boldsymbol{B}_r \mathcal{L}_j \tau} e^{\boldsymbol{B}_r \mathcal{L}_{j-1} \tau} \ldots e^{\boldsymbol{B}_r \mathcal{L}_0 \tau} \boldsymbol{x}(0)$$

According to the structure of $\boldsymbol{B}_r$ (i.e., all of its diagonal elements are -1 except for the element corresponding to the reference individual which is zero), it can be easily checked that $x_{\text{ref.}}(t) = x_{\text{ref.}}(0), \forall t > 0$ (the power series can be used to see that in the overall state-transition matrix of (6.8) all elements in the row corresponding to the reference individual are zero except the diagonal value which is one).

Furthermore, as described in the proof of Theorem 9 the network associated with system (6.16) contains a spanning tree through the evolution, and each $\boldsymbol{\mathcal{L}}_k$ matrix has one singular trivial eigenvalue and positive real parts for the nontrivial eigenvalues. Therefore, following a similar procedure to that proving Theorem 8, it can be shown that all state variables coverage to a final agreement as $x_i(t) \to x_{\text{ref.}}$ as $t \to \infty$. □

Experimental Validation:

Consider the sample networks in Figure 6.1. In Subfigures 6.6a and 6.6c one defector is chosen as the reference, and it can be seen that all the individuals of the evolutionary network with feedback loop eventually agree on pure defection. This verifies the results of Theorem 9.

To reflect upon the potential extension of the introduced framework, a tracking scenario is designed. A cooperator i is chosen as the reference state. However, its strategy varies, say according to $x_i(t) = \frac{1}{2} + \frac{1}{2}\sin(\frac{t}{1500})$. It is clear from Subfigures 6.6b and 6.6d that the whole network follows the reference state throughout the evolution. The phase-shift observed for the small world network can be explained by the absence of hubs in this case, causing the behavior to spread slowly.

6.6 Discussion

In this chapter we thoroughly analyzed the proposed CAIPD model, thereby gaining a broader understanding of the evolution of cooperation on complex social networks. Distinguishing between *evolutionary* networks, in which the interaction dynamics are fixed, and the more general case of *evolutionary* networks with time-varying dynamics, three main contributions can be listed. Firstly, convergence to agreement in evolutionary networks has been proven (Theorem 5). Moreover, it has been proven that this final agreement is a weighted average of the initial state, and that these weights can be computed explicitly using the trivial left eigenvector of the Laplacian matrix associated with the network in the very first iteration. Secondly, these proofs have been extended to the more general case of evolutionary networks with feedback loop (Theorem 8). Thirdly, an extension to CAIPD has been proposed that allows to model influence of the evolution of social networks towards states of specific individuals. It has been proven that individuals with lower adaptation rates contribute most to the final agreement. Moreover, all proofs have been validated empirically for both scale-free and small world networks.

These results provide a first step towards active control of complex social networks, by studying how certain individuals may influence the convergence and final agreement reached in the network. Moreover, the thorough analysis presented in this

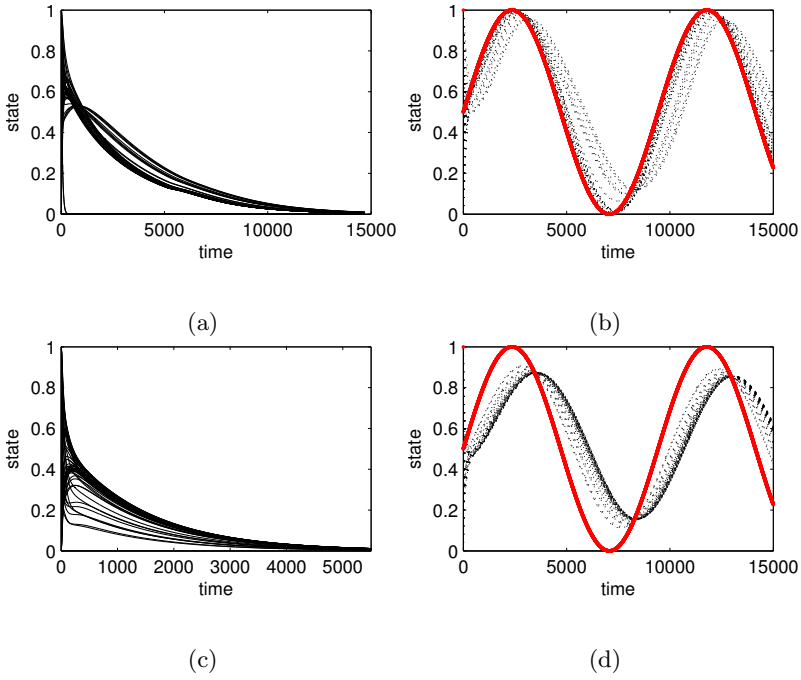


Figure 6.6: State-reference agreement in evolutionary networks with feedback loop (a) scale-free network with ref.= 14 (b) scale-free network with ref.=2 and $x_2(t)$ smoothly changing between cooperation and defection (c) small world network with ref.= 34 (d) small world network with ref.=50 and $x_{50}(t)$ smoothly changing between cooperation and defection.

work may constitute the basis for various directions in the study and understanding of such types of networks.

7

Simultaneous Evolution of Topology and Behaviors

This chapter is based on:
B. Ranjbar-Sahraei, D. Bloembergen, H. Bou-Ammar, K. Tuyls, G. Weiss, "Effects of Evolution on the Emergence of Scale Free Networks", In Proceedings of the fourteenth International Conference on the Synthesis and Simulation of Living Systems (ALIFE), New York, USA, 2014.

The evolution of cooperation in social networks, and the emergence of these networks using simple rules of attachment, have both been studied extensively although mostly in separation. In real-world scenarios, however, these two concepts are typically intertwined, where individuals' behaviors affect the structural emergence of the network and vice versa. Although much progress has been made in understanding each of the aforementioned fields, many joint characteristics are still unrevealed.

In the previous chapter, we provided the study of Evolution of Cooperation on a social network that was remaining fixed throughout the evolution. In this chapter, we study the Evolution of Cooperation over a network that is being emerged from a single node. This allows us to study the effect of network emergence on evolution of agents' behaviors and also study the effect of behavior evolution on the emergence of social networks. We call this method as Simultaneous Emergence and Evolution (SEE) model. To be more precise, the SEE model combines the continuous action prisoner's dilemma (modeling the evolution of cooperation) with preferential attachment (used to model network emergence), enabling the simultaneous study of both structural emergence and behavioral evolution of social networks. A set of empirical experiments show that the SEE model is capable of generating realistic complex networks, while at the same time allowing for the study of the impact of initial conditions on the evolution of cooperation.

7.1 Background

Many studies have targeted the discovery of structural properties of networks that promote cooperation. For instance, Santos and Pacheco (2005) show that cooperation has a higher chance of survival in scale-free networks; Ohtsuki et al. (2006) find a relation between the cost-benefit ratio of cooperation and the average node degree of a network that determines whether cooperation can be sustained; and Van Segbroeck et al. (2010) look at heterogeneity and clustering to find that these structural properties influence behavior on the individual rather than network-wide level. Others have focused on the role of the particular interaction model between neighboring nodes in determining the success of cooperation. For example, Hofmann et al. (2011) simulate various update rules in different network topologies and find that the evolution of cooperation is highly dependent on the combination of update mechanism and network topology. In **Chapter 6** we proposed a mathematical model, based on control theory, that allowed individuals to choose their actions from a continuous range between pure defection and pure cooperation. Control theory was also used by Bloembergen et al. (2014) aiming at ways of influencing the behaviors in social networks.

These studies have assumed the network to be fixed, looking only at the evolution of cooperation over time. In contrast, real-world social networks are not fixed, but continuously change as individuals make and break their ties (Kossinets and Watts, 2006). To this end, Zimmermann and Eguíluz (2005) and Santos et al. (2006) allow individuals to choose with whom to interact, e.g. by giving them the possibility to break ties with 'bad' neighbors and replacing them with a random new connection, and show that such a mechanism may promote cooperation. However, these work still assume a network to be in place, only modifying the connections between nodes over time.

Here, we investigate what happens when nodes are added to the network during interaction. Specifically, we start with an empty network, and add a new node at each time step. Simultaneously, the existing nodes in the network interact following the Continuous Action Iterated Prisoner's Dilemma (CAIPD) model of proposed in **Chapter 6**; new nodes are attached following preferential attachment. Usually, preferential attachment is assumed to follow the Barabási-Albert model Barabási and Albert (1999) where links are formed to existing ones proportional to their degree. However, in many social scenarios it intuitively makes sense to look at other individuals' performance rather than their degree when determining with whom to interact - connecting with high performing individuals may give you an edge. We empirically compare both methods of preferential attachment, looking at the structure of the networks formed in detail.

7.2 The SEE Model

Aiming at a unification, the Simultaneous Emergence and Evolution (SEE) model incorporates two evolutionary procedures. The first is concerned with the evolution of behaviors in the network, which follows from the CAIPD model. The second deals with the construction of the network itself; here, preferential attachment is used. Contrary to previous work, however, the links that each new individual forms with existing nodes depend on the current fitness of those nodes under the CAIPD dynamics, rather than on their degree. Next, an in-depth description of the SEE algorithm is presented.

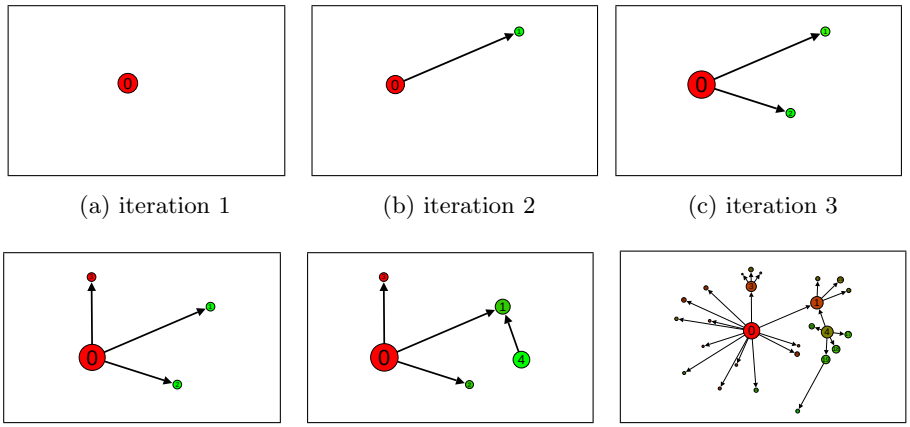
Starting from m initially connected individuals, new nodes are added one at a time. The initial state of these m nodes, as well as of each new node, are set randomly to either pure defection or pure cooperation with equal probability. Each new node is connected to m existing ones with a probability proportional to the fitness of the existing nodes, computed according to Equation 6.5. The connection probability p_i (i.e., the probability that a new node is connected to i) is defined as

$$p_i = \frac{f_i}{\sum_j f_j} \tag{7.1}$$

where f_i is the fitness of node i and the sum runs over all N pre-existing nodes $j = 1, 2, \ldots, N$. Therefore, nodes with high fitness tend to quickly accumulate more neighbors, while nodes with low fitness are unlikely to be chosen as the connector for a new node. An upper limit size of $N_{\max}$ is defined. This ensures that the network halts its expansion after reaching size $N_{\max}$.

In parallel to the structural emergence of the network, the CAIPD model is used to evolve the individual behaviors of the existing nodes. At each iteration, the adjacency matrix $\mathcal{W}$ is updated. Fitnesses are then computed according to Equation 6.5. These new fitness values are then used to update the state of each node, and therefore of the network as a whole, using the dynamical model of Equation 6.7. This is in practice performed with an adequately small step size. The SEE model allows to vary the update rates of both evolutionary processes independently. For example, the behavior of the individual nodes might evolve faster or slower than the rate at which new nodes are added. This ratio between the update rate of the behavior and the update rate of the network is defined by R_{Evo}, such that when at each time step k a new node is added, the CAIPD model progresses R_{Evo} steps.

To illustrate, consider a network initiated with a single node with pure defection state $x_0 = 0$, as depicted in Subfigure 7.1a. At the second iteration, Subfigure 7.1b, a cooperating individual is entering the environment (i.e. individual 1), and gets



(a) iteration 1 (b) iteration 2 (c) iteration 3

(d) iteration 4 (e) iteration 5 (f) iteration 25

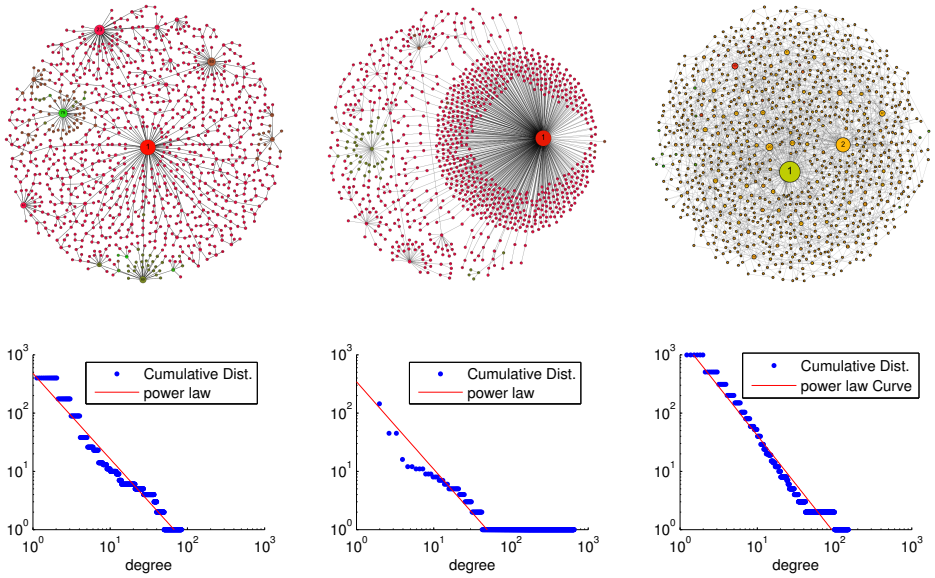
Figure 7.1: Emergence and evolution of a network according to the SEE model. Node size reflects individual's fitness and its color denotes the state (red for defection to green for cooperation). The direction of arrows shows how individuals influence each other.

attached to the defector (i.e., individual 0). At this stage, the defector acquires some benefits from the cooperator, while imposing a cost on the cooperator. This results in a higher fitness for the defector than the cooperator (depicted using the node size in the figure).

For further illustration, Subfigures 7.1c-7.1e show the attachment of three more individuals with defecting or cooperating states (chosen randomly) after joining the network. In parallel to this network emergence, individuals influence each other as described by the CAIPD model, resulting in a simultaneous evolution of their behavior. Subfigure 7.1f shows the structure and behavioral state of the network after the 25^{th} iteration.

7.3 Experiments and Results

In this section we first illustrate sample networks generated using the proposed SEE model, and show the scale-free characteristics that emerge. Hereafter, the cumulative degree distribution of 8000 different networks generated for 8 different settings of the SEE model will be studied in detail by computing the power law exponent in these networks. Finally, the evolution of cooperation resulting from the proposed SEE model is compared to the standard Barabási-Albert model of preferential attachment. In all experiments, the upper limit for network size is set to $N_{max} = 1000$. The



(a) $\alpha = 2.46$ (b) $\alpha = 2.53$ (c) $\alpha = 2.67$

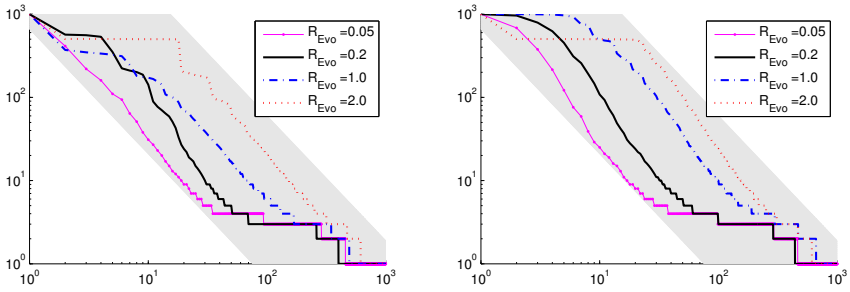
Figure 7.2: Sample network topologies generated by the SEE model (a) and (b) for $m = 1$ and (c) for $m = 2$, after 1000 iterations. The state and degree of the individuals are denoted by the color (red for pure defection to green for pure cooperation) and size of the nodes. The cumulative degree distribution of each network, shown as blue dots, shows how close this network follows a power law curve, shown as red line, with exponent α.

number of links added for each new individual, m, is set to either 1 or 2 (indicated where applicable). In the CAIPD model the step size is 0.1; $b = 4$, $c = 1$ and $\beta = 1$.

Sample Networks Generated by the SEE Model

Consider an evolution ratio R_{Evo} of 1 in a network that initiates from a single individual which is set initially to either pure defection or pure cooperation. When applying the SEE model, various different network structures can be expected to emerge, as there is stochasticity involved in both initialization of the nodes' states and their attachment. Three samples of such networks are illustrated in the top portion of Subfigures 7.2a-7.2c.

In order to study whether the networks generated by the SEE model follow a power law degree distribution, the cumulative degree distribution, i.e., the number of nodes with degree greater than or equal to k, of the sample networks in Subfigures 7.2a-7.2c



(a) m = 1 (b) m = 2

Figure 7.3: Cumulative Degree distribution of nodes for different evolution rates. The gray region contains the power law distributions corresponding to the scale free networks with exponent $\alpha = 2.5$ and different coefficients of the power law.

are shown on a log-log scale[1]. The results indeed show a power-law degree distribution with exponent close to $\alpha = 2.5$ for the sample networks in Figure 7.2. Next, we study the average cumulative degree distribution of networks generated by the SEE model in more detail.

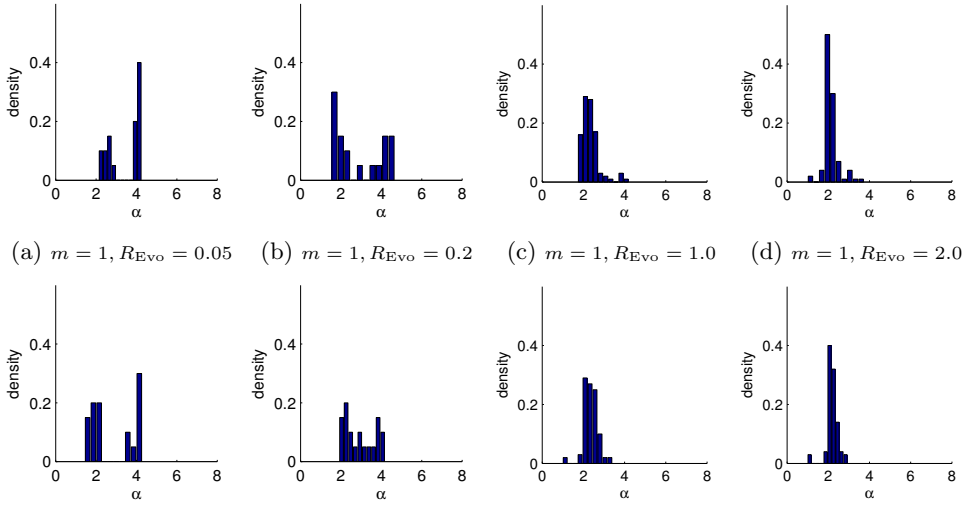
Degree Distribution in the SEE Model

In this section, we provide an empirical study on a large number of different networks generated using the proposed SEE model. We analyse different settings for R_{Evo}, ranging from 0.05 (slow evolution) to 2 (fast evolution). Subfigures 7.3a and 7.3b show the average cumulative degree distribution of these networks for $m = 1$ and $m = 2$, respectively. For each combination of settings, the SEE model runs for 1000 iterations, with initial nodes randomly set to either cooperation or defection, and the results are averaged.

Subfigures 7.3a and 7.3b illustrate emergence of networks under SEE model. On average, the networks largely follow a power law degree distribution with exponent close to $\alpha = 2.5$. When evolution is slow (i.e., $R_{\text{Evo}} \to 0$) the power law is less clearly present, in particular towards the high end of the degree distribution. A possible explanation is that, as the CAIPD evolution slows down, the fitness of the nodes gets updated less frequently as there are fewer interactions. Hence, having more neighbors does not immediately translate to a potential higher fitness.

To get a more detailed insight, the distribution of the exponent of power law distribution that is fit to the constructed networks is illustrated in Subfigures 7.4a-7.4d

[1]The cumulative distribution of a power law distribution is also power law; see Lemma 4 in **Chapter 5**.



(a) $m = 1, R_{\text{Evo}} = 0.05$ (b) $m = 1, R_{\text{Evo}} = 0.2$ (c) $m = 1, R_{\text{Evo}} = 1.0$ (d) $m = 1, R_{\text{Evo}} = 2.0$

(e) $m = 2, R_{\text{Evo}} = 0.05$ (f) $m = 2, R_{\text{Evo}} = 0.2$ (g) $m = 2, R_{\text{Evo}} = 1.0$ (h) $m = 2, R_{\text{Evo}} = 2.0$

Figure 7.4: Distribution of the power law exponent for various evolution ratios in the SEE model.

and Subfigures 7.4e-7.4h for $m = 1$ and $m = 2$, respectively. These figures show that the SEE model with slow evolution rates exhibit power law degree distribution with exponents $1 < \alpha < 5$ (i.e., some of the networks fall outside the range of typical real-world complex networks). Increasing the evolution ratio shrinks the range of α values that are observed and centers their distribution around $\alpha = 2.5$, yielding realistic scale-free networks. Moreover, it is interesting to note that a bifurcation seems to take place when R_{Evo} decreases (in Subfigures 7.4a, 7.4b and 7.4e): the distribution of power law exponents splits into two parts with their mass centered around 2 and 4. This phenomenon warrants a closer inspection in future work.

Comparison with Existing Model

In the previous section, the scale-free characteristic of the SEE model was studied and it was shown that the degree distribution of these networks follows a power law degree distribution with $\alpha \approx 2.5$. In this section, we study the influence of the SEE model on the evolution of behavior in the network. We compare the proposed SEE model, which uses preferential attachment based on fitness (see Equation 7.1), with the standard Barábasi-Albert model that uses the degree (see Equation 2.1). For all experiments, $N_{\text{MAX}} = 1000$, and the evolution ratio R_{Evo} is set to 1.

Subfigures 7.5a and 7.5b show the evolution of cooperation under the SEE model,

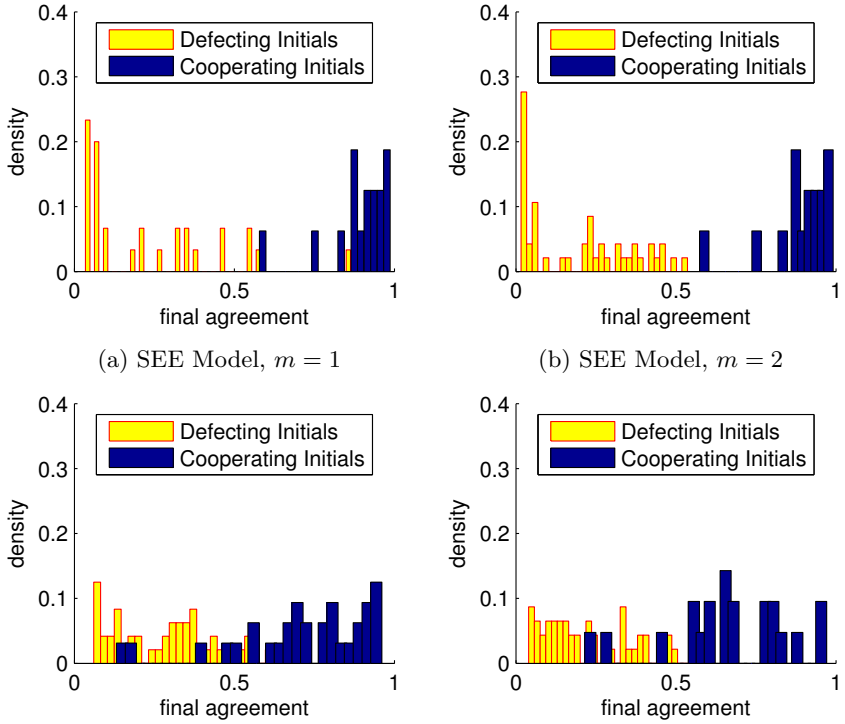
specifically the figures show the final cooperation level in the network depending on whether the initial nodes where either defectors or cooperators. Similarly, Subfigures 7.5c and 7.5d show the same results when the Barábasi-Albert (B-A) model is used for the preferential attachment. It is clear from these figures that the final cooperation level in the network greatly depends on the initial state of the first individuals. When the initial nodes are cooperators, the network tends to cooperate to a large (> 0.5) degree, whereas the situation reverses when the initial nodes are defectors. This effect is most clear under the SEE model, where a large fraction of the networks eventually reaches either a high (≈ 1) or low (≈ 0) degree of cooperation. When preferential attachment according to Barábasi-Albert is used, this effect is less strong. Here we observe a broad mix of final cooperation levels; moreover the divide between initial cooperators and initial defectors is less clear.

Clearly, the above results demonstrate that final agreements depend on the initial state of the first individuals in the network. This phenomenon is manifested in both the SEE and the Barábasi-Albert model. Having proposed a generalized and formal framework for analysing evolution of cooperation and network emergence, this aspect constitutes a major direction in our future work, where SEE can be used to acquire analytical conclusions describing the effects of such a dependence.

7.4 Discussion

The recent interest to study social networks and their behavior has led to many studies, which can roughly be divided in two streams. The first stream has studied the emergence of these networks, and has in particular tried to find generative models that can explain certain structural properties of real-world complex networks, such as a scale-free degree distribution. The second stream of research has focussed on the evolution of behaviors on such networks, when the nodes represent individuals that interact according to some rules. Most notably, interest has been in the evolution of cooperation in social networks, aiming to identify properties of both the network and the interactions that sustain cooperation.

This chapter aims to unify these two streams, by studying the simultaneous evolution of behavior on a social network, and the structural emergence of the network itself. The Simultaneous Emergence and Evolution (SEE) model proposed in this chapter combines a modified version of preferential attachment, used to generate scale-free networks, with the continuous action iterated prisoner's dilemma (CAIPD) model, describing the evolution of cooperation. Using the proposed model, a number of different networks, emerging from different initial conditions, have been studied. It has been shown that the SEE model yields realistic scale-free networks, despite the fact that the preferential attachment is based on individual's fitness rather than



(a) SEE Model, $m = 1$

(b) SEE Model, $m = 2$

(c) B-A Model, $m = 1$

(d) B-A Model, $m = 2$

Figure 7.5: Final degree of cooperation as a function of the initial members' state in the SEE model in (a) and (b) and the Barábasi-Albert (B-A) model in (c) and (d). The colors indicate the state of the initial members: blue for cooperators, and yellow for defectors.

degree.

Moreover, results show that both structural emergence and behavioral evolution are intertwined, mutually influencing each other, and should therefore be studied in tandem. Aiming at a better understanding of such phenomena, the SEE model provides a fundamental and general framework that allows the analysis of these processes as they co-evolve.

An interesting direction for future work is to include the possibility of rewiring as well in the SEE model, whereby existing nodes may break or create links at any time.

8 Conclusions

This chapter concludes this dissertation. First, the research questions that were posed in **Chapter 1** are answered in order to clarify the contributions done for solving the problem statement. Then, various perspectives for future research are offered. These perspectives include extensions of the presented methods in each chapter and the identification of new research avenues that are opened as a result of this research.

8.1 Answers to the Research Questions

Here we return to the research questions defined in **Chapter 1** and use the results given in **Chapters 3-7** to discuss each research question.

Question 1: *To what extent cross-matching of existing sources of structured and unstructured data can eliminate the need for human inputs during an information extraction process?*

In **Chapter 3**, we proposed a relation extraction method called REDS. REDS takes advantage of relationship fingerprints that provide a reliable and scalable signature for relationships. Using the concept of fingerprints allowed to exploit the information in a structured knowledge repository for extracting relationships for unstructured data. In a real world genealogical setting, we showed that this approach successfully reached a precision of 90% in extracting relationships that had overlaps with the knowledge repository. Furthermore, REDS was able to extract 75% of the relationships in a different dataset on which it was not trained. Specifically, REDS provided a framework to use distant supervision in domains where there is no redundancy in named entities and relationships.

Question 2: *How to develop a simple, interpretable and yet scalable identity resolution tool for heterogeneous social data?*

We developed a tool called HiDER which was introduced in **Chapter 4**. According to the evaluations of domain experts, this tool performs the task of identity resolution with reliable results. Yet, the overall architecture of HiDER is very scalable and interpretable. HiDER takes advantage from the well-established concept of inverted indexing and, in particular, from the concept of unique relationship fingerprints that were proposed as a solution to the previous research question. This makes HiDER easy to integrate with existing searching platforms in various domains. We elaborated on the benefits of using HiDER in dealing with a real-world heterogeneous social data and illustrated the outcomes of this tool.

Question 3: *What are the origins of the heavy-tailed distributions in social graphs?*

Due to the importance of a deeper understanding of social graphs, in **Chapter 5** we proposed two new models for the emergence of social graphs based on the theory of dynamical systems. These two models, DBEM and PBEM, follow simple rules to capture the effect of dominance and prestige in a hierarchical social network, respectively. Interestingly, the equilibrium state of the former model recovers an exponential distribution and the equilibrium state of the latter model generates a power-law distribution in the social graph. Due to the deterministic nature of both models, the exponential and power-law distributions are derived in closed-form. The analytical results given in this chapter and the final evaluations backtrack the origin of heavy-tailed distributions to the concept of prestige in hierarchical structures. In addition, the model proposed in this chapter sheds light on the reasons behind prevalence of hierarchies in social structures.

Question 4: *To what extent can the theory of dynamical systems contribute to modeling the feedback loop between behaviors and interactions in social graphs?*

Chapter 6 employed the theory of dynamical systems to develop a state-space model for the evolution of behaviors in social graphs. This model unravels the feedback loop between behaviors and interaction strengths, and thus it allows for a detailed analytical study. This chapter specifically proposes the CAIPD model for modeling the iterated Prisoners' Dilemma among multiple social agents. Using CAIPD, 1) we provided a detailed study of the model steady state and revealed the connections of this steady-state with eigenvectors of the network adjacency matrix; 2) we used the concept of dwell time to study the behaviors of evolving social graphs with the theory of linear systems; and 3) we used the theory of linear systems to provide a method of imposing indirect influence on the evolution of the network.

Question 5: *What are the mutual effects of evolving behaviors and emerging topologies?*

Two separate research topics *a)* analysis of social behaviors and *b)* analysis of social structures are defined for a better understanding of the evolution of social networks. Aiming at bridging these two research topics, in **Chapter 7**, we apply the CAIPD model, which was developed for answering the previous research question, over a network that is just emerging from one single node. The new model that we propose, SEE, captures the mutual effects of evolving behaviors and emerging topologies. The benefits of using SEE model are shown to be two-fold; SEE allows us to make a deeper analysis of evolution of behaviors in social networks and also helps us to generate diverse social structures. A key finding of SEE model is the high influence of initial members' behaviors over the steady state behavior of a network in long run. In other words, the overall behavior of societies is highly biased toward the behavior of initial/founding members.

8.2 Answers to the Problem Statement

This thesis aimed at studying dynamic social graphs from both a data-driven and an analytical perspective. The data-driven approaches proposed in **Chapters 3** and **4** provided powerful practical tools for the integration of heterogenous sources of social data. These tools resulted in the automatic generation of a set of rich social graphs that could not be analyzed with existing analytical methods. Therefore, two novel analytical models were proposed in **Chapter 5** that allowed for a deeper understanding of the topological properties of social graphs. A new analytical model was proposed in **Chapter 6** to model the evolution of behaviors in such graphs. The proposed models successfully captured important properties of dynamic social graphs. A further research step was taken in **Chapter 7**: the simultaneous evolution of topology and behaviors in social graphs was studied and the effect of this co-evolution on topological diversity of the social graphs and their equilibrium state was demonstrated.

To conclude, effective methods for turning heterogenous social data into social graphs had been developed and novel models for the analysis of such dynamic social graphs were proposed. Several possible promising extensions of the proposed methods as well as new research avenues opened up by the research described in this thesis are described in the next section.

8.3 Perspectives for Future Research

In this thesis, we discussed the solution of two main challenges: (a) how to extract the structure and dynamics of social graphs form heterogeneous social data; and (b) how to model and analyze the properties of dynamic social graphs. In each chapter, we explored a particular aspect of these two challenges by (1) introducing a framework to extract information from unstructured data; (2) developing an interpretable and scalable information retrieval tool for retrieving information from heterogeneous social data; (3) analyzing the origins of heavy-tailed distributions in social networks; (4) modeling the feedback loop between agent behaviors and interactions, and (5) studying the simultaneous evolution processes in a network.

Due to variety of domains and ever-changing requirements in each domain, the work presented in this thesis can be extended and further studied along different directions. In addition, this thesis opens up new research venues that are elaborated on at the end of this section.

In-Chapter Improvements

In each chapter, we proposed new methods to tackle the considered challenges. These methods can be further studied and extended in other domains and application contexts.

Chapter 3 - Future work can follow two directions. First, the REDS method can be implemented on domains other than historical data. For instance, REDS can be used to extract the relations between uncommon entities from Wikipedia documents or scientific articles. The main advantage of REDS — specifically when extraction of hyper-relationships (i.e., relationships among more than two entities) is considered — is its lower computation complexity compared to existing techniques. Second, classification techniques other than Decision Tree, such as the Naive Bayes classifiers and Logistic Regression, can be used for learning the relation patterns.

Chapter 4 - There are several interesting future directions for the developed data product of this chapter. First, principles of HiDER can be applied to domains other than historical information (e.g., scientific articles) to generate story lines for entities. In particular, the use of fingerprints to adopt the information retrieval framework can increase the accuracy and make the method scalable. Second, HiDER extracts valuable knowledge from heterogeneous raw data, an extraction that demands new visualization techniques. Therefore, exploring possible ways for visualization of genealogical data is an interesting line of future research and extension. Third, HiDER is a data product that allows users to interact with historical data. Extracting insights

from user data by itself is an interesting research topic. For instance, the reliability of search results and the preferences of users can be explored by analyses of HiDER logfile.

Chapter 5 - Future work can involve four directions. First, the DBEM and PBEM models follow simple principles, yet generate very interesting results. Inspired by the real world, more sophisticated rules (e.g., by considering adaptation and mutation) can be designed, and their behaviors might lead to new findings on the origin of social graph properties. Second, in addition to the closeness centrality of individuals, network measures, such as betweenness centrality, HITS, and clustering, should be tackled. Finding a closed form formula for each of these characteristics can lead to very valuable findings. Third, the use of DBEM and PBEM for distinguishing between dominance and prestige were slightly discussed throughout the real-world validations. This property can be further studied and the models can be exploited to validate large-scale networks. Fourth, the data processing phase required for validation of real-world data can be further improved. While we used a simple method for the extraction of hierarchies and the normalization of the adjacency matrix, more advanced and optimized techniques can be used for this phase.

Chapter 6 - Future work can can have different goals. First, CAIPD can be extended to games other than the Prisoner's Dilemma (e.g., Stag Hunt and Public Goods Game) to propose a *generic model for evolution of behaviors and interactions in complex networks*. Second, the multi-rate update strategy that indirectly influences the evolution of social graph can be further studied. Specifically, this strategy can be mathematically analyzed in the presence of a feedback loop. Third, the *dwell time* concept used to prove stability of the time varying evolution can be replaced by a more elegant proof. Any effort in this direction can contribute to the System Theory community as well as the Social Network Analysis community. Fourth, the structure and values of the CAIPD system matrix can be further studied. An interesting related research question is finding an appropriate method which uses the initial state of this matrix to predict the steady state behaviors of the model.

Chapter 7 - Future work can include three directions. First, by combining the CAIPD model and PBEM/DBEM under the umbrella of SEE, a unique mathematical model can be developed, where the former model explains the dynamics of behaviors and the latter explains the dynamics of topology. This mathematical model can lead to interesting findings on the characteristics of real-world networks. Second, under SEE, a variety of network properties emerges. These network properties can be further explored; these properties are generated through the formation of novel topologies and

new evolution trends. Third, SEE allows for choosing the ratio between evolution rate and emergence rate. The effect of this hyper-parameter on behavior of the system can be further studied.

New Research Avenues

This thesis opens up at least four new promising research avenues. First, it allows for conducting of *Advanced Genealogical Research*. Second, it paves the way to *Trace the Origins of Heavy-Tailed Distributions* in social networks using theory of dynamical systems. Third, it provides new methods for studying *Controllability of Behaviors in Social Networks*. Fourth, it establishes a bridge between *Swarm Robotics* and *Experimental Economics*. Next, these research avenues are explained in more detail.

Advanced Genealogical Research: Traditionally, heterogeneous genealogical data is maintained in different corpora and different information retrieval systems are used for each type. For instance, civil registers are usually indexed based on their field names (e.g., name of father, name of mother, date of birth, etc.) and the unstructured information, such as notarial acts, are indexed by metadata (e.g., date and place of issue and category). Furthermore, the retrieved information is often represented in form of flat tables. Although, a transition to more advanced visualization techniques in form of interactive family trees can be observed, successful and sophisticated visualizations are rare.

The first part of this thesis focused on developing a scalable Identity Resolution technique for the integration of heterogeneous sources of social data, with specific application to genealogical data. We demonstrated how to use structured data for extracting named entities and relationships from unstructured data, and how to use network-based identity resolution methods to integrate available pieces of information. This thesis shows how the retrieved information in response to a user query can be represented in form of story lines and graphs, in contrast to the traditional representation in form of flat tables. The developed tool allows genealogists to come up with new research questions and makes them capable of studying the population under study with detail and in large scale.

Tracing the Origins of Heavy-Tailed Distributions in Social Networks: Heavy-tailed distribution is widely observed in complex networks, such as air traffic, friendship networks, and article citations. Although the research community has immensely studied the structural properties, characteristics and applications of these distributions, the origins of these distributions are unknown. In other words, the

fundamental question of "why do humans generate heavy-tailed distributions by their actions" is still unanswered.

In this thesis, based on the theory of dynamical systems, we made a connection between the hierarchical structure of a network and the heavy-tailed distribution of individuals' strengths. We mathematically argued that as soon as a prestige-based hierarchy replaces a dominance-based hierarchy, the exponential distributions turn into heavy-tailed ones. This mathematical link between hierarchy and distribution of strengths paves the way for further research. Based on the work of this thesis, we can trace the origins of a social network properties to dominance-oriented behavior or prestige-oriented behavior, and thus we can gain insight into the evolutionary foundations of social networks.

Controllability of Behaviors in Social Networks: Studying the Controllability of Complex Networks has attracted much interest in the recent decades. A majority of the proof-oriented work in this field is developed based on the theory of dynamical systems and it exploits the theory of linear systems. Therefore, the linearity of complex networks is one of the main prerequisites for employing controllability methods. In practice, however, no dynamical model exists for most of the real-world networks and if any model exists, the behaviors are very often nonlinear. Therefore, the recently proposed methods cannot be directly applied to real-world settings.

In this thesis, inspired by the evolutionary game theory, we proposed a time-varying linear model for the evolution process of behaviors, interactions, and network topology. As we showed in this thesis by means of a *multi-rate update* process, this framework allows us to introduce the concept of control into social networks. In addition, Bloembergen et al. (2014) adopted our proposed framework and applied the optimal control principles to an arbitrary social network. Therefore, we have provided a bridge between *social network analysis* and *controllability of linear systems* — this can be further explored in the future.

Use of Swarm Robotics in Facilitating the Experimental Economics: In experimental economics, often a group of human subjects are invited to a lab to play a game (e.g., the prisoner's dilemma, ultimatum game or public goods game), and the behavior of lab subjects is recorded and used to test the validity of theories. This process is, however, costly and time-consuming. In addition, exposing human subjects to different situations should follow strict ethical guidelines. An alternative approach, which is being used vastly by computer scientists, is to simulate social agents and allow them to play games. Computer simulations are cheap, fast, and relatively easy to analyze. A third alternative is to use multi-robot systems for conducting similar type of experiments. Not only is this approach relatively cheap, easy, and interpretable,

but it also contains realistic uncertainties that are mandatory for practically relevant testing.

To use multi-robot systems for experimental economics, we need a mathematical framework capable of capturing the characteristics of economic games and the dynamics of robots. The CAIPD model proposed in this thesis provides such a framework. By using this mathematical framework, a group of robots can interact in an uncertain environment and their behaviors can be translated into properties of the game. E.g., we have made a first attempt at such research, results of which were published in Ranjbar-Sahraei et al. (2014c).

Bibliography

Agichtein, E. and Ganti, V. (2004). Mining reference tables for automatic text segmentation. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 20–29. ACM.

Agichtein, E. and Gravano, L. (2000). Snowball: Extracting relations from large plain-text collections. In *Proceedings of the fifth ACM conference on Digital libraries*, pages 85–94. ACM.

Albert, R. and Barabási, A.-L. (2002). Statistical mechanics of complex networks. *Reviews of modern physics*, 74(1):47.

Altwaijry, H., Kalashnikov, D. V., and Mehrotra, S. (2013). Query-driven approach to entity resolution. *Proceedings of the VLDB Endowment*, 6(14):1846–1857.

Amblard, F. and Deffuant, G. (2004). The role of network topology on extremism propagation with the relative agreement opinion dynamics. *Physica A: Statistical Mechanics and its Applications*, 343:725–738.

Appleby, M. C. (1983). The probability of linearity in hierarchies. *Animal Behaviour*, 31(2):600–608.

Axelrod, R. and Hamilton, W. D. (1981). The evolution of cooperation. *Science*, 211:1390–6.

Bach, N. and Badaskar, S. (2007). A review of relation extraction. *Literature review for Language and Statistics II*.

Banko, M., Cafarella, M. J., Soderland, S., Broadhead, M., and Etzioni, O. (2007). Open information extraction for the web. In *IJCAI*, volume 7, pages 2670–2676.

Barabási, A.-L. and Albert, R. (1999). Emergence of scaling in random networks. *Science*, 286(5439):509–12.

Barrat, A., Barthélemy, M., and Vespignani, A. (2004a). Modeling the evolution of weighted networks. *Physical Review E*, 70(6):066149.

Barrat, A., Barthélemy, M., and Vespignani, A. (2004b). Weighted evolving networks: coupling topology and weight dynamics. *Physical review letters*, 92(22):228701.

Bartunov, S., Korshunov, A., Park, S.-T., Ryu, W., and Lee, H. (2012). Joint link-attribute user identity resolution in online social networks. In *Proceedings of the 6th International Conference on Knowledge Discovery and Data Mining, Workshop on Social Network Mining and Analysis. ACM.*

Baxter, R., Christen, P., and Churches, T. (2003). A comparison of fast blocking methods for record linkage. In *ACM SIGKDD '03 Workshop on Data Cleaning, Record Linkage, and Object Consolidation*, pages 25–27.

Bhattacharya, I. and Getoor, L. (2007). Collective entity resolution in relational data. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 1(1):5.

Bilenko, M., Kamath, B., and Mooney, R. J. (2006). Adaptive blocking: Learning to scale up record linkage. In *Data Mining, 2006. ICDM'06. Sixth International Conference on*, pages 87–96. IEEE.

Bilge, L., Strufe, T., Balzarotti, D., and Kirda, E. (2009). All your contacts are belong to us: automated identity theft attacks on social networks. In *Proceedings of the 18th international conference on World wide web*, pages 551–560. ACM.

Bloembergen, D., Ranjbar-Sahraei, B., Ammar, H. B., Tuyls, K., and Weiss, G. (2014). Influencing social networks: An optimal control study. *Proceedings of the 21st European Conference on Artificial Intelligence (ECAI 2014).*

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

Bloothooft, G. (1994). Corpus-based name standardization. *History and Computing*, 6(3):153–167.

Boal, S. R. (2013). Identity resolution for consumers with shared credentials. US Patent App. 13/944,486.

Boccaletti, S., Latora, V., Moreno, Y., Chavez, M., and Hwang, D.-U. (2006). Complex networks: Structure and dynamics. *Physics reports*, 424(4):175–308.

Bollobás, B. and Riordan, O. (2003). Coupling scale-free and classical random graphs. *Internet Mathematics*, 1(2):215–225.

Borkar, V., Deshmukh, K., and Sarawagi, S. (2001). Automatic segmentation of text into structured records. In *ACM SIGMOD Record*, volume 30, pages 175–186. ACM.

Brin, S. (1999). Extracting patterns and relations from the world wide web. In *The World Wide Web and Databases*, pages 172–183. Springer.

Buccafurri, F., Lax, G., Nocera, A., and Ursino, D. (2015). Discovering missing me edges across social networks. *Information Sciences*.

Bunescu, R. C. and Mooney, R. J. (2005). A shortest path dependency kernel for relation extraction. In *Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing*, pages 724–731. Association for Computational Linguistics.

Cancho, R. F. i. and Fernández, A. H. (2008). Power laws and the golden number. *Problems of general, germanic and slavic linguistics*, pages 518–523.

Choi, F. Y. (2000). Advances in domain independent linear text segmentation. In *Proceedings of the 1st North American chapter of the Association for Computational Linguistics conference*, pages 26–33. Association for Computational Linguistics.

Christen, P. (2012). *Data matching: concepts and techniques for record linkage, entity resolution, and duplicate detection*. Springer Science & Business Media.

Christen, P. and Gayler, R. (2008). Towards scalable real-time entity resolution using a similarity-aware inverted index approach. In *Proceedings of the 7th Australasian Data Mining Conference-Volume 87*, pages 51–60. Australian Computer Society, Inc.

Clauset, A., Moore, C., and Newman, M. E. (2008). Hierarchical structure and the prediction of missing links in networks. *Nature*, 453(7191):98–101.

Codd, E. F. (1990). *The Relational Model for Database Management: Version 2*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA.

Cucchiarelli, A. and Velardi, P. (2001). Unsupervised named entity recognition using syntactic and semantic contextual evidence. *Computational Linguistics*, 27(1):123–131.

de Vries, H. (1995). An improved test of linearity in dominance hierarchies containing unknown or tied relationships. *Animal Behaviour*, 50(5):1375–1389.

de Vries, H. (1998). Finding a dominance order most consistent with a linear hierarchy: a new procedure and review. *Animal Behaviour*, 55(4):827–843.

DOMO (2014). Data never sleeps 2.0. `https://www.domo.com/blog/2014/04/data-never-sleeps-2-0/`.

Durrett, R. (2006). *Random Graph Dynamics (Cambridge Series in Statistical and Probabilistic Mathematics)*. Cambridge University Press, New York, NY, USA.

Efremova, J., Ranjbar-Sahraei, B., and Calders, T. (2014a). A hybrid disambiguation measure for inaccurate cultural heritage data. In *the 8th Workshop on LaTeCH*, pages 47–55.

Efremova, J., Ranjbar-Sahraei, B., Oliehoek, F. A., Calders, T., and Tuyls, K. (2014b). A baseline method for genealogical entity resolution. In *Workshop on Population Reconstruction*.

Efremova, J., Ranjbar-Sahraei, B., Rahmani, H., Oliehoek, F., Calders, T., Tuyls, K., and Weiss, G. (2015). Multi-source entity resolution for genealogical data. In *Population Reconstruction*, pages 129–154. Springer International Publishing.

Elmasri, R. A. and Navathe, S. B. (1999). *Fundamentals of Database Systems*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 3rd edition.

Estrada, E. and Vargas-Estrada, E. (2013). How peer pressure shapes consensus, leadership, and innovations in social groups. *Scientific reports*, 3.

Etzioni, O., Banko, M., Soderland, S., and Weld, D. S. (2008). Open information extraction from the web. *Communications of the ACM*, 51(12):68–74.

Etzioni, O., Cafarella, M., Downey, D., Kok, S., Popescu, A.-M., Shaked, T., Soderland, S., Weld, D. S., and Yates, A. (2004). Web-scale information extraction in knowitall:(preliminary results). In *Proceedings of the 13th international conference on World Wide Web*, pages 100–110. ACM.

Etzioni, O., Cafarella, M., Downey, D., Popescu, A.-M., Shaked, T., Soderland, S., Weld, D. S., and Yates, A. (2005). Unsupervised named-entity extraction from the web: An experimental study. *Artificial intelligence*, 165(1):91–134.

Feingold, D. G. and Varga, R. S. (1962). Block diagonally dominant matrices and generalizations of the Gershgorin circle theorem. *Pacific J. Math.*, 12:1241–1250.

Ferrara, E., De Meo, P., Fiumara, G., and Baumgartner, R. (2014). Web data extraction, applications and techniques: A survey. *Knowledge-Based Systems*, 70:301–323.

Freeman, L. C. (1978). Centrality in social networks conceptual clarification. *Social networks*, 1(3):215–239.

Froehlich, J. W. and Thorington, Richard W., J. (1981). The genetic structure and socioecology of howler monkeys (alouatta palliata) on barro colorado island. *Ecology*

of Barro Colorado Island: Seasonal Rhythms and Long Term Changes in a Tropical Forest, ed. E. G. Leigh and A. S. Randi.

Garlaschelli, D., Battiston, S., Castri, M., Servedio, V. D., and Caldarelli, G. (2005). The scale-free topology of market investments. *Physica A: Statistical Mechanics and its Applications*, 350(2):491–499.

Gibbons, R. (1992). *A Primer in Game Theory.* Pearson Education.

Granovetter, M. S. (1973). The strength of weak ties. *American journal of sociology*, pages 1360–1380.

Grant, T. (1973). Dominance and association among members of a captive and a free-ranging group of grey kangaroos (*Macropus giganteus*). *Animal Behaviour*, 21(3):449–456.

Guimera, R., Mossa, S., Turtschi, A., and Amaral, L. N. (2005). The worldwide air transportation network: Anomalous centrality, community structure, and cities' global roles. *Proceedings of the National Academy of Sciences*, 102(22):7794–7799.

GuoDong, Z., Jian, S., Jie, Z., and Min, Z. (2005). Exploring various knowledge in relation extraction. In *Proceedings of the 43rd annual meeting on association for computational linguistics*, pages 427–434. Association for Computational Linguistics.

Hauert, C. and Szabó, G. (2005). Game theory and physics. *American Journal of Physics*, 73:405.

Hegselmann, R. and Krause, U. (2002). Opinion dynamics and bounded confidence models, analysis, and simulation. *Journal of Artificial Societies and Social Simulation*, 5(3).

Hofmann, L.-M., Chakraborty, N., and Sycara, K. (2011). The Evolution of Cooperation in Self-Interested Agent Societies: A Critical Study. *Proc. of 10th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2011)*, pages 685–692.

Jackson, M. O. (2008). *Social and Economic Networks.* Princeton University Press.

Jadbabaie, A., Lin, J., and Morse, A. S. (2003). Coordination of groups of mobile autonomous agents using nearest neighbor rules. *Automatic Control, IEEE Transactions on*, 48(6):988–1001.

Kambhatla, N. (2004). Combining lexical, syntactic, and semantic features with maximum entropy models for extracting relations. In *Proceedings of the ACL 2004*

on Interactive poster and demonstration sessions, page 22, Morristown, NJ, USA. Association for Computational Linguistics.

Kendall, M. G. (1948). Rank correlation methods.

Killingback, T. and Doebeli, M. (2002). The continuous prisoner's dilemma and the evolution of cooperation through reciprocal altruism with variable investment. *The American Naturalist*, 160(4):421–438.

KONECT (2015a). Kangaroo network dataset – KONECT. `http://konect.uni-koblenz.de/networks/moreno_kangaroo`.

KONECT (2015b). Us airports network dataset – KONECT. `http://konect.uni-koblenz.de/networks/opsahl-usairport`.

Köpcke, H., Thor, A., and Rahm, E. (2010). Evaluation of entity resolution approaches on real-world match problems. *Proceedings of the VLDB Endowment*, 3(1-2):484–493.

Kossinets, G. and Watts, D. J. (2006). Empirical analysis of an evolving social network. *Science*, 311(5757):88–90.

Kretschmer, H. (2004). Author productivity and geodesic distance in bibliographic co-authorship networks, and visibility on the web. *Scientometrics*, 60(3):409–420.

Lazer, D., Pentland, A., Adamic, L., Aral, S., Barabási, A.-L., Brewer, D., Christakis, N., Contractor, N., Fowler, J., Gutmann, M., Jebara, T., King, G., Macy, M., Roy, D., and Van Alstyne, M. (2009). Computational social science. *Science*, 323(5915):721–723.

Leskovec, J., Backstrom, L., Kumar, R., and Tomkins, A. (2008). Microscopic evolution of social networks. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 462–470. ACM.

Leung, C. and Chau, H. (2007). Weighted assortative and disassortative networks model. *Physica A: Statistical Mechanics and its Applications*, 378(2):591–602.

Levine, W., editor (1996). *The Control Handbook*. CRC Press, Boca Raton, FL.

Linton C. Freeman (2015a). Datasets - Froelich, Thorington, Sailer, Gaulin – Howler Monkey groups. `http://moreno.ss.uci.edu/data.html#howler`.

Linton C. Freeman (2015b). Datasets - van Hooff, Wensing – Wolf Dominance. `http://moreno.ss.uci.edu/data.html#wolf`.

Lotka, A. J. (1926). The frequency distribution of scientific productivity. *Journal of Washington Academy Sciences*.

McCallum, A. (2005). Information extraction: Distilling structured data from unstructured text. *Queue*, 3(9):48–57.

Michelson, M. and Knoblock, C. A. (2006). Learning blocking schemes for record linkage. In *Proceedings of the National Conference on Artificial Intelligence*, volume 21, page 440. Menlo Park, CA; Cambridge, MA; London; AAAI Press; MIT Press; 1999.

Mintz, M., Bills, S., Snow, R., and Jurafsky, D. (2009). Distant supervision for relation extraction without labeled data. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 2-Volume 2*, pages 1003–1011. Association for Computational Linguistics.

Mones, E., Vicsek, L., and Vicsek, T. (2012). Hierarchy measure for complex networks. *PloS one*, 7(3):e33799.

Motoyama, M. and Varghese, G. (2009). I seek you: searching and matching individuals in social networks. In *Proceedings of the eleventh international workshop on Web information and data management*, pages 67–75. ACM.

Muchnik, L., Pei, S., Parra, L. C., Reis, S. D., Andrade Jr, J. S., Havlin, S., and Makse, H. A. (2013). Origins of power-law degree distribution in the heterogeneity of human activity in social networks. *Scientific reports*, 3.

Müller, H. and Freytag, J.-C. (2005). *Problems, methods, and challenges in comprehensive data cleansing*. Professoren des Inst. Für Informatik.

Nadeau, D. and Sekine, S. (2007). A survey of named entity recognition and classification. *Lingvisticae Investigationes*, 30(1):3–26.

Nadeau, D., Turney, P., and Matwin, S. (2006). Unsupervised named-entity recognition: Generating gazetteers and resolving ambiguity.

Newman, M. (2008). The physics of networks. *Physics today*, 61(11):33–38.

Newman, M. E. (2001). Scientific collaboration networks. ii. shortest paths, weighted networks, and centrality. *Physical review E*, 64(1):016132.

Newman, M. E. (2002). Assortative mixing in networks. *Physical review letters*, 89(20):208701.

Newman, M. E. (2003). The structure and function of complex networks. *SIAM review*, 45(2):167–256.

Newman, M. E. (2005). Power laws, pareto distributions and zipf's law. *Contemporary physics*, 46(5):323–351.

Nguyen, T.-V. T. and Moschitti, A. (2011). End-to-end relation extraction using distant supervision from external semantic repositories. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: short papers-Volume 2*, pages 277–282. Association for Computational Linguistics.

Nowak, M. A. (2006). Five rules for the evolution of cooperation. *science*, 314(5805):1560–1563.

Nowak, M. A. and May, R. M. (1992). Evolutionary games and spatial chaos. *Nature*, 359(6398):826–829.

Ohtsuki, H., Hauert, C., Lieberman, E., and Nowak, M. A. (2006). A simple rule for the evolution of cooperation on graphs and social networks. *Nature*, 441(7092):502–505.

Olfati-Saber, R. and Murray, R. M. (2004). Consensus problems in networks of agents with switching topology and time-delays. *Automatic Control, IEEE Transactions on*, 49(9):1520–1533.

Opsahl, T. (2011). Why anchorage is not (that) important: Binary ties and sample selection. `http://wp.me/poFcY-Vw`.

Pinheiro, F., Santos, F., and Pacheco, J. (2012). Tracking the Evolution of Cooperation in Complex Networked Populations. *Evolutionary Computation, Machine Learning and Data Mining in Bioinformatics*, pages 86–96.

Price, M. E. and van Vugt, M. (2014). The evolution of leader–follower reciprocity: the theory of service-for-prestige. *Frontiers in human neuroscience*, 8.

Purohit, H., Dow, A., Alonso, O., Duan, L., and Haas, K. (2012). User taglines: Alternative presentations of expertise and interest in social media. In *Social Informatics (SocialInformatics), 2012 International Conference on*, pages 236–243. IEEE.

Purohit, H., Dow, P. A., Duan, L., and Alonso, O. (2013). Derivation and presentation of expertise summaries and interests for users. US Patent App. 13/797,914.

Quinlan, J. R. (1986). Induction of decision trees. *Machine learning*, 1(1):81–106.

Quinlan, J. R. (2014). *C4. 5: programs for machine learning*. Elsevier.

Rahmani, H., Ranjbar-Sahraei, B., Weiss, G., and Tuyls, K. (2016). Entity resolution in disjoint graphs: an application on genealogical data. *Intelligent Data Analysis*, 20(2).

Ranjbar-Sahraei, B., Ammar, H. B., Bloembergen, D., Tuyls, K., and Weiss, G. (2014a). Theory of cooperation in complex social networks. In *Proceedings of the 25th AAAI Conference on Artificial Intelligence (AAAI-14)*.

Ranjbar-Sahraei, B., Bou-Ammar, H., Bloembergen, D., Tuyls, K., and Weiss, G. (2014b). Evolution of Cooperation in Arbitrary Complex Networks. In *13th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2014)*.

Ranjbar-Sahraei, B., Groothuis, I. M., Tuyls, K., and Weiss, G. (2014c). Valuation of cooperation and defection in small-world networks: A behavioral robotic approach. In *BNAIC 2014: Proceedings of the 26th Benelux Conference on Artificial Intelligence, Nijmegen, the Netherlands*, pages 103–110.

Ratanamahatana, C. A. and Gunopulos, D. (2003). Feature selection for the naive bayesian classifier using decision trees. *Applied Artificial Intelligence*, 17(5-6):475–487.

Ratinov, L. and Roth, D. (2009). Design challenges and misconceptions in named entity recognition. In *Proceedings of the Thirteenth Conference on Computational Natural Language Learning*, pages 147–155. Association for Computational Linguistics.

Ravichandran, D. and Hovy, E. (2002). Learning surface text patterns for a question answering system. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, pages 41–47. Association for Computational Linguistics.

Ren, W. and Beard, R. W. (2005). Consensus seeking in multiagent systems under dynamically changing interaction topologies. *Automatic Control, IEEE Transactions on*, 50(5):655–661.

Ren, W., Beard, R. W., and Kingston, D. B. (2005a). Multi-agent kalman consensus with relative uncertainty. In *American Control Conference, 2005. Proceedings of the 2005*, pages 1865–1870. IEEE.

Ren, W., Beard, R. W., and McLain, T. W. (2005b). Coordination variables and consensus building in multiple vehicle systems. In *Cooperative Control*, pages 171–188. Springer.

Ridley, M. (1994). *The red queen: Sex and the evolution of human nature*. Penguin UK.

Riedel, S., Yao, L., and McCallum, A. (2010). Modeling relations and their mentions without labeled text. In *Machine Learning and Knowledge Discovery in Databases*, pages 148–163. Springer.

Sailer, L. D. and Gaulin, S. J. C. (1981). Proximity, sociality and observation: the definition of social groups. *American Anthropologist*, 86:91–98.

Sales-Pardo, M., Guimera, R., Moreira, A. A., and Amaral, L. A. N. (2007). Extracting the hierarchical organization of complex systems. *Proceedings of the National Academy of Sciences*, 104(39):15224–15229.

Santos, F. and Pacheco, J. (2005). Scale-Free Networks Provide a Unifying Framework for the Emergence of Cooperation. *Physical Review Letters*, 95(9):1–4.

Santos, F. C., Pacheco, J. M., and Lenaerts, T. (2006). Cooperation prevails when individuals adjust their social ties. *PLoS computational biology*, 2(10):e140.

Sarawagi, S. (2008). Information extraction. *Foundations and trends in databases*, 1(3):261–377.

Schraagen, M. and Hoogeboom, H. J. (2011). Predicting record linkage potential in a family reconstruction graph. In *23th Benelux Conference on Artificial Intelligence (BNAIC2011)*, pages 199–206.

Schulz, K. U. and Mihov, S. (2002). Fast string correction with levenshtein automata. *International Journal on Document Analysis and Recognition*, 5(1):67–85.

Schutt, R. and O'Neil, C. (2013). *Doing data science: Straight talk from the frontline*. O'Reilly Media, Inc.

Shinyama, Y. and Sekine, S. (2006). Preemptive information extraction using unrestricted relation discovery. In *Proceedings of the main conference on Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics*, pages 304–311. Association for Computational Linguistics.

Shizuka, D. and McDonald, D. B. (2012). A social network perspective on measurements of dominance hierarchies. *Animal Behaviour*, 83(4):925–934.

Slotine, J.-J. E., Li, W., et al. (1991). *Applied nonlinear control*, volume 60. Prentice-Hall Englewood Cliffs, NJ.

Stibel, J. M. and Stibel, A. B. (2014). Method and system for directly targeting and blasting messages to automatically identified entities on social media. US Patent 8,762,473.

Surdeanu, M. and Ciaramita, M. (2007). Robust Information Extraction with Perceptrons. In *Proceedings of the NIST 2007 Automatic Content Extraction Workshop (ACE07)*.

Tjong Kim Sang, E. F. and De Meulder, F. (2003). Introduction to the conll-2003 shared task: Language-independent named entity recognition. In *Proceedings of the seventh conference on Natural language learning at HLT-NAACL 2003-Volume 4*, pages 142–147. Association for Computational Linguistics.

van den Heuvel, M. P. and Sporns, O. (2013). Network hubs in the human brain. *Trends in cognitive sciences*, 17(12):683–696.

van Hooff, J. A. R. A. M. and Wensing, J. A. B. (1987). Dominance and its behavioral measures in a captive wolf pack. *Chapter 11 in Harry Frank, ed., Man and Wolf. Dordrecht: Junk*, pages 219–252.

Van Segbroeck, S., de Jong, S., Nowe, A., Santos, F. C., and Lenaerts, T. (2010). Learning to coordinate in complex networks. *Adaptive Behavior*, 18(5):416–427.

van Vugt, M. and Tybur, J. M. (in press - 2016). *The Evolutionary Foundations of Hierarchy: Status, Dominance, Prestige, and Leadership*. Handbook of Evolutionary Psychology.

Vosecky, J., Hong, D., and Shen, V. Y. (2009). User identification across multiple social networks. In *Networked Digital Technologies, 2009. NDT'09. First International Conference on*, pages 360–365. IEEE.

Watts, D. J., Dodds, P. S., and Newman, M. E. (2002). Identity and search in social networus patent appks. *science*, 296(5571):1302–1305.

Watts, D. J. and Strogatz, S. H. (1998). Collective dynamics of 'small-world' networks. *nature*, 393(6684):440–442.

Wilk, G. and Włodarczyk, Z. (2013). On possible origins of power-law distributions. *arXiv preprint arXiv:1307.7855*.

Winkler, W. E. (2006). Overview of record linkage and current research directions. In *Bureau of the Census*. Citeseer.

Wolfowitz, J. (1963). Products of indecomposable, aperiodic, stochastic matrices. *Proceedings of the American Mathematical Society*, 14(5):733–737.

Xie, Y.-B., Wang, W.-X., and Wang, B.-H. (2007). Modeling the coevolution of topology and traffic on weighted technological networks. *Physical Review E*, 75(2):026111.

Yates, A., Cafarella, M., Banko, M., Etzioni, O., Broadhead, M., and Soderland, S. (2007). Textrunner: open information extraction on the web. In *Proceedings of Human Language Technologies: The Annual Conference of the North American Chapter of the Association for Computational Linguistics: Demonstrations*, pages 25–26. Association for Computational Linguistics.

Yule, G. U. (1925). A mathematical theory of evolution, based on the conclusions of dr. jc willis, frs. *Philosophical Transactions of the Royal Society of London. Series B, Containing Papers of a Biological Character*, pages 21–87.

Zelenko, D., Aone, C., and Richardella, A. (2003). Kernel methods for relation extraction. *The Journal of Machine Learning Research*, 3:1083–1106.

Zhao, S. and Grishman, R. (2005). Extracting relations with integrated information using kernel methods. In *ACL*. The Association for Computer Linguistics.

Zimmermann, M. G. and Eguíluz, V. M. (2005). Cooperation, social networks, and the emergence of leadership in a prisoner's dilemma with adaptive local interactions. *Physical Review E*, 72(5).

Publications List

2016

Bijan Ranjbar-Sahraei, Haitham Bou Ammar, Karl Tuyls, and Gerhard Weiss. "On the Prevalence of Hierarchies in Social Networks". In: Social Network Analysis and Mining Journal. 2016, 6(58).

Hossein Rahmani, Bijan Ranjbar-Sahraei, Gerhard Weiss, and Karl Tuyls. "Entity Resolution in Disjoint Graphs: an Application on Genealogical Data". In: Intelligent Data Analysis Journal. 2016, 20(2), pp. 455-475.

2015

Bijan Ranjbar-Sahraei, Julia Efremova, Hossein Rahmani, Toon Calders, Karl Tuyls, and Gerhard Weiss. "HiDER: Query-Driven Entity Resolution for Historical Data". In: Proceedings of the joint European Conference on Machine Learning and Principles and Practice of Knowledge Discovery (ECML PKDD). 2015, pp. 281-284.

Julia Efremova, Bijan Ranjbar-Sahraei, Hossein Rahmani, Frans A Oliehoek, Toon Calders, Karl Tuyls, and Gerhard Weiss. "Multi-Source Entity Resolution for Genealogical Data". In: Population Reconstruction. Springer International Publishing, 2015, pp 129-154.

Bijan Ranjbar-Sahraei, Haitham Bou Ammar, Karl Tuyls, and Gerhard Weiss. "On the Skewed Degree Distribution of Hierarchical Networks". In: Proceedings of the IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM). 2015, pp. 298-301.

Bijan Ranjbar-Sahraei, Karl Tuyls, Ipek Caliskanelli, Bastian Broeker, Daniel Claes, Sjriek Alers, and Gerhard Weiss. "Bio-inspired Multi-robot Systems". In: Biomimetic Technologies. Woodhead Publishing, 2015, pp 273 - 299.

2014

Julia Efremova, Bijan Ranjbar-Sahraei, Frans A Oliehoek, Toon Calders, and Karl Tuyls. “A Baseline Method for Genealogical Entity Resolution”. In: Population Reconstruction Workshop. 2014, pp. 129-154.

Julia Efremova, Bijan Ranjbar-Sahraei, and Toon Calders. “A Hybrid Disambiguation Measure for Inaccurate Cultural Heritage Data”. In: Proceedings of the14th Conference of the European Chapter of the Association for Computational Linguistics (EACL). 2014, pp. 47-55.

Hossein Rahmani, Bijan Ranjbar-Sahraei, Gerhard Weiss, and Karl Tuyls. “Contextual Entity Resolution Approach for Genealogical Data”. In: Proceedings of the Workshop on Knowledge Discovery, Data Mining and Machine Learning (KDML). 2014, pp. 168-179.

Bijan Ranjbar-Sahraei, Daan Bloembergen, Haitham Bou Ammar, Karl Tuyls, and Gerhard Weiss. “Effects of Evolution on the Emergence of Scale Free Networks”. In: Proceedings of the 14th International Conference on the Synthesis and Simulation of Living Systems (ALIFE). 2014, pp. 376-383.

Bijan Ranjbar-Sahraei, Haitham Bou-Ammar, Daan Bloembergen, Karl Tuyls, and Gerhard Weiss. “Evolution of Cooperation in Arbitrary Complex Networks”. In: Proceedings of the 13th International Conference on Autonomous Agents and Multiagent Systems (AAMAS). 2014, pp. 677-684.

Daan Bloembergen, Bijan Ranjbar-Sahraei, Haitham Bou Ammar, Karl Tuyls, and Gerhard Weiss. “Influencing Social Networks: An Optimal Control Study”. In: Proceedings of the 21st European Conference on Artificial Intelligence (ECAL). 2014, pp. 105-110.

Bijan Ranjbar-Sahraei, Haitham Bou-Ammar, Daan Bloembergen, Karl Tuyls, and Gerhard Weiss. “Theory of Cooperation in Complex Social Networks”. In: Proceedings of the 28th AAAI Conference on Artificial Intelligence (AAAI). 2014, pp. 1471-1477.

Bijan Ranjbar-Sahraei, Irme M. Groothuis, Karl Tuyls, and Gerhard Weiss. “Valuation of Cooperation and Defection in Small-World Networks: A Behavioral Robotic Approach”. In: Proceedings of the 26th Benelux Conference on Artificial Intelligence (BNAIC). 2014, pp. 103-110.

2013

Bijan Ranjbar-Sahraei, Gerhard Weiss, and Karl Tuyls. “A Macroscopic Model for Multi-robot Stigmergic Coverage”. In: Proceedings of the 2013 International Conference on Autonomous Agents and Multi-Agent Systems (AAMAS). 2013, pp. 1233-1234.

Julia Efremova, Bijan Ranjbar-Sahraei, Frans A Oliehoek, Toon Calders, and Karl Tuyls. “An Interactive, Web-based Tool for Genealogical Entity Resolution”. In: The 25th Benelux Conference on Artificial Intelligence (BNAIC). 2013.

Bijan Ranjbar-Sahraei, Katerina Stanková, Karl Tuyls, and Gerhard Weiss. “Stackelberg-based Coverage Approach in Nonconvex Environments”. In: Proceedings of the Advances in Artificial Life (ECAL). 2013, 12, pp. 462-469.

Bijan Ranjbar-Sahraei, Sjriek Alers, Katerina Stanková, Karl Tuyls, and Gerhard Weiss. “Toward Soft Heterogeneity in Robotic Swarms”. In: Proceedings of the 25th Benelux Conference on Artificial Intelligence (BNAIC). 2013, pp. 384-385.

Summary

The abundance of data has changed the way we conduct our personal and social life. In fact, in recent decades, we have faced an explosion of *social data*, in which unprecedented variety of personal information has become accessible to the public. Social data consists of two different data categories: First is the social data on individuals. This category captures the history of their actions, their habits, and their future plans. The second category is the social data on interactions among individuals, such as friendship interactions, professional connections and co-location observations. Once we integrate these two categories of social data, *social graphs* emerge. The importance of taking control over integration, modeling and manipulation of the social graphs has attracted interest from various research communities. As a result, much progress has been made in the analysis of social graphs. However, a faster progress has been made in data collection technologies resulting in a higher volume of data, more variety in forms of data and less veracity in the available sources.

With the rise of online social networks and widely accessible historical archives, the complexity and dynamics of social graphs have led to the following two basic research challenges. First, how to extract the structure and dynamics of social graphs form heterogeneous social data, that is, how to turn social data into social graphs. Second, how to model and analyze the properties of dynamic social graphs, that is, how to analyze evolving social graphs.

The analysis of dynamic social graphs is a common theme in this thesis. Chapter 2 provides the background knowledge on information retrieval, social graphs, and the theory of dynamical systems that are required for a better understanding of this thesis. It also introduces the datasets that are used throughout the thesis for experimental analysis and numerical verifications. The data-driven approaches proposed in Chapters 3 and 4 provide powerful practical tools for the integration of heterogenous sources of social data. These tools resulted in the automatic generation of a set of rich social graphs that could not be analyzed with existing analytical methods.

Two novel analytical models are proposed in Chapter 5 that allow for a deeper understanding of the topological properties of social graphs. A new analytical model is proposed in Chapter 6 to model the evolution of behaviors in such graphs. The proposed models successfully capture important properties of dynamic social graphs. A further research step is taken in Chapter 7: the simultaneous evolution of topology and behaviors in social graphs is studied and the effect of this co-evolution on topological diversity of the social graphs and their equilibrium state is demonstrated.

To conclude, in this thesis, effective methods for turning heterogenous social data into social graphs are developed and novel models for analysis of such dynamic social graphs are proposed. This thesis opens up at least four new promising research avenues. First, it allows for the conduction of *Advanced Genealogical Research*. Second, it paves the way to *Trace the Origins of Heavy-Tailed Distributions* in social networks using theory of dynamical systems. Third, it provides new methods for studying *Controllability of Behaviors in Social Networks*. Fourth, it establishes a bridge between *Swarm Robotics* and *Experimental Economics*.

Samenvatting

De overvloed aan data heeft ons persoonlijke en sociale leven veranderd. In de afgelopen decennia zijn we geconfronteerd met een explosie van *sociale data*, waarin een ongekende hoeveelheid aan persoonlijke informatie publiekelijk toegankelijk is geworden. Sociale data bestaan uit twee verschillende categorien. De eerste is sociale data over individuen. Deze categorie bevat hun gedrag, hun gewoonten en hun toekomstplannen. De tweede categorie beschrijft de interactie tussen individuen, zoals vriendschap, professionele relaties en ontmoetingen. Wanneer we deze twee categorien van sociale gegevens samenvoegen ontstaan *sociale grafen*. Het belang van het controleren van de integratie, modellering en manipulatie van deze sociale grafen heeft de belangstelling gewekt vanuit verschillende onderzoeksgemeenschappen. Hierdoor is veel vooruitgang geboekt in het analyseren van sociale grafen. Er is echter een nog snellere vooruitgang geboekt in data verzamelingstechnieken, wat geresulteerd heeft in een grotere hoeveelheid data, meer verschillende soorten data en minder waarheidsgetrouwheid in de beschikbare bronnen.

Met de opkomst van sociale netwerken en de breed toegankelijke historische archieven, hebben de complexiteit en dynamica van de resulterende sociale grafen geleid tot de volgende twee onderzoeksvragen. Ten eerste, hoe kan de structuur en dynamiek uit de sociale heterogene data gehaald worden en worden omgezet in grafen, oftewel hoe kan sociale data worden weergegeven als een sociale graaf. Ten tweede, hoe kunnen de eigenschappen van dynamische sociale grafen, ofwel evoluerende sociale grafen, gemodelleerd en geanalyseerd worden.

De analyse van dynamische sociale grafen is een overkoepelend thema in dit proefschrift. Hoofdstuk 2 geeft de benodigde achtergrondkennis betreffende *information retrieval*, sociale grafen en de theorie van dynamische systemen, die nodig is om dit proefschrift beter te kunnen begrijpen. Verder worden hier de datasets gentroduceerd die op verschillende plaatsen in dit proefschrift worden gebruikt voor experimentele analyses en numerieke verificatie. De data-gedreven benaderingen, voorgesteld in Hoofdstuk 3 en 4, leiden tot krachtige praktische gereedschappen voor de integratie van heterogene sociale databestanden. Deze gereedschappen resulteerden in het automatisch genereren van een set van verrijkte sociale grafen, welke met bestaande analytische methoden niet geanalyseerd konden worden.

Twee nieuwe analytische modellen worden voorgesteld in Hoofdstuk 5 die een diepgaand begrip van de topologische eigenschappen van sociale grafen mogelijk maken. In hoofdstuk 6 wordt een nieuw analytisch model gentroduceerd waarmee de evolutie

van gedrag in zulke grafen kan worden gemodelleerd. Dit model beschrijft met succes verschillende belangrijke eigenschappen van dynamiche sociale grafen. Een volgende onderzoeksstap wordt genomen in Hoofdstuk 7: de gelijktijdige evolutie van topologie en gedrag in sociale grafen wordt bestudeerd, en het effect van deze co-evolutie op de topologische diversiteit en de evenwichtstoestanden van de sociale grafen wordt aangetoond.

Concluderend worden er in dit proefschrift effectieve methoden ontwikkeld om heterogene sociale data te verwerken tot sociale grafen en worden er nieuwe modellen voorgesteld ter analyse van deze dynamische sociale grafen. Uit dit proefschrift volgen ten minste vier nieuwe, veelbelovende onderzoeksrichtingen. Ten eerste maakt dit *Geavanceerd Genealogisch Onderzoek* mogelijk. Ten tweede plaveit dit de weg voor het *Traceren van de Oorsprong van 'Heavy-Tailed' Verdelingen* in sociale netwerken door middel van dynamische-systeemtheorie. Ten derde levert het nieuwe methoden op voor het bestuderen van de *Regelbaarheid van Gedragingen in Sociale Netwerken.* Ten vierde wordt er een brug geslagen tussen de vakgebieden van de *Zwermrobotica* en de *Experimentele Economie.*

Acknowledgements

I would like to start by thanking Gerhard Weiss for his continual support during the time of my Ph.D. I am grateful for the chance to join the group and for the invaluable freedom that helped me to grow as a researcher.

I offer sincere appreciation to Karl Tuyls who supported my work and positioned my research via various strategic decisions over the past few years.

My completion of this thesis could not have been possible without the immense technical support of Haitham Bou Ammar. Apart from being a great friend since my first days in the Netherlands, Haitham acted as an informal supervisor and collaborator who helped shape my work. Haitham has also showed me a new way of life full of passion and determination.

I would like to also thank Daan Bloembergen who has been a great friend. Daan was always there in one way or another. Not only was his work vital in completion of this thesis, but he actively facilitated a smooth integration to the Netherlands. He is a real representative for Dutch politeness and support of knowledge migrants.

Kateřina Staňková has had a great impact on my research career. She has continuously helped me expand my professional network to different communities such as mathematicians, economists, and game theorists. She has introduced me to many great researchers and has helped me in finding my way for my future career. That not to mention her great technical help with my research.

My gratitude goes to Jacob van der Woude who I met during the last year of my PhD. Meeting him consisted a major breakthrough in my scientific career. Jacob showed me how to be a mathematician. He was kind-enough to listen, help, and teach.

I also thank Toon Calders who was a nice, kind and knowledgeable person. Toon and I have had many pleasant and illuminating conversations. He had a major role in steering a part of my research that constituted the first half of this thesis.

Further, I would like to thank the following friends. Sjriek Alers with his high technical knowledge which taught me a lot. Firat Ismailoglu has always been kind and available for chats. Siqi Chen was my first office mate, and was always kind and supportive. Li You was indeed a kind and nice second-office mate with whom I had a pleasant time and nice discussions. Thanks to Fredrick Schadd for all his help and support. Special thanks to Hossein Rahmani with whom we had nice scientific

discussions and his role was vital in the work provided in first half of this thesis.

In addition, Swarmlab has been a great working environment for me. Once many founding members of Swamrlab had already left, Robert Stevens joined us to carry out his internship. He was a great talented friend, always reminding me of my enthusiasm in doing practical work during my B.Sc. studies. The arrival of Jerry Spanakis at Swamrlab boosted the scientific atmosphere who was a dedicated researcher. From Jerry, I learnt how to keep my life-work balance.

Furthermore, I am thankful to Nasser Davarzani for his support during my PhD life. Also great thanks to Paras Arora, Kirill Tumanov, Stelios Asteriadis, Rico Mockel and to Evgueni Smirnov, Wei Zhao, Shuang Zhou, Zhenglong Sun and Chiara Sironi. I am also thankful to Mansoureh Jesmani, my good friend, for her pleasant and positive problem solving attitude, and Maryam Salehijam, for her efforts in proofreading parts of this thesis.

Moreover, I want to mention that Kurt Driessens has been an asset to our department. Kurt has been always cheerful, supportive and up-for doing anything cool. It has been a pleasure for us to jointly organize a few events. Ali Nakisaei, my talented friend, has supported me prior to starting my PhD. He developed the SurvAnts simulator, which was a great help in initiating my PhD research. Thanks to Amin Haghpanah for designing the cover page of this thesis.

Likewise, my special gratitude goes to Rien Wols and Anton Schuttellars from the Brabant Historical Information Centre whose efforts and patience were instrumental to the first half of this thesis. Thanks to Julia Efremova for her fruitful collaboration on MiSS project, and thanks to Anna Zseleva who introduced me to Quantitative Economics. Thanks to all staff members at DKE and the CATCH Community.

Hoda, my wife, has been the highlight of my life, since the first day of our marriage. In every aspect she was always with me, cheering me up and keeping me motivated to work harder. Over the past four years, Hoda has always supported me by adapting herself to my working conditions. I am so proud of her for working very hard to find her own life as a PhD candidate. My highest gratitude goes to my parents Mahin and Shokrollah who have always supported me since childhood. During my life abroad, they appeared infinitely patient, understanding and supportive.

Additionally, during the difficult days of making a transition, either geographically or professionally, the hospitality of Ahmad Salahnejad and Somaye Zorae and their attitude in sharing their knowledge was very helpful. Thanks to Nick Gaiko, Mohit Kumar, Adelaide Barbey, Yoeri Dijkstra, Corine Meerman, Xiaoyan Wei, Guangliang Fu, Kaihua Xi, Marco ten Eikelder, Xiaobo Zhang, Matin Hosseini, and others who made my life in Delft very pleasant.

About the Author

Bijan Ranjbar-Sahraei was born in Shiraz, Iran, on April 14th 1986. He graduated from Shiraz University with a B.Sc. in Electrical Engineering (2009) and an M.Sc. in Control Engineering (2011). He started his PhD on 1st of April 2012 at Department of Data Science and Knowledge Engineering, Maastricht University. Since then he has carried out his research in Data Mining and Social Network Analysis. As a member of an NWO funded project called MiSS (for Mining Social Structures in genealogical data) he has developed new methods for information extraction in historical archives. In particular, he has developed a knowledge discovery tool for the historical information center of North Brabant (BHIC); the software integrates historical archives of the 18^{th} and 19^{th} centuries and generates rich event timelines and informative family networks for genealogists. Also, during his PhD studies, Bijan has been engaged in a proof-oriented work on analysis of topology and dynamics of social networks. He has proposed a few novel models for a better understanding of interplay between behaviors and interactions in social networks.

So far, his scientific contributions have led to various publications in high-level journals such as Communications in Nonlinear Science and Numerical Simulation, IEEE Transactions on Industrial Electronic, Nonlinear Dynamics, and Social Network Analysis and Mining, and the high-level conferences AAMAS, AAAI, ALIFE, and ECML-PKDD. Bijan has served as associate editor for Paladyn. Journal of Behavioral Robotics, and as reviewer for ACM Transactions on Autonomous and Adaptive Systems, IEEE Transactions of Automatic Control, IET Control Theory & Applications, Robotics and Autonomous Systems, and Mathematical Problems in Engineering. During his PhD studies, he has also been involved in various teaching responsibilities for courses such as Databases, Intelligent Systems and Scientific Writing. He has supervised handful of B.Sc. and M.Sc. students, and has been the daily supervisor of industrial interns.