Learning From Networked-data: Methods and Models for Understanding Online Social Networks Dynamics

Thesis approved by
the Department of Computer Science
Technische Universität Kaiserslautern
for the award of the Doctoral Degree
Doctor of Natural Sciences (Dr. rer. nat.)
to

Mohammed Abufouda

Date of Defense: 17.07.2020
Dean: Prof. Dr. Jens Schmitt
Reviewer: Prof. Dr. Katharina A. Zweig
Reviewer: Prof. Dr. Reda Alhajj

D 386

You can't be perfect,
but if you don't try, you won't be good enough.
Paul Halmos

To Adam

Abstract

Nowadays, people and systems created by people are generating an unprecedented amount of data. This data has brought us data-driven services with a variety of applications that affect people's behavior. One of these applications is the emergent online social networks as a method for communicating with each other, getting and sharing information, looking for jobs, and many other things. However, the tremendous growth of these online social networks has also led to many new challenges that need to be addressed. In this context, the goal of this thesis is to better understand the dynamics between the members of online social networks from two perspectives. The *first* perspective is to better understand the process and the motives underlying link formation in online social networks. We utilize external information to predict whether two members of an online social network are friends or not. Also, we contribute a framework for assessing the strength of friendship ties. The *second* perspective is to better understand the decay dynamics of online social networks resulting from the inactivity of their members. Hence, we contribute a model, methods, and frameworks for understanding the decay mechanics among the members, for predicting members' inactivity, and for understanding and analyzing inactivity cascades occurring during the decay. The results of this thesis are: (1) The link formation process is at least partly driven by interactions among members that take place outside the social network itself; (2) external interactions might help reduce the noise in social networks and for ranking the strength of the ties in these networks; (3) inactivity dynamics can be modeled, predicted, and controlled using the models contributed in this thesis, which are based on network measures. The contributions and the results of this thesis can be beneficial in many respects. For example, improving the quality of a social network by introducing new meaningful links and removing noisy ones help to improve the quality of the services provided by the social network, which, e.g., enables better friend recommendations and helps to eliminate fake accounts. Moreover, understanding the decay processes involved in the interaction among the members of a social network can help to prolong the engagement of these members. This is useful in designing more resilient social networks and can assist in finding influential members whose inactivity may trigger an inactivity cascade resulting in a potential decay of a network.

Acknowledgments

This work would not have been accomplished without the help of many wonderful people whom I have been very lucky to meet. First, I want to express my deep gratitude to my supervisor Prof. Dr. Katharina Zweig (Nina) for giving me the wonderful chance to pursue my Ph.D and for the freedom in research she granted me. On the one hand, our discussions, the advices she gave me, and the trust she placed in me are invaluable. The pleasant environment she established in our workplace made it a joy to work under her supervision. On the other hand, her support, encouragement, and understanding for me as a father who took two parental leaves during my research period show how kindhearted she is.

I'd like also to thank my thesis reviewer Prof. Dr. Reda Alhajj for reviewing my thesis. His valuable comments and feedback contributed a lot to improve this thesis.

Before my Ph.D research period, I already met people who were very helpful for my journey in academia. I want to thank Prof. Dr. Lars Grunske for our discussions about research and academia during my Master studies, and for his advice to do a Ph.D. I would also like to express my sincere appreciation to Dr. Gunnar Aastrand Grimnes for the chance he gave me at DFKI during my Master studies. Furthermore, I want to thank Prof. Dr. Mohammed Awad for his support and trust he placed in me during my Bachelor studies years ago. Your encouraging words during the stressful situations we faced in Gaza were major motives for my journey and helped me a lot.

During my Ph.D, I enjoyed being surrounded by nice and smart people who made it even more joyful to be working on my Ph.D. I want to thank the following colleagues and friends for the nice chats we had over coffee breaks on floor 6 and for our conversations about many random topics: Sude Tavassoli, Mareike Bockholt, Ingrid Romani, Marsha Kleinbauer, Wolfgang Schlauch, and Sebastian Wild.

I am also full of gratitude for my friend Malik Mlitat for his support during my first days in Germany. I will always remember my first day in Kaiserslautern when he hosted me at his place.

Last but not least, I thank my wife Hadil from the bottom of my heart for all her support and encouragement during my tough moments. This work would have never reached this point without you being in my life. Thank you so much, Hadil, for making my Ph.D possible!

Contents

Listing of figures

List of Tables

List of Algorithms

1 Prologue

The rapid increase in the amount of data generated by humans and systems has pushed technology and research to be more data-driven than ever before. Thus, multidisciplinary research areas, such as data science and algorithmic accountability, have emerged in the last decade in an attempt to cope with the real new challenges caused by the availability of data and systems built mainly through learning from data. One of these disciplines is *Network Science*, which has rapidly expanded following the seminal work of Watts and Strogatz [WS98] and Barabási and Albert [BA99], with new advances in understanding complex systems by modeling their components as nodes of a graph and the interactions among these components as the edges of the graph. This abstraction of complex systems has enabled useful analysis of a complex system being studied, such as finding the most important nodes in the system (centrality analysis [KLP+05]), finding the nodes that are connected to each other more closely than to other nodes (community detection [For10]), inferring links that are not observed in the data and predicting links that may appear in the future (Link prediction [LNK03]), and many others.

In this thesis, we explore the area of online *Social Network Analysis* as the context, employ and extend *Network Science* as an abstraction tool for understanding complex systems, and apply *Machine Learning* techniques as a method for achieving the goals of this thesis. Thus, this thesis is positioned at the crossroads of these three components.

In the following sections, the context of this thesis will be defined, as well as the problems it addresses and the goals it achieves, and the contributions it will make. Additionally, the benefits of

the results obtained from this research will be provided, including the potential target audience that may benefit from these.

1.1 Context of this thesis

In recent years, technology has shaped and changed our daily lives to an extreme extent. One aspect of this change is the emergence of *Online Social Networks* (OSNs), which have become a convenient and powerful alternative for human communication, knowledge acquisition, political campaigns, jobs hunting, and many others. Thus, gaining a thorough understanding of the dynamics of the interactions among the members of OSNs is indispensable for various reasons, such as for sustaining these OSNs and their economic benefits for the owners and customers; and for understanding the societal impacts caused by the online activities of their members.

In this thesis, OSNs are approached as *complex network systems* where nodes are the members of the OSN and edges are the members' interactions within the OSN. The area of social network analysis is an active research area with researchers from different domains, various perspectives, and various challenges and problems. This thesis falls in the area of *Networked-Data Science*, as we employ probabilistic modeling and machine learning for networked data in order to better understand the dynamics of social interactions in online social networks and, correspondingly, predict online human behaviors. In this thesis, we perceive the dynamics of the members of an OSN from two perspectives. The first is link dynamics, which is the process of formation, persistence and intensity, or removal of links in a social network between its members (nodes). The second is node dynamics, which is the process of a member (node) of a social network being active or inactive over time. We assume that we are dealing with a network of a predefined set of nodes; thus, the emergence of new nodes over time is ignored and is not within the scope of this thesis. Also, the existence of external information other than friendship relations is assumed for the contributions related to link dynamics; and the existence of temporal networks for the contributions related to node dynamics.

1.2 Problems and motivations

The integral part that OSNs play in present-day life and the continuous effects they manifest pose many challenges that need to be addressed. These challenges are evident in different dimensions, such as social science and computer science. In this thesis, we are concerned with the following problems of *social network dynamics*:

P1- **Link Prediction**: Link formation [BG00, JW02] constitutes the main part of OSN link dynamics. Many studies have shown that link formation is driven by internal homophily[1] in a social network, such as the works by McPherson et al. [MSLC01], by Thelwall [The09],

[1]The etymology of the word "Homophily" is self (Homo) like (philia).

and by De Choudhury [DC11]. However, the effect of external[2] homophily on the link formation process has not been examined yet, which limits link prediction to the topological information contained only in friendship social networks. The prediction of links between members who are already part of the social network can be inferred from the structural measures of the social network itself. This approach, which is mainly used in the literature (see the literature reviews by Lü et al. [LZ11] and by Martínez et al. [MBC16]), has one limitation though: The structural measures of the social network are mainly based on the local neighborhood and the distance between each pair of nodes, which limits predicting links that may exist between two members who have no common friends and/or the distance between them is large. This limitation hinders finding links that were mainly formed because of external factors. In this thesis, external information is utilized to predict the links of an *entire* social network. This provides an opportunity to gain more (of the missing) topological network information, i.e., finding the false-negative edges. Moreover, utilizing external information helps to understand to which degree link formation in social networks is affected by a network's external factors.

P2- **Link Assessment and Tie Strength Ranking**: Social networks, similar to many other networks such as protein-protein interaction networks [Dea02], contain edges that do not reflect real relationships or represent relationships of low intensity. We call these edges *noisy edges*. The problem of link assessment is crucial because performing network analysis on networks with much noise leads to inaccurate results and uncertain conclusions. To the best of our knowledge, link assessment has not been addressed before for social networks. In the literature, there exist a few methods for tie strength ranking[3]; however, these methods (see for example the work by Jones et al. [JSB+13] and by Spiliotopoulos et al. [SPO14]) do not incorporate external information. In this thesis, a new link assessment method is proposed. The method identifies noisy edges and additionally ranks the ties in social networks based on their strength.

P3- **Inactivity modeling, analysis, and prediction**: The dynamics of social networks may contain decay processes, where nodes become inactive, triggering potential inactivity cascades. The problem of decayed social networks (social networks that turned out of service) has happened to several famous OSNs in recent years, such as MySpace and Friendster, causing a massive drop in the economic value of the decayed social networks. The reasons for the decay, its mechanics, and possible prevention mechanisms are subjects not well-studied in the literature. In this thesis, we address this problem extensively by providing a theoretical model for understanding the decay mechanics, an empirical analysis of decayed social

[2]The term "external" refers to any information about members of a social network other than their friendship links.

[3]We will stick to the term "tie strength ranking" as a term indicating ranking or weighting edges because it is used frequently in the literature.

networks, and prediction models for predicting node inactivity and some characteristics of inactivity cascades. The benefit gained from studying decay dynamics is an understanding of how the decay takes place, which enables preventing social decay and engineering resilient social networks. Additionally, understanding how the decay happens makes it possible to identify vulnerable nodes that may initiate a disruptive inactivity cascade.

1.3 Goals of this Thesis

Figure 1.3.1 shows an abstracted overview of the goals of this thesis. We assume that this abstraction is a sound representation of the life cycle of a social network over time. The following description provides more information about how we approach the social network dynamics problems described in the previous section as well as the goal of each transition from a panel to another.

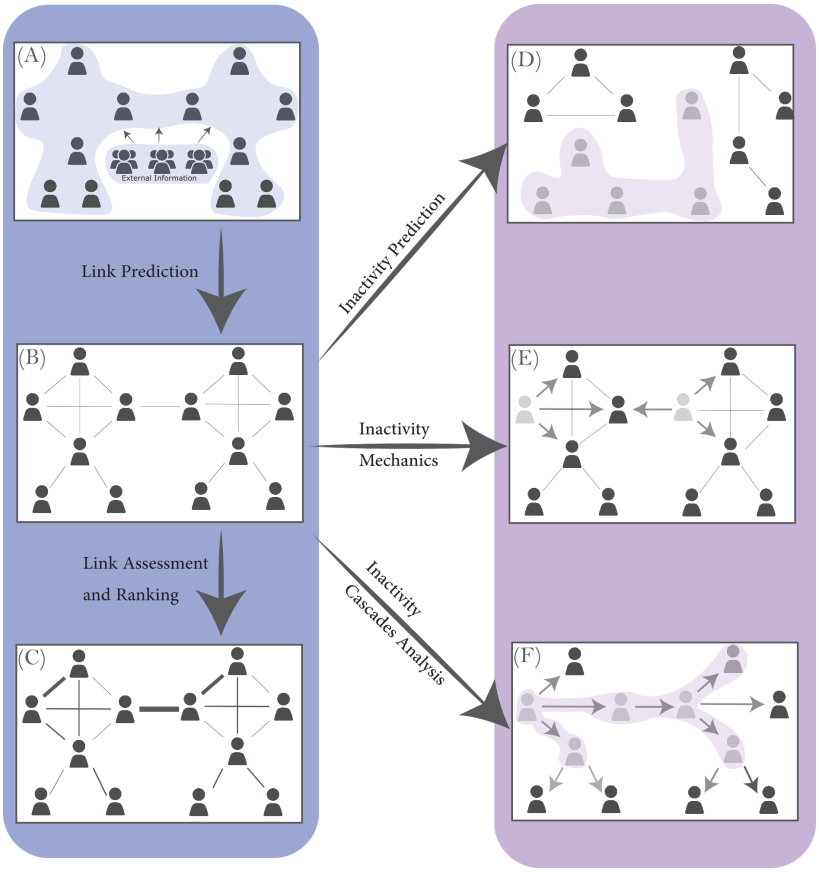


Figure 1.3.1: Schematic and abstracted overview of the goals of this thesis.

Panel A: The network in panel A shows an initial social network at the very start time of network formation, where there are members who are not connected to any other members. The panel also contains external information about the interactions among the members of the link-

less network. This information is supposed to be used later for predicting all the links of the network as well as for link assessment and tie strength ranking.

Panel B: The network in panel B shows that the links of the network are inferred. The goal of this thesis is to predict these edges using external information.

Panel C: The network in panel C shows a network with ranked edges. The goal of this thesis is to assess and rank the strength of the edges of a social network using external information.

Panel D: The network in panel D shows a network where some of the nodes are inactive; i.e., some nodes were observed in one network at one time but were not observed in one or more of the networks at a later time. The goal of this thesis is to predict these nodes and to find inactivity patterns in the underlying social networks.

Panel E: The network in panel E shows some inactive nodes and their neighbors. The goal of this thesis is to investigate how a node's inactivity affects its neighbors as a way to understand the mechanics of inactivity decay and to provide a theoretical model that captures a node's inactivity in a network.

Panel F: The network in panel F shows an inactivity cascade of some members of the network. The goal of this thesis is to formally define inactivity cascades and their properties. Additionally, we aim at predicting cascade size and cascade virality as two main characteristics of an inactivity cascade.

1.4 Contributions overview

The contributions of this thesis are categorized and described as follows:

- **Engineering:** *First:* a simple topology-based transformation of a network into a vector space. This transformation, which can be based on the network's nodes or its edges, enables employing the topological structure of a network for prediction models, especially machine learning prediction. *Second:* rigorously defined frameworks and methods for utilizing networked-data for building machine learning prediction models. The contributed frameworks and methods include the following: engineered features derived from networks, models for classification problems (for link prediction and link assessment) and regression problems (for predicting cascade's size and virality) problems, tuned parameters for improving prediction performance, and validated results. These frameworks and methods are mainly used for: (1) predicting the entire social network's links from data outside the social network itself (addressing P1); (2) assessing and ranking the strength of friendship ties (addressing P2); (3)

predicting user inactivity (addressing P3); and (4) predicting inactivity patterns and cascade (addressing P3). The contributed methods and models have been published in [AZ14, AZ15, AZ18b, Abu18a, Abu18b].

- **Theory:** *First*, a theoretical model for describing the decay dynamics of online social networks. The model is built based on probability theory and the simulation of the model reveals insights regarding the decay mechanics in online social networks that can be utilized for building and maintaining active social networks. The model partly tackles the previously stated research problem P3 and has been published in [AZ18a, AZ17]. *Second*, a theoretical prediction model for predicting users' inactivity is contributed and supported by a linear-time optimization technique. The model partly tackles research problem P3 and has been published in [Abu18a]. The results of the prediction model are satisfactory in terms of prediction accuracy measures and show many insights regarding the prediction of users' inactivity using network-based features. For example, we were able to (1) predict inactivity of members of alive social networks using information from decayed social networks, and (2) identify network-based measures that are highly correlated to members' activity/inactivity.

A considerable amount of research is related to automate software for the aforementioned two contributions. Thus, an important subaltern contribution of this work is that the code is provided publicly and can be easily used by others from interdisciplinary domains[4]. In line with this, a Python repository will be provided to facilitate the process of utilizing networks for machine learning. The code includes customizable and fully automated software code units for selecting classifiers and regressors, data preprocessing that respects the nature of networked-data, extensive parameter tuning, and validation methods and baseline comparisons.

1.5 Benefits and target audience

The work in this thesis can be beneficial to different parties. Here we list the target audiences we expect will benefit from this work and/or build on it.

- *Social network maintainers and owners*: Sustaining a successful online social network requires both continuous analysis and forecasting methods for the interactions among its members. The work of this thesis serves this goal. The contributions of this work and the insights obtained from the analysis can directly help analysts responsible for maintaining a social network. For instance, the method provided for link prediction helps to sustain the growth dynamics of a social network by increasing the number of meaningful links among the members; this, in turn, increases members' engagement in the social network, which is the ultimate goal of any social network from the owner's perspective. Furthermore, link assessment

[4]The code used in this thesis is available here: https://github.com/abufouda for reproducibility.

and tie strength ranking help to categorize friends, set privacy circles, and improve feeds. Finally, the models, insights, and results obtained from the decay dynamics part of this thesis will presumably help sustain resilient social networks that are more robust against disruptions caused by inactivity cascades. For example, realizing when a user is going to become inactive, knowing the effects of a user's inactivity on its friends, and knowing in advance when an inactivity cascade may become viral are major concerns for sustaining a growing social network.

- *Members of social networks*: Understanding how the members of a social network interact with each other is mandatory for providing purposeful services by social networks service providers. From the members' perspective, more effective friend recommendations, better online advertisements, and better content, which are some of the main aspects that members seek in online social networks, are direct applications of the findings of the link prediction and link assessment contributions of this thesis

- *Researchers*: The work of this thesis can be utilized for future analysis by researchers addressing the same problems as this work. For example, the link assessment and tie strength ranking method provided in this work can be used as a preprocessing step in subsequent network analysis. This issue and other related research problems will be addressed in more detail in Chapter 9, where we talk about future work and research directions.

- *Practitioners*: Social network analysis can be applied in different directions, for example, applications and games built using the APIs provided by social network providers, such as Facebook, on top of the social network itself. The developers of such applications can also benefit from this work by gaining a better understanding of their users from the network perspective. For example, the prediction model provided for predicting members' inactivity can be used by practitioners to know when users may become inactive so that they can take counter actions in order to prolong the presence and activity of these users.

1.6 Thesis structure

Figure 1.6.1 shows the structure of this thesis, highlighting the type of contribution (theory and/or engineering), the chapter topic, and the author's related publications.

Each chapter is briefly described in the following:

- Chapter 2 contains descriptions regarding the foundations used in this thesis, from the perspective graph theory and network science. The chapter starts with definitions of graph theory elements, followed by the foundations and definitions of network science, followed by

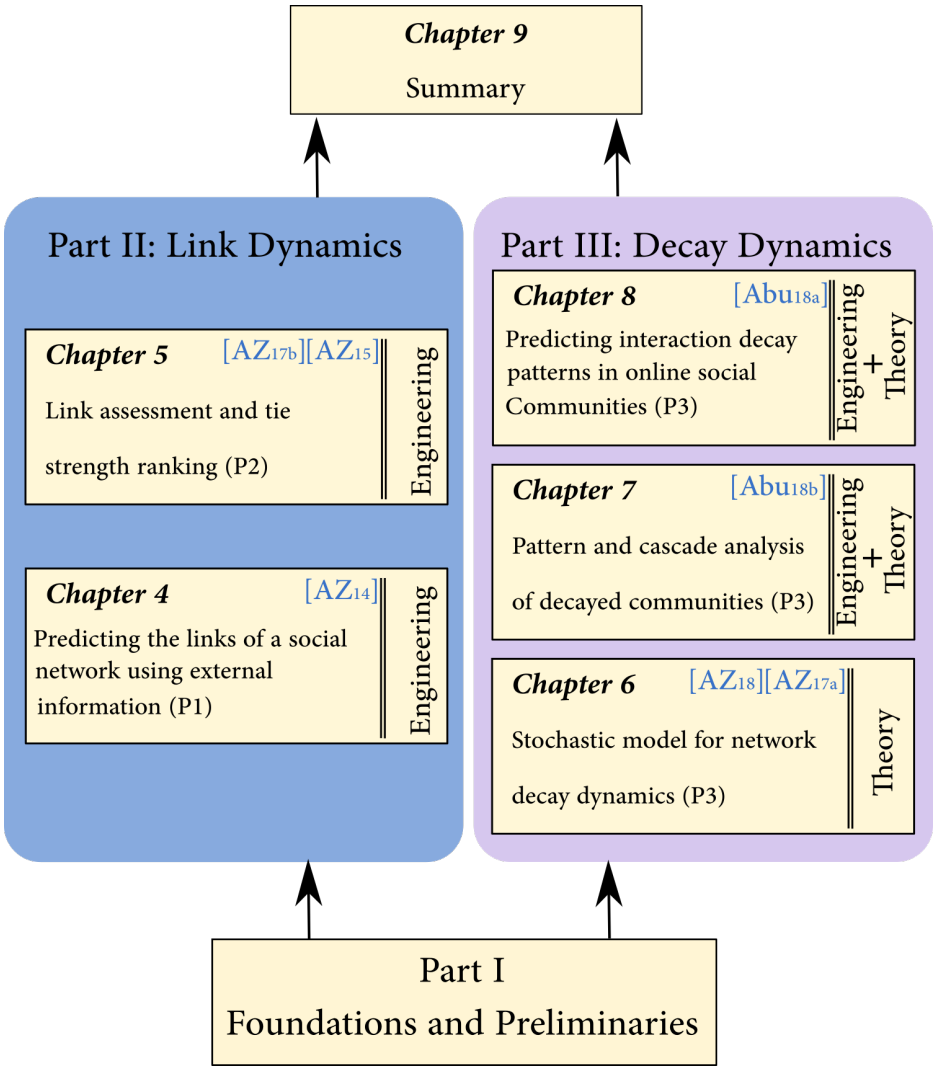


Figure 1.6.1: Thesis overview.

the transformation framework contributed in this thesis, and ends with some general characteristics of social networks. This chapter aims at providing the minimal yet sufficient details required to follow the rest of this thesis smoothly.

- Chapter 3 contains the mathematical and the formal preliminaries used and extended in this thesis. The chapter provides details about learning theory as a basis for the machine learning techniques we used, definitions of submodularity, and a glimpse of optimization theory.

- Chapter 4 presents the link prediction contribution of this thesis. The chapter includes related work, the description of the method and the datasets, as well as the results and findings.

- Chapter 5 presents the link assessment and tie strength ranking contribution of this thesis. The chapter includes related work, the motivation of the problem and its importance, the description of the method and the dataset, and ends with a summary and future work.

- Chapter 6 presents part of the theoretical work of this thesis. It contains a theoretical model for understanding the decay of social networks based on node inactivity. The properties and some theoretical findings and their implications are presented in this chapter.
- Chapter 7 presents the modeling and analysis of inactivity cascades. The chapter includes a formal definition of inactivity cascades followed by the analysis and prediction of cascade size and virality.
- Chapter 8 presents two prediction methods for predicting node inactivity in social networks. The testing and validation of the method are also presented in this chapter, which ends with conclusions and findings regarding the conducted experiments.
- Chapter 9 concludes this thesis by providing a summarization of the contributions and possible future work.

1.7 Publications

This thesis is based on the following peer-reviewed publications of the author:

1. Mohammed Abufouda and Katharina A. Zweig. Interactions around social networks matter: Predicting the social network from associated interaction networks. In 2014 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM 2014), pages 142–145, Aug 2014 [AZ14].
2. Mohammed Abufouda. Community aliveness: Discovering interaction decay patterns in online social communities. In the 4th European Network Intelligence Conference, Lecture Notes on Social Networks, Springer. Springer International Publishing, 2017 [Abu18a].
3. Mohammed Abufouda. Postmortem analysis of decayed online social communities: Cascade pattern analysis and prediction. In Complexity International Journal, 2018 [Abu18b].
4. Mohammed Abufouda and Katharina A. Zweig. Are we really friends?: Link assessment in social networks using multiple associated interaction networks. In Proceedings of the 24th International Conference on World Wide Web, WWW '15 Companion, pages 771–776, 2015, ACM [AZ15].
5. Mohammed Abufouda and Katharina A. Zweig. Stochastic modeling of the decay dynamics of online social networks. In Complex Networks VIII, Bruno Gonçalves, Ronaldo Menezes, Roberta Sinatra, and Vinko Zlatic, editors, pages 119–131. Springer International Publishing, 2017 [AZ17].

6. Mohammed Abufouda and Katharina A Zweig. A theoretical model for understanding the dynamics of online social networks decay. In arXiv preprint arXiv:1610.01538. In preparation, 2018 [AZ18a].

7. Mohammed Abufouda and Katharina Anna Zweig. Link classification and tie strength ranking in online social networks with exogenous interaction networks. In Behavioral Analytics in Social and Ubiquitous Environments, Springer, pages 1-27, 2017 [AZ18b].

Part I

Foundations and Preliminaries

2

Graphs, Networks, and Social Networks

2.1 Synopsis

In this chapter, we will present the required definitions and preliminaries of graph theory and networks that are needed to follow this thesis. This chapter also includes a description of social networks, as these are the main topic of this thesis. Furthermore, this chapter introduces one of the contributions in this thesis, which is the transformation of a network from nodes and edges nature to a vector space.

2.2 Graph theory

In 1735, Leonhard Euler came across the problem of finding a walk through the city of Königsberg via the seven bridges over the Pregolya river with one constraint: walking over each bridge only one time. The work of Euler, which provided a proof that there is no such walk solving the problem, constitutes the first known work in graph theory. Figure 2.2.1 shows part of the city of Königsberg and the bridges along with the graph representation of the problem as provided by Euler[1]. In the following subsections, we will present the basics of graph theory that are required to follow this thesis.

[1] The historical drawing of the city in the figure was taken from this link.

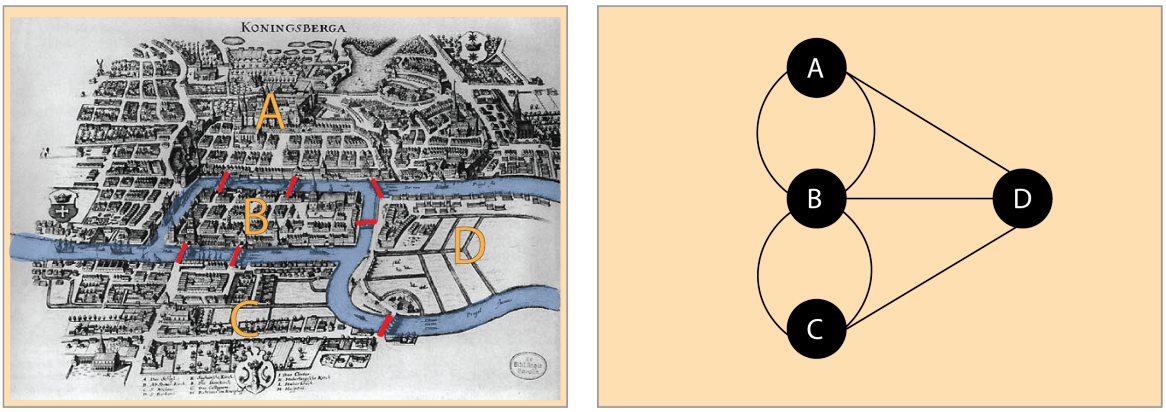


Figure 2.2.1: The figure shows (in the left panel) part of the city of Königsberg with the Pregolya river running through it and seven bridges over the river. The right panel shows the corresponding graph abstraction of the land areas around the river, depicted as nodes, and the bridges, depicted as edges.

2.2.1 What is a graph?

In this section, we will present basic definitions for graphs and their families and properties.

Definition 2.2.1. An undirected graph G is defined as a tuple $G = (V_G, E_G)$, where V_G is a finite nonempty set of nodes and E_G is the set of edges in the graph that is defined as: $E_G \subseteq V_G \times V_G$.

The number of nodes in a graph is denoted as n and the number of edges is denoted as m. An undirected edge $e = \{u, v\}$ is a connection between nodes u and v, where $u, v \in V_G$. A graph can also be directed (Digraph), in which case the order of the nodes at the endpoint of an edge is relevant. A directed graph is defined in the same way as in Definition 2.2.1. For digraphs, an edge is identified by the source node and the target node; for example, the directed edge $e_1 = (u, v)$ is not the same as the directed edge $e_2 = (v, u)$.

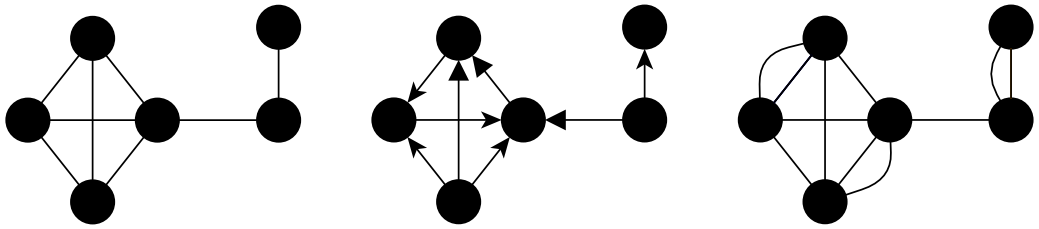
Definition 2.2.2. A *multigraph* is a graph that has multiple edges between a pair of nodes.

Definition 2.2.3. A graph is *simple* if it is undirected, has no multiple edges, and has no self-edges.

Definition 2.2.4. A *complete* graph is a graph whose nodes are pairwise connected by an edge, i.e, a complete undirected graph has $\binom{n}{2}$ edges. A complete graph of n nodes is denoted by K_n. Figure 2.2.2 shows some types of graphs.

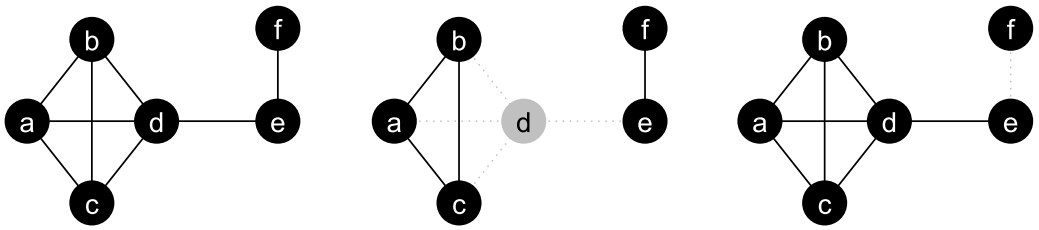
Definition 2.2.5. A *weighted* graph is a graph whose edges are mapped to weights with a mapping function $\omega : E_G \to \mathbb{R}$. Conventionally, an edge of an unweighted graph has a weight of 1.

The neighbors of a node v is the set of nodes incident to it, which is denoted by $\Gamma(v)$. The cardinality of the set $\Gamma(v)$ is called the degree of a node, $deg(v)$. It can be seen that $\sum_{v \in V_G} deg(v) = 2m$. This is called the *handshaking* lemma.



(a) A simple graph. **(b)** A directed graph. **(c)** A multi-edge graph.

Figure 2.2.2: Graph families.



(a) G **(b)** $G - \{d\}$ **(c)** $G - \{e = \{f, g\}\}$

Figure 2.2.3: A graph G, panel a, and examples of its subgraphs, panels b and c.

Definition 2.2.6. Given a graph $G = (V_G, E_G)$, $G_1 = G - v$ is a subgraph of G which is defined as $G_1 = (V_{G_1}, E_{G_1})$, where $V_{G_1} = V - \{v\}$ and $E_{G_1} \subseteq E_G$ such that E_{G_1} includes all edges of E_G except those that are incident to v.

Analogously, the subgraph can be defined by removing an edge from the graph G. Figure 2.2.3 shows a graph and two of its subgraphs.

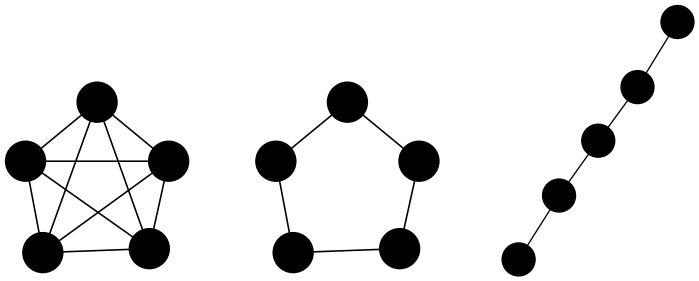
Definition 2.2.7. A component H of a disconnected graph G is a connected subgraph whose nodes are not connected to any other node in G. A connected graph contains one *component*.

Definition 2.2.8. A *cut-node* or a *cut-edge* of a graph G is a node or an edge whose removal increases the number of components of G.

2.2.2 Roaming through a graph

Definition 2.2.9. A *walk* is a sequence $(v_0, e_1, v_1, \cdots, e_k, v_k)$ of alternating nodes and edges such that $1 \leq i \leq k$, where an edge e_i is defined as $e_i = (v_{i-1}, v_i)$. The length of a walk is the number of edges it contains.

Definition 2.2.10. A *trail* is a walk without repeated edges. A *path* is a walk without repeated nodes.



(a) K_5 **(b)** C_5 **(c)** P_5

Figure 2.2.4: Five-node graph variations.

Definition 2.2.11. A *cycle* is a path where the start node and the end node are the same.

Figure 2.2.4 shows a variation of a graph with 5 nodes showing a complete graph, a cycle, and a path.

Definition 2.2.12. A *tree* is a graph with no cycles. Consequently, $m = n - 1$ for a tree.

Definition 2.2.13. For a connected graph G, the distance between two nodes u and v, $d_G(u, v)$ is defined as the number of edges in the shortest path between the two nodes.

Definition 2.2.14. The *eccentricity* of a node v is defined as the largest distance between node v and any other node in the graph. Thus, $ecc(v) = max_{u \in V_G}\{d(v, u)\}$. Additionally, the diameter of a graph, $diam_G$ is the largest eccentricity of all nodes in the graph.

2.3 From graphs to network science

The era of digitalization in which we are living and the availability of data traces generated by humans and systems has allowed graph theory to be applied on a wide scale. *Network Science* is a recent emergent field of science that basically builds on graph theory and its algorithms for modeling the interaction between the components of a complex system. This application of graphs helps to understand how these systems are working, predict the future behavior of these systems, and possibly control them. In what follows, we will describe the basics of complex network analysis, which are required to follow this thesis.

What is network science? We define networks as "*graphs in action*" within a defined context. This means that we know what a graph's node represents in reality and what an edge between two nodes represents in reality. Thus, network science builds on providing a graph abstraction of the system of interest, where the graph's nodes and edges are realized by entities and the relationships between the entities of the studied system. Thus, we know what a node and an edge actually are. Based on

that, we define network science as the set of theories, methods, models, and tools used to study a complex system by abstracting it to a network.

Interdisciplinarity and implications: The entities and the relationships of a network can be a useful representation for a vast number of systems and phenomena that are composed of interacting pieces. Thus, networks have been used for understanding different complex systems, such as the Internet [FFF99, HA99], power grids [WS98, ASBS00], transportation [Kan63, SDC+03], social networks [Mor53, Mil67, KJB+90, DYB03], the World Wide Web [AJB99], scientific collaboration [Pri65], biology (protein networks [JMBO01]), and collaboration among software developers [Sin10, AA17]. So, networks seem to be ubiquitous, which has contributed to an explosion of the literature and makes network science a very active research area in the last few years. This has enriched our understanding of the studied systems significantly.

In the following sections, we will provide a description of node and edge related measures as well as some network macroscopic measures. Then, we will present a section that describes network models[2].

2.3.1 Node-related measures

In this section, we present some measures that are related to the nodes of a network. We introduce these measures because we will use them extensively throughout this thesis. We will use these measures mainly in the network vectorization method we contributed to this thesis, which will be described in details Section 2.4. Below, we list the main measures used for a network's nodes.

Farness

Measuring the distances between a pair of nodes helps to gain insights regarding how certain processes work on top of a network. Distance measurement first appeared in the work of Shimbel [Shi53]. The farness of a node v is the summation of the distances between v and all other nodes in a network G. It is defined as:

$$\mathcal{FAR}(v) = \sum_{w \in V_G} d(v, w) \tag{2.1}$$

Closeness centrality

The closeness centrality is a measure that quantifies how close a node v is to all other nodes in a network G. Another way to calculate that is to inverse the farness of node v. Thus, the closeness of

[2]In the field of network science, the terminology is not standardized due to the interdisciplinary nature of the field. For example, a node may be called an agent, a node, a site, a member, or something else altogether, depending on the studied domain. The same applies to the terms used for an edge, which may be called an interaction, a relationship, a tie, a bond, or something else, depending on the studied domain. We will stick as much as possible to the terms node and edge.

a node is defined as:

$$\mathcal{C}(v) = (\mathcal{FAR}(v))^{-1} \tag{2.2}$$

Eigenvector centrality

The eigenvector centrality [Ruhoo] of a node is defined as:

$$Evec(x_i) = \frac{1}{\lambda} \sum_{j \in V_G} a_{ij} xj, \tag{2.3}$$

where λ is a constant and a_{ij} is a location defined by i, j in the adjacency matrix that represents the network. The measure can be written in matrix form as: $\lambda x = A \cdot x$.

Average Minimum cut

A minimum cut of two nodes u and v, $MinCut(u, v)$, is the minimum number of edges that need to be removed in order to separate the two nodes; i.e., one node will be in a subgraph that is disconnected from another subgraph containing the other node. The average minimum cut of a node v is defined as:

$$\mathcal{MC}(v) = \frac{1}{n} \sum_{u \in V_G, u \neq v} MinCut(u, v), \tag{2.4}$$

where n is the number of nodes in a graph.

Coreness

A k-core subgraph of a graph G is the maximal subgraph such that each node has at least k-edges. The *coreness* of a node $Core(v) = k$ if the node v is in the k-core subgraph and not in the $k+1$-core subgraph. Figure 2.3.1 shows an example of a graph with its k-core decomposed into subgraphs where $k \in \{1, 2, 3\}$. This measure was first introduced by Seidman [Sei83].

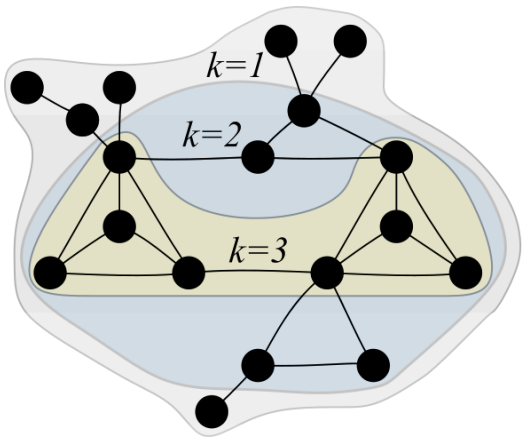
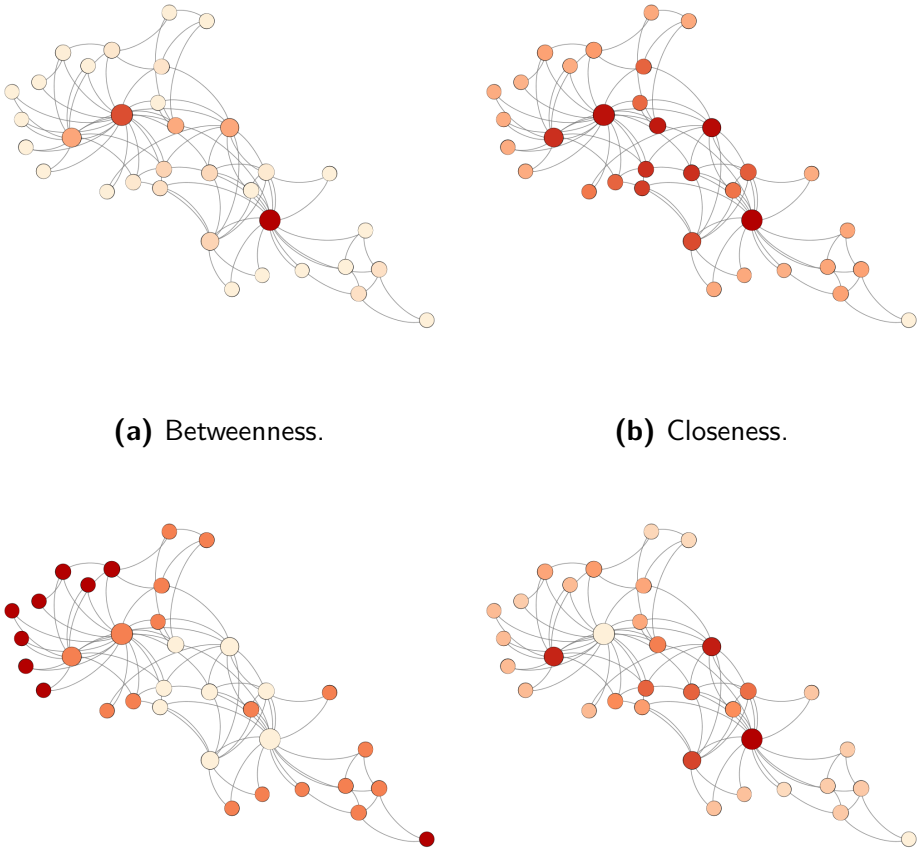


Figure 2.3.1: An example graph showing different coreness levels of a graph.



(a) Betweenness. **(b)** Closeness.

(c) Eccentricity **(d)** Eigenvector centrality

Figure 2.3.2: The figure shows different centrality measures of the Karate club dataset [Zac77]. The color is directly proportional to the measured value. The node size in all panels is directly proportional to its degree.

Betweenness centrality

The betweenness centrality measures how central a node is in terms of being in the shortest paths between other pair of nodes. It was firstly introduced by Freeman [Fre77]. Betweenness centrality is defined as:

$$\mathcal{B}(v) = \sum_{s,t \in V_G} \frac{\sigma_{st}(v)}{\sigma_{st}}, \tag{2.5}$$

where $\sigma_{st}(v)$ is the number of shortest paths between nodes s and t that includes the node v, and σ_{st} is the number of shortest paths between nodes s and t.

Figure 2.3.2 shows how the same network looks different with different lenses of centrality measures.

2.3.2 Edge-related (edge proximity) measures

In this section, we present some measures that are related to the edges of a network. We introduce these measures because we will use them extensively throughout this thesis. We will use these measures mainly in the network vectorization method we contributed to this thesis, which will be described in details Section 2.4.

Edge betweenness

Similar to the betweenness of nodes, edge betweenness measures the number of times an edge e appears in the shortest path between any two nodes in a graph [GN02]. It is defined as:

$$\mathcal{B}(e) = \sum_{v,u \in V_G} \frac{\sigma_{uv}(e)}{\sigma_{u,v}} \tag{2.6}$$

Number of Common Neighbors

For any node z in a network G, the neighbors of z, $\Gamma(z)$, are the set of nodes that are adjacent to z. For each pair of nodes v and w, the number of common neighbors of these two nodes is the number of nodes that are adjacent to both node v and node w.

$$\mathcal{CN}(v, w) = |\Gamma(v) \cap \Gamma(w)| \tag{2.7}$$

Resource Allocation

Zhou et al. [ZLZ09] proposed this measure to address link prediction and showed that it provided slightly better performance than $\mathcal{CN}$. This measure assumes that each node has some resources that will be distributed equally among its neighbors. This idea is then adapted for two nodes v and w as follows:

$$\mathcal{RA}(v, w) = \sum_{\substack{z \in \Gamma(v) \cap \Gamma(w) \\ z \neq v \neq w}} \frac{1}{|\Gamma(z)|} \tag{2.8}$$

Adamic-Adar coefficient

Ever since this measure was proposed by Adamic and Adar [AA03], the Adamic-Adar Coefficient has been used in different areas of social network analysis, such as link prediction.

$$\mathcal{AAC}(v, w) = \sum_{\substack{z \in \Gamma(v) \cap \Gamma(w) \\ z \neq v \neq w}} \frac{1}{log|\Gamma(z)|} \tag{2.9}$$

Jaccard Index

This measure was first proposed in information retrieval [SM86] as a method for quantifying the similarity between the contents of two sets. This idea is applied to the neighbors of any two nodes as follows:

$$\mathcal{JI}(v,w) = \frac{|\Gamma(v) \cap \Gamma(w)|}{|\Gamma(v) \cup \Gamma(w)|} \tag{2.10}$$

Preferential Attachment

Newman [New01] showed that in collaboration networks, the probability of collaboration between any two nodes (authors) v and w is correlated to the product of $\Gamma(v)$ and $\Gamma(w)$. We used the definition proposed by Liben-Nowell and Kleinberg [LNK03]:

$$\mathcal{PA}(v,w) = |\Gamma(v)| \cdot |\Gamma(w)| \tag{2.11}$$

Sørensen-Dice Index

This measure has been used in ecology to find the similarity between species in ecological data [Dic45] and is defined as:

$$\mathcal{SD}(v,w) = \frac{2 \times |\Gamma(v) \cap \Gamma(w)|}{|\Gamma(v)| + |\Gamma(w)|} \tag{2.12}$$

Hub Promoted Index

This measure was used to find the similarity between two nodes in a networks with hierarchical structures [RSM+02], and is defined as:

$$\mathcal{HPI}(v,w) = \frac{|\Gamma(v) \cap \Gamma(w)|}{min(|\Gamma(v)|, |\Gamma(w)|)} \tag{2.13}$$

Hub Depressed Index

Similar to $\mathcal{HPI}$, the hub depressed index is defined as:

$$\mathcal{HDI}(v,w) = \frac{|\Gamma(v) \cap \Gamma(w)|}{max(|\Gamma(v)|, |\Gamma(w)|)} \tag{2.14}$$

Local community degree measure

Measuring the similarity between two nodes can also be done by looking at how the common neighbors are connected to each other. The common neighbors measure based on the local community

degree measure defined by Cannistraci et al. [CALR13] as:

$$\mathcal{CRA}(v,w) = \sum_{\substack{z\in\Gamma(v)\cap\Gamma(w)\\ z\neq v\neq w}} \frac{|\Gamma(v)\cap\Gamma(w)\cap\Gamma(z)|}{|\Gamma(z)|} \tag{2.15}$$

2.3.3 Macroscopic measures

Density

The density of a network is the fraction of maximum possible edges that are actually observed in a network. It is defined as:

$$\rho = \frac{m}{\binom{n}{2}} = \frac{2m}{n(n-1)} \approx \frac{2m}{n^2}, \tag{2.16}$$

where $0 \leq \rho \leq 1$, and for large networks, i.e., $n \gg 1$, it can be approximated to $\frac{2m}{n^2}$.

Clustering coefficient

In this thesis we used the definition given by Barrat and Weigt [BW00]:

$$\mathcal{CC} = 3 \cdot \frac{N(C_3)}{N(P_3)}, \tag{2.17}$$

where $N(C_3)$ is the number of cycles of length three (this is called triangles), and $N(P_3)$ is the number of paths of three nodes (this is called triples).

2.3.4 Network basic models

Providing a network model that models real systems was what jump-started the field of network science. In this section, we will provide descriptions of the most famous network models in this field. Figure 2.3.3 illustrates the models that will be described in the following subsections.

Regular networks

Regular networks are networks with a deterministically defined structure of the network. Figure 2.3.3a shows a regular lattice of two dimensions. Although regular networks do not look realistic, they have some limited applications such as for modeling the magnetic interaction of particles [Isi25]. Noticeably, regular networks have been used to derive more complex and realistic networks such as the small-world model [WS98] as we will see in the section on small-world networks (Section 4).

Random graphs

Although random graphs were theoretically developed well before the emergence of network science [SR51, ER59], they have been used extensively as a reference for comparing the results ob-

tained from the analysis of real networks, which are not random, in order to find out which patterns exist in the real networks. Basically, a random graph is defined as a set of graphs where any graph G on nodes n and with m edges appears with certain probability:

$$P(G) = p^m(1-p)^{\binom{n}{2}-m}, \tag{2.18}$$

where p is the probability that each pair of nodes are incident by an edge. This probabilistic definition of a graph allows analytical analysis of random graphs. For example, we can determine the probability of having a random graph with m edges, G_m, by the following equation:

$$P(G_m) = \binom{\binom{n}{2}}{m} p^m(1-p)^{\binom{n}{2}-m} \tag{2.19}$$

Thus, the mean number of edges in a random graph model is defined as:

$$\langle m \rangle = p \cdot \binom{n}{2} \tag{2.20}$$

The degree distribution of a random network is defined as:

$$P_k = \binom{n-1}{2} p^k(1-p)^{n-1-k}, \tag{2.21}$$

where P_k is the probability that a node is connected to k other nodes.

Algorithm 2.1 shows how to generate a random graph based on the $G(n,p)$ model [ER59].

Algorithm 2.1: Generating a random graph based on the $G(n,p)$ model by [ER59].

Input: $n \geq 2, p \in [0,1]$

Init: $G = (V,E)$, where $|V| = n$ and $E = \emptyset$

1 **forall** $e = \{u,v\}$, *where* u *and* $v \in V$ **do**

2 $q \in [0,1]$

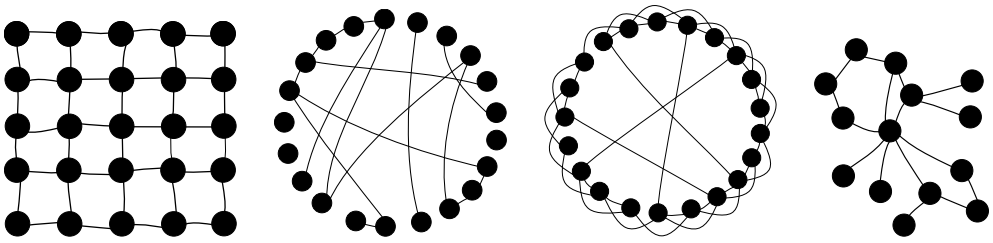
3 **if** $q \leq p$ **then**

4 $E = E \cup \{e\}$

Output: G

Small-world networks

The term small-world networks was coined by Watts and Strogatz [WS98] when they developed a model derived from a regular network by rewiring its edges. The result was a network with a small average distance between two randomly selected nodes and large clustering coefficient. This is also known as *six degrees of separation*, after the work of Milgram [Mil67].



(a) Regular network.

(b) Random network.

(c) Small world network.

(d) Scale-free network.

Figure 2.3.3: Four different network models. (a) The network in 2.3.3a shows a regular network; a 2-D lattice of 25 nodes. (b) The network in 2.3.3b shows a random network of 21 nodes and 16 edges, meaning that $p = 0.08$. The average degree of the network $\langle k \rangle = p(n-1) \approx 1.6$. (c) A small-world network based on the work of Watts-Strogatz [WS98]. The network in 2.3.3c is a modification of a regular network where each node is connected to the four other nodes in a circular manner. Then, a modification to the network is performed by rewiring some edges randomly. In the network, there are four randomly rewired edges, which shortens the average of the distance between each pair of nodes. (d) The network in 2.3.3d shows a scale-free network with 14 nodes and 16 edges with $\gamma = 2$. The network clearly contains a node (hub) with degree noticeably larger than the degrees of the other nodes.

Scale-free networks

A power-law function is defined as $f(x) = x^{\gamma}$, where γ is a constant. Thus, a power-law distribution is defined as:

$$P_k \sim k^{-\gamma} \tag{2.22}$$

A *scale-free* network is a network whose degree distribution follows a power-law function. Many real networks have been shown to have a power-law degree distribution such as the WWW network [AJB99], the Internet network [FFF99, HA99], and a metabolic network [JTA+00]. The main topological characteristic of a scale-free network is that, it contains a few nodes with significantly larger degrees than the degrees of most nodes in a network. These nodes are called *hubs*.

2.4 Network vectorizing: From network to learnable structure

In this thesis, we devise a new method that allows transforming networks from their traditional nature (nodes and edges) into another universe that is composed of vectors. This transformation enhances the applicability of networked data. For instance, using this transformation, it is easy to enable machine learning for networks. In the following, we will present how this transformation is performed. Concretely, this transformation has two types (cf. Figure 2.4.1), and they are described in the following sections.

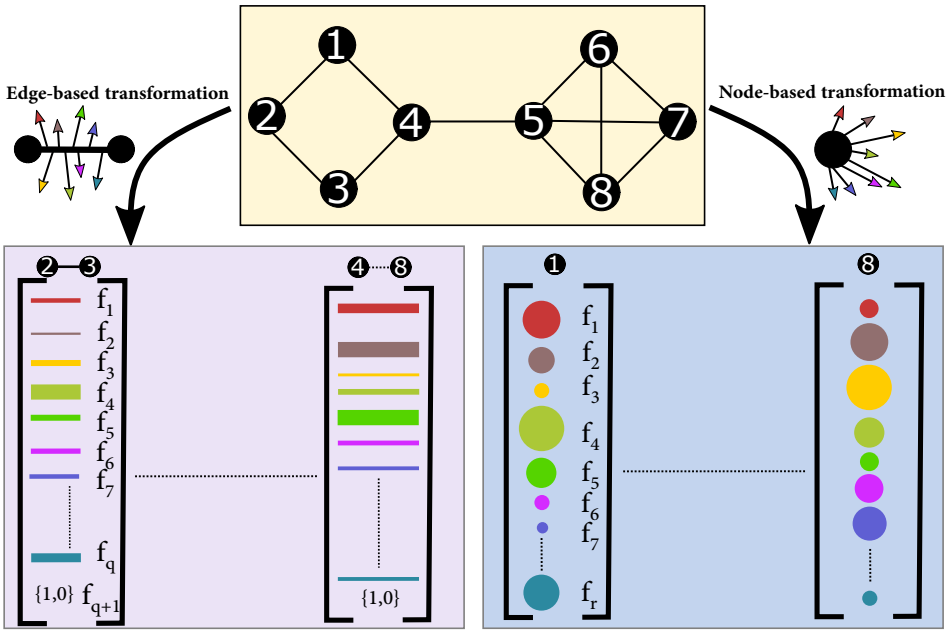


Figure 2.4.1: This illustration shows how the transformation of a network to a vector space structure is performed. There are two types of this transformation (1) Edge-based transformation: which transforms all possible edges in a network into a vector of edge-based measure values of the edge-proximity measures described in Section 2.3.2. Note that non-existing edges are also included in the representation, such as the edge $e = \{4, 8\}$. (2) Node-based transformation is similar to the previous type except that the nodes are represented as vectors.

2.4.1 Edge-based transformation:

In this type, each pair of nodes is represented as a vector of length q where each value in this vector, $f_1, \cdots, f_q$, represents a value of a measure like, but not limited to, those described in Section 2.3.2 by Equations 2.6 to 2.14. Thus, we have $\binom{n}{2}$ vectors each of length $q + 1$. That is because we have q features and an additional value that indicates whether there is a link between the two nodes or not (cf. Figure 2.4.2). Algorithm 2.2 shows how edge-based transformation is performed and how the

features data model (FDM) is generated using edge-based transformation.

Algorithm 2.2: Edge-based transformation algorithm of a graph G using the set of edge-proximity features $\mathbf{f}$.

Input: $G = (V, E), \mathbf{f} = (f_1, \cdots, f_q)$
Init: $FDM = \emptyset$
for $u, v \in V$ **do**
  $values = ()$
  for $f \in \boldsymbol{f}$ **do**
    $values = values \oplus f(u, v)$[a]
  if $e = \{u, v\} \in E$ **then**
    $values = values \oplus$ **True**
  else
    $values = values \oplus$ **False**
  $FDM = FDM \cup \{values\}$
Output: FDM

[a] $\oplus$ is an append operation on sequence, for example $(1, 2, 3) \oplus 4 = (1, 2, 3, 4)$

2.4.2 Node-based transformation:

In this type, each node is represented as a vector of length r where each value in this vector, $f_1, \cdots, f_r$, represents a value of a measure like, but not limited to, those described in Section 2.3.1 from by Equations 2.1 to 2.5. Thus, we have n vectors each of length r (cf. Figure 2.4.2).

Algorithm 2.3: Node-based transformation algorithm of a graph G using the set of node-related features $\mathbf{f}$.

Input: $G = (V, E), \mathbf{f} = (f_1, \cdots, f_r)$
Init: $FDM = \emptyset$
for $v \in V$ **do**
  $values = ()$
  for $f \in \boldsymbol{f}$ **do**
    $values = values \oplus f(v)$
  $FDM = FDM \cup \{values\}$
Output: FDM

2.4.3 Exemplary network transformation

Figure 2.4.2 shows an example on how network transformation is performed. For this example, the features $\mathcal{CN}$ (common neighbors), $\mathcal{PA}$ (preferential attachment), $\mathcal{JI}$ (Jaccard index), $\mathcal{RA}$

(resource allocation), $\mathcal{AAC}$ (Adamic-Adar coefficient), and $\mathcal{SD}$ (Sørensen-Dice Index) are used for the edge-based transformation for each pair of nodes in the graph. In addition, a True/False label is used to indicate whether there is an edge between these pair of nodes in the graph or not. The node-based transformation is also shown using the features *deg* (degree), $\mathcal{B}$ (betweenness), $\mathcal{C}$ (closeness), *Core* (coreness), and *Evec* (eigenvector centrality).

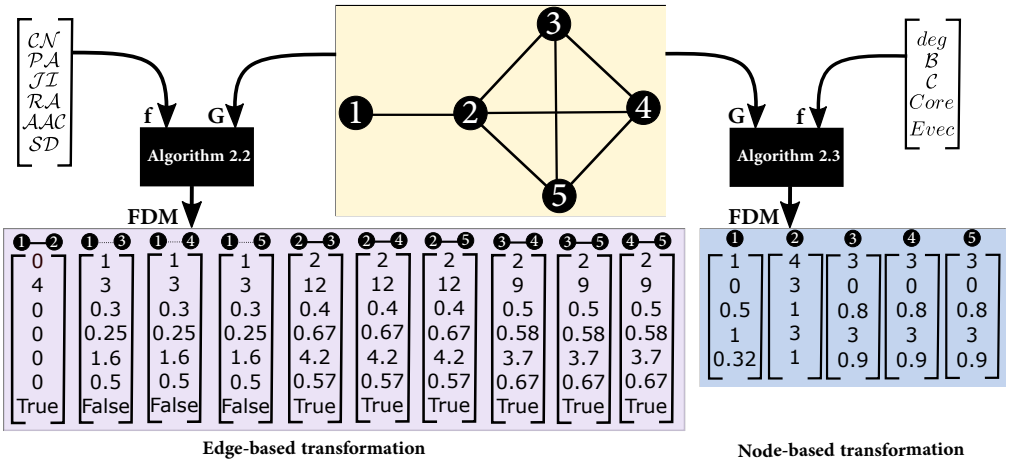


Figure 2.4.2: This illustration shows an example network and its corresponding edge-based and node-based transformations.

2.5 Social networks

Online social network (OSNs) as a term and technology is relatively new. As a result of the recent technological advances, online social networks have become a central part of today's life. The context of this thesis is about social networks. Thus, it is essential to give some background and connection to the origin of social networks from the social science perspective.

2.5.1 It began as "Sociometry"

Social networks are networks whose nodes represent humans and whose edges are the social interactions between humans in the social network. The representation of humans and their interactions as a network is not new. In Social Science, the term *sociogram* is, in essence, the same as a social network. The term sociogram first appeared in the work of Moreno and Jennings [MJ38] and Moreno [Mor53] as an adjacency matrix of a group of members studying emotions. After the emergence of network science, social networks have been studied extensively not only by social scientists but also by other scientists interested in network science. Of course, the existence of online social networks facilitates the availability of interactions between humans in online environments, making the analysis of their social interaction a popular research topic.

2.5.2 From 6 to 4 degrees of separation

In 1967, Milgram conducted an experiment to study the distance between humans in a social network [Mil67]. The experiment, which was later cloned multiple times by Milgram [TM77, KM70], was conducted as follows: Milgram asked 96 people to deliver a package to a target person. If the person who currently had the package knew the target, then the package should be sent directly to the target; otherwise, it should be sent to someone who would probably know the target. The results of this experiment showed that 18 people were able to send the package to the target successfully and that the average path length of the delivery was 5.9. This led to the coining of the term "*six degrees of separation*". The small-world network model provided by Watts and Strogatz [WS98] mimics this property.
A recent work by Backstrom et al. [BBR+12] repeated the experiment on the entirety of Facebook users. Amazingly, the results revealed that the current degree of separation between Facebook users is, on average, a little bit less than *four*.

2.5.3 Birds of a feather flock together (aka. Homophily)

One aspect in social science is the tendency of people sharing common behavior to prefer interacting with each other. This is basically the definition of *homophily* coined by Lazarsfeld et al. [LM54] in social theory and has recently been connected to social networks and the tie formation process by McPherson et al. [MSLC01]. This property has gained much attention in the area of social analysis. The availability of large datasets of online social interactions took the term to a wide range of possible experimentation. For instance, Aral et al. [AMS09] differentiate between influence and homophily in the dynamic social networks of 27 million users. It has also been used for prediction models in social media [ABS+12]. Figure 2.5.1 shows how social groups are interconnected in groups.

2.5.4 Socializing in triangles

The social behavior of humans seemingly tends to form new acquaintances. One way to do that is to get to know our friends' friends. From a social perspective, two people who have a common friend are more likely to become friends in the future. The term triadic closure means the tendency to form triangles, K_3, in our social connections. This property was first reported by Simmel [Sim08] and later by Rapoport [Rap53]. Figure 2.5.1 shows how the green edges form triangles in the network.

2.5.5 Weak ties can be strong

In social networks, ties can be strong in terms of intensity, duration, and reciprocity. On the other side, weak ties are typically formed between two people of different characteristics or attitudes, such as people separated geographically by large distance or people from different ethnicities. Although

these ties (the weak ties) are in essence weak, their implications may be strong. For example, a weak tie between two humans belonging to different social groups can be strong from many perspectives. From the network perspective, weak ties keep the entire network connected as they are a component that prevents a network from containing disconnected groups. From the social perspective, weak ties can be strong in terms of an individual gaining more information from other groups and being exposed to different cultures, and thereby gaining knowledge not available in the local group. This property was first studied by Granovetter [Gra77] who discovered in a survey that many of the people who changed their jobs found their new jobs through an acquaintance, not through a friend. Figure 2.5.1 shows how links between different groups make it possible to connect one group to another. Additionally, the small-world phenomenon can be seen in the figure; it is easy, for example, to forward a message from one node to another in just a few steps.

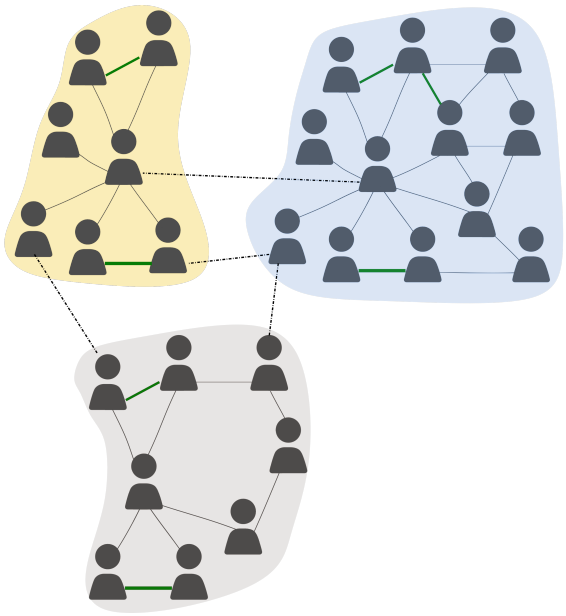


Figure 2.5.1: An example of a social network showing people connected in circles. These circles represent the homophily effect of group formation. The green links represent the triadic closure property. The dashed links between members of different groups represent weak ties.

3

Formal and Theoretical Toolkit

3.1 Synopsis

In this chapter, a formal framework for optimization techniques used for machine learning algorithms will be presented. Next, an introduction of learning theory will be provided, followed by detailed information about linear regression models and Support Vector Machines classifier. In this chapter, we will also present information about the validation and testing method that will be used in this thesis.

3.2 A Glimpse of optimization

Mathematical optimization is all about finding the extreme points (global maximum and global minimum) of a function (cf. Figure 3.2.1.). Mathematical optimization has a very large number of applications [BV04]. In this thesis, we are concerned with mathematical optimization for two reasons. First, we use machine learning in some of the methods and models we contribute to this research. Thus, it is necessary to discuss a certain level of mathematical optimization so that the reader can follow the related parts easily. Second, the model contributed in Chapter 6 has an optimization part related to submodular function optimization. Thus, this chapter includes the basics of optimizing submodular functions, which facilitates thr understanding of Chapter 6 .

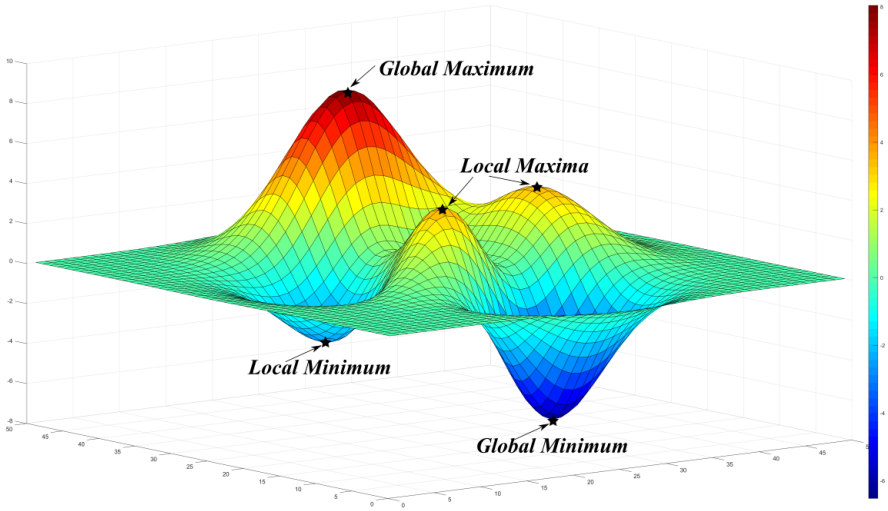


Figure 3.2.1: The figure shows an objective function of two parameters with its extreme points.

Definition 3.2.1. A constrained multivariable optimization problem is:

$$\begin{aligned} \text{minimize} \quad & f(\mathbf{w}) \\ \text{subject to} \quad & g_1(\mathbf{w}) \leq c_1 \\ & g_2(\mathbf{w}) \leq c_2 \\ & \vdots \\ & g_m(\mathbf{w}) \leq c_m, \end{aligned} \tag{3.1}$$

where the function f is the *objective function*; the function we are optimizing and it is defined as $f : \mathbb{R}^n \to \mathbb{R}$, and its n-parameters (or variables) are the elements of the set $\mathbf{w} = \{w_1, \cdots, w_n\}$. The set of m-functions $\mathbf{g} = \{g_1, \cdots, g_m\}$ is the set of *constraint functions* such that each g_i, where $i \in \{1, \cdots, m\}$, is defined as $g_i : \mathbb{R}^n \to \mathbb{R}$. If there is no constraint, i.e., if the constraint set is empty, then the optimization problem is called *unconstrained optimization*, which is typically easier to solve than a constrained optimization problem. A variable c_i, where $i \in \{1, \cdots, m\}$, is a constant boundary for the constraint functions. The definition in equation 3.1 is defined for minimization problems and can also be used for maximization problems if we minimize the negative of the objective function. The solution of the problem is $\mathbf{w}^\star$, which is the optimal value of the parameters $\mathbf{w}$.

Definition 3.2.2. A point $a^\star = (a_0, \cdots, a_n)$ is a *critical point* of the function $f(\mathbf{w})$ if $\nabla f(a_1, \cdots, a_n) = 0$.

The definition means that the partial derivative of all variables equals *zero* at any of its critical

points, i.e., $\frac{\partial}{\partial w_1} f(w_1, \cdots, w_n) = \frac{\partial}{\partial w_2} f(w_1, \cdots, w_n) = \cdots = \frac{\partial}{\partial w_n} f(w_1, \cdots, w_n) = 0$.

3.2.1 Solving unconstrained multivariable optimization problems

Usually, optimization problems are multivariable, which means that $|\mathbf{w}| > 1$. A classical way to solve unconstrained multivariable optimization problems is to solve the gradient of the objective function when it equals *zero*. The result will give only the critical point of a function, so we need to determine whether a point is a maximum, minimum, or saddle point (a point with zero gradient that is neither a maximum nor a minimum).

Definition 3.2.3. The Hessian matrix of a multivariable function is defined as:

$$\mathbf{H}_{f(\mathbf{w})} = \begin{bmatrix} \frac{\partial^2}{\partial w_1^2} f(\mathbf{w}) & \frac{\partial^2}{\partial w_1 \partial w_2} f(\mathbf{w}) & \cdots & \frac{\partial^2}{\partial w_1 \partial w_n} f(\mathbf{w}) \\ \frac{\partial^2}{\partial w_2 \partial w_1} f(\mathbf{w}) & \frac{\partial^2}{\partial w_2^2} f(\mathbf{w}) & \cdots & \frac{\partial^2}{\partial w_2 \partial w_n} f(\mathbf{w}) \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2}{\partial w_n \partial w_1} f(\mathbf{w}) & \frac{\partial^2}{\partial w_n \partial w_2} f(\mathbf{w}) & \cdots & \frac{\partial^2}{\partial w_n^2} f(\mathbf{w}) \end{bmatrix}$$

Theorem 3.1 (The second derivative test [AS64]). If $a^\star$ is a critical point of a function f, then we have:

- If $\det \mathbf{H}_{f(a^\star)} > 0$, then $a^\star$ is a local minimum.
- If $\det \mathbf{H}_{f(a^\star)} < 0$, then $a^\star$ is a local maximum.
- If $\det \mathbf{H}_{f(a^\star)}$ is undefined, then $a^\star$ is a saddle point.

3.2.2 Solving constrained multivariable optimization problems

Solving constrained multivariable optimization is more sophisticated than solving unconstrained optimization. To show how to solve constrained optimization problems with multivariables, let us modify the constraints in Equation 3.1 and have them as: $g_i(\mathbf{w}) - c_i = 0,\ \forall i \in \{1, \cdots, m\}$. As a result, the critical points that we are looking for are now restricted to some points on the surface of the constraint functions $\mathbf{g}$. If the optimization problem can be formulated like that, we can use the *Lagrange Multipliers* theorem [Ber95] to solve the problem. The following equation shows how this can be done.

$$\mathcal{L}(\mathbf{w}, \boldsymbol{\lambda}) = f(\mathbf{w}) - \sum_{i=1}^{m} \lambda_i \left(g_i(\mathbf{w}) - c_i\right), \tag{3.2}$$

where $\boldsymbol{\lambda} = \{\lambda_1, \cdots, \lambda_m\}$ are the *Lagrangian multipliers*. The main idea of the Lagrangian method is to get rid of the constraints, $g_1, \cdots, g_m$, by introducing additional variables, $\boldsymbol{\lambda}$, to a new function,

$\mathcal{L}$, which includes the original variables, $\mathbf{w}$. So, in order to find the critical points of Equation 3.2, we simply use its derivative. Thus we get:

$$\frac{\partial \mathcal{L}(\mathbf{w}, \boldsymbol{\lambda})}{\partial \mathbf{w}} = \frac{\partial f(\mathbf{w})}{\partial \mathbf{w}} - \boldsymbol{\lambda} \frac{\partial g(\mathbf{w})}{\partial \mathbf{w}} = 0. \tag{3.3}$$

Unpacking Equation 3.3, we get the following system of equations:

$$\begin{aligned} \frac{\partial \mathcal{L}}{\partial w_1} &= \frac{\partial f}{\partial w_1} - \lambda_1 \frac{\partial g_1}{\partial w_1} - \cdots - \lambda_m \frac{\partial g_m}{\partial w_1} = 0 \\ &\vdots \qquad\qquad\qquad\qquad\qquad\qquad \vdots \\ \frac{\partial \mathcal{L}}{\partial w_n} &= \frac{\partial f}{\partial w_n} - \lambda_1 \frac{\partial g_1}{\partial w_n} - \cdots - \lambda_m \frac{\partial g_m}{\partial w_n} = 0 \end{aligned} \tag{3.4}$$

Thus, we have a system of $m + n$ equations to solve $m + n$ variables .

3.2.3 Discrete optimizing

In this section, we will discuss discrete optimization of a special function type called *submodular functions*. We provide this section because the model presented in Chapter 6 is proven to be submodular with an application based on the optimization of submodular functions (cf. Figure 3.2.2 for the intuition of a submodular function).

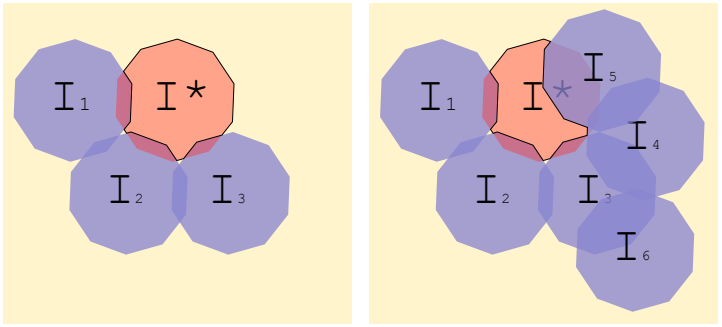
Definition 3.2.4 (Submodularity)**.** A function $f : 2^{\mathcal{V}} \to \mathbb{R}$ is said to be *submodular* over a finite ground set $\mathcal{V}$ if $f(A) + f(B) \geq f(A \cup B) + f(A \cap B),\ \forall A, B \subseteq \mathcal{V}$.

Another standard definition of a submodular function [Lov83] is:

Definition 3.2.5 (Submodularity [Cun85])**.** A function $f : 2^{\mathcal{V}} \to \mathbb{R}$ is said to be *submodular* if: $f(A \cup \{v\}) - f(A) \geq f(B \cup \{v\}) - f(B)$, where $A \subseteq B \subset \mathcal{V}$ and $v \in \mathcal{V} \setminus B$.

Definition 3.2.6 (Monotonicity)**.** A function $f : 2^{\mathcal{V}} \to \mathbb{R}$ is *monotone* if $f(A) \leq f(B),\ \forall A \subseteq B \subset \mathcal{V}$.

Maximizing a submodular function without constraints is trivial. However, the maximizing a submodular function becomes NP-hard when there is a constraint, such as the size of the set that we can use to maximize the submodular function. Thus, an approximation algorithm can be used in practice. We start first with a greedy algorithm for maximizing a submodular function as shown



(a) $A = \{I_1, I_2, I_3\}$. **(b)** $B = \{I_1, \cdots, I_6\}$.

Figure 3.2.2: A figurative example showing the intuition of a submodular function. Assume that we have two sets, A and B, as shown in panels 3.2.2a and 3.2.2b, representing the areas covered by a sensor such that $f(I_1, \cdots, I_n) = \bigcup_{i=1}^{n} area(I_i)$. Now, we want to add a new sensor $I^\star$ and let a function f represent the total area covered by a given set of sensors. In this example, we are interested in the following sets: A, B, $A \cup \{I^\star\}$, and $B \cup \{I^\star\}$. It is clear from the figure that $f(A \cup \{I^\star\}) - f(A) \geq f(B \cup \{I^\star\}) - f(B)$. That is, in essence, what a submodular vanishing return means; the red area that does not intersect with any blue area is larger in panel 3.2.2a than in panel 3.2.2b.

in Algorithm 3.1.

Algorithm 3.1: A greedy algorithm for maximizing a function f using k elements of the ground set $\mathcal{V}$.

Input: $\mathcal{V}$

Init: $S = \emptyset$

1 **for** k *iterations* **do**

2 $\quad s^\star = \underset{s \in \mathcal{V} \setminus S}{\arg\max}\ f(S \cup \{s\})$

3 $\quad S = S \cup \{s^\star\}$

Output: S

Theorem 3.2 (Nemhauser et al. [NWF78]). If f is a monotone and submodular function and S_{greedy} is the solution of the greedy algorithm 3.1, then: $f(S_{greedy}) \geq (1 - \frac{1}{e}) f(S^\star)$, where $S^\star$ is the optimal solution of size k and e is Euler's number.

Minimizing a submodular function can be performed in polynomial time [IFF01]. Although the lower bound of the maximization has not been discovered yet, the best algorithm is still not practical.

3.2.4 Optimization in practice

In many cases encountered in practice, the function that we want to optimize may not be differentiable or its closed analytical solution is extremely computationally expensive. Those two reasons make the techniques presented above in Sections 3.2.1 and 3.2.2 not viable.

The gradient descent algorithm (the very first version of this algorithm was introduced by Cauchy in 1847 [CAU47]) is an iterative algorithm for optimizing a continuous function. It starts with an arbitrary point $\mathbf{w}^0$, which represents random values of the variables of the function and then goes to other K points $\mathbf{w}^1 \rightarrow \mathbf{w}^2 \rightarrow \cdots \rightarrow \mathbf{w}^K$ such that $\mathbf{w}^k = \mathbf{w}^{k-1} + a\mathbf{d}^k$, where a is the step size (sometimes it is called learning rate) of the move and $\mathbf{d}^k$ is the direction of the move.

Algorithm 3.2: The gradient descent algorithm.

Input: $f(\mathbf{w})$, a, and K

Init: $\mathbf{w}^0$

1 **for** K *iterations* **do**

2 $\quad \mathbf{w}^k = \mathbf{w}^{k-1} - a\nabla f(\mathbf{w}^{k-1})$

Output: $\mathbf{w}$

Algorithm 3.2 shows the gradient descent algorithm, where the direction of the move is the opposite of the direction of the gradient, which makes this algorithm suitable for minimization problems. Figure 3.2.3 exemplifies the gradient descent optimization steps of two different starting points leading to different convergent stationary points.

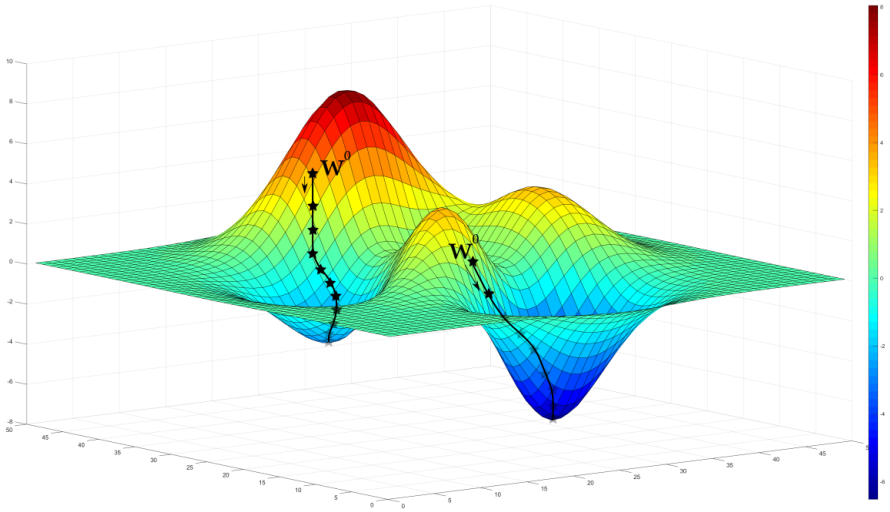


Figure 3.2.3: The figure shows two scenarios of how gradient descent works. It is evident that the global minimum is not guaranteed and depends on the initial $\mathbf{w}^0$, which is selected arbitrarily.

The performance of the gradient descent algorithm becomes computationally expensive with a larger number of points (because the gradient descent algorithm calculates the gradient of the function for all the points at each step), and also has a problem of converging to saddle points. The *Stochastic* gradient descent [RM51] algorithm is another algorithm that overcomes these two

problems; it has proven to be efficient for large datasets and for escaping from the saddle points. The stochastic gradient descent algorithm is very similar to the gradient descent algorithm with one change: we calculate the gradient of the function to be optimized for a uniformly randomly sampled subset of the data points, $|\Psi| = n$. Algorithm 3.3 shows the stochastic gradient descent algorithm.

Algorithm 3.3: The stochastic gradient descent algorithm.

Input: $f(\mathbf{w})$, α, n, and K

Init: $\mathbf{w}^0$

1 **for** K *iterations* **do**

2 $\quad \mathbf{w}^k = \mathbf{w}^{k-1} - \alpha \nabla f(\mathbf{w}^{k-1}, \Psi)$

$\quad$ // Ψ is a randomly sampled set of length n from the whole training dataset

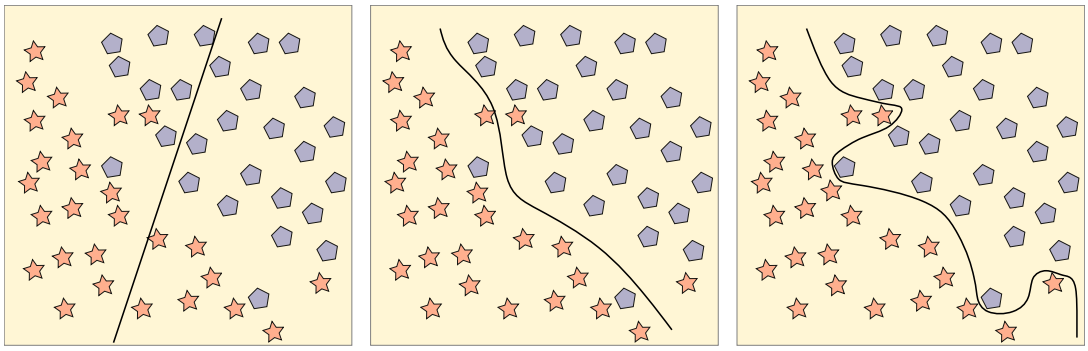
Output: $\mathbf{w}$

3.3 Learning theory

The work in this thesis applies machine learning in many places. Thus, we think it is useful to introduce some elementary basics of the theory of learning. Thus, the main goal of this section is to answer the question: *why learning is possible?*. Additionally, this section introduces two examples of machine learning algorithms and also two methods for validation that are used in the experiments of Chapter 4, Chapter 5, Chapter 7, and Chapter 8.

3.3.1 Basics of supervised learning

Suppose that we have n observed data points $\mathcal{D}$, each composed of two parts: (1) the features $\mathbf{x} = \{x_1, \cdots, x_m\}$, and (2) a *target* value y that represents a label or a value. A *supervised* machine learning algorithm selects the best possible function $\hat{f} : \mathbf{x} \to y$ such that $\hat{f} \approx f$, where $\hat{f}$ is a function that is selected by the *learning algorithm* from finitely many other function set $\mathbb{F}$ (see $\hat{f}_1$, $\hat{f}_2$, and $\hat{f}_3$ in Figure 3.3.1) and f is an unknown target function. The function $\hat{f}$ (which is also called hypothesis) is used as a prediction model for predicting the unknown target value y given the set of features the function $\hat{f}$ accepts, i.e., $\mathbf{x}$. This form of supervised machine learning is called *classification* if the target value y has finite discrete values, i.e., labels; if the number of these labels is two, then we have a binary classification problem (In Chapter 4 and Chapter 5 we will use this type of learning for link prediction and link assessment, respectively). On the other hand, if the target value y is a continuous variable, then this form of supervised machine learning is called *regression* learning (In Chapter 7 we will use this type of learning for predicting cascade virality and size).



(a) $\hat{f}_1$: Under-fitted model **(b)** $\hat{f}_2$: Good-fitted model **(c)** $\hat{f}_3$: Over-fitted model

Figure 3.3.1: The figure shows three typical types of binary classification model-fitting in a two-dimensional example. In Panel 3.3.1a, the classification model underfits the data points because it is an oversimplified linear model for linearly non-separable data, whereas in Panel 3.3.1c the model overfits the data points because every data point is modeled correctly with a complex model. The model in Panel 3.3.1b is a good fit because it classifies almost every point correctly with a simple, yet not trivial, model.

Why is learning from data attainable?

The learned function $\hat{f}$, which approximates the target function f, is deduced from a sample of the data, not from the entire population of the data (think of a function that should tell whether an image is a cat or not and how many images are required for that function to learn to identify cat images well). This means that the function $\hat{f}$ should be good enough to classify new data points that the function $\hat{f}$ has never seen before. As a result, the question now is why we believe that the function $\hat{f}$ is a good approximation of the target function? In other words, why is this function can be learned in the first place? Moreover, why should the learned function $\hat{f}$ work for data that it was not trained on? To answer these questions, we provide the following formalization of the learning problem[1].

Theorem 3.3 (Hoeffding's inequality [Hoe63]). Let μ be the mean value of a random variable and ζ the mean of its sample. Then, Hoeffding's inequality states that:
$\mathbb{P}[|\mu - \zeta| > \varepsilon] \leq 2e^{-2\varepsilon^2 n}$, where n is the length of the sample we have and ε is an error tolerance value.

In practice, Hoeffding's inequality can be used to find the bound for the in-sample prediction error (E_{in}) and the out-of-sample error (E_{out}), which can be seen as the data set we have and the whole possible space of data, respectively. Thus, the left side of Hoeffding's inequality can be written as $\mathbb{P}[|E_{out} - E_{in}| > \varepsilon]$. Based on that, theorem 3.3 provides a bound in a probabilistic sense on how many data samples we need in order to guarantee a certain deviation from the best prediction within ε.

[1] A detailed explanation on learning theory can be found in [AMMIL12, MRT18].

Definition 3.3.1 (PAC learnable function)**.** A function $\hat{f}$ is called PAC (probably approximately correct) learnable if the following inequality holds:
$\mathbb{P}[|E_{out}(\hat{f}) - E_{in}(\hat{f})| < \varepsilon] \leq 1 - \delta$, where ε is an error tolerance value and $\delta \in (0, 1]$.

Hoeffding's inequality as defined in Theorem 3.3 is applied only for one hypothesis. Thus, if we have M hypotheses, the formula becomes:

$$\mathbb{P}[|E_{out}(\mathbb{F}) - E_{in}(\mathbb{F})| > \varepsilon] \leq \sum_{m=1}^{M} 2e^{-2\varepsilon^2 n} = 2Me^{-2\varepsilon^2 n} \tag{3.5}$$

Obviously, having M on the right side of the inequality renders the bound in Equation 3.5 very loose. However, the inequality in Equation 3.5 is still PAC-learnable for multiple finite hypotheses according to the Definition 3.3.1.

In the following subsections, two models of supervised machine learning classifiers will be presented that will be used later on in this thesis[2].

3.3.2 Linear discriminator model

In this section, we present a simple linear regression model for learning from data. The linear regression target function is defined as:

$$\hat{f} = w_0 + \sum_{i=1}^{n} w_i x_i, \tag{3.6}$$

where $x_i \in \mathbf{x}$ is a feature and its weight is $w_i \in \mathbf{w}$. This definition generates an indefinite number of hypotheses (see Figure 3.3.2) and we need to select the one that has the smallest error. We define the error as a *cost function*. Using the optimization techniques introduced earlier in Section 3.2, we find the best hypothesis by finding the best weights $\mathbf{w}$. For the linear regression model, there are many cost functions. We select the *Mean Squared Error* as an example, which is defined as:

$$MSE = \frac{1}{N} \sum_{i=1}^{N} (y_i - \hat{y}_i)^2, \tag{3.7}$$

where y_i is the actual value, $\hat{y}_i = w_0 + w_i x_i$ is the predicted value, and N is the test sample size.

3.3.3 Increasing margins (aka. Support Vector Machines)

In Figure 3.3.2, the data are linearly separable and it is obvious that we can draw many lines to separate the points. However, one may ask which one of these lines is the best? Vapnik provided

[2]The goal of presenting these two models is to give an example of how a classifier is built from the bottom up with its corresponding optimization. However, these two models are not the only models used in this thesis. Other models will be used as well; and their technical description can be found here [FHT01, AMMIL12, JWHT13]

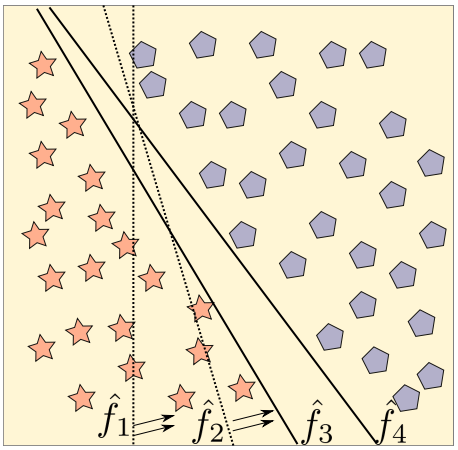


Figure 3.3.2: The figure shows how a linear model works for classifying two classes in 2D space. The learning algorithm starts with an arbitrary hypothesis, $\hat{f}_1$, and evaluates the cost function for it. Then, the learning algorithm tries to minimize the error, generating $\hat{f}_2$, and finishes with hypothesis $\hat{f}_3$ or $\hat{f}_4$ depending on the starting hypothesis and the optimization algorithm's parameters. Note that there is an infinite number of good linear (and non-linear) classifiers for this example, and selecting the best among them is doable, yet not trivial. This will be discussed in Section 3.3.3.

.

a nice solution for this problem by introducing margin maximization of the separation boundaries [CV95]. Figure 3.3.3 shows the idea of increasing the margins of the separating plane, which intuitively provides a generalizable model that performs well on the out-of-sample data points. Now, the optimization problem can be informally defined as finding the $\mathbf{w}$ that maximizes the margin.

To derive the Support Vector Machines (SVMs) optimization formula, let us first define the distance between any point $x_i \in \mathbf{x}$ and a plane $\mathbf{w}^\intercal\mathbf{x} + w_0 = 0$ as $\frac{1}{\|w\|}$. Thus, the optimization problem of the SVM becomes:

$$\begin{aligned} &\text{Maximize} && \frac{1}{\|w\|} \\ &\text{subject to} && \min_{j=1,2,\cdots,n} \quad |\mathbf{w}^\intercal x_j + w_0| = 1. \end{aligned} \tag{3.8}$$

Equation 3.8 can be simplified to a friendly form as follows.

$$\begin{aligned} &\text{Minimizing} && \frac{1}{2}\mathbf{w}^\intercal\mathbf{w} \\ &\text{subject to} && y_j(\mathbf{w}^\intercal x_j + w_0) \geq 1, \quad \textit{for } j = 1, 2, \cdots, n. \end{aligned} \tag{3.9}$$

Applying the Lagrangian method on Equation 3.9 now becomes easy, as described in Section 3.2.2, thus we get:

$$\text{Minimize} \quad \mathcal{L}(\mathbf{w}, w_0, \boldsymbol{\lambda}) = \frac{1}{2}\mathbf{w}^\intercal\mathbf{w} - \sum_{j=1}^{n} \lambda_j[y_j(\mathbf{w}^\intercal x_j + w_0) - 1] \tag{3.10}$$

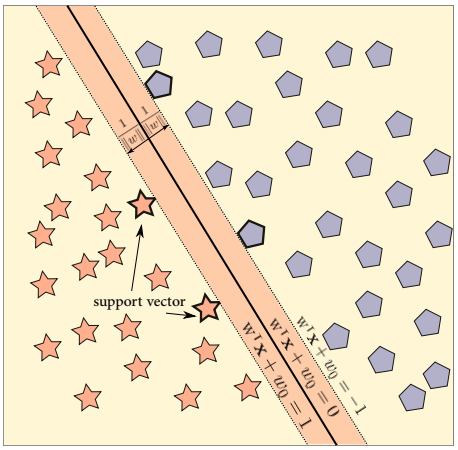


Figure 3.3.3: The figure shows the support vector points (in bold borders) and the line with the maximum margin between these support vector points from the two classes and the line itself.

Equation 3.10 can be further simplified by eliminating the variables (i.e., the duality of the equation) $\mathbf{w}$ and w_0 and replacing them with their partial derivatives, where $\frac{\partial \mathcal{L}}{\partial \mathbf{w}} = \mathbf{w} - \sum_{j=1}^{n} \lambda_j y_j x_j$ and $\frac{\partial \mathcal{L}}{\partial w_0} = -\sum_{j=1}^{n} \lambda_j y_j$. Thus, we get the final friendly-optimizable problem:

$$\begin{aligned} &\text{Minimize} \quad \mathcal{L}(\boldsymbol{\lambda}) = \sum_{j=1}^{n} \lambda_j - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} y_i y_j \lambda_i \lambda_j x_i^{\mathsf{T}} x_j \\ &\text{subject to} \quad \sum_{j=1}^{n} \lambda_j y_j = 0 \end{aligned} \tag{3.11}$$

3.3.4 Model validation and testing

Testing and validating a model with a given dataset need to be performed to obtain some guarantees regarding the generalizability of the model. In the following, we present two classical methods for model validation and testing.

Percentage split

In this method, we split the entire dataset into two disjoint sets, a learning (training) set and a testing set. Then a model is built by training on the learning set and is tested on the other set that the model has never trained on. Figure 3.3.4 illustrates this process, which ends by selecting the model with the smallest error.

k-fold cross-validation

In this method, the entire dataset is split into k subsets where the training is performed on $k-1$ subsets and the testing is performed on one subset. The following steps shows how cross validation

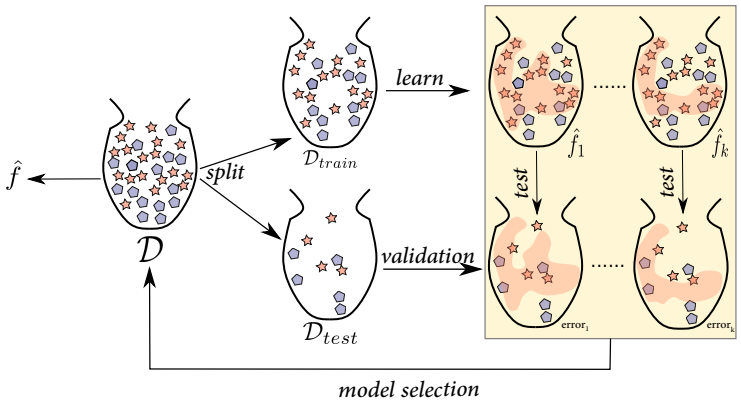


Figure 3.3.4: Validation of the learned model by splitting the data into two disjoint sets for training and validation.

is performed using k folds.

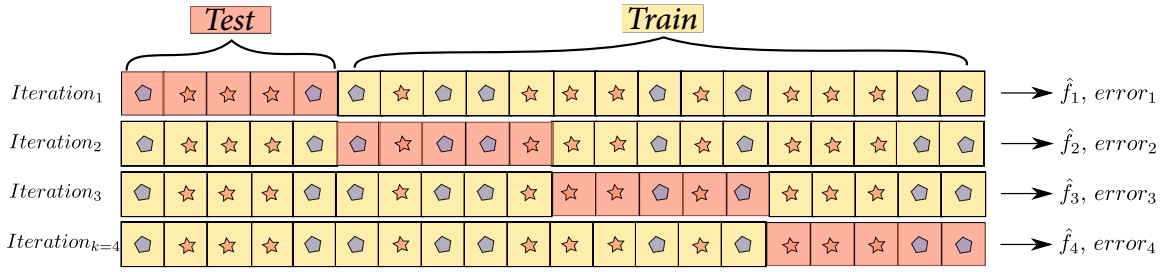


Figure 3.3.5: Model validation using k-fold cross-validation for $k = 4$. Each iteration produces a model and an error value. The overall error of the k-fold cross-validation is the average of these errors.

Step 1: Split the whole dataset into k subsets (folds) of equal length (or as close to equal).

Step 2: For $i \in \{1, \cdots, k\}$, holdout the i^{th} fold for testing, and train the model on the remaining $k - 1$ folds.

Step 3: Test each of the trained models on its corresponding holdout fold.

Step 4: The prediction error is then calculated by averaging the errors of the k models.

Part II

Link Dynamics

4

Predicting the links of a social network using external information

4.1 Synopsis

In this chapter[1], a framework for predicting all the links of a social network is presented. This framework uses the structure of associated exogenous interaction networks, which are any interactions that take place outside the friendship social network itself. The framework employs machine learning to improve the prediction performance of the links of the social network. The goal of this chapter is to provide empirical evidence that the link formation process in social networks can be driven not only by internal homophily but also by external factors, i.e., external homophily. The results obtained from the analysis conducted over multiple datasets support this hypothesis.

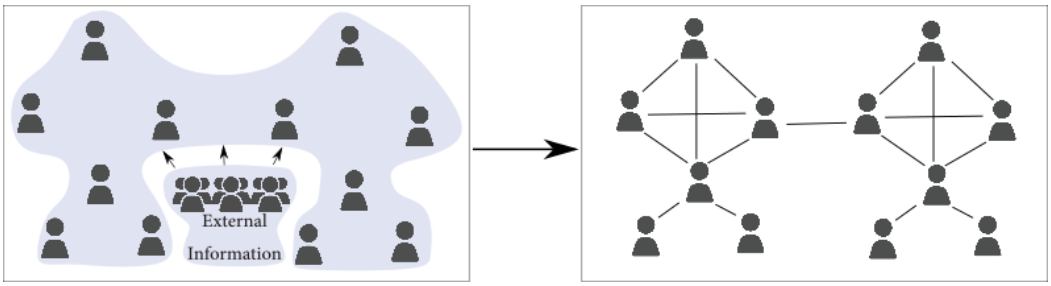


Figure 4.1.1: The goal of this chapter is to predict the links of a social network from external information.

[1]This chapter is based on the work [AZ14].

4.2 Introduction

The link formation process in social networks is voluntary and motivated mainly by personal interests. However, it is undoubtedly influenced by notions of homophily [MSLC01], which may produce relationships between persons based on external and not entirely controllable factors (cf. Section 2.5.3). Homophily is defined as the tendency to connect with similar people. This similarity is based on internal factors such as having many common friends. However, this similarity can also be influenced by external factors, such as working in the same company, sharing the same hobby, or participating in the same political party. Based on that premise, considering only the structure of a friendship social network for a group of people is not enough to understand the existing relationships among the members of this group. Also, the structure of a friendship social network alone is not enough to predict links that may be formed in the future. Thus, in order to provide a comprehensive and informative view of a social network (SN), it is important to consider all possible and available information about its members and their interactions in other contexts. These interactions are represented by additional networks $\mathcal{G}$, so-called *exogenous interaction networks*, of the same set of members of the social network SN. The term external or exogenous means that these interactions are not part of the friendship social network itself. To show that such networks $\mathcal{G}$ are informative with respect to the links of the social network they accompany, we will provide evidence that they at least partly drive the process of link formation in the SN. In this chapter, we will show that harnessing the information of the associated networks $\mathcal{G}$ makes it possible to predict the entire link structure of the SN.

4.2.1 Motivating example

Let us consider a social coding platform like *github.com*, whose members are software developers. In addition to providing the possibility to share their work, the developers can also follow each other as friends to build a social network SN. Also, several interactions may influence link formation in the SN. These exogenous interactions are represented as networks and include:

- *The collaboration development network* (g_1): The nodes of this network represent developers and a directed edge appears between two developers when one of them has committed to the other's software repository at least once.
- *The watcher network* (g_2): The nodes of this network represent developers and a directed edge appears between two developers when one of them is watching the software repository of the other developer.
- *The fork network* (g_3): The nodes of this network represent developers, and a directed edge appears between two developers when one of them forks a repository of the other developer.

- *The pull requests network* (g_4): The nodes of this network represent developers, and a directed edge appears between two developers when one of them sends a pull request to the other developer.

In order to analyze the link formation of the *SN*, we build a model that *predicts* links in a given social network *SN*. The closer the predicted link structure is to the real network's structure, the more convincing is the idea that the model captures the main motivations for link formation. So far, link prediction approaches have assumed that the information given in a social network at time t is enough to infer future link formation at a time $t' > t$. In this chapter, we test to which extent the links in the social network, e.g., the github friendship social network, can be predicted by the links found in the exogenous networks $\mathcal{G}$ described above without using any information from the social network itself. The work presented in this chapter is related to the link prediction problem initially proposed by Liben-Nowell et al. [LNK03], namely how to predict the formation of new links between actors in a time interval t based on the already existing network structure in the same social network in an earlier time interval. Here, we use the following variant of the link prediction problem: Given a set of exogenous interaction networks $\mathcal{G}$ and a social network *SN* of the same actors at any point of time t, predict the network structure of the *SN* at time t without using *any* information from the *SN* itself. This prediction is not only helpful for revealing latent links among the members of the *SN*, but also for providing information regarding the correlation between the *SN* and each network $g_i \in \mathcal{G}$.

4.3 Related work

In their seminal work, Liben-Nowell and Kleinberg [LNK03] modeled and addressed the link prediction problem in social interaction networks by providing a set of proximity measures as predictors in an unsupervised machine learning approach. The authors used different co-authorship datasets to predict future coauthor-relationships based on a set of proximity measures. These are still the main proximity measures used in later work by several researchers, particularly those employing machine learning techniques. Since the work of Liben-Nowell and Kleinberg, the area of link prediction has witnessed extensive studies in that area. However, many related works exist in the literature under different names and different contexts. Thus, link prediction in social networks has become a very active research area with many applications, leading to a plethora of papers in the area. Many surveys and reviews were conducted to discuss or compare the link prediction problem, or to provide an outlook about how to address it. These surveys and reviews include the work by Linyuan and Tao [LZ11], Al Hasan and Zaki [AHZ11], Peng et al. [WXWZ15], Yang et al. [YLC15], and Martinez et al. [MBC16]. To narrow down the related work to our contribution in this chapter, we relate to the following two link prediction categories:

4.3.1 Link prediction using only one social interaction network

In this flavor of link prediction, a prediction model uses the information only from one social interaction network without any additional information. The training and testing are performed either on one temporal snapshot of the network (percentage split or k-fold cross-validation) or on different temporal snapshots over multiple time points where training is performed on a snapshot at time point t, and testing is performed on another snapshot at $t+1$. In the following, we present the related work from the perspective of the used method in each related work.

Machine learning methods

Al Hassan et al. [HCSZ06] were among the first to apply supervised machine learning to predict links in co-authorship networks, which is still an active field of research for predicting a different kind of social relationships [BKR10, DSP11, FTL+11, BZT13]. The prediction problem was also generalized to other types of networks such as weighted and bipartite graphs. For example, Sá et al. [DSP11] proposed a supervised machine learning link prediction for weighted networks. They used a set of measures as features for a machine learning features model. Their method showed satisfactory results for both weighted and unweighted co-authorship datasets. Supervised machine learning has also been employed for the specific link prediction problem for bipartite graphs of a DBLP bibliographical dataset by Benchettara et al. [BKR10]. Mengshoel et al. [MDCT13] provide a general framework for machine learning feature model, algorithm selection, and filtering for improving prediction. They used an ACM digital library dataset to test their framework which showed that a supervised learning approach based on Decision Trees or Logistic Regression performed well.

Probabilistic and statistical methods

Link prediction has also been approached from probabilistic and statistical perspectives. Methods and models using Exponential Random Graph Models (ERGMs) [HL81] and link probability have been presented. Getoor et al. [GFKT02] provided a probabilistic relational model that describes any interaction between two relational entities from a probabilistic perspective. Their model also handles the uncertainty aspect of the interaction between the entities. Guo et al. [GHFX07] presented an extension of ERGMs that handles attributes of nodes over time to predict the structure of a network at a specific point of time. Kashima and Abe [KA06] and Wang et al. [WSP07] presented probabilistic graphical models that estimate the probability of the occurrence of a link between two nodes.

Similarly and as an alternative to ERGMs, McCulloh et al. [MLC10] provided a link probability model to generate networked data. The authors provided a comparison between their approach and the classical ERGMs revealing that ERGMs are better when dealing with a single graph at one point of time and ERGMs are explainable. On the other hand, their probabilistic model was computa-

tionally efficient and more accurate than ERGMs. Leskovec et al. [LBKT08] (and similar work by Desmarias and Cranmer [DC12]) presented a similar model using maximum-likelihood method for modeling the evolution of nodes and links in social networks.

4.3.2 Link prediction using multiple social interaction networks

In this flavor of link prediction, multiple networks or multiple attributes are used for link prediction. Popescul and Ungar [PU03] applied statistical learning methods for multiple relational data sets (like citation, co-authorship, and publish venue). Taskar et al. [TWAK04] applied relation Markov network framework, a framework they developed earlier [GFT01], to estimate the probability of links for relational data with attributes. Ahmad et al. [ABSC10] utilized information from social communication theories, namely Multi-Theoretical Multi-Level (MTML), to predict links in one network based on this network at an earlier time point and other networks, together. They used data from multi-player online games to generate three different networks and then split these networks into training and testing sets. The prediction performance was not good; in some cases, it was worse than the random classifier. Similarly, Backstrom and Leskovec [BL11] presented a supervised random walk approach for predicting which link may appear in the future using Facebook social network and node attributes. Researchers also started to use more than one relationship to predict the network structure of a complex network. For example, Lu et al. [LSTD10] used references, co-authorship, and co-citation information at one time point to predict the formation of new co-authorship-relationships at a later time point.

4.4 Contribution

The difference between our work and the work in Section 4.3.1 is that we utilize external information; thus, we use multiple networks in the prediction. The related work presented in Section 4.3.2 is similar to our work in that both are using multiple networks. However, the aforementioned works in Section 4.3.2 followed the same paradigm for link prediction, namely dividing the friendship social network into two independent temporal snapshots for training and testing. So, for each snapshot, the associated networks were used. We followed a different approach to performing link prediction with multiple networks; we train on only the interaction networks and then test on social network using our new network transformation presented in Section 2.4. Thus, we are not considering the temporality of the network. Here, we aim to identify the influence of a **single** interaction network on the social network's structure without using any information from the social network structure itself. Thus, the work in this chapter differs significantly from the related work as we predict the links of the **entire** social network, not only the structure of possible newly added links. This enables insights regarding the influence of semi-controllable interaction networks and the voluntarily built structure in a given social network.

4.5 The proposed method overview

The approach presented in this chapter is based on an extension of the classical link prediction problem defined by Liben-Nowell and Kleinberg [LNK03], where we use multi-networks (the interaction networks) as a training set to predict the links in the *SN* using machine learning classification algorithms. Thus, the structure of the social network to be predicted is never used during the training process. Figure 4.5.1 shows the general framework based on edge-based transformation, as presented in Section 2.4. In this transformation, the set of networks $\mathcal{G} = \{g_1, \cdots, g_k\}$ is used to build a set of features data models $FDM_{\mathcal{G}} = \{FDM_{g_1}, \cdots, FDM_{g_k}\}$. The algorithm used to generate each FDM in this set is Algorithm 4.1, which is basically the algorithm described in Section 2.3, but with concrete feature input. Each $FDM_{g_i} \in FDM_{\mathcal{G}}$ is composed of a set of topological feature values for each pair of nodes (v, w) from a network $g_i \in \mathcal{G}$ together with a label that represents whether there is an edge between v and w or not in g_i. Then, each of $FDM_{g_i} \in FDM_{\mathcal{G}}$ can be used to train a classifier that predicts the links of *SN*.

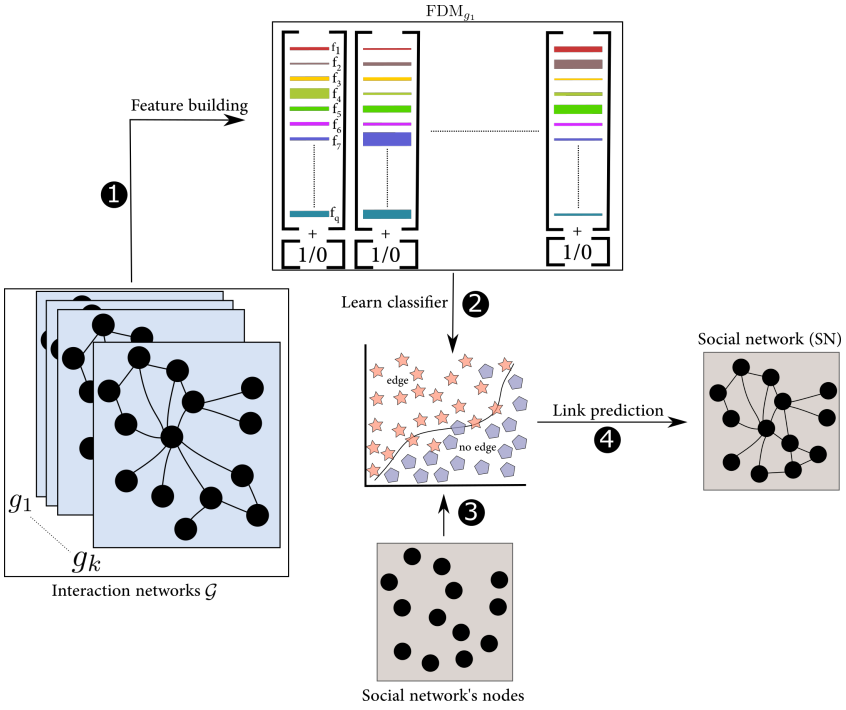


Figure 4.5.1: Overview of the prediction framework that predicts the entire social network's links using the topological structure of the interaction networks $\mathcal{G}$. Each network g_i of the interaction networks $\mathcal{G} = \{g_1, \cdots, g_k\}$ is used to build the FDM$_{g_i}$, which is a vector-based representation of each pair of nodes in g_i. This means we have $\sum_{i=1}^{k} \binom{|V_{g_i}|}{2}$ vectors for the undirected set of networks $\mathcal{G}$. Each of these vectors also contains a binary label that represents whether there is an edge between the two nodes or not.

Algorithm 4.1: Edge-based transformation algorithm of a graph G using the set of edge-proximity features $\mathbf{f}$.

Input: $G = (V, E), \mathbf{f} = (\mathcal{CN}, \mathcal{RA}, \mathcal{AAC}, \mathcal{JI}, \mathcal{PA})$
Init: $FDM = \emptyset$
for $u, v \in V$ **do**
  $values = ()$
  for $f \in \boldsymbol{f}$ **do**
    $values = values \oplus f(u, v)$
  if $e = \{u, v\} \in E$ **then**
    $values = values \oplus True$
  else
    $values = values \oplus False$
  $FDM = FDM \cup \{values\}$
Output: FDM

In general, the features of the vectors of an FDM contain any subset of edge-related measures (edge proximity measures) described in Section 2.3.2. Thus, the features data model FDM_{g_i} for a single network g_i contains $\binom{|V_{g_i}|}{2}$ vectors each with $q + 1$ values for each of the q proximity measures used and the label. Once constructed, the FDM_{g_i} can be used in a machine learning classification problem denoted by $\psi(FDM_{g_i}, SN)$, which means we use the FDM_{g_i} to train a machine learning classifier that is used to predict the links of the SN. For directed networks, two versions are used for each measure by providing two versions of the neighborhood set of node v, $\Gamma(v)$: the in-neighbors $\Gamma(v)_{in}$ and the out-neighbors $\Gamma(v)_{out}$. Based on this, an *in* and an *out* version of the proximity measures used can be constructed. For example, the in-cooccurrence for two nodes v and w is: $coocc(v, w)_{in} = |\Gamma(v)_{in} \cap \Gamma(w)_{in}|$. We emphasis here that the network SN was never used until testing. The training was performed on the interaction networks only.

4.6 Datasets and evaluation metrics

In this section, we will provide information about the datasets used to validate the method and the evaluation metrics used to evaluate the performance of the prediction.

4.6.1 Dataset description

Below, the various datasets used in our experimentation are described.

- *Research Group* [MMR13]: Includes the *Facebook* social network along with four external interaction networks built among the employees of the research group. The relations in these other networks are co-working, co-author, going out to lunch, and leisure.

- *International Internet* [BP14]: Includes three different networks for the Internet relations of 75 nations. *Hyperlinks* is a directed network such that an edge exists between two nodes (countries) if there is a website in one of these countries' domains that points to a website from a domain of the other country. We consider this network as the social network among countries. *Bandwidth* is a network among countries where edges represent the existence of an Internet connection between two countries. In the *shared website* network, an edge appears between two countries if they share at least one common most-frequently visited website. The original hyperlinks network is directed (with reciprocity of 0.92), while the other two networks are undirected. To overcome this problem, only the reciprocal edges in the original hyperlinks network are considered.

- *Terrorist Network* [Eve12]: Includes the friendship network of 79 individuals together with information on associated interaction networks like trainings performed together, meetings between them, places commonly visited by two persons, and business links.

- *Github*: A social network of software developers with a set of external interaction networks as compiled by Gousi et al. [Gou13] and described earlier in Section 4.2.

- *Brightkite* [CML11]: A location-based social network[2]. Originally, check-in is a bipartite network of actors and places where an actor can check-in to the software to let it know that he or she visited that place. We performed one-mode projection to construct the check-in network such that there is an edge between two persons if they were at the same place at least once.

- *Law Firm* [Laz12]: A social network of law firm partners with information on two other interaction networks: *co-working* and *advice seeking*.

Table 4.6.1 shows the network statistics for all of the networks we used. These statistics include the number of nodes n, the number of edges m, the clustering coefficient $\mathcal{CC}$ [WS98, BW00], and the network's density ρ.

4.6.2 Classification evaluation metrics

In this section, a set of classical classification evaluation metrics is presented. These metrics are used for evaluating the classification results of the experiment. In a binary classification scheme, only four types of results can be obtained: True Positive (TP), True Negative (TN), False Positive (FP), and False Negative (FN). Table 4.6.2 shows these variations of the prediction measures, which is also called the confusion matrix.

[2]Data is available here: http://snap.stanford.edu/data/loc-brightkite.html

Dataset	Networks	n	m	CC	ρ
Research group	*SN* **Facebook**	32	248	0.48	0.24
	g1 Work	60	194	0.34	0.1
	g2 Co-author	25	21	0.43	0.08
	g3 Lunch	60	193	0.57	0.1
	g4 Leisure	47	88	0.34	0.08
Internet	*SN* **Hyperlinks**	75	2550	0.99	0.84
	g1 Bandwidth	75	448	0.42	0.16
	g2 Shared websites	75	2360	0.92	0.86
Terrorist Network	*SN* **Friends**	61	91	0.2	0.04
	g1 Financial	13	15	0.88	0.2
	g2 Places	31	82	0.61	0.18
	g3 Business	44	458	0.75	0.48
	g4 Meeting	26	63	0.41	0.2
	g5 Training	38	147	0.72	0.2
	g6 Organization	63	416	0.84	0.22
	g7 Operations	39	267	0.78	0.36
Github (directed)	*SN* **Followers**	595232	2551900	0.13	$\approx$0
	g1 Commits	322461	909125	0.2	$\approx$0
	g2 Watchers	274597	2478561	0.02	$\approx$0
	g3 Forks	220443	673396	0.35	$\approx$0
	g4 Pull requests	156688	379207	0.08	$\approx$0
Brightkite	*SN* **Friendship**	11655	63664	0.172	$\approx$0
	g1 Check-in	13029	1378862	0.75	0.016
Law firm (directed)	*SN* **Friends**	69	339	0.43	0.09
	g1 Co-work	71	726	0.41	0.15
	g2 Advice	71	717	0.42	0.14

Table 4.6.1: Statistics of the datasets used for link prediction.

	$e \in E_{SN}$	$e \notin E_{SN}$
Predicted $e \in E_{SN}$	TP	FP
Predicted $e \notin E_{SN}$	FN	TN

Table 4.6.2: Confusion matrix of a binary classification for the link prediction problem.

Based on these basic metrics, the following evaluation metrics are used to compare the prediction results:

Precision ($\mathcal{P}$): is the ratio of TP to the number of all positive classifications. $\mathcal{P} = \frac{TP}{TP+FP}$.

Recall ($\mathcal{R}$): is also called the *true positive rate* and the *sensitivity*. $\mathcal{R} = \frac{TP}{TP+FN}$.

F1-score ($\mathcal{F}$): is the harmonic mean of precision and recall. $\mathcal{F} = \frac{2.\mathcal{P}.\mathcal{R}}{\mathcal{P}+\mathcal{R}}$.

4.6.3 Ground truth data

Let the ground truth $SN = (V, E)$ be a network with the node set V and the edge set E that contains only true-positives and true-negatives. Let $SN_{predicted} = (V, E')$ be the predicted social network on the same node set of V, with the edge set E' being the predicted edges. Accordingly, the set $E - E'$ contains the false-negative links, i.e., links that exist in reality (in E) but are not found in the $SN_{predicted}$ (in E'). Similarly, $E' - E$ contains the false-positive links, i.e., those that do not exist in SN but are found in $SN_{predicted}$. The goal is now to get a classification result that is as close to SN as possible.

4.7 Empirical results

In this section, we will present the results of the experiments, along with the first introduction of and reporting on simplistic network prediction without a supervised machine learning approach. Afterwards, we will present the results obtained when supervised machine learning was used.

4.7.1 Simplistic network prediction ($\mathcal{SP}$)

In the current settings, the first question to be answered is to which extent a single external interaction network, g_i, can predict a social network's entire structure. We call the prediction of a SN's links based on a single associated network g_i without applying machine learning *simplistic prediction*. Simplistic Prediction $\mathcal{SP}(g_i, SN)$ simply predicts that each edge in g_i also exists in the SN and that nodes not connected in g_i are not connected in SN either. Thus, TP, TN, FP, and FN are defined as follows:

1. TP is given by the number of edges contained in both networks at the same time.
2. TN is given by the number of pairs of nodes not connected by an edge in both SN and g_i.
3. FP instance means that an $e\{v, w\}$ does not exist in the SN but does exist in g_i.
4. FN instance means that an $e\{v, w\}$ exists in the SN but does not exist in g_i.

We calculate these four measures for each pair of nodes in the set $V(g_i) \cup V(SN)$. The results of this simplistic prediction are shown in Table 4.7.1. The F1-scores are surprisingly high in some cases. For example, the correlation between advice-seeking and being friends in the law firm dataset is already 0.45 and sharing lunch is also correlated with being friends in a research group by 0.51. The F1-score is very high for the financial links and the social links in the terrorist dataset (0.72).

Baseline predictor

There is always the possibility that such a result is caused merely by the number of nodes and edges in the graph. For example, if g_i and SN are both complete graphs, the "prediction" is perfect by virtue

of their structure. To exclude this possibility, i.e., to show that the correlation shown using simplistic prediction between each g_i and the SN is significant, 100 random graphs with the same number of nodes and edges as in g_i were built, using Algorithm 2.1, and used in the simple prediction approach to predict the *SN*'s links. The results are also shown in Table 4.7.1; in more than half of the cases, this prediction is worse than the simplistic prediction by at least a factor of 10. Notable exceptions are the two interaction networks of the Internet: Here, the densities are so high overall that a good prediction result is inevitable. This effect is less pronounced but still visible in both interaction networks in the law firm dataset and the business link network with respect to the terrorist social network: All of them show a rather high density to begin with and a very low number of nodes. Here, the general structure of the two networks, the respective g_i and the *SN*, seems to dictate parts of the success of the simplistic prediction approach.

4.7.2 Predicting an *SN*'s links with machine learning based on g_i

The simplistic prediction approach yielded surprisingly high congruence between interaction networks and their social network. However, the simplistic prediction does not look for patterns that may provide a better prediction. Machine learning can help by learning patterns from the interaction networks that make a link in the *SN* likely. For example, a new lawyer in the law firm might seek advice from the senior partner of the company but he never actually had the chance to meet that senior partner yet, which means that they are not friends in the friendship network. The advice network will contain such a link, but the machine learning classifier might notice that most of the individuals in the same (network) position as the new lawyer do not claim to have a friendship connection and thus the classifier will assign this claimed relation a low probability of actual existence. The classifier can, for example, learn that most edges exist between people who have many neighbors in common. If the new lawyer and the senior partner do not have any neighbors in common, the classifier will predict that this pair of nodes is not connected as friendship connection in the *SN*, despite the claim.

The method illustrated in Figure 4.5.1 is described as follows:

Step 1: For each network $g_i \in \mathcal{G}$, the corresponding FDM_{g_i} is constructed as described in Section 2.4.1 using Algorithm 4.1. The inputs to the algorithm are:

1. the interaction network g_i, and
2. the set of features: $\mathcal{CN}$ (common neighbors), $\mathcal{RA}$ (resource allocation), $\mathcal{AAC}$ (Adamic-Adar coefficient), $\mathcal{JI}$ (Jaccard-index), and the $\mathcal{PA}$ (preferential attachment). We selected these features because they were computationally not expensive, especially for a large network.

Figure 4.5.1 contains only FDM_{g_1} for readability reasons.

Step 2: We train a machine learning classification model for each FDM_{g_i}. The training set is the whole FDM_{g_i}. As the network SN may have nodes that are not in the set of nodes of g_i, the FDM is built on the set of nodes $V_{SN} \cap V_{g_i}$. The used classifier for this step is the Logistic regression, which provides a probabilistic value for each pair of nodes representing the edge existing likelihood. Then, a threshold of 0.5 is used to binarizing the output such that if the probability given by the classifier is larger than or equal 0.5, then the label is True. Otherwise, it is False.

Step 3: Using the output of the trained model from step 2, which is an in-sample prediction of the whole FDM_{gi}, we perform a simplistic prediction by testing the model's output labels on the corresponding pair of nodes in the SN in the same manner as described in Section 4.7.1. Clearly, the structure of the *SN* is never used in the training process; hence, no validation process is required (like percentage split).

Step 4: The results of the prediction from step 3 are binary values associated to each pair of nodes in the *SN*, i.e., we predict whether there is a link between any pair of nodes of *SN* or not. As we already know the ground truth edges of the *SN*, we can now calculate the prediction evaluation metrics.

Based on that, we will test the following claim:

Claim: The links of an *SN* can be more effectively predicted using a machine learning classifier learned only from the structure of g_i than the simplistic prediction.

Here, we use the simplistic prediction result as a good baseline for the prediction performed by machine learning. Thus, we formulate the prediction problem as $\psi(\text{FDM}_{g_i}, SN)$ which mean that we used FDM_{g_i} as a training set to learn a classifier. Then, we test this classifier on the test set SN, as described in Section 4.5. We used Weka [3] machine learning tool version 3.7 for the training.

[3] https://www.cs.waikato.ac.nz/ml/weka/

The used algorithm was Weka's implementation of logistic regression with its default values, i.e., no parameter tuning was performed. We selected logistic regression without parameter tuning for simplicity, later in the next chapter, we will present more information about different classifiers, parameter selection, and classifiers decision boundaries.

Table 4.7.1 shows the quality of this prediction quantified by the evaluation metrics described earlier. Overall, the quality of the prediction is significantly high, compared to the simplistic prediction, knowing that the SN was never exposed during the training process. This supports our claim that machine learning can uncover more link patterns in the SN. It is evident that a prediction $\psi(FDM_{g_i}, SN)$ model is more effective than the simplistic prediction $\mathcal{SP}(g_i, SN)$ performed in Section 4.7.1: In no case is the prediction using machine learning of the *SN* worse than the simplistic prediction. However, the increase in quality varies strongly: The prediction of the social links between terrorists based on their business links does not improve by using a machine learning approach. The largest improvement is seen in the co-author network in the Research Group dataset. The best prediction with the machine learning approach is achieved for the co-working relationship between lawyers: the simplistic prediction achieves an F-score of 0.54 compared to 0.76 achieved by the machine learning approach.

Dataset	Interaction Network	$\mathcal{SP}_{random}$ $\mathcal{F}$	$\mathcal{SP}(g_i, SN)$ $\mathcal{F}$	$\psi(\text{FDM}_{g_i}, SN)$ $\mathcal{F}$
Research group	Work	0.021	**0.52**	0.53
	Co-author	≈0	0.472	**0.72**
	Lunch	0.029	0.51	0.63
	Leisure	0.03	0.46	0.67
Internet	Bandwidth	0.27	0.28	0.35
	Shared website	0.98	**0.84**	**0.9**
Terrorist Networks	Financial	0.16	**0.72**	**0.76**
	Places	0.07	0.35	0.55
	Business	0.042	0.13	0.13
	Meeting	≈0	0.62	0.69
	Training	0.039	0.38	0.6
	Organization	0.03	0.198	0.42
	Operations	0.039	0.275	0.35
Github	Commits	≈0	0.1	**0.25**
	Watchers	≈0	0.1	0.16
	Forks	≈0	**0.15**	0.18
	Pull requests	≈0	0.02	0.13
Brightkite	Check-in	≈0	0.3	0.42
Law firm	Co-worker	0.13	**0.54**	**0.76**
	Advice	0.1	0.45	0.63

Table 4.7.1: F-score of different types of prediction.

4.8 Conclusion

In this chapter, we have shown that the link formation process in a social network cannot only be predicted from the social network itself but that the whole structure of a social network can be satisfactorily predicted from other external interaction networks. This high correlation between interaction networks and social networks does not tell us the direction of causality. However, it is clear that links in the social network are largely voluntary: Nobody is forced to be another person's friend (although some cultural pressure might apply). Some of the interaction networks are not fully controllable by the actors of the social network. For example, co-working structures are often determined by the hierarchy of the company or by the sheer necessity of having people from different departments in a project team. If such a non- or semi-controllable interaction network shows a large similarity with the associated social network structure, this indicates that part of the social link formation is not so much guided by internal homophily but rather by external homophily: We are highly likely to be friends with those with whom we spend a lot of time; whether by choice or dictated by circumstances.

5

Link assessment and tie strength ranking

5.1 Synopsis

In this chapter[1], a framework for assessing the links in a social network is presented. Link assessment means checking whether a link in a social network is a true-positive or a false-positive link when comparing to real friendship relations, i.e., to check whether two friends in social networks are really friends in reality. Besides, the framework is employed to rank the links of a social network. The framework was tested on two datasets containing ground-truth social networks (surveyed friendship of lawyers and validated Facebook network of a group of researchers). The framework was tested again on the same networks with added edges to simulate noise. The results were satisfactory in terms of classification metrics. The results of this chapter reveal insights regarding the use of machine learning for networked data.

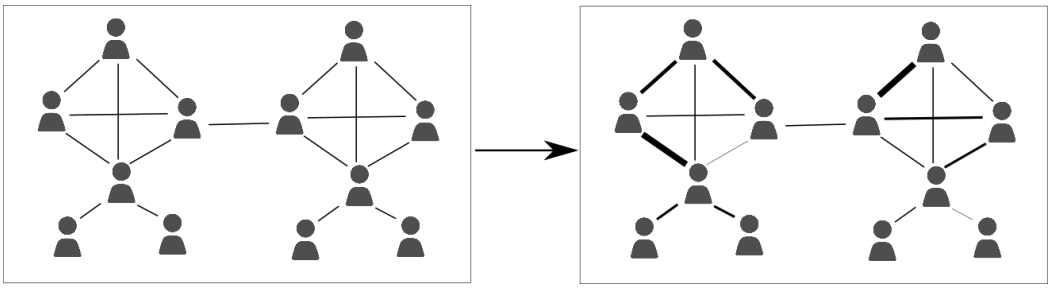


Figure 5.1.1: The goal of this chapter is to assess and rank the links of a social network with the help of external information.

[1]This chapter is based on the work [AZ15, AZ18b]

5.2 Introduction

Like many other complex networks, online social networks contain noise, which are links that do not reflect a real relationship or links with low intensity. These noisy links change the real structure of the network and decrease its quality. Accordingly, having a network with a lot of noise impedes accurate analysis of these networks [Zwe14]. In biology, for example, researchers often base their analysis of protein-protein interaction networks on so-called high-throughput data. This process is highly erroneous, generating up to 50% false-positives and 50% false-negatives [Dea02] and thus introducing noisy links into the constructed protein-protein interaction networks. As a result, assessing how real a link is in these networks is indispensable in order to get a high-quality representation of the studied system. Therefore, accurate analysis results are hard to attain without an assessment process. Based on that, many researchers have started assessing the quality of these biological networks [GR03, CHLN04] by assessing the structure of their links.

In online social networks, the situation is quite similar, as many online social networks experience such noisy relationships. A friend on Facebook, a follower on Twitter, or a connection on LinkedIn does not necessarily represent a real-life friend, a real person, or a contact from someones professional work, respectively. One possible reason for the noisy relationships in these OSNs is the low cost of forming a link on online social network platforms, which results in a large number of connections for a member. Another reason for the existence of noisy relationships is the automatic sending of invitations when a member first registers on one of the social network platforms; these invitations may contribute to connecting you with persons you really do not know in real-life but whom you have contacted once for whatever reason. Another example is the follow relationships in the Twitter social network, where it is easy to be followed by a fake account or by a real account whose owner seeks a possible follow-back to get more connections. Such fake accounts were recently removed from Twitter[2] and Facebook[3].

In this chapter, we aim at assessing the relationships within a friendship social network (SN) based on the structure of networks related to the friendship social network of interest SN. Like the exogenous (external) networks in Chapter 4, these networks are called *Exogenous Interaction Networks:* $\mathcal{G} = \{g_1, \cdots, g_k\}$. We have presented results in Chapter 4 that show that social networks can be well predicted by exogenous[4] networks. While it is possible that friendships induce collaboration or other interactions in exogenous networks, it seems plausible that the formation of a friendship link on social networks between colleagues is initiated, e.g., by work collaboration. In this chapter, we are using the network structure of these exogenous networks to identify *real* friendship links in the friendship social network.

[2] https://www.nytimes.com/2018/07/11/technology/twitter-fake-followers.html

[3] https://www.recode.net/2018/5/15/17349790/facebook-mark-zuckerberg-fake-accounts-content-policy-update

[4] Exogenous means any external information that is not based on the structure of the friendship social network.

Looking merely at one individual network, in this case the *SN*, is a rather simplistic abstraction of social interaction, which is not sufficient for understanding its dynamics [Bea14]. Thus, utilizing the exogenous interaction networks as a medium for assessing the quality of the links in a friendship social network (i.e., identifying real friendship connection) is our concern in this chapter.

Motivating example: To better understand the concept of link assessment using exogenous interaction networks, let us consider a real dataset from a research center environment that we will use later in the experiments. In addition to the Facebook friendship network, the members of the research group have different social interactions that can be used to check the quality of the structure of their *Facebook* friendship network *SN*, i.e., telling whether an edge in this Facebook network is a real friendship relation or not. These exogenous interaction networks $\mathcal{G}$ include:

- Work (g_1): Where a link exists between two members if they work/ed in the same department.
- Co-author (g_2): Where a link exists between two members if they have co-authored a publication.
- Lunch (g_3): Where a link exists between two members if they had lunch together at least once.
- Leisure (g_4): Where a link exists between two members if they have participated in the same leisure activity at least once.

From a network perspective, the interactions in the exogenous networks presumably affect the structure of the social network *SN*. That is because the link formation process within any social network is not only driven by its structure but is also influenced to some extent by external factors (exogenous interaction networks $\mathcal{G}$), which was the topic of Chapter 4. For example, it is highly probable that any two persons who have had lunch together and/or have spent some leisure time together will be friends in the *SN*. However, if there is a friendship link between two members A and B in the *SN* and there is no link between A and B in any of the networks in $\mathcal{G}$, then this relationship might be a noisy one, or it may be a very low-strength link that does not qualify as a real friendship relation. In Figure 5.2.1, the links that exist both in the social network *SN* and in any other network $g_i \in \mathcal{G}$ are presumably real links. On the other hand, there are 44 links that are not in any $g_i \in \mathcal{G}$, which leads to the question: *How likely is it that these edges to be noise?* In fact, it is hard to capture all of the possible relationships between the members of this dataset in real-life. For example, one of the 44 links might be between two researchers who are living in the same building or who are members of the same political party, which is data that we do not have or that is hard to collect. Thus, these links are potential noise or relationships with very low intensity, by virtue of the data we have. Later, we will discuss why this dataset, in particular, is considered as a gold-standard for the real friendship among its members.

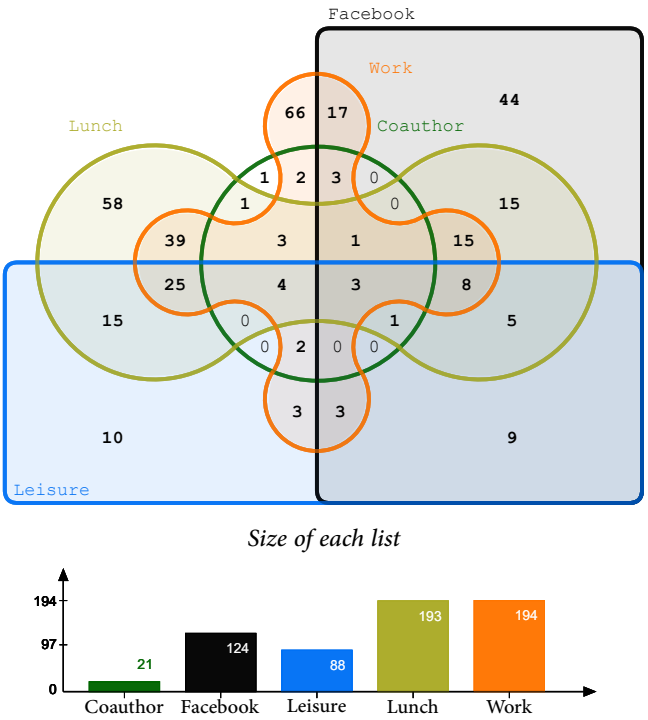


Figure 5.2.1: Venn diagram [BME+14] for edge overlapping between the Facebook social network SN and the other exogenous interaction networks $\mathcal{G}$ for the Research Group dataset.

5.3 Related work

The work in this chapter can be regarded from different perspectives. In the following, we present the related work to our work in this chapter.

5.3.1 Link prediction

The work in this chapter is related to the link prediction problem using external information that is associated with a social network. The problem of link prediction was initially defined in the seminal work of Liben-Nowell and Kleinberg [LNK03], which has been followed by a plethora of research in the area of link prediction. Surveys and literature reviews such as [AHZ11, LZ11, WXWZ15, MBC16] provide an overview of the methods used in link prediction. The work that is most relevant to ours is link prediction using a social network plus additional information. Wang and Sukthankar [WS14] provided a link prediction model for predicting the collaboration among researchers of the DBLP using different types of relations. Yang et al. [YCSH12] and Negi and Chaudhury [NC16] introduced link prediction models for multi-relational networks where the edges have different types of interactions. Similarly, Davis et al. [DLC11] demonstrated link prediction of YouTube following relationships using different types of interactions captured on YouTube, such as sharing videos and sharing subscriptions. Similar work was done by Horvat et al. [HHHZ12] on inferring the structure of a social network using the structures of other social networks. A re-

cent work by Lakshmi and Bhavani [JLDB17] incorporates temporal data into the multi-relational dataset to provide effective link prediction.

5.3.2 Network construction from noisy observation

The work in this chapter is related to noisy network structure. Many networks are constructed from relational data or interaction data. These networks normally suffer from noisy edges, which requires methods to identify these noisy edges so that any subsequent network-based analysis becomes more reliable. In biological networks, many works aim at assessing the quality of these biological networks [GR03, CHLN04] by identifying noisy edges in the structure of their links. Network construction from relation data and/or dynamical observations has been an active research area where noise in these observations is the main challenge. Thus, many studies focused on network construction under noisy observation. Tu et al. [TCC13] provided a method for constructing networks from time series relational data with noise. Similarly, Ouyang et al.[OJT16] were interested in link prediction in networks that contain noise. They contributed a method for identifying noisy edges using neighborhood measures. In the same vein, Zhang et al. [ZCH17], Shandilya et al. [ST11], Tam et al. [TCL18], Newman [New18] provided methods for constructing networks from relational interactions with noise reduction. Constructing directed networks from noisy interactional observation has also been addressed by Ching and Tam [CT17].

5.3.3 Tie strength ranking

The work in this chapter is related to tie strength ranking research. A recent study has shown that at least 63% of Facebook users have unfriended at least one friend for different reasons [Sib14]. According to Sibona [Sib14], the reasons for unfriending include frequent or useless posts, political and religious polarization, inappropriate posts, and others. These reasons for the deletion of a friendship connection indicate that social networks contain noisy relationships that need to be eliminated in order to keep only the desired friends and, consequently, their feeds. Accordingly, online social networks contain many false-positive links that push the members to use the unfriend/unfollow feature or, as a less extreme reaction, categorize unwanted connections as restricted members. Thus, a member of an online social network can easily connect to another member based on strong motivation, such as being a real-life friend or being a member of the same political party, or based on weak motivation, such as being a friend of someone they know. This variation in the type of friendship links in social networks has led many researchers to quantify the strength of the relationships [GK09, XNR10, Zea12, GER12, Gil12, JSB+13] within social networks as general requirement for friend recommendation. Pappalardo et al. [PRP12] proposed a multidimensional model for capturing the strength of the ties in the social networks of the same actors. A very related work was done by Xie et al. [XLZ+12], where the authors studied Twitter users to identify real friends. Also, Spitz et al. [SGS+16] assessed the low-intensity relationships in complex bipartite

networks using node-based similarity measures. Pratima and Kaushal [PK16] presented a prediction model for predicting tie strength between any two connected users of OSNs as an alternative to the binary classification of being a friend or not. Kumar et al. [KSSF16] studied weight prediction, as a form of tie strength, in signed networks. Some researchers have been interested only in quantifying strong ties. Jones et al. [JSB+13] studied the interactions among the users of Facebook to identify the strong ties in the network. A similar recent work by Rotabi et al. [RKKS17] employed network motifs to detect strong ties in social networks. Some applications of tie strength have been applied in different domains. Wang et al. [WLE+16] presented a social recommendation system based on tie strength prediction. McGee et al. [MCC13] predicted the location of users using the tie strength of the members of Twitter.

5.4 Contribution

Our contribution in this chapter differs from the previous related work as follows. The method presented in this chapter considers not only the online activities of an *SN* as social relationships but also some other offline interactions or interactions that are platform-independent, i.e., interactions that take place outside the social network platform. This provides more significant insights into the motives underlying tie formation in online social networks. Additionally, to the best of our knowledge, the link assessment application of social network analysis has not been addressed before in the context of *social networks* as complex systems. In this chapter, we are neither interested in constructing networks from relational data nor in predicting links in a network. However, we are interested in assessing the links in friendship network that may contain noise. In this chapter, we will define the link assessment problem and present a method for quantifying the noisy links in them using the network vectorization method described in Chapter 2.4. Besides, one contribution of this chapter is extensive details on using machine learning for networked data.

5.5 The proposed method

This chapter aims to assess and rank the links in a social network *SN* using the exogenous interaction networks of the same members of the *SN*. The method we will propose benefits from the structure of these networks in order to infer with the help of a supervised machine learning classifier whether a link in the *SN* is a true-positive or a false-positive. The idea of the proposed framework is to convert the link assessment problem into a machine learning classification problem, which can be seen as an extension to the work presented in Chapter 4. The proposed method for link assessment and tie strength ranking is similar to what we proposed in Section 4.5. The difference is manifold: (1) as we are interested in link assessment of social friendship networks, we restricted our experiments to datasets with friendship social networks that are very close to real offline friendship networks; (2) results of probabilistic classifiers' output are used as an indicator for the tie strength ranking of

the nodes; (3) in-depth investigation on the properties of the FDM and the correlation between the used features; (4) insights into using machine learning for networked-based data models; (5) multiple classifiers are used and parameter tuning was performed. Those differences extend our understanding regarding the best practices of using machine learning in networked-data models and also the limitation as we will see later on this chapter.

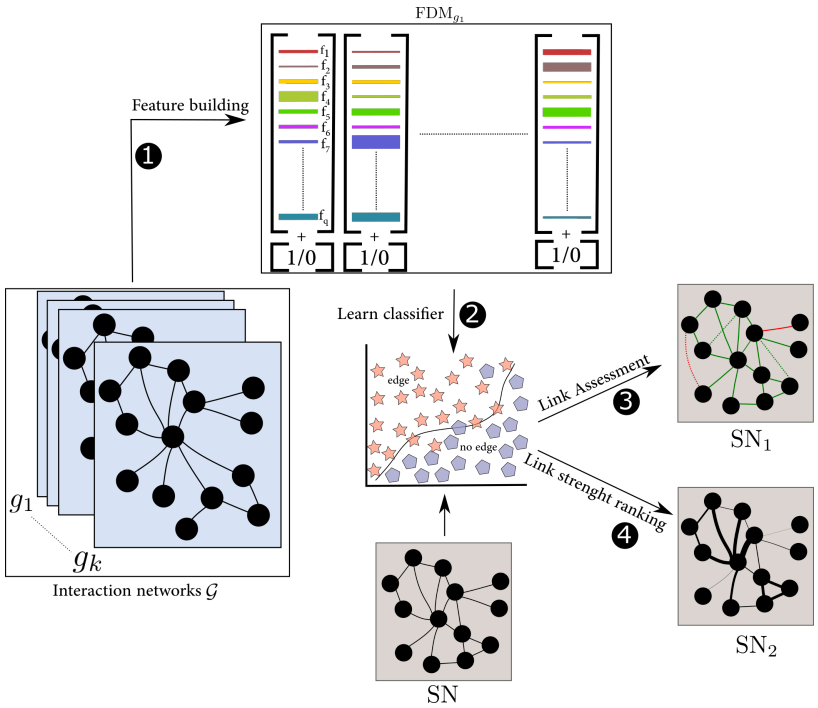


Figure 5.5.1: The framework for link assessment and tie strength ranking using exogenous interaction networks and machine learning. SN_1 is the friendship social network with assessed links and SN_2 is the friendship social network with ranked links.

Figure 5.5.1 depicts the process of assessing the links of a social network SN using exogenous interaction networks $\mathcal{G}$. The process is described as follows (the particularities of training and testing is described later):

Step 1: The FDM_{g_i} is constructed for each network $g_i \in \mathcal{G}$, in the same manner as explained in Section 4.5. Algorithm 5.1 is used for constructing the FDM with the given set of features. The used features in this chapter are: common neighbors ($\mathcal{CN}$), resource allocation ($\mathcal{RA}$), Adamic-Adar coefficient ($\mathcal{AAC}$), Jaccard index ($\mathcal{JI}$), hub depressed index ($\mathcal{HDI}$), hub promoted index ($\mathcal{HPI}$), Sørensen-Dice Index ($\mathcal{SD}$), preferential attachment ($\mathcal{PA}$), and the local community degree ($\mathcal{CRA}$).

Step 2: Each constructed FDM_{g_i} is used to train machine learning classifier (or classifiers when comparing the performance of different classifiers) such that for each FDM_{g_i} we have a trained classifier.

Step 3: The trained classifier from step 2 is used to assess the links of the *SN* by identifying each of them as TP, FP, TN, or FN.

Step 4: The trained model from step2 is also used to assign weights to each pair of nodes in the *SN*, which represent the strength of the connection between pairs of nodes. The weights are simply the probabilistic output of the classifier, which means a probabilistic classifier must be used for step 2.

Algorithm 5.1: Edge-based transformation algorithm of a graph G using the set of edge-proximity features $\mathbf{f}$.

Input: $G = (V, E), \mathbf{f} = (\mathcal{CN}, \mathcal{RA}, \mathcal{AAC}, \mathcal{JI}, \mathcal{HDI}, \mathcal{HPI}, \mathcal{SD}, \mathcal{PA}, \mathcal{CRA})$
Init: $FDM = \emptyset$
for $u, v \in V$ **do**
  $values = ()$
  for $f \in \boldsymbol{f}$ **do**
    $values = values \oplus f(u, v)$
  if $e = \{u, v\} \in E$ **then**
    $values = values \oplus True$
  else
    $values = values \oplus False$
  $FDM = FDM \cup \{values\}$
Output: FDM

5.6 Experimental setup

5.6.1 Ground-truth

Let the ground truth $SN = (V, E)$ be a network with the node set V and the edge set E that contains only true-positives and true-negatives. Let $SN_{predicted} = (V, E')$ be the predicted social network on the same node set of V, with the edge set E' being the predicted edges. Accordingly, the set $E - E'$ contains the false-negative links, i.e., links that exist in reality (in E) but are not found in the $SN_{predicted}$ (in E'). Similarly, $E' - E$ contains the false-positive links, i.e., those that do not exist in SN but are found in $SN_{predicted}$. The goal is now to get a classification result that is as close to SN as possible. Therefore, the more accurate the machine learning classifier, the more efficient the link assessment method.

It is not easy to define a real-life friend. Calling someone a friend or not a friend is a very subjective issue. Thus, getting a dataset with ground truth real-life friendship relations seems to be very challenging. To overcome this issue, we restricted our experiments to datasets that contain a friendship social network that is believed to be very close to offline friendship relations. We used the research group (RG) and the law firm (LF) datasets described in Section 4.6. For the RG dataset, the Facebook network is for a small group of people which was acquired by the maintainer of the dataset [MMR13]. Although it is hard to conclude that this Facebook network is *the* real offline friendship network of the members of the RG dataset, the links of the network presumably contain neither false-positive links nor false negative links according to the dataset provider[5]. We consider the Facebook network in the RG dataset as gold-standard for the offline friendship network among the RG dataset members. For the LF dataset, the friendship social network is based on a survey. Thus it is hard not to believe the survey participants about their friendship relations opinions. Hence, we used the friend social network in the LF dataset as a ground-truth friendship network.

5.6.2 Training and testing

Based on the information provided in the previous section, the machine learning problem $\psi(X, Y)$ means that the dataset X is used to train a machine learning classifier to classify the links in Y. In this case, X is the FDM_{g_i} and Y is the FDM_{SN}; which is the ground truth. To test the effectiveness of this method, a social network with ground truth data will be assessed. We have different scenarios for the experiments in this chapter:

[5]This may sound contradicting to what we conjectured in the introduction concerning the noise in social networks. However, we contacted the owner of the data set and made sure that there is neither false-positives nor false-negatives in the Facebook network, which is congruent with our experiment. Additionally, this Facebook network is a closed (in terms of its members) social network for *certain known* people where any member certainly knows the other members at least by name and/or face. That is why we considered it as a gold-standard dataset.

1. ***Scenario 1:*** *Train on one exogenous network, test on SN.* If the links of the social network *SN* are assessed using a network $g_i \in \mathcal{G}$, then the machine learning problem becomes: $\psi(FDM_{g_i}, FDM_{SN})$, which means that the training phase uses the FDM generated only from a single network g_i to assess the links in the *SN*. This assessment enables us to determine whether the structure of a network $g_i \in \mathcal{G}$ is sufficient for efficiently assessing the links in the *SN* or not. The used features are the same as described in Algorithm 5.1.

2. ***Scenario 2:*** *Train on all external networks, test on SN.* Similar to scenario 1, if the links of the *SN* are assessed using the whole set of interaction networks, then the machine learning problem becomes: $\psi(FDM_{\mathcal{G}}, FDM_{SN})$, where $FDM_{\mathcal{G}}$ is the FDM for the combined networks that are generated from an aggregation of all exogenous networks in a dataset. The used features are the same as described in Algorithm 5.1.

3. ***Scenario 3:*** *Train and test on SN.* For this scenario, we split the FDM_{SN} into 70% and 30% for training and testing, respectively. The split was performed after shuffling the data with the scikit-learn Python library [PVG+11] random seed *zero*. This scenario is used to compare the assessment using only external networks using only the *SN* itself. The used features are the same as described in Algorithm 5.1.

For all scenarios, the values of the features $\mathcal{CN}$, $\mathcal{RA}$, $\mathcal{AAC}$, $\mathcal{JI}$, $\mathcal{HDI}$, $\mathcal{HPI}$, $\mathcal{SD}$, $\mathcal{PA}$, and $\mathcal{CRA}$ are used as input to Algorithm 5.1, which constructs the FDM_{g_i} for every network g_i of the exogenous networks $\mathcal{G}$. Also, the FDM is constructed for the social network of interest FDM_{SN}, where *SN* is the ground truth to test on. It is clear that the two first scenarios used two disjoint datasets one for training and the other for testing. Thus, neither percentage split nor k-fold cross-validation was required for these two scenarios, unlike the third scenario. In the results section, we will present the results of each scenario and discuss them.

5.6.3 Evaluation metrics

In addition to the evaluation metrics used in Section 4.6.2 (accuracy, precision, recall, and the F1-score), we also used the Area under Receiver Operating Characteristics curve (*AU-ROC*). The ROC curve [HM82] plots the true-positive rate against the false-positive rate. The area under this curve reflects how good a classifier is and is used to compare the performance of different classifiers.

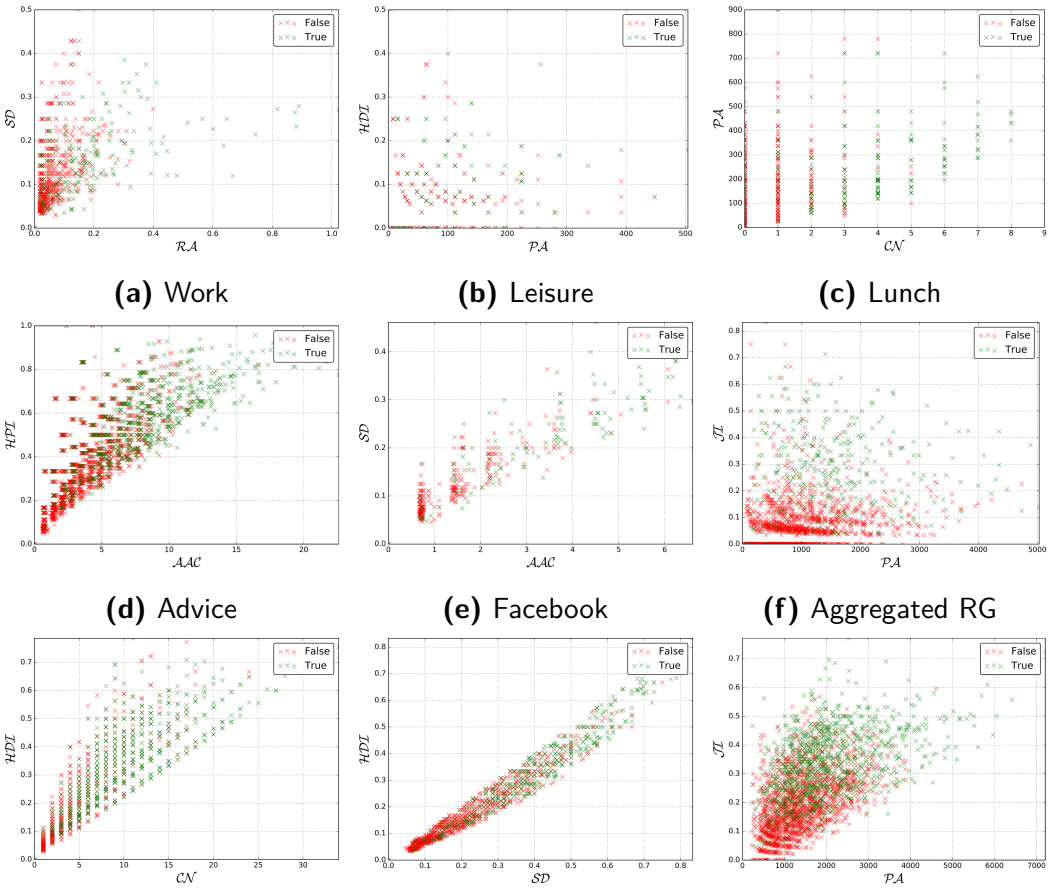
5.7 Empirical results

In this section, the properties of the constructed *FDMs* and the classification results will be presented. Also, we will present the noise identification method and the corresponding results. This

section ends with presenting the tie strength ranking results.

5.7.1 The properties of the FDM

In this subsection, we will present properties of the features of the constructed *FDM* (those used in Algorithm 5.1) and what they look like. Figure 5.7.1 shows two selected dimensions (2-D) of the FDMs constructed from the networks used. The figure shows that the *FDM* is not linearly separable, which renders the classification problem non-trivial. The figure also shows that there are some features that are highly correlated to each other; for example, Figure 5.7.1h shows a strong correlation between the $\mathcal{SD}$ and the $\mathcal{HDI}$ features. Later, we will discuss the correlation between the features and their impact on the classification process. There are many machine learning classifiers,



(a) Work **(b)** Leisure **(c)** Lunch

(d) Advice **(e)** Facebook **(f)** Aggregated RG

(g) Coworker **(h)** Friendship **(i)** Aggregated LF

Figure 5.7.1: Selected 2-D scatter plots of the *FDM* for the networks used. The x-axis and the y-axis represent the selected features presented in Section 2.3.2. The red markers are the *False* instances, and the green markers are the *True* instances, which indicate the existence, respectively non-existence, of an edge.

each with their own assumptions, limitations, and parameters to tune. For example, some classifiers like *Logistic Regression* assumes that there is no correlation between the features used to train

it. This makes logistic regression unsuited for classification with highly correlated features. On the other hand, there are classifiers, such as the *Support Vector Machines* (cf. Section 3.3.3) with kernels, that can perform well with correlated features; others assume normalized feature values, and so on. Thus, it is crucial to understand the data that is being used in the classification process.

Figures 5.7.3 and 5.7.4 depict a deeper analysis of the *FDM*'s features. In Figure 5.7.3, the correlations between the features of the *FDM* are not the same across all networks of the *Research Group* dataset. In Figure 5.7.2a (the Work network), there is less correlation between the features when compared with, for example, Figure 5.7.2e (Facebook). In Figure 5.7.3, panels 5.7.2a, 5.7.2b, 5.7.2c, 5.7.2d, 5.7.2e, and 5.7.2f, the feature that is correlated the least with the other features is $\mathcal{PA}$. It turned out that the FDM's features are intrinsically correlated. The reason is that all the used features, except the $\mathcal{PA}$, are dependent on $\mathcal{CN}$. The correlation is more apparent in the corresponding correlation scatter plots in Figure 5.7.3, panels 5.7.3a, 5.7.3b, 5.7.3c, 5.7.3d, 5.7.3e, and 5.7.3f. These panels show a strong correlation between $\mathcal{JI}$ and $\mathcal{SD}$, between $\mathcal{JI}$ and $\mathcal{HDI}$, and between $\mathcal{ACC}$ and $\mathcal{CN}$. Also, from the distribution of the feature in the diagonals of Figure 5.7.3, panels 5.7.3a, 5.7.3b, 5.7.3c, 5.7.3d, 5.7.3e, and 5.7.3f, it is obvious that the distribution of these features is not Gaussian. Most features of all *FDMs* show low variance, except for the FDM of Facebook in Figures 5.7.2e and 5.7.3e.

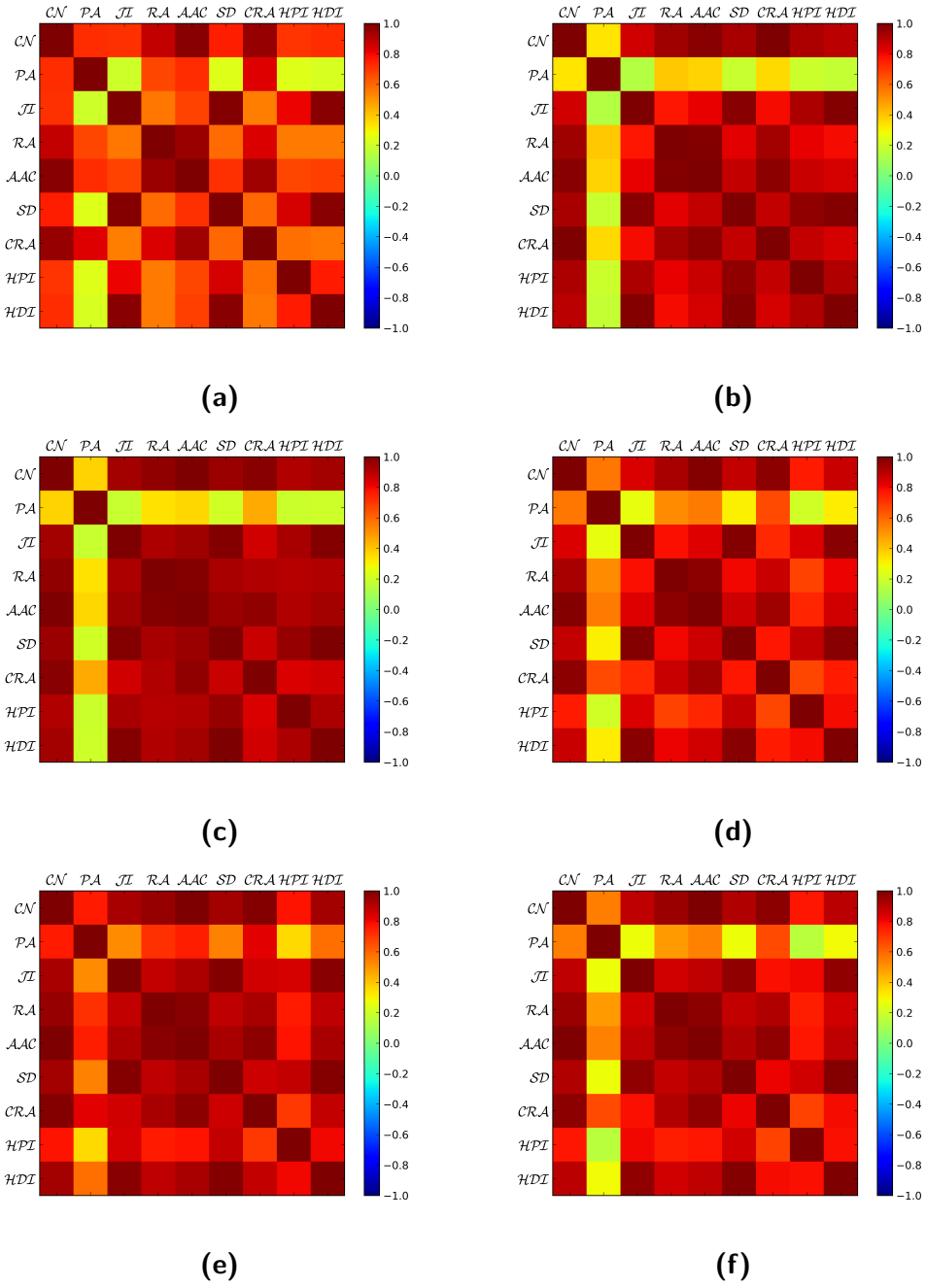


Figure 5.7.2: The feature correlation matrix of the features of the *FDM* for the Research Group (RG) dataset. Panels a, b, c, d, e, and f show the correlation matrix for the *FDM* of the networks Work, Co-Author, Lunch, Leisure, Facebook, and Aggregated RG, respectively.

Figure 5.7.4 shows the same analysis as presented in Figure 5.7.3 but for the *Law Firm* dataset. However, there are some differences in the properties of the features of the *FDM*s of the Law Firm networks. For example, the networks' *FDM*s have more variance for all features of the *FDM*s of all networks. Also, the features are more correlated with each other compared to the Research Group dataset.

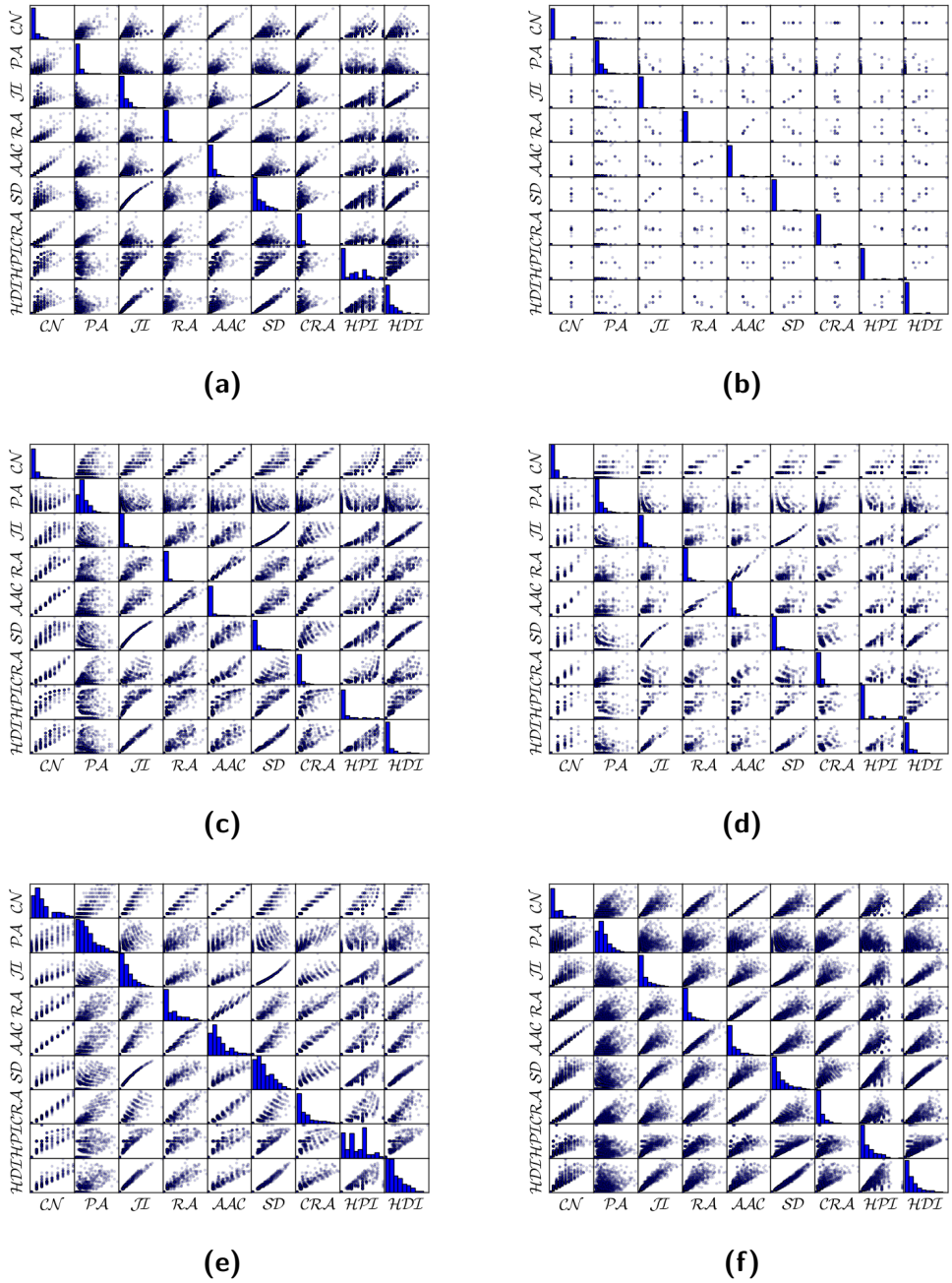


Figure 5.7.3: The feature correlation scatter plots of the features of the *FDM* for the Research Group (RG) dataset. Panels a, b, c, d, e, and f show the correlation scatter plots between two feature of the *FDM* for the networks Work, Co-Author, Lunch, Leisure, Facebook, and Aggregated RG, respectively, with the distribution of each feature in the diagonal.

5.7.2 Link Assessment Results

In this section, we present the results for each of the scenarios defined in Section 5.6.2. The results presented in this section are based on the $\mathcal{SVM}$ classifier [CV95] with a Gaussian kernel. We used the implementation from *scikit-learn* Python package [PVG+11] with its default parameters.

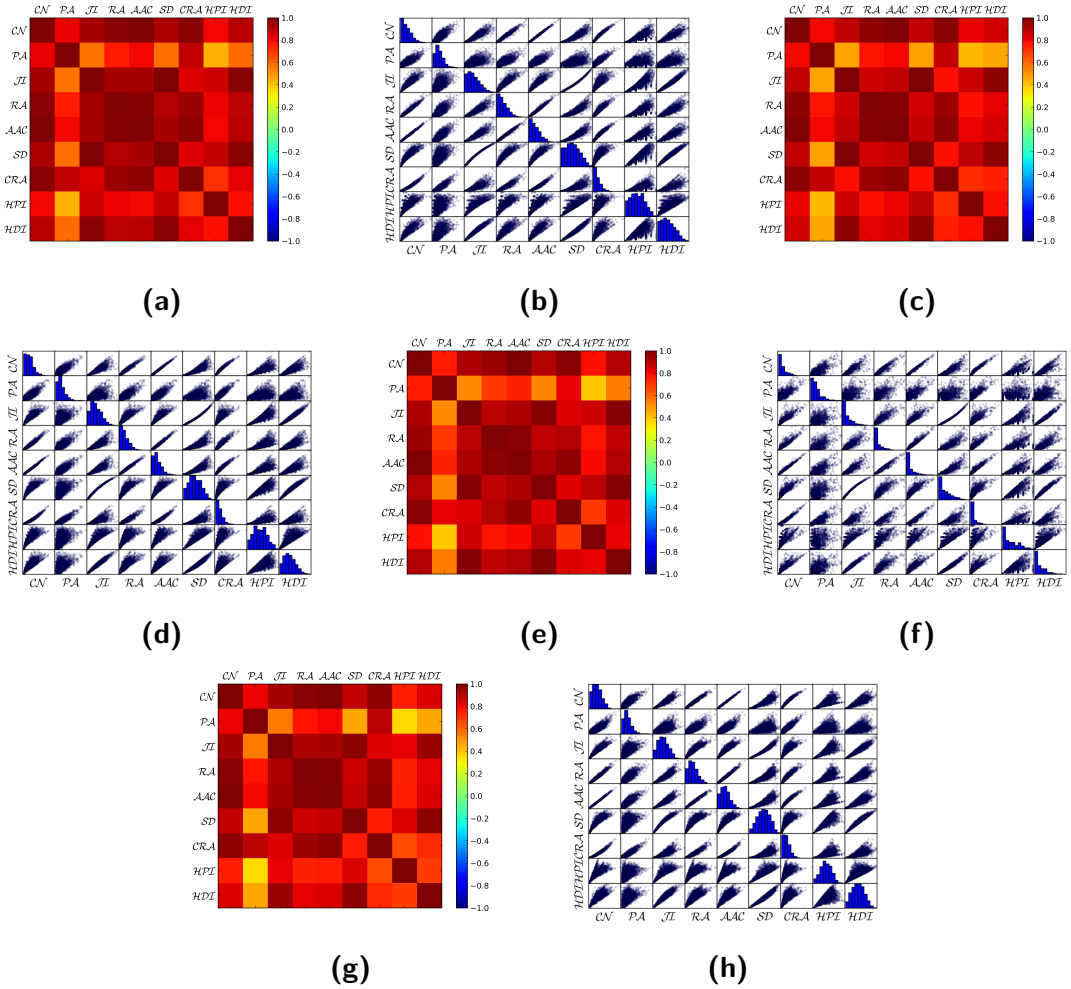


Figure 5.7.4: The feature correlation matrix and the feature correlation scatter plots of the *FDM*'s features for the Law Firm (LF) dataset. Panels a, c, e, and g show the correlation matrix for the *FDM* of the networks Advice, Coworker, Friend, and Aggregated LF, respectively. Panels b, d, f, and h show the correlation scatter plots between two features each of the *FDM* for these networks, with the distribution of each feature in the diagonal.

Table 5.7.1 shows the results of the assessment for the research group dataset (RG) and the law firm data set (LF) for the three different scenarios presented earlier in Section 5.6.2. In the following, we present the results and discuss them for different claims.

Claim 1: Each network of the exogenous interaction networks exhibits sufficient structure to assess the links of the corresponding SN.

The results shown in Table 5.7.1 for scenario 1 for both datasets supports this claim. For the RG dataset, the assessment results are satisfactory in terms of the evaluation metrics; the results are close for all networks in scenario 1. The lower bound for the assessment is 0.824 considering the F1-score ($\mathcal{F}$), which is reasonable considering the very small amount of data the *FDM* of the network co-Author contains. The best performance is for the network g_4 (Lunch), which showed

a slightly better prediction results. For the LF dataset, the prediction performance of scenario 1 is also very close over different networks. The lower bound for the assessment in the LF dataset is 0.882 considering the F1-score.

Claim 2: Aggregating networks of different types of interactions does not improve the assessment.

The results shown in Table 5.7.1 for scenario 2 for both datasets support this claim. For scenario 2, using the aggregated networks, the assessment did not provide any improvement when comparing to the assessment performance using a single network. In fact, the assessment using scenario 2 is slightly worse than the assessment using scenario 1. We think that aggregating networks introduces some noisy edges in the aggregated network that mislead the classifier. For example, assume we have three nodes A, B, and C such that the triple A-C-B exist in the lunch network (A and B are not connected in the lunch network, but both are connected to C). Additionally, assume that A and B are connected in the coauthor network. Thus, when aggregating the two networks we have a triangle of the three members of mixed type of interactions that appear to the classifier as one type. This triangle in the aggregated network makes the classifier to give a high probability for A and B as friends. It seems that the performance of the link assessment using the aggregated network is upper-bounded by the performance of the link assessment using one network. Overall, the assessment using the aggregated network is still very good compared to the baseline predictors as we will see later on Section 5.8.3.

Dataset	Scenario	Trained On	Tested On	Performance			
				$\mathcal{ACC}$	$\mathcal{P}$	$\mathcal{R}$	$\mathcal{F}$
RG	Scenario 1	g_1: Work	SN (Facebook)	0.841	**0.842**	0.841	**0.841**
		g_2: Co-author		0.822	0.827	0.822	0.824
		g_3: Lunch		**0.843**	0.835	**0.843**	0.839
		g_4: Leisure		0.837	0.835	0.836	0.836
	Scenario 2	Aggregated		0.834	0.834	0.830	0.832
	Scenario 3	SN (Facebook)		0.833	0.829	0.830	0.832
LF	Scenario 1	g_1: Coworker	SN (Friend)	0.889	0.884	0.889	0.886
		g_2: Advice		0.893	0.887	0.893	0.889
	Scenario 2	Aggregated		0.885	0.879	0.885	0.882
	Scenario 3	SN (Friend)		**0.972**	**0.984**	**0.919**	**0.950**

Table 5.7.1: Assessment results for the Research Group (RG) and the Law Firm (LF) datasets. The table shows the training and testing using the three different scenarios defined in Section 5.6.2 for each dataset. The bold numbers are the best assessment performance for each measure across different scenarios.

5.7.3 Comparing different classifiers for link assessment

There are dozens of machine learning classifiers, and each has its advantages, limitations, and parameters to tune, which makes the selection of the appropriate classifier a difficult task. Thus, we want now to check the performance of the method with different classifiers.

Claim 3: The performance of the link assessment method is robust with different classifiers.

Table 5.7.2 shows a comparison of the performance of different classifiers. Based on the results in the table, the presented method resulted in a closely similar performance for most classifiers. Once again, the results of the LF dataset are slightly better than those of the RG dataset for all of the compared classifiers.

Dataset	Classifier	Performance			
		$\mathcal{ACC}$	$\mathcal{P}$	$\mathcal{R}$	$\mathcal{F}$
RG	$\mathcal{KN}$	0.800	0.795	0.800	0.797
	$\mathcal{SVM}$	0.821	**0.833**	0.821	0.827
	$\mathcal{DT}$	0.800	0.806	0.800	0.804
	$\mathcal{NB}$	0.778	0.821	0.778	0.798
	$\mathcal{LR}$	**0.827**	0.825	**0.827**	**0.827**
LF	$\mathcal{KN}$	0.843	0.823	0.843	0.833
	$\mathcal{SVM}$	0.816	0.858	0.816	0.836
	$\mathcal{DT}$	0.880	0.870	0.880	0.875
	$\mathcal{NB}$	**0.883**	0.875	**0.883**	**0.877**
	$\mathcal{LR}$	0.868	**0.878**	0.868	0.872

Table 5.7.2: Comparison of the performance of different classifiers for the aggregated versions of the RG and the LF datasets (scenario 3). The compared classifiers are: $\mathcal{KN}$: k-Nearest Neighbors vote [Alt92]; $\mathcal{SVM}$: Linear Support Vector Machines [CV95]; $\mathcal{DT}$: Decision Trees [Qui86]; $\mathcal{NB}$: Naive Bayes [Zha04]; $\mathcal{LR}$: Logistic Regression [WD67]. We used the *scikit-learn* package [PVG+11] of Python for the previous algorithms with their default parameter values.

Another aspect that is important when talking about different classifiers is the resulting *decision boundaries* and how good they are. Figure 5.7.5 shows the decision boundaries for different classifiers. The figure shows that linear models, like linear $\mathcal{DT}$ and $\mathcal{LR}$, are not able to properly discriminate between the False and the True instances efficiently. Additionally, the figure shows that the accuracy metric is a useless measure as it is not informative for the case of the *FDM* whose labels are highly imbalanced. For example, let us take a closer look at the $\mathcal{QDA}$ classifier for the second row, the attributes $\mathcal{HDI}$ vs. $\mathcal{AAC}$. The accuracy of the classifier is 0.91, which is considered high. Having said that, the panel shows that *all* of the points were classified in the red area, which ignores the True instances and makes it hard to find a binary threshold to produce binary results. This behavior indicates that accuracy is not a good measure to use if we have imbalanced

data. On the other hand, classifiers that use kernels (a method for transferring non-linearly separable data into linearly separable data by transforming the data into a higher dimension) showed good discrimination between the False and the True instances. An example of this is the $\mathcal{SVM}$ with a Gaussian kernel [CV95], the second column in Figure 5.7.5. From the figure, it is clear that the $\mathcal{SVM}$ with a Gaussian kernel was able to find disjoint areas for the data points, which helps to produce good classification results. Obviously, no conclusions can be withdrawn from this figure as it shows only two dimensions of the FDM. However, it gives insights about how a classifier behaves under non-linearly data.

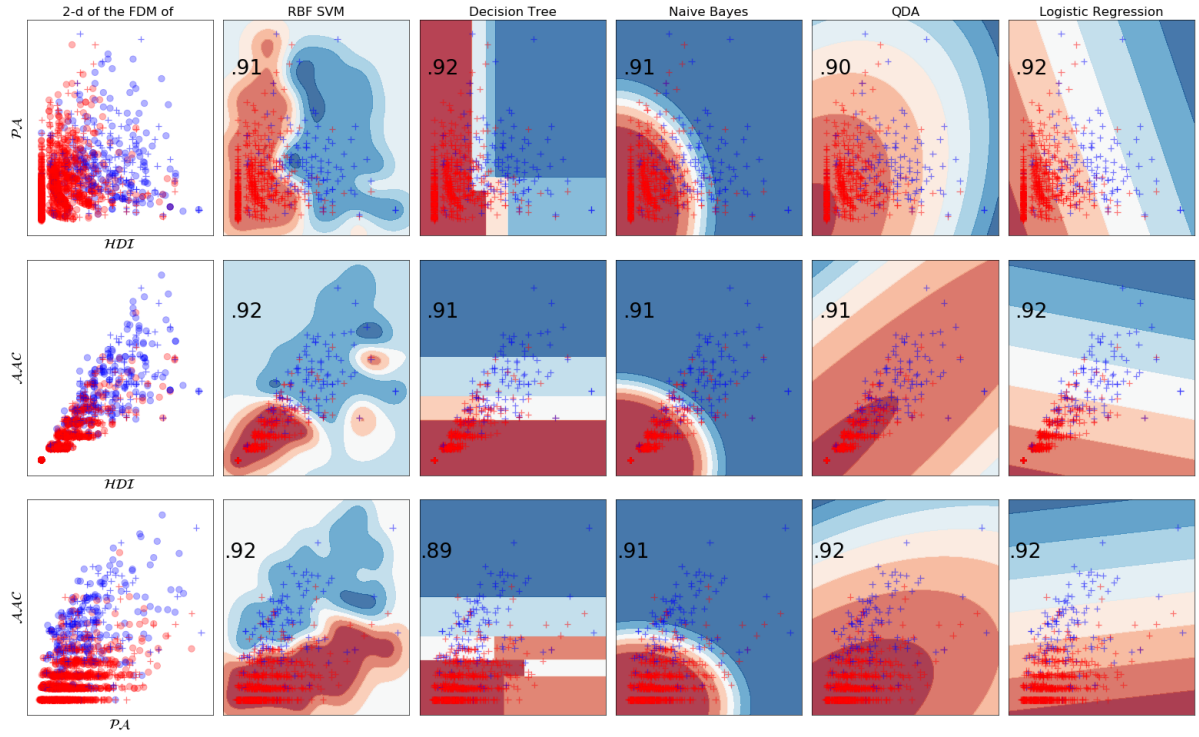
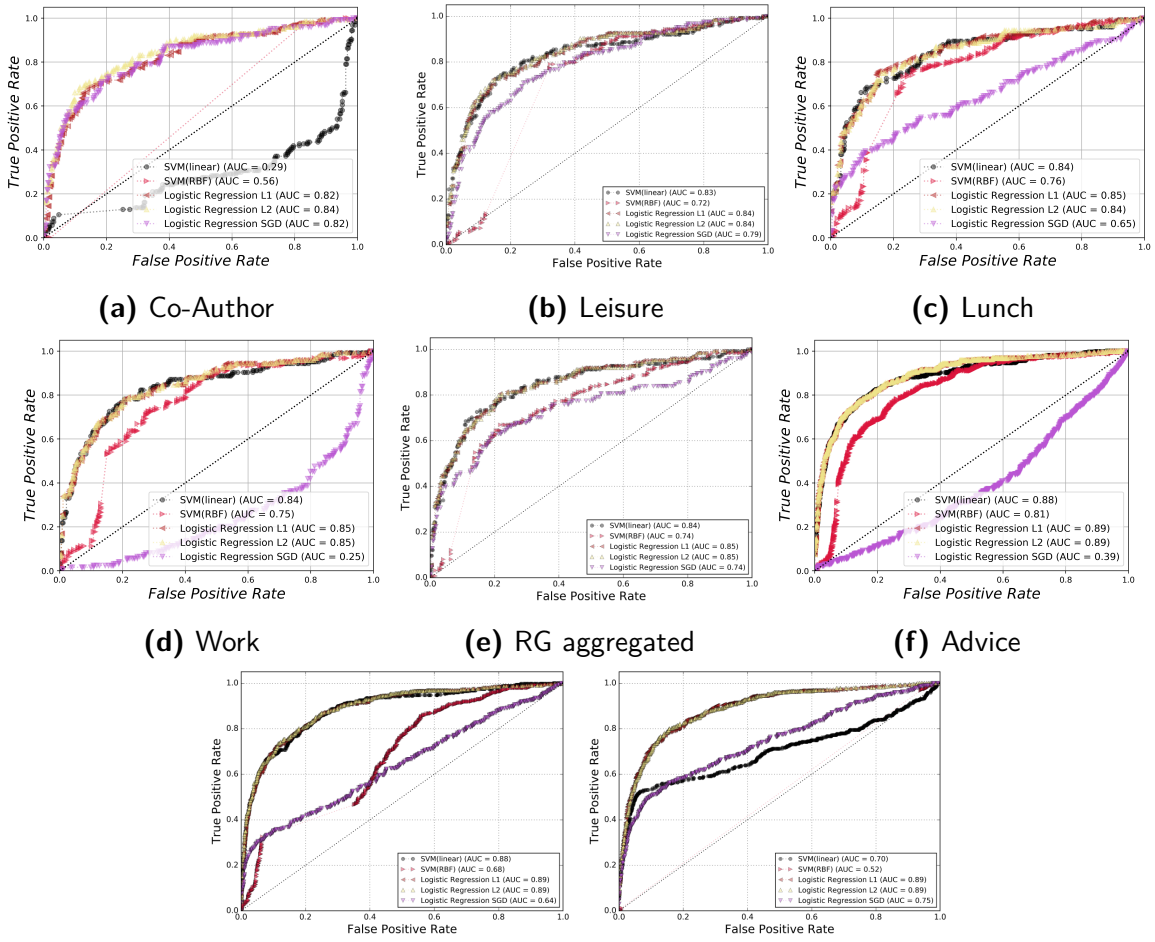


Figure 5.7.5: Decision boundaries for different probability-based classifiers. We used a 2-D scatter plot of the *FDM* constructed from the aggregated networks of the RG dataset as an illustration. The leftmost panels are the 2-D features before the classification was performed. The red points are False instances, and the blue points are True instances. The red "+" markers and the blue "+" markers are the False and True instances to be classified by the classifier, i.e., the test samples. The other points, none of which are "+" points, are the training points, where the training and the testing points were randomly split with a ratio of 60 to 40 for training and testing, respectively, with Python's random seed *zero*. The other panels represent the classification results with the decision boundaries for the test sample only. The number at the top left is the accuracy of the classification, and the gradient of the colored areas represents the probability of the classification. For example, the darker the blue area, the higher the probability that the points in this area are true instances. The classifiers used are those classifiers that give a probability as a classification result, namely: $\mathcal{SVM}$ with Gaussian kernel [CV95]; $\mathcal{DT}$: decision trees [Qui86]; $\mathcal{NB}$: Naive Bayes [Zha04]; $\mathcal{QDA}$: the Quadratic Discriminant Analysis [Cov65]; $\mathcal{LR}$: Logistic Regression [WD67]. We used the *scikit-learn* package [PVG+11] of Python for the previous algorithms with their default parameter values.

Another way to compare the performance of different classifiers is to use the area under the ROC curve for each classifier. Figure 5.7.6 shows the AUC for $\mathcal{SVM}$ and $\mathcal{LR}$ with different parameters. For the $\mathcal{SVM}$, we have two kernel parameter values: (1) $\mathcal{SVM}$ with Gaussian kernel and (2) linear $\mathcal{SVM}$. For the $\mathcal{LR}$, we have different parameters: (1) gradient descent with $L1$ regularization; (2) gradient descent with $L2$ regularization; (3) and stochastic gradient descent without regularization. Figure 5.7.6a again shows that the linear models are not robust and are not able to provide a good classification.



(a) Co-Author **(b)** Leisure **(c)** Lunch

(d) Work **(e)** RG aggregated **(f)** Advice

(g) Coworker **(h)** LF aggregated

Figure 5.7.6: The area under the ROC curve for $\mathcal{SVM}$ with Linear and Gaussian kernels and $\mathcal{LR}$ with $L1$ and $L2$ regularization and with a stochastic gradient descent optimization algorithm. The training dataset used in this figure are the corresponding GDM_{g_i}, and the testing dataset is the FDM_{SN} of the corresponding g_i.

5.7.4 Identification of noisy edges

The *SN*s used in this chapter are assumed to be very close to the real-life friendship (as discussed in Section 5.6). This does not allow proper validation for noise (false-positives) identification only because the networks used do not contain any. To overcome this issue, we injected additional k edges into the *SN* that are assumed to be noise. Then, we test the method to find out how good it is in finding *only* these k noisy edges.

Claim 4: The performance of the link assessment method is robust under randomly injected noisy edges.

To test how well the method performs for identifying noisy edges, we conducted experiments with the following general steps:

1. **Adding noisy edges:** We add $k = \left\lfloor \left(\binom{n}{2} - m \right) \times r \right\rfloor$ edges to the *SN*, where m is the number of edges in the *SN* and r is the fraction of edges to be added. We make sure that each added edge is not in the set E_{SN}. The resulted network is called $SN_{disguised}$. For example, if $r = 1$ then $SN_{disguised}$ is a complete network.

2. **Training:** For training, we used the FDM_{g_i} to train a classifier. The FDM_{g_i} is constructed using Algorithm 5.1. Additionally, we train also on an aggregated network $\mathcal{G}$ to see how the aggregated version of the exogenous networks is able to identify noise.

3. **Testing:** For testing, we used the test set $FDM_{SN_{disguised}}$ which is constructed using Algorithm 5.1.

4. **Success rate:** We only check whether the learned classifier identifies the added k edges as false-positive or not. I.e., we restrict our assessment performance to these k edges to see how good the method is in identifying these edges as noise. Based on that, we defined the *success rate* as the number of edges that were predicted as false-positives divided by k, the whole number of added edges.

We repeat the previous steps for different values of $r \in \{0.1, 0.2, 0.3, 0.4, 0.5\}$. For each value, we performed 100 different runs, each with a different random seed, and for each run, we calculate the success rate. Figure 5.7.7 shows the results of this method. In general, the results are not comparable to a random binary classifier as a baseline (which will have 0.5 success rate on average). In some cases, such as the networks of LF dataset in Figure 5.7.7 Panels 5.7.7f, 5.7.7g and 5.7.7h, the success rate is very high. On the other hand, the success rate is not that high for the RG dataset networks when comparing to the LF dataset. However, it is still very good except for the network co-author (cf. Figure 5.7.7c) which is comparable to a random binary classifier. The poor perfor-

mance for the co-author network is due to it being a very small network; it hardly captures a good structure for the relationships among its members. The noise identification success rate in the LF dataset was higher than in the RG dataset, seemingly because the networks of the LF dataset are nearly 10 times denser than the RG. Noticeably, the success rate for the aggregated networks of the RG and the LF dataset was better than the best network, the work network and coworker network, respectively. We think the reason is that the aggregated version contains more information that helped the classifier to learn more patterns to distinguish real and noisy edges in the *SN* network. It is also noticeable that the method is robust under different values of r. There is almost no variation in the results (the success rate) when comparing the results for different values of r for the same network. I.e., the mean of the success rate r values for one network are very close.

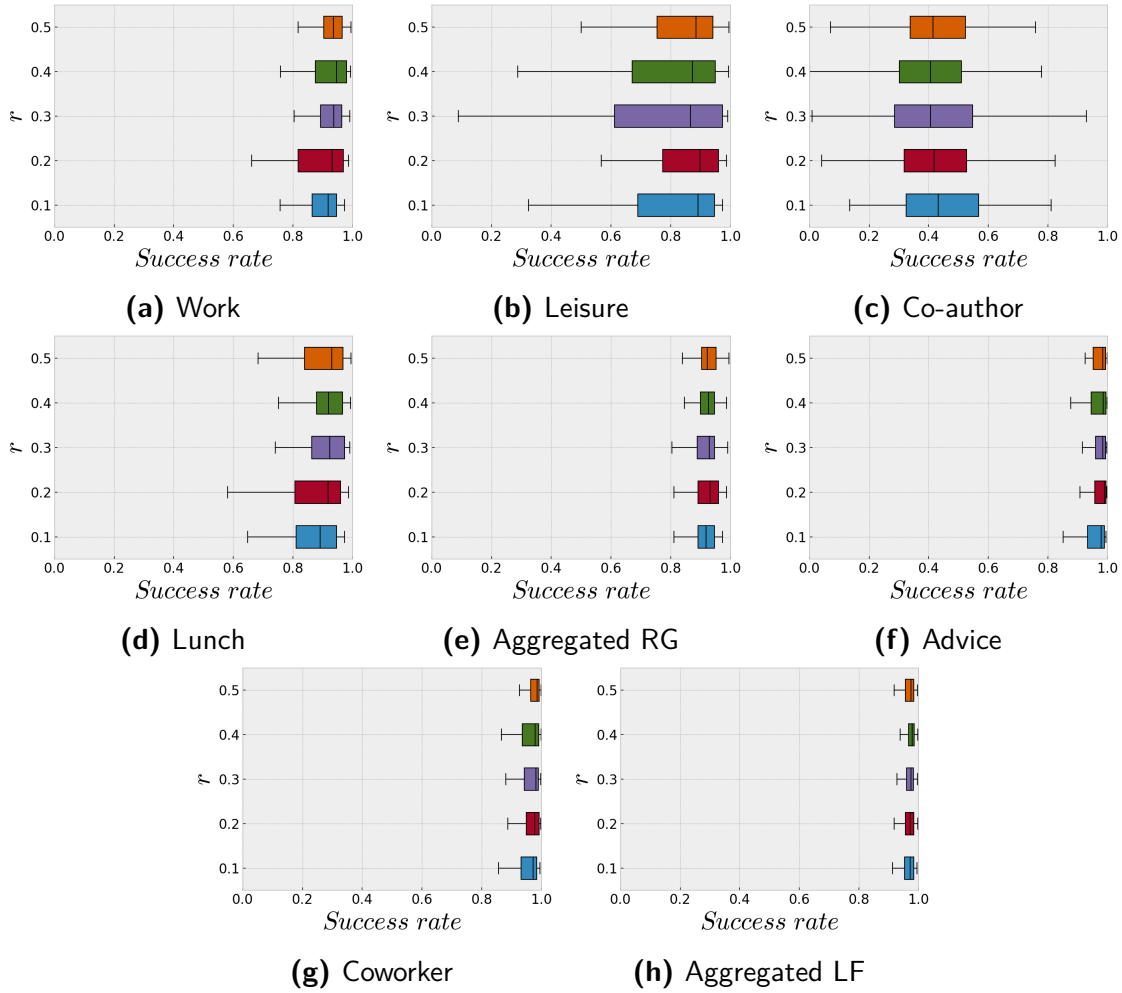


Figure 5.7.7: The figure shows the success rate of identifying the added edges to the *SN* using different networks. Each panel in the figure shows the success rate when training using the corresponding network, e.g., work network from RG dataset, and testing on $SN_{disguised}$. We used different values of r, and for each value, we performed 100 runs. Each run has different random seed value (from zero to 99) and added randomly different k edges. As we have 4 thousands experiments to generate the results in this figure, we used for training a linear SVM classifier because it is faster than the SVM with Gaussian kernel. We used the implementation from the *scikit-learn* Python package [PVG+11] with its default parameters.

5.7.5 From binary classification to tie strength ranking

In some scenarios, the links of a social network need to be ranked by the tie strength between the members. The proposed method can also give a continuous range of values between 0 and 1 (instead of having two classes), using probabilistic classifiers, i.e., classifiers that produce a probability value instead of a binary class, then find a threshold for binarizing the resulted probabilities. These probabilities are used here as the *tie strength rank* of the edges in the social network being assessed. Figure 5.7.8 shows the ranking results of the *SN* in the RG and LF datasets using different classifiers. Our assumption here is that the *best* ranking for the edges of the *SN* is a step function that changes

its value from *zero* to *one* on the number of true-negative edges in the ground-truth network (see the red line in Figure 5.7.8). This means that for undirected *SN* with n nodes and m edges, we have m edges with a tie strength of *one* and $\binom{n}{2} - m$ edges with a tie strength of *zero*. Then, the predicted tie strength for all edges, including the true-negatives in the ground truth, is compared to the best ranking using the following error measure:

$$\sum_{e=\{u,v\},\forall u,v \in V_{SN}} |e_{predicted} - e_{real}|, \tag{5.1}$$

where V_{SN} is the set of nodes of the network *SN*, $e_{predicted} \in [0, 1]$ is the probability of having an edge e in the *SN* according to probability classifier, and $e_{real} \in \{1, 0\}$, which indicates whether e is an edge in E_{SN} or not. The closer the results to the step function, the better the ranking (cf. Figure 5.7.8).

Figure 5.7.8 shows the results of the ranking using the proposed method. $\mathcal{NB}$ and $\mathcal{KN}$ provided the best ranking among all of the classifiers we used. For the RG dataset, the best link ranking, in terms of error ranking as explained earlier, was achieved using the Lunch network with an error of 20% and the Work network with an error of 21% using the $\mathcal{DT}$ and $\mathcal{NB}$, respectively. For the LF dataset, the best ranking was achieved using any of the networks in the dataset with an error of 12% using the $\mathcal{NB}$. As in the assessment results presented in the previous section, the ranking results of the *SN* of the LF is better than the ranking of the RG's *SN*.

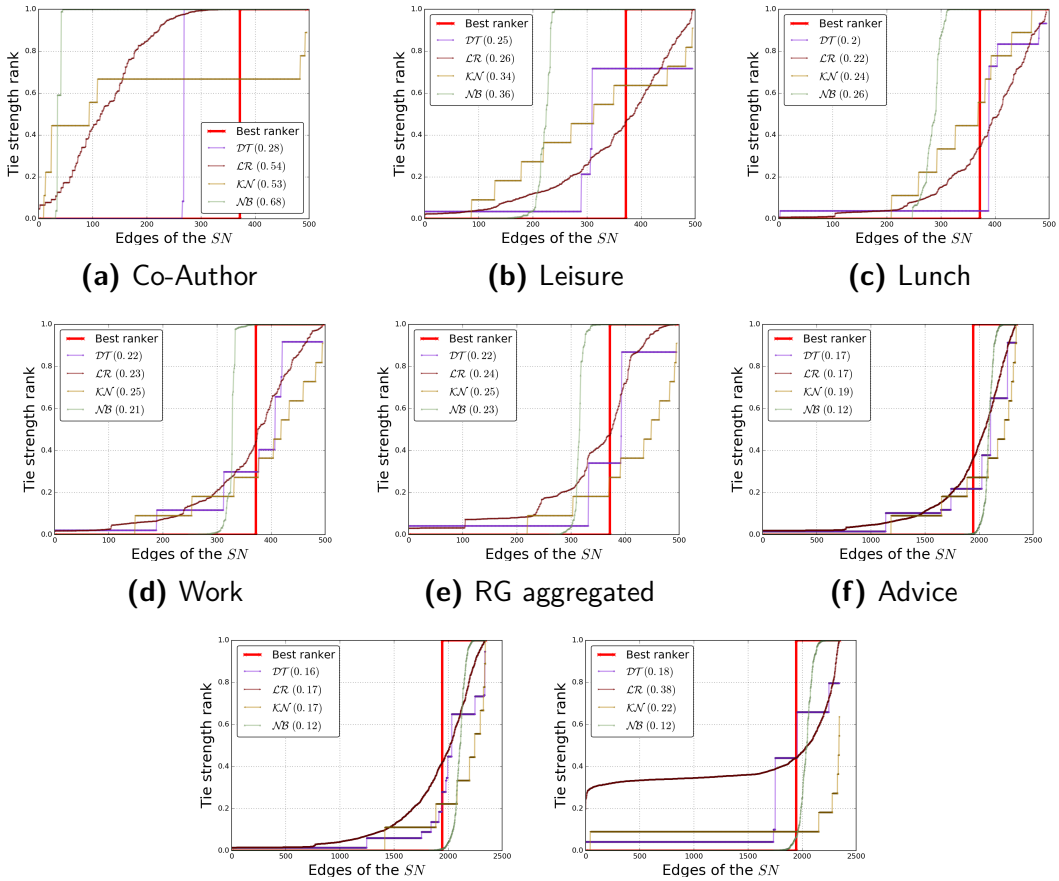


Figure 5.7.8: Tie strength ranking for the social network using the exogenous networks. The x-axis represents the edges in the SN ranked by their strength according to the ranking results; the y-axis is the tie strength rank. The best ranker, in bold red, is simply the step function on the number of edges in the social network. The best ranker is used to compare the goodness of the ranking using the proposed method. In the legend, different classifiers are used to estimate the probabilities. The numbers beside the names of the classifiers represent the errors in the ranking. This error is calculated as defined in Equation 5.1.

5.8 Discussion

The proposed method showed a good potential regarding both link classification and tie strength ranking. It seems that machine learning can be used effectively for the network-based features. In this section, we will offer our final thoughts about the problem addressed in this chapter, our method, and its limitations.

5.8.1 The importance of link assessment and tie ranking

Addressing the link assessment problem is crucial today, where online social media contain a lot of spam, ads-intensive websites, and fake news. We strongly believe that identifying noisy links in

social networks contributes to eliminating these problems and reducing their impact. Tie strength ranking, on the other hand, can also improve the quality of the information spread via online social networks. For example, an automatic ranking of the friends list on Facebook might lead to better news feeds, more reliable friend recommendations, and better-targeted ads, to name but a few. Thus, the work presented in this chapter has actionable insights on online social networks.
It seems that the tie strength problem explicitly includes link assessment. However, the two problems should be handled separately because the cost of link assessment may be lower than the cost of tie strength ranking. One reason for that is the difficulty of getting the ground truth of the real tie strength between the nodes of a network. This reason has caused some researchers to focus only on the strong ties, like the work presented in the related work section. Moreover, link classification can be a preprocessing step for many network-based analysis tasks, such as community detection, where eliminating noisy edges may provide more meaningful communities. Thus, we emphasize the distinction between the two problems.

5.8.2 Classification methods for network-based features

Feature correlation: The major network-based features for link proximity are based on common neighbors $\mathcal{CN}$, which makes most of the features highly correlated with each other. Having said that, highly correlated features might be a problem in classification, especially with a small amount of training data. Thus, devising new link proximity measures that are not based on the number of common neighbors is important.
Classifier selection: Decision boundaries help select a good classifier for the dataset used. Learning and optimization processes are computationally expensive, and experimenting with different classifiers with different parameters is always a laborious task. Thus, experimenting on a sample of the data to select the best classifier is crucial. To handle this, decision boundaries, like those presented in Figure 5.7.5, are very helpful for understanding the data that we have as well as to selecting the best classifier for subsequent optimization. Linear classifiers showed poor performance as the constructed *FDM* is not linearly separable, whereas classifiers with kernels showed better performance than the others. Additionally, $\mathcal{KN}$ and $\mathcal{DT}$ showed promising results for the ranking problem. It turned out that these two classifiers provided good probabilities for approximating the tie strength in the *SN*, but bad thresholds for the binary classification.

5.8.3 Baseline comparison

To provide more confidence for the results, the experiments were conducted on random graphs as a null model. We generated 50 random graphs of the same number of nodes and edges in each $g_i \in \mathcal{G}$ using Algorithm 2.1. Then, for each of these random graphs, we performed the assessment. The averaged results, e.g., the average accuracy, of the random graphs were incomparable to the results of the real datasets used. Additionally, we tested the model against a classifier that uses one simple rule

(like a threshold over one of the used features' values) as a baseline assessment. The results of the presented method using the classifiers presented in Section 5.3 were significantly better than those using the baseline classifier. Finally, a random classifier was used as another baseline classifier (cf. Figure 5.7.6). The results of the classifiers used were significantly better than those of the random classifier. Thus, we strongly believe that the results provided in this chapter are significant and are not due to any random chances.

5.8.4 Limitation

The presented method used data from a social network itself in addition to external information. The external (offline) information may not always be available, which represents a challenge. That is why we restricted the datasets to those with friendship networks that are as close to the real life friendship as possible. Moreover, the existence of the ground truth data for the tie strength ranking is hard to attain. Thus, we resort to the binary ranker as the best possible option to evaluate the tie strength ranking provided by the method.

5.9 Conclusion

In this chapter, we presented a method for link assessment and tie strength ranking of the links of online social networks using exogenous social interaction networks. The proposed method employs machine learning classification techniques to perform the link assessment via label classification based on edge-proximity measures. We conducted experiments on two different datasets that contain a friendship social network in addition to exogenous social interactions. The link assessment results, in terms of the F1-score and the accuracy, were satisfactory compared to baseline predictors. The results show that it is possible to assess the links in a social network using exogenous social interactions. Additionally, we also performed tie strength ranking using probabilistic binary classifiers. The intensive study of the features used in this chapter and the conducted experiments revealed insights about the use of machine learning for network-based features. These insights concern (1) feature correlation and its effect on the classification; (2) goodness of the decision boundaries of the classifiers used; (3) classifier selection for both link assessment and tie strength ranking.

From a network perspective, the results of the datasets used suggest that directed networks embrace more building structures that enable better link assessment and tie strength ranking than undirected networks. Also, the results suggest that a single exogenous interaction network contain enough information to assess or rank the links in a social network. It seems that for a set of persons, their social interaction outside the social network provides enough information to predict their real social relationships.

From a machine learning perspective, the results achieved in this work, regarding both link assessment and tie strength ranking, show that network-based features can be used for analyzing net-

works and building prediction models. Additionally, we discovered that some classifiers are good at providing a binary classification for link assessment, while others are good at providing a probability range for tie strength ranking.

Part III

Decay Dynamics

6

Stochastic model for network decay dynamics

6.1 Synopsis

In this chapter[1], a theoretical stochastic model for capturing the mechanics of the decay dynamics in a social interaction setting will be presented. The main equations of the model are proven to be submodular and monotone. A simulation of the model was performed using a temporally decayed dataset of the StackExchange website, and results will be presented.

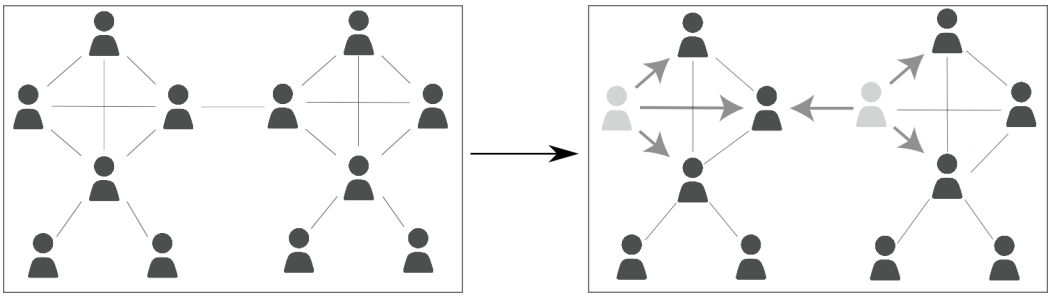


Figure 6.1.1: The goal of this chapter is to provide a theoretical model for capturing the inactivity dynamics in a network.

[1]This chapter is based on the work [AZ17, AZ18a].

6.2 Introduction

Today's online social networks represent a major source of communication and information exchange among people all over the world. Many online social networks, such as Facebook, Twitter, and LinkedIn, have proven their usefulness in connecting people and facilitating an exquisite new medium for sharing news, forming groups of people with same interests, and gathering knowledge. The growth of these online social networks in terms of user activity shows that they have become a vital part of today's human activities.

Growth dynamics

One well-studied aspect of online social networks dynamics is the *growth dynamics* of a network. The work by Barabási and Albert [BA99] has presented a simple model for understanding the growth dynamics of a network, namely the *Preferential Attachment Model* (PAM), which is a the-rich-get-richer model. Jin et al. [JGN01] noticed that the model by Barabási and Albert [BA99] and other similar models, like the work by Dorogovtsev and Mendes [DM00] for modeling the growth of random networks, are not suitable for understanding the growth dynamics of social networks. Thus, they developed a model that considers the particularities of social networks without any power law distribution and with a large clustering coefficient [JGN01]. When online datasets became available, Newman [New01] empirically studied the growth of social networks using scientific collaboration networks against the PAM model. Bala and Goyal [BG00] developed a non-cooperative game-based model for understanding network formation from the perspective of game theory. Later, Jackson [Jac03] surveyed the models and methods that were being used to capture the network formation process and compared them in terms of stability and efficiency. Leskovec et al. [LKF05] first showed on dynamic network data that networks densify and their diameter shrinks over time. They also provided another growth dynamics model that was able to produce networks with these properties. The prior work and the availability of rich datasets motivated researchers to perform an in-depth investigation of the properties of networks over time. Kumar et al. [KNT06] studied the growth of a large social network in terms of network component analysis; Kossinets and Watts [KW06] studied the tie formation process within social networks, and Capocci et al. [CSC+06] studied the statistical properties of the growth characteristics of Wikipedia collaboration social networks. Likewise, Backstrom et al. [BHKL06] empirically studied how groups are formed and evolve over time in the MySpace social network, while Mislove et al. [MKG+08] presented a study on the growth of the Flicker social network.

Decay dynamics

Even though there are many successful social networks, the evolution of a social network also incorporates *decay*. In the last decade, some online social networks were shut down after suffering from

colossal loss or inactivity of their members. Online social networks, such as Friendsfeed, Friendster, MySpace, Orkut, and many websites of the StackExchange platform are now out of service. Even though some of these online social networks, e.g., Orkut and MySpace, showed tremendous growth [AHK+07] just a decade ago. The decay of these networks poses many questions about the reasons for their downfall. Garcia et al. [GMS13] and Chhabra et al. [CBS14] studied the static properties of Friendster and MySpace, respectively, in order to understand the network-related properties of these networks as an example of a decayed network. Recent studies by Malliaros and Vazirgiannis [MV13] and Bhawalkar et al. [BKL+15] provided theoretical models for understanding social engagement in online social networks with the potential to predict social inactivity. Torkjazi et al. [TRW09] performed an analysis of the MySpace online social network and examined the activity and inactivity of its users, offering some insights into the reasons behind the decline of MySpace. Similarly, Ribeiro [Rib14] studied the activity and inactivity of users by providing a model that uses the number of daily active users as a proxy of the dynamics on membership-based websites. Kairam et al. [KWL12] developed machine learning prediction models to predict community *longevity*, i.e., how long a community in an online social network will survive. In the same context, Asur et al. [AHSW11] discuss the persistence and decay of Twitter tweets. While investigating the reasons behind the inactivity of members of an online social network is not within the scope of this chapter, some recent studies have proposed some answers [SBBV13, KLSS+15], suggesting that the main reason behind this decay is the inactivity of the members of an online social network.

What is missing?

To develop a sound understanding of the decay dynamics of networks, not only the static properties of these networks need to be studied, but their dynamics and properties must also be investigated over time – this is precisely what we are interested in. As a scenario, we consider the StackExchange sub-websites, which were shut down after some time as there was not enough activity to keep the sub-website alive. The closed sub-websites are an example of social network decay, where we model the members of a sub-website as the nodes of the network and an edge exists between any two nodes if they post, comment, or answer to the same question on the sub-website. While we cannot offer an answer to why a person starts losing interest in a social network, we can try to analyze and model the effect of this behavior on other members of the network. Such a model might, in turn, hint at the causes of social decay or at least explain part of it.

6.3 Contribution

In this chapter, we provide a probabilistic model for understanding the social decay phenomenon in online social networks. The model we will present provides insights regarding the effect that a departing node has on its neighboring nodes. Our contribution to this chapter is threefold:

1. A longitudinal network analysis of the StackExchange sub-websites showing their decay.

2. A probabilistic model for social network decay, which is a *step-by-step* mechanistic model for a departing node and the effect of its depart

6.4 Model and notations

As we are considering a dynamic system, the notation G^t represents a network at time t, and G^0 is the initial network. The model assumes that each node has three time references: (1) at t, which is the current status (now), (2) $t+1$, which is *one* step in future, and (3) $t-1$, which is *one* steps in the past. We assume that every node $w \in V$ has an initial *Depart Probability* $\pi_w^{t=0}$, which denotes the probability of node w departing the network at time point 1 (which means that w is not connected at any other nodes at time point 1), and generally at $t+1$. If a node w did not depart at $t+1$, i.e., if $w \in V(G^{t+1})$, then its current depart probability, π_w^t, will increase depending on its neighbors who departed at $t-1$. The *tie strength* at time $t-1$, representing some possibly dynamic measure of the relationship strength, is denoted by $\delta_{v,w}^{t-1}$ and assumed to be $\in (0,1]$.

The tie strength can be any measure that reflects the intensity of the interaction between two nodes, e.g., the frequency of the interactions between these two nodes over time or the number of common neighbors over time. We think incorporating tie strength is a necessary design decision. That is because social influence among the members of any social network is undoubtedly affected by the tie strength (interaction intensity) between any two members[2].

Definition 6.4.1. A dynamic network G is called a "*Decaying Network*" if $|E(G)^{t-1}| \geq |E(G)^t|$, $|V(G)^{t-1}| \geq |V(G)^t|$, and $V(G)^t \subseteq V(G)^{t-1}, \forall t > 0$.

[2]This design decision will not add any complexity beyond necessity for the model because it is not an additional parameter if the tie strength is inferred from the network structure itself.

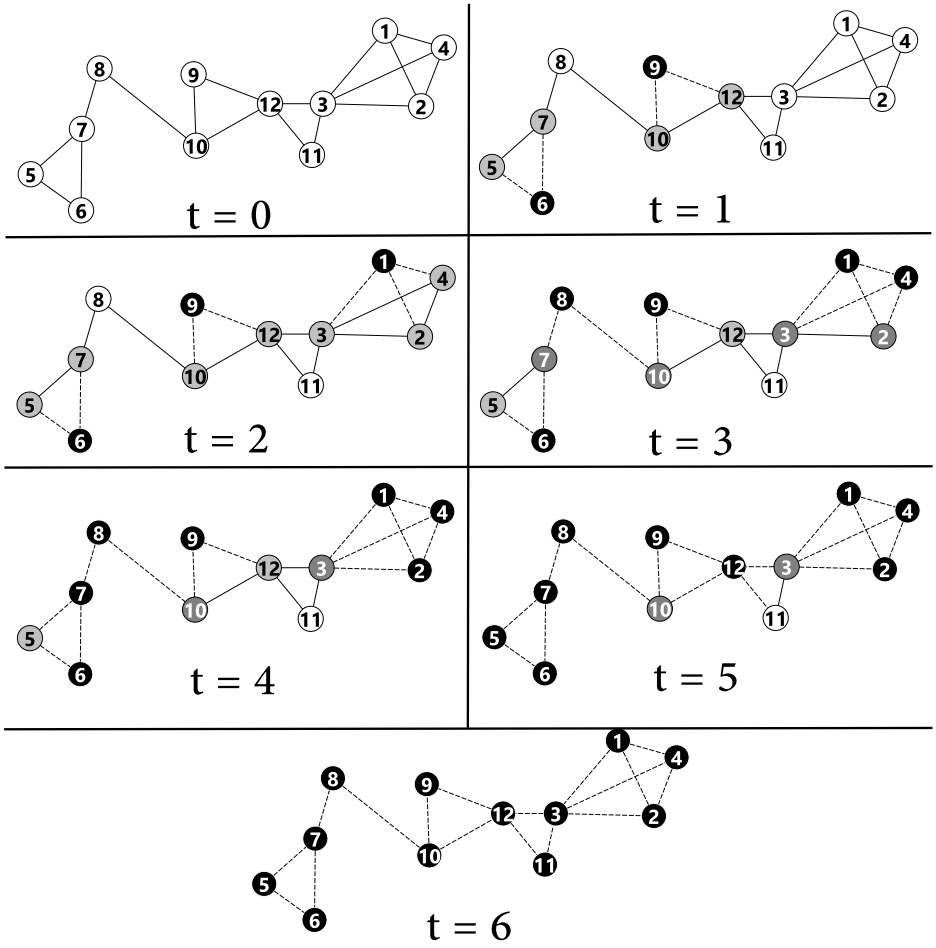


Figure 6.4.1: An illustration of the model. The color of the nodes represents how likely a node is to depart in the future, where white nodes are very unlikely to depart, and the level of grayness correlates with the probability departing. Whenever a node departs the network, it is marked as black, all its incident edges are removed, and all of its neighbors get affected by its departure by increasing their depart probability. The dotted edges are the removed edges. The color of the labels of the nodes is irrelevant; it is just for readability issue.

We assume the model starts with a *Decaying Network*, i.e, no further nodes or edges are added to the network. The main idea of the model is shown in Figure 6.4.1.

6.4.1 Probability Gain

At any point of time t where $t > 0$, the depart probability of a node w, that did not depart, changes from π_w^{t-1} to π_w^{t+1}, by adding *Probability Gain* $\Delta\pi_w^t$. Thus, a node w will depart at time $t+1$ with a probability π_w^{t+1} such that:

$$\pi_w^{t+1} = min\{1,\ \pi_w^{t-1} + \Delta\pi_w^t\} \tag{6.1}$$

If a node w did not depart the network at time, then we have two sets: $\overline{\Gamma}_w^{t-1}$ and $\underline{\Gamma}_w^{t-1}$, which are the sets of w's neighbors who departed and did not depart the network at $t-1$, respectively.

Probability gain due to one node departure:

We first define the probability gain due to the departure of a single neighbor v of node w at time point $t-1$, and then generalize it to w's neighbors that departed the network: $\overline{\Gamma}_w^{t-1}$. Now, the probability gain that node w will get due to the departure of its neighbor node v at $t-1$ is defined as:

$$\Delta\pi_w^t(v) = 1 - (1 - \pi_v^{t-1})(1 - \delta_{v,w}^{t-1}) \tag{6.2}$$

where the edge $e = \{v, w\} \in E(G^{t-2})$ and $e = \{v, w\} \notin E(G^{t-1})$ as $v \in \overline{\Gamma}_w^{t-1}$ and $w \in V(G^{t-1})$. We define the total probability gain produced by the departure of node v for all of its neighbors that did not depart (see Figure 6.4.2 for an illustration) is given by:

$$\Delta\pi^t(v) = \sum_{w \in \underline{\Gamma}_v^{t-1}} 1 - (1 - \pi_v^{t-1})(1 - \delta_{v,w}^{t-1}) \tag{6.3}$$

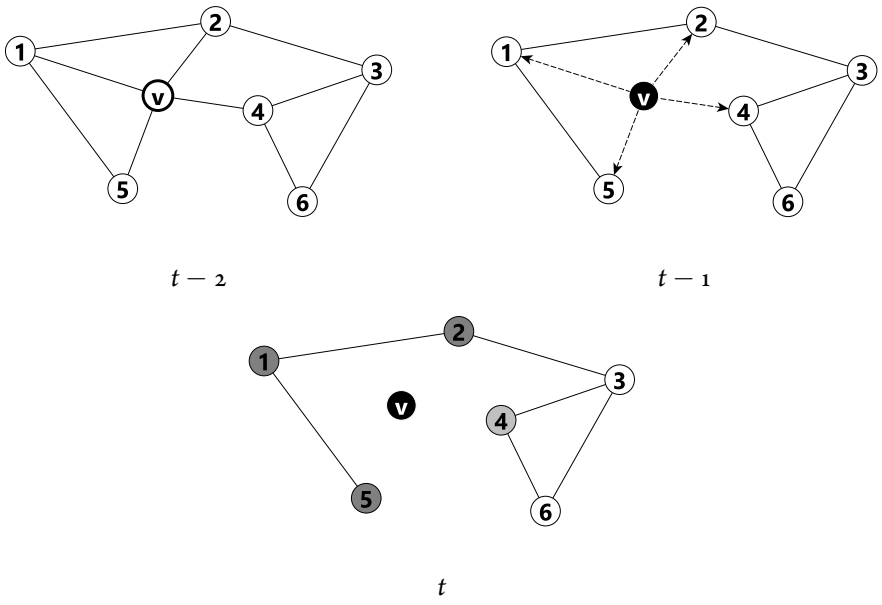


Figure 6.4.2: This figure shows how a node v affects all of its neighbors when it departs. At $t-2$, the node v has a depart probability of π_v^{t-2} which was gained by v's initial depart probability π_v^0 and possible probability gains caused earlier by departing neighbors, i.e., $\pi_v^{t-2} = \pi_v^0 + \sum_{t=1}^{t=t-3} \Delta\pi_v^t$. At time $t-1$, node v departs the network, which affects its neighbors by increasing the depart probability of nodes $1, 2, 4, 5$. Here we assume that the tie strength between v and nodes $1, 2, 5$ is greater than the tie strength between v and 4. That is why the nodes $1, 2, 5$ gain more depart probability than node 4, which is represented by the darker color of nodes $1, 2, 5$.

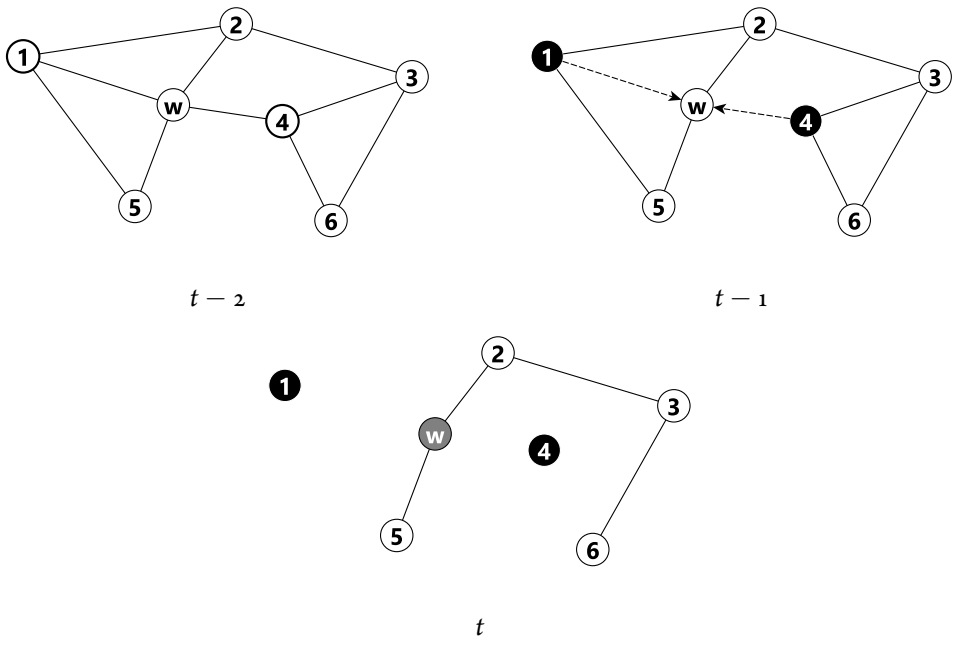


Figure 6.4.3: This figure shows how a node w is affected by the departure of its neighbors. At $t-2$, nodes $1, 4$ have the depart probabilities π_1^{t-2} and π_4^{t-2}, respectively, which were gained by the nodes' initial depart probabilities π_1^0 and π_4^0 and possible earlier probability gains. At time $t-1$, nodes $1, 4$ departed the network, affecting their neighbors. Here we are interested in node w. The departure of nodes $1, 4$ left node w with an increased depart probability at time t. Note that nodes $2, 3, 5, 6$ are also affected also by the departure of nodes $1, 4$, but for the sake of simplicity and visualization traceability, we focus on node w.

Probability gain due to multiple nodes departure:

We now generalize the probability gain induced by the departure of a single node to capture the impact of all the neighbors that have departed, i.e., $\overline{\Gamma}_w^{t-1}$.

$$\begin{aligned}\Delta\pi_w^t &= 1 - [\underbrace{(1-\xi_w^{t-1})}_{\text{Assures depart}}(\underbrace{\prod_{u\in\overline{\Gamma}_w^{t-1}}(1-\pi_u^{t-1})}_{\text{Depart probabilities effect}})(\underbrace{\prod_{u\in\overline{\Gamma}_w^{t-1}}(1-\delta_{u,w}^{t-1})}_{\text{Tie strength effect}})] \\ &= 1 - [(1-\xi_w^{t-1})(\prod_{u\in\overline{\Gamma}_w^{t-1}}(1-\pi_u^{t-1})(1-\delta_{u,w}^{t-1}))]\end{aligned} \tag{6.4}$$

where $\xi_w^{t-1} = \frac{|\overline{\Gamma}_w^{t-1}|}{|\Gamma_w^{t-1}|}$ and the quantity $1-\xi_w^{t-1}$ assures that when all of the neighbors of the node w departs, node w will (be forced to) depart, too, as it will be disconnected. Thus, Equation 6.1 becomes:

$$\pi_w^{t+1} = min\{1,\ \pi_w^{t-1} + 1 - [(1-\xi_w^{t-1})(\prod_{u\in\overline{\Gamma}_w^{t-1}}(1-\pi_u^{t-1})(1-\delta_{u,w}^{t-1}))]\} \tag{6.5}$$

6.5 Monotonicity and submodularity

In this section, we will show the monotonicity and submodularity properties of the model's equations.

Observation 6.5.1 (Monotonicity of the probability gain sum)**.** *The probability gain is always at least zero as all the parameters of Equation 6.3 are between zero and one. It follows that the probability sum is a monotone function.*

Observation 6.5.2 (Monotonicity of the probability gain products)**.** *For Equation 6.4, it is clear that the product of the probabiltiy gain is also a monotone function. That is because the probability gains and the parameters of the equation are between zero and one.*

Theorem 6.1. The depart probability gain function, Equation 6.3, is submodular.

Proof. Assume that a node w departed the network and the set $\underline{\Gamma}_w^{t-1}$ is the set of w's neighbors who did not depart. Using the Definition 3.2.5 and Equation 6.3, we prove the theorem by proving the following inequality:

$$\sum_{u\in S^*} 1-(1-\pi_w^{t-1})(1-\delta_{u,w}^{t-1}) - \sum_{u\in S} 1-(1-\pi_w^{t-1})(1-\delta_{u,w}^{t-1}) \geq \sum_{u\in T^*} 1-(1-\pi_w^{t-1})(1-\delta_{u,w}^{t-1}) - \sum_{u\in T} 1-(1-\pi_w^{t-1})(1-\delta_{u,w}^{t-1})$$

where $S^* = S\cup\{v\}$, $T^* = T\cup\{v\}$, and $S\subseteq T\subset \underline{\Gamma}_w^{t-1}$. For $S=T$, the equality holds.
Now we need to show that the inequality is correct for the case where $S\subset T$. Simplifying the previous equation, we get:

$$|S^*| - \sum_{u\in S^*}(1-\pi_w^{t-1})(1-\delta_{u,w}^{t-1}) - |S| + \sum_{u\in S}(1-\pi_w^{t-1})(1-\delta_{u,w}^{t-1}) \geq |T^*| - \sum_{u\in T^*}(1-\pi_w^{t-1})(1-\delta_{u,w}^{t-1}) - |T| + \sum_{u\in T}(1-\pi_w^{t-1})(1-\delta_{u,w}^{t-1})$$

Simplifying the previous inequality we obtain:

$$1+\sum_{u\in S}(1-\pi_w^{t-1})(1-\delta_{u,w}^{t-1}) - \sum_{u\in S\cup\{v\}}(1-\pi_w^{t-1})(1-\delta_{u,w}^{t-1}) \geq 1+\sum_{u\in T}(1-\pi_w^{t-1})(1-\delta_{u,w}^{t-1}) - \sum_{u\in T\cup\{v\}}(1-\pi_w^{t-1})(1-\delta_{u,w}^{t-1})$$

Using the fact that the sets $S\cup\{v\}$ and $T\cup\{v\}$ are larger than the sets S and T, respectively, which have one additional item, namely v, we can further simplify the previous inequality to:

$$\Delta\pi_v^{t-1}(w) + \sum_{u\in S}(1-\pi_w^{t-1})(1-\delta_{u,w}^{t-1}) - \sum_{u\in S}(1-\pi_w^{t-1})(1-\delta_{u,w}^{t-1}) \geq \Delta\pi_v^{t-1}(w) + \sum_{u\in T}(1-\pi_w^{t-1})(1-\delta_{u,w}^{t-1}) - \sum_{u\in T}(1-\pi_w^{t-1})(1-\delta_{u,w}^{t-1})$$

which always holds. Therefore, the original inequality for $S\subset T$ holds. □

Lemma 1. The depart probability gain function, Equation 6.4, is monotone, i.e., for a node w we have $\pi_w^t \leq \pi_w^{t+1}$ if node w did not depart the network at $t+1$. □

The second theorem in the technical paper [AZ18a] is not included here because its proof is flawed.

6.6 Analysis and simulation results

In this section, we will provide the analysis of the decayed StackExchange sub-websites and the results of the simulation of the model. The goal of this section is to show the potential of the model in capturing few major properties of interaction decay over time.

6.6.1 Dataset description

The StackExchange[3] is a network of question & answer website that contains sub-websites for specific topics, such as Computer Science, German Language, or Workplace, to name but a few. Before being available to the public permanently, each of these sub-websites must go through a beta version, where these beta versions become permanent for the public if they sustain a certain level of activity. If the sub-website does not meet the activity requirement, it is shut down. Some of these sub-websites go back and forth between being beta and closed. As a result, all of the users' accounts and their interactions (such as adding a new comment on a post and up-voting) are saved. This information is the dataset used for this chapter. For the analysis, we used the alive websites: (1) Statistics[4], (2) Latex[5], (3) German[6], (4) Apple[7] and (5) Music[8]. We used also the decayed websites: (1) Literature, (2) Theoretical physics and (3) Astronomy[9].

Business Startups decayed sub-website

We constructed undirected networks from this dataset for Business Startups closed sub-website as follows. Nodes are the users of the sub-website and an edge (interaction) between two nodes (users) A and B appears if user A commented on a question (or comment) posted by user B. Each edge has a timestamp that reflects its creation time (and generally the last time A and B interacted with each other in case of multiple interactions between A and B). This network is called G_o. The temporal networks are then extracted from the network G_o such that each network at time t (G_t) contains the edges (with their incident nodes) that have timestamps $\geq t$. As the original timestamps were continuous, we discretized on every 45 days. This number was chosen based on different ex-

[3] https://StackExchange.com/
[4] https://stats.stackexchange.com/
[5] https://tex.stackexchange.com/
[6] https://german.stackexchange.com/
[7] https://apple.stackexchange.com/
[8] https://music.stackexchange.com/
[9] A list of the closed sub-websites can be found here: https://archive.org/details/stackexchange and here https://area51.stackexchange.com/.

periments such that we get temporal networks that are not empty and are neither sparse nor very dense.

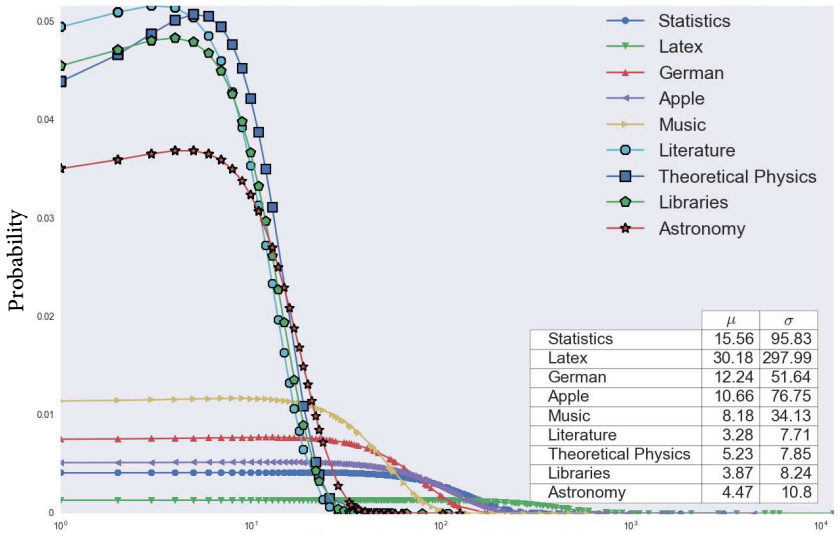
6.6.2 Users activity properties

Figure 6.6.1a shows the probability distribution function (PDF) of the number of user comments for alive and decayed sub-websites. The figure shows that the decayed sub-websites clearly have different distribution characteristics with a low mean and low standard deviation. A similar difference is found in Figure 6.6.1b and Figure 6.6.1c, which represent the PDF of the users' total received *Reputation* score and *Upvotes* score, respectively. These two properties reflect the level of knowledge and experience that the members of a website have. For the decayed websites, it is clear that, on average, the members have a much lower reputation score and fewer upvotes than those in the alive sub-websites. The three figures 6.6.1 (Panel a), 6.6.1b, and 6.6.1c show that there is less social activity on the decayed sub-websites, which may be used as an indication for studying the future of the alive sub-websites and decay patterns found in the decayed sub-websites, which will be the topic of Chapter 8 and Chapter 7.

Model simulation

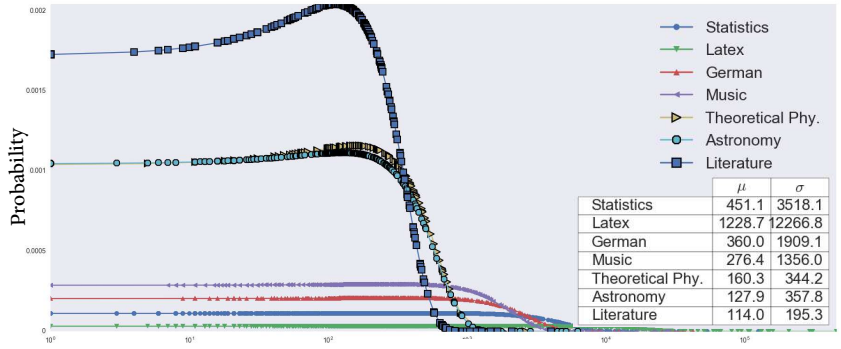
Algorithm 6.1 describes the steps we followed in our experiments for simulating the model described in this chapter. We simulated the model using Equation 6.5, which is the probability gain due to multiple nodes departure. Line 2 initializes the initial depart probability π_v°. We selected values from 0.0005 to 0.045 in 0.0005 increments. For each of these values, the model was run and simulated the probability gain as described in Equation 6.4. The update step in line 13 simulated Equation 6.5. The result of the algorithm was a set of graphs that are used for the analysis. The output of this algorithm resulted in a large number of graphs. For example, in the case of the Startup Business sub-website, we analyzed more than 200*k* graphs for 250 runs for each probability to get higher confidence in the results. The tie strength was a normalized edge weight, with the weight being the frequency of the interaction between two nodes before the departure of one.

In Figure 6.6.2, we show the macro properties of the *real* networks of the Startup Business website over time. The network evolution shows a clear decay, which is represented as a decrease in the number of nodes. This decrease was associated with a decrease in the average degrees of the nodes over time and also with a decrease in the nodes' coreness [BZ03]. Another macro measure we used was network density. Figure 6.6.2c shows an increase in density over time. This increase is due to the early departure of nodes with fewer degrees, i.e., nodes that are part of dense subgraphs appear to depart the network later than others (this property will be further investigated in Chapter 7).



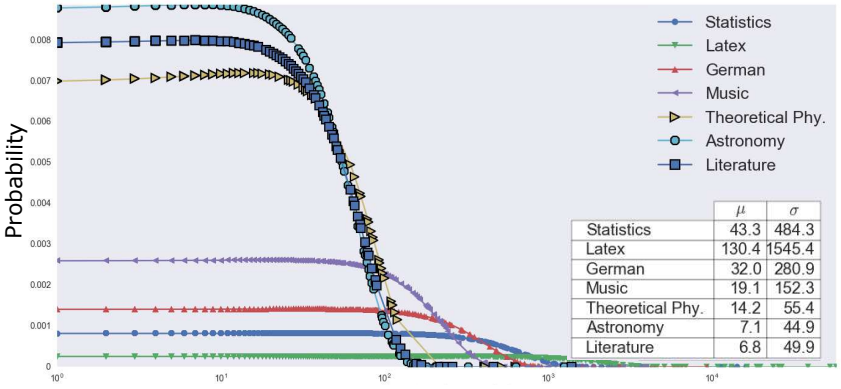
	μ	σ
Statistics	15.56	95.83
Latex	30.18	297.99
German	12.24	51.64
Apple	10.66	76.75
Music	8.18	34.13
Literature	3.28	7.71
Theoretical Physics	5.23	7.85
Libraries	3.87	8.24
Astronomy	4.47	10.8

(a) The PDF for the number of comments by user (log-scaled)



	μ	σ
Statistics	451.1	3518.1
Latex	1228.7	12266.8
German	360.0	1909.1
Music	276.4	1356.0
Theoretical Phy.	160.3	344.2
Astronomy	127.9	357.8
Literature	114.0	195.3

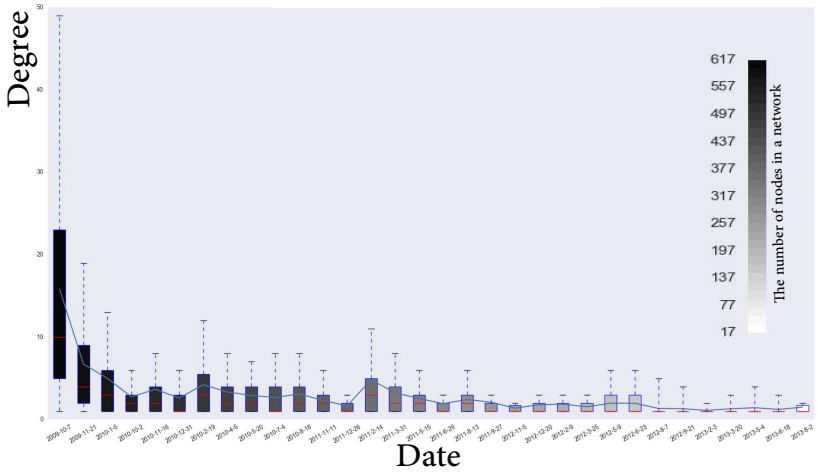
(b) The PDF for the reputation of the users (log-scaled)



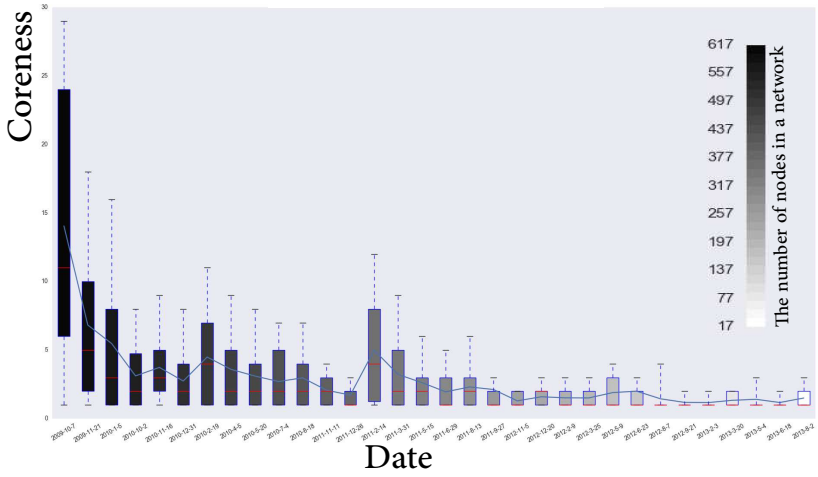
	μ	σ
Statistics	43.3	484.3
Latex	130.4	1545.4
German	32.0	280.9
Music	19.1	152.3
Theoretical Phy.	14.2	55.4
Astronomy	7.1	44.9
Literature	6.8	49.9

(c) The PDF for the upvotes received by the users (log-scaled)

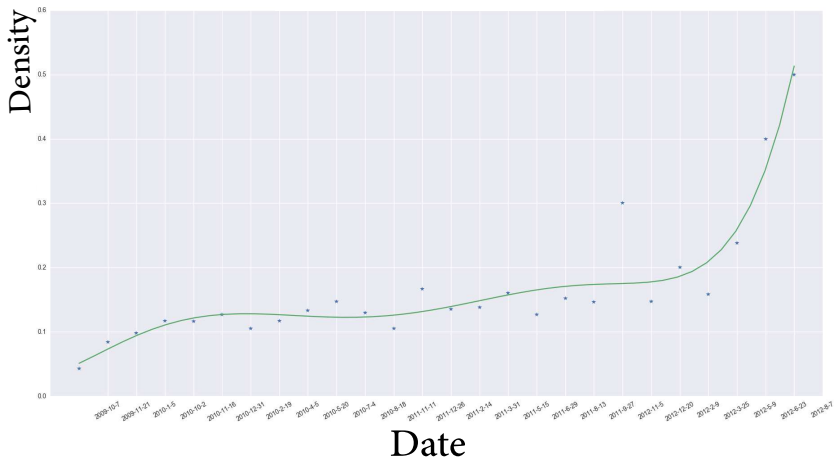
Figure 6.6.1: Characteristics of the interaction decay on the decayed and alive sub-websites of different StackExchange sub-websites. The figures show the probability distributions function (PDF) of different types of interactions on different sub-websites. Markers with bold borders are decayed sub-websites, μ is the mean, and σ is the standard deviation. The figures clearly show that the decayed networks have different distribution properties than the alive networks.



(a) Network degrees over time for Startups sub-website



(b) Network coreness over time for Startups sub-website



(c) Network density over time for Startups sub-website

Figure 6.6.2: Macro properties of the real networks under decay for the Startup business site. Figures 6.6.2.a, 6.6.2.b, and 6.6.2.c show the degrees of the nodes, the node coreness, and the network density over time.

Algorithm 6.1: Model simulation.

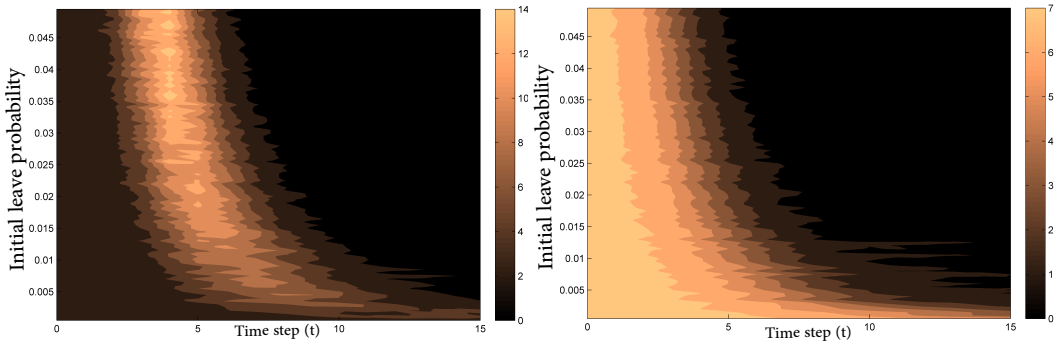
Input: Graph G_0

forall $v \in V_{G_0}$ **do**
  initialize π_v^0 // assign initial depart probability to each node
$t = 0$,
$G_t = G_0$
$\mathbf{G} = \{G_t\}$
while G_t *is not empty* **do**
  $\mathbb{D}_t = \emptyset$ // $\mathbb{D}_t$ contains the nodes that departed at time t
  $t = t + 1$
  forall $v \in V_{G_t}$ **do**
    // decide randomly whether v will depart or not based on π_v^t
    if *Depart*$(v,\ \pi_v^t)$ *is True* **then**
      $\mathbb{D}_t = \mathbb{D}_t \cup \{v\}$
  forall $u \in V_{G_t}\ \&\ u \notin \mathbb{D}_t\ \&\ \overline{\Gamma}_u^{t-1} \neq \emptyset$ **do**
    Update$(\pi_u^t,\ \overline{\Gamma}_u^{t-1})$// update the depart probability for the remaining nodes (using Equation 6.5)
  $V_{G_t} = V_{G_t} \setminus \mathbb{D}_t$ // remove the departed nodes and their incident edges from G_t
  $\mathbf{G} = \mathbf{G} \cup \{G_t\}$

Output: $\mathbf{G} = \{G_0, G_1, \cdots, G_{n-1}\}$ where G_n is an empty graph

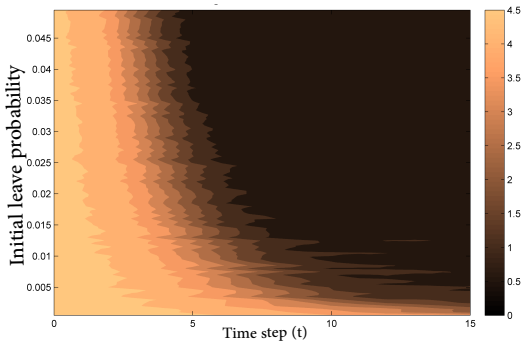
Next, we will show the results of the *model simulation*. Figure 6.6.3a shows the number of components in the network over time for different values of π_v^0. The number of components starts to increase to a maximum value before starting to decrease. The reason is that at the beginning, the model starts with a one-connected component graph and after each step, some nodes are removed due to the depart probability. The departure of some nodes results in a disconnected graph with more disconnected components. The number of these disconnected components increases until they are composed only of triples or simple edges. As a result, a node that departs from these triples or these edges will no longer increase the number of components.

Figure 6.6.3b and Figure 6.6.3c show similar behavior for the average degree and the average coreness over time, respectively. The more nodes that are being removed from the network, the fewer edges remain, and thus the average degree and the average coreness decrease uniformly over time. This behavior of the model is similar to the real data presented in Figure 6.6.2. The last global measure we used is the network density, as shown in Figure 6.6.3d. The density of the simulated networks increases over time for the same reason stated for the real networks in Figure 6.6.2. These results show that the model exhibits a behavior that is close to the real behavior of networks under decay.

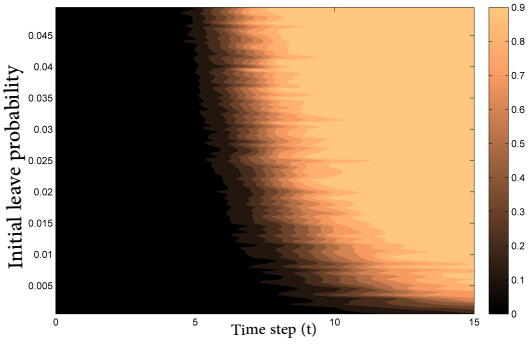


(a) #Connected components over time.

(b) Average degree over time.



(c) Average coreness over time.



(d) Density over time.

Figure 6.6.3: The results of multiple global measures of the simulation of the model. Figures 6.6.3a, 6.6.3b, 6.6.3c, and 6.6.3d show the number of components, the average degree, the average coreness, and the density of the network over time for different values of the initial depart probability π_v^o, respectively. The model started with G_o of the business startups as the input network and simulates the decay over time.

6.7 Applications of the model

There are different applications in which the model can be utilized.

- *Social network resilience:* Resilience against huge disruptions in social networks is not a well-studied subject. We think that the model provides a first step towards engineering a resilient social network by understanding the decay dynamics of a network.
- *Depart cascade detection*: The departure of one member is not as harmful to networks that seek growth as a cascade of members departing. The model captures the dynamics of depart cascades by observing the depart probabilities of the nodes and their increase.

6.8 Conclusion

In this chapter, we presented a preliminary empirical analysis of the social decay dynamics of the closed (decayed) StackExchange sub-websites. The closed sub-websites showed inactivity of interactions among the members of these sub-websites, which might have caused their decay. We modeled these interactions among the members of these sub-websites as a network, which enabled us to build a model for understanding the decay dynamics. Then, we presented a model for capturing the decay dynamics in social networks. The model is a probabilistic model that assumes that the departure of a member in a social network affects the depart of its neighbors. In this chapter, we also presented some mathematical properties and proved them. Also, we presented the macro network properties of real networks under decay and compared these results with the results of the model simulation. The comparison of the model and the real networks under decay showed similar behavior for four macro properties, which demonstrates the potential of the model for different usage purposes.

7

Pattern and cascade analysis of decayed communities

7.1 Synopsis

In this chapter[1], an extensive analysis and modeling of the inactivity cascades in the StackExchange websites will be presented. The work in this chapter includes defining a cascade's *size*, *virality*, *duration*, *coreness*, and *similarity*. The analysis will reveal valuable insights regarding the patterns found in inactivity cascades and a comparison between these patterns in decayed and alive websites. In addition, a prediction model will be presented that aims at predicting cascade *size* and *virality* of inactivity cascades.

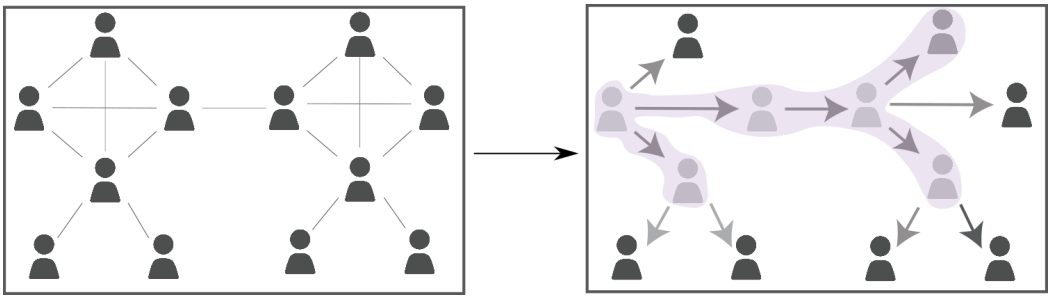


Figure 7.1.1: The goal of this chapter is to model and analyze the inactivity cascades that occur in social networks.

[1]This chapter is based on the work [Abu18b].

7.2 Introduction

In recent years, online social networks (OSNs) have proven their suitability as a new medium for sharing news and knowledge, expressing opinions, finding jobs, and many other things. In the literature, there are many studies that focus on the growth dynamics of a network, starting with the seminal works of Barabási and Albert [BA99] and Watts and Strogatz [WS98], which formed the basis for the field of network science, via many studies examining the growth dynamics of social networks [JGN01, KW06, CSC+06, MKG+08] to community membership evolution [BHKL06]. All of these works provide methods and models for analyzing and understanding growth dynamics in social networks. However, the dynamics of members' interactions in social networks is not always growth dynamics; many online social platforms have gone through *decay* dynamics in terms of low activity among their members and/or members leaving or deleting their accounts. Online social platforms such as MySpace and Orkut are now out of service after being very active for years, and they are now examples of decayed online social networks. This phenomenon has not been studied well in the literature; the causes and mechanics of decay, as well as the prevention of decay, are still open questions that need to be answered.

In this chapter, we approach the decay dynamics problem from a network perspective by modeling the members as network nodes and their social interactions as temporal edges. We aim to better understand the patterns that occur during the decay process by investigating what we call *inactivity cascades*, which were extracted from decayed StackExchange sub-websites. These inactivity cascades are mainly constructed from the structure of the used social networks. The network structure has already been shown to be crucial for understanding the dynamics of any process that takes place on top of a network such as the structure of the World Wide Web networks [Kle98, HA99], and social network analysis [Mor53, Mil67, KJB+90, DYB03]. Moreover, network structure is used in many studies for understanding the dynamics of the processes that take place over networks, such as epidemic dynamics [Kee05, ZLZ+18], knowledge spread [CJ04], and knowledge transfer [RM03]. The information produced and evolved on the StackExchange website as an information exchange platform also connects this chapter to the area of *information dynamics* [HA04, LPP11, ZLZ+18]; thus, the work in this chapter can be seen from the perspective of the decay of the information production process on the StackExchange website as a medium of knowledge production and sharing.

7.3 Related work

This chapter is related to studies and works that are concerned with decay or inactivity dynamics in social networks. In this section, we present the related work and show how this chapter compares these related work.

7.3.1 Theoretical models for interaction decay

Due to limitations on existing data about interaction decay, researchers have focused on the theoretical aspects of the decay process based on random networks. For example, Dorogovtsev and Mendes [DM00] presented a model for understanding the properties of random networks if edges are removed, signaling that the dynamics of a network is not limited to adding nodes and/or edges. Fenner et al. [FLL06] contributed a theoretical model for generalizing the *the-rich-get-richer* model of network evolution, which focuses mainly on growth dynamics, with an extension to link deletion in the Web network. Their model implicitly assumes that dynamics is not limited to growth dynamics, but may include link removal. Decay dynamics modeling also raised some computational aspects of the decay dynamics problem. Bhawalkar [BKL+15] and Zhang et al. [ZZQ+17] provided a theoretical model and mathematical framework for finding the set of nodes whose deletion generates the smallest k-core sub-graph of a network. Both studies focus on the computational challenge of the decay dynamics modeling. Their works assure that the node removal problem is relevant in social and other networks.

7.3.2 Interaction decay as a common phenomenon in social networks dynamics

Burt [Bur00] was among the first to analyze the decay of the interactions and used a financial network for among bank members. The author modeled a decay function that captures the decay rate of social interactions over four years. Later, with the rise of many social networks and social platforms, research primarily focused on growth dynamics, with very few works dealing with decay dynamics. Torkjazi et al. [TRW09] studied users' migration from MySpace to Facebook when the latter was getting more attention from users. Their study suggests that OSNs have a life cycle that may end with service decay. Dev et al. [DGH+18] studied the reasons behind the failure of what they call *Knowledge Markets*, such as StackExchange. They utilized economic production models in order to understand the dynamics of knowledge generated in these knowledge markets. Asur et al. [AHSW11] approached the activity of users from a trend analysis perspective on Twitter, shedding light on what causes some tweets to be trendy. They also found that the decay dynamics of a trend follows a linear function. Wu et al. [WDSF+13] predicted the activity and inactivity of members of the DBLP co-authorship dataset by modeling the dynamics of the social engagement of the members of DBLP. They also provide insights regarding the characteristics of the members who left the networks using network measures. Community activity has also been studied by Kairam et al. [KWL12], who developed machine learning prediction models to predict community longevity. The authors also provide insights into the factors that contribute to keeping online communities active. Similarly, Patil et al. [PLG13] provided a machine learning framework for investigating group stability in online social networks. Cannarella and Spechler built an epidemic model for predicting the dynamics of the members of Facebook [CS14]. The results showed that Facebook would lose

80% of its users between 2015 and 2017[2].

Ribeiro [Rib14] studied user activity and inactivity by providing a model that uses the number of daily active users as an indicator of the dynamics in membership-based websites. This author also presented a prediction model for predicting whether a community will continue to grow or not, similar to the work by Kairam et al. [KWL12]. Malliaros and Vazirgiannis [MV13] and Bauckhage et al. [BK14] contributed models for social engagement describing the activity and inactivity of members of social networks based on game theory. Similar to the work in [MV13], Garcia et al. [GMS13] investigated the decay of the Friendster social network using game theory. As one of the results of their work, Garcia et al. argue that decay has a direction, which starts from nodes with less coreness; this was later refuted by Seki and Nakamura [SN17], who developed a model that shows that decay starts from nodes with higher coreness.

7.4 Contribution

The previous works fall into two categories: (1) studies that consider both growth and decay processes as typical behavior of online social networks, and (2) studies that approach the decay process in a social context only via models, which were not validated with real decayed inactivity data using temporal snapshots. Although the first category seems to be more realistic, none of the related work in this category provides any thorough analysis of the mechanics of the decay process compared to the rich analysis of growth dynamics. This means there is little insight into the decay process of online social interaction that would serve to better understand online behavior. As a result, the authors of the work in the second category of the related work realized that decay dynamics needs to be considered as a separate process and requires further thorough investigation, particularly after the decline of many online social networks such as MySpace and Friendster. However, these works used either synthesized data or did not consider the temporal aspect of the problem. The use of synthesized data led to contradictory conclusions on the same research question (see the work by Garcia et al. [GMS13] and an opposing argument by Seki and Nakamura [SN17] regarding the decay direction and our attempt to resolve this issue in Section 7.7).

This contribution of this chapter fills the gap by focusing only on decay dynamics using real temporal data from decayed online social communities. Furthermore, we enhance our analysis using inactivity cascades, which, to the best of our knowledge, have not been covered before. This enables us to better understand the characteristics of *real* inactivity cascades and, hence, helps us gain more insights into the online behavior of humans.

Based on that, the contributions of this chapter are summarized as follows:

[2]The same model was used by Facebook researchers and predicted that Princeton University would lose half of its students by 2018. See: https://www.Facebook.com/notes/mike-develin/debunking-princeton/10151947421191849/

- Extracting and analyzing inactivity cascades from the decayed and alive sub-websites of StackExchange.
- Devising measures for understanding the decay process and the decay patterns in both decayed and alive sub-websites.
- Identifying different inactivity patterns in alive and decayed sub-websites.
- Finding empirical evidence that an inactivity cascade cannot be described by only one network measure.
- Building a machine learning framework for predicting the size and virality of inactivity cascades.

The previous contributions can be seen as two parts: (1) *analysis* of the decay process via cascade modeling, and (2) *prediction* of cascade's properties. These two parts are complementary because analysis without prediction limits our control over these platforms, and because predicting the properties of decay requires a better understanding of the decay process itself so that we can provide a good prediction model.

7.5 Definitions and methods

7.5.1 Networks and measures

Considering temporal graphs, a graph G that is observed at a specific point of time t is denoted as $G_t = (V_{G_t}, E_{G_t})$, where V_{G_t} and E_{G_t} are the set of nodes and edges, respectively, that are observed at time point t in the graph G_t. Thus, the set of graphs $\mathbf{G} = \{G_{t_0}, G_{t_1}, \cdots, G_{t_k}\}$ is a temporal structure of a graph at equally separated discrete time points $\{t = 0, t = 1, \cdots, t = k\}$. We call G_{t_0} the *initial network* and its vertices *core* nodes. The last observed time of a node v is denoted by $\tau(v)$, where v has a degree of *zero* in the graph $G_{\tau(v)+1}$. The edges of the graph G_{t_0} have timestamps that refer to the last time an edge is activated between the incident nodes, that is because an edge can appear multiple times at different time points. We define $\phi(e) = t \in \{0, 1, \cdots, k\}$ as a mapping function that maps the real creation time of an edge $e \in E_{G_{t_0}}$ to one of the time points $\{0, 1, \cdots, k\}$, such that the creation time of the edge e appears in the smallest possible time interval $[t, k]$ (i.e., the largest possible t). Now, given G_{t_0} (whose edges are associated with timestamps), we construct a set of graphs $\mathbf{G} = \{G_{t_0}, G_{t_1}, \cdots, G_{t_k}\}$ as follows. For each time point t there is an associated network G_t such that:

- $V(G_t) \subseteq V(G_{t-1})$,
- $E(G_t) \subseteq E(G_{t-1})$, and
- $\forall e \in E(G_t), \phi(e) \geq t$

7.5.2 Inactivity cascades

An *Inactivity cascade tree* $\mathcal{I}$, a *cascade* for short, is a rooted tree where each directed edge $e = (u, v)$ contains two nodes such that the *last observed time* points of nodes v and u were $\tau(v) = t'$ and $\tau(u) = t''$, respectively, such that $t' \geq t''$ and $e = \{u, v\} \in E_{G_o}$. The construction of an inactivity cascade is an iterative process that generates only trees. The specific steps for constructing the inactivity cascades is described in Algorithm 7.1. The root of a cascade $\mathcal{I}$ is called *cascade initiator*, which is any node that becomes inactive first. The notation $\mathcal{I}_v$ is a cascade $\mathcal{I}$ that was initiated by the node v and $V_{\mathcal{I}}$ and $E_{\mathcal{I}}$ are the set of nodes and the set of edges in a cascade $\mathcal{I}$, respectively. We could have multiple initiators, and thus multiple cascades, and it could happen that two of them are connected in G_o. The number of nodes in a cascade is called cascade *size*. Algorithm 7.1 describes the steps we followed for extracting inactivity cascades, and Figure 7.5.1 shows a toy example on how the algorithm works. For the set of graphs $\mathbf{G}$, a set of inactivity cascade trees $\mathbf{I}$ is extracted.

Algorithm 7.1: The steps for extracting inactivity cascades, **I**, from the set of temporal networks networks **G**.

Input : $G_0, G_1, \cdots, G_k$ // The set of temporal networks

Init : $\mathbf{I} = \emptyset, S = (L_0, L_1, \cdots, L_k)$

// $L_i \in S$ is defined as: $\{v \mid \tau(v) = i, \forall v \in V_{G_0}\}$. L_0 contains the initiators of the cascades

foreach $v \in L_0$ **do**
  $\mathcal{I} = (V_{\mathcal{I}} = (v), E_{\mathcal{I}} = \emptyset)$ // Start a new cascade $\mathcal{I}$. $V_{\mathcal{I}}$ is a temporally ordered sequence
  foreach $q \in L_0$ **do**
    // Check if the initiator v is connected to another initiator q
    if $e = \{v, q\} \in E_{G_0}$ **then**
      $V_{\mathcal{I}} = V_{\mathcal{I}} \oplus q$ // Append the node q to $V_{\mathcal{I}}$
      $E_{\mathcal{I}} = E_{\mathcal{I}} \cup \{e = (v, q)\}$ // Add a directed edge (v,q) to the cascade
  // Check if the initiator v is connected to any non-initiator node
  foreach $L \in (L_1, \cdots, L_k)$ **do**
    foreach $u \in L$ **do**
      // Check if u is connected to any other nodes in the cascade $\mathcal{I}$
      foreach $w \in V_{\mathcal{I}}$ **do**
        if $e = \{u, w\} \in E_{G_0}$ **then**
          $V_{\mathcal{I}} = V_{\mathcal{I}} \oplus u$
          $E_{\mathcal{I}} = E_{\mathcal{I}} \cup \{e = (w, u)\}$
          // The break prevents cycles from being formed in $\mathcal{I}$
          break
  // Add the extracted cascade $\mathcal{I}$ to the set of all cascades **I**
  $\mathbf{I} = \mathbf{I} \cup \{\mathcal{I}\}$

Output: I

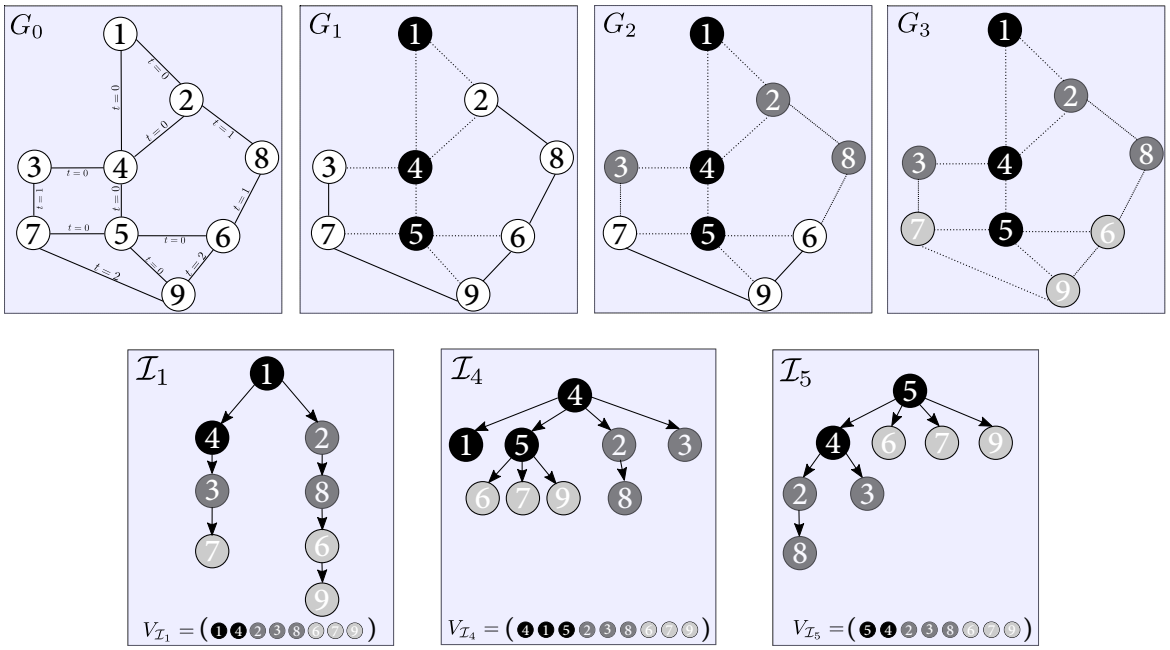


Figure 7.5.1: An example on how Algorithm 7.1 works. Network G_o is the initial network from which we construct the temporal networks G_1 to G_3. The network G_o includes all nodes and edges we are observing. The network G_o also contains edge creation time, which refers to the last time an edge is activated. The white nodes are nodes that are not inactive, and the label color of nodes is irrelevant. In the network G_1 nodes 1, 4, and 5 were not observed because all of their incident edges were not observed, i.e., nodes 1,4, and 5 were observed lastly in G_o because all of their incident edges have timestamp *zero* in the network G_o. Thus, $L_o = \{1, 4, 5\}$ (black nodes) is the set of the initiators for the cascades to be extracted. Later in the network G_2, more nodes become inactive; thus, we have $L_1 = \{2, 3, 8\}$ (gray nodes). Likewise, we have $L_2 = \{6, 7, 9\}$ (light gray nodes). The three trees in the bottom are the cascades $\mathcal{I}_1$, $\mathcal{I}_4$, and $\mathcal{I}_5$ initiated by nodes $1, 4, 5$, respectively. Note that nodes in a sequence $V_{\mathcal{I}_i}$ are ordered according to their leave time ascendingly such that all nodes in L_j appear before the nodes in L_{j+1} in any $V_{\mathcal{I}_i}$. This affects which source we are using when adding a new node (edge) to the tree. For example, in cascade $\mathcal{I}_1$, node 2 is connected to node 1 and not node 4; that is because node 1 appears first in the set $V_{\mathcal{I}_1}$. Though nodes 1 and 4 became inactive at the same time. For the same reason, node 2 is connected to node 4, not to node 1, in cascade $\mathcal{I}_4$.

7.5.3 CASCADE DURATION

Edge formation period for an edge $e = (u, v)$, where $e \in E_{\mathcal{I}}$ (the set of the edges in a cascade $\mathcal{I}$), is defined as $\tau(v) - \tau(u)$. Based on that, we measure the normalized *cascade duration*, which is defined as:

$$CD_{\mathcal{I}} = \frac{1}{k \cdot |E_{\mathcal{I}}|} \sum_{e=(u,v)\in E_{\mathcal{I}}} \tau(v) - \tau(u), \tag{7.1}$$

It is clear that the term $\tau(v) - \tau(u)$ can be larger than 1 as we have multiple time steps, thus we need to normalize by dividing by k.

7.5.4 CASCADE VIRALITY (WIENER INDEX)

The *virality* of a cascade $\mathcal{I}$ measures how far the effect of the initiator of a cascade goes [GAHW15]. The measure[3] is defined as:

$$v_{\mathcal{I}} = \frac{1}{n(n-1)} \sum_{v,u \in V_{\mathcal{I}}} d(u,v), \tag{7.2}$$

where $d(u,v)$ is the length of the shortest path between nodes u and v, and n is the number of nodes in a cascade. Throughout this chapter, the terms Wiener Index and virality are used interchangeably.

7.5.5 CASCADE SIMILARITY

Although the node inactivity time is fixed, the extracted cascades based on Algorithm 7.1 are different mainly because each cascade has different initiator. We think that having multiple cascades is more realistic than having only one cascade per network because the inactivity process can be initiated by any node and possibly by multiple nodes at the same time. For example, the extracted cascades in Figure 7.5.1 are clearly not identical, though, there is some structural similarity between them. Thus, we can find the similarity between any two cascades so that we get more insights regarding the extracted cascades. We propose a Jaccard-like similarity measure of two cascades. To get more structural similarity, we consider the structural properties of a cascade by considering the neighborhood of nodes in cascades. That is, if a node is shared between two cascades and has many shared neighbors in the two cascades, then the two cascades are assumed to be more similar. Thus, we define:

$$sim(\mathcal{I}_1, \mathcal{I}_2) = \frac{1}{|V_{\mathcal{I}_1} \cap V_{\mathcal{I}_2}|} \sum_{z \in V_{\mathcal{I}_1} \cap V_{\mathcal{I}_2}} \frac{|N(z_{\mathcal{I}_1}) \cap N(z_{\mathcal{I}_2}|)}{|N(z_{\mathcal{I}_1}) \cup N(z_{\mathcal{I}_2})|} \tag{7.3}$$

7.5.6 STATISTICAL DIVERGENCE

In this section, we introduce statistical measures that will be used later in the experiments. We will use the following measures to get statistical significance of the results in Section 7.7.

Definition 7.5.1. The *cumulative distribution function* (CDF) for a discrete random variable X is defined as:

$$F_X(x) = P(X \leq x) = \sum_{t \leq x} f(t). \tag{7.4}$$

If X is continuous, then the CDF is defined as: $F_X(x) = \int_{-\infty}^{x} f_X(t)dt$. Similarly, the Complementary CDF is defined as $\overline{F}_X(x) = P(X > x) = \sum_{t > x} f(t)$.

Definition 7.5.2. The *Kolmogorov-Smirnov Test* (KS-Test) is a statistical test[4] that tells whether two different samples were drawn from the same distribution or not. The test is used to compare two

[3] This measure was originally proposed as *Wiener Index* [Wie47].
[4] Throughout this chapter the term statistical test refers to KS-Test.

patterns in order to determine whether they are the same or different. Informally, it is the maximum absolute distance between the two CDFs of the two samples. More formally, for two CDFs, F_1 and F_2, the KS-Test statistics D is defined as:

$$D_{KS} = \sup_{-\infty<x<\infty} |F_1(x) - F_2(x)|, \tag{7.5}$$

where $\sup_{-\infty<x<\infty}$ is the supremum of a set.

Definition 7.5.3. *Entropic similarity of patterns*: Shannon Entropy [Sha01] quantifies the information in a discrete random variable $x \sim p(x)$ as follows:

$$H(P) = -\sum_{i=1}^{n} p(x_i) \cdot log\, p(x_i). \tag{7.6}$$

Given two probability distributions P and Q, the *Kullback-Leibler divergence* [KL51] (D_{KL}) is a measure that finds how similar these two distributions are; it is defined as:

$$D_{KL}(P||Q) = \sum_{i=1}^{n} p(x_i) \cdot log\frac{p(x_i)}{q(x_i)}. \tag{7.7}$$

Definition 7.5.4. The *Jensen-Shannon divergence* [Lin91] is then defined as:

$$D_{JS}(P, Q) = \frac{1}{2}[D_{KL}(P||R) + D_{KL}(Q||R)], \tag{7.8}$$

where $R = \frac{1}{2}(P + Q)$, which is a symmetric distance variation of D_{KL}.

7.6 Dataset

The StackExchange[5] is a network of question & answer website that contains sub-websites for specific topics, such as Computer Science, German Language, or Workplace, to name but a few. Before being available to the public permanently, each of these sub-websites must go through a beta version, where these beta versions become permanent for the public if they sustain a certain level of activity. If the sub-website does not meet the activity requirement, it is shut down. Some of these sub-websites go back and forth between being beta and closed. As a result, all of the users' accounts and their interactions (such as adding a new comment on a post and up-voting) are saved. This information is the dataset used for this chapter[6].

[5] https://StackExchange.com/

[6] A list of the closed sub-websites can be found here: https://archive.org/details/stackexchange and here https://area51.stackexchange.com/.

7.6.1 Network construction

We parsed, structured, and analyzed a set of closed (decayed) sub-websites as an example of communities that underwent decay dynamics and alive sub-websites. The decayed sub-websites we considered in this chapter are *Business Startups* and *Economics*. In addition to that, we also have data for alive websites, such as *Statistics*, *Latex*, and *Music*. We used both types in order to make a comparison, if possible, between the patterns and cascades found in the alive and the decayed communities. One advantage of this dataset is that it contains all the temporal information needed to construct temporal social networks based on the interactions among the users. So, we constructed the networks based on the following steps:

- **Network nodes:** The nodes are the members of the StackExchange. Each member could have multiple user accounts, each on a different sub-website.
- **Network edges:** The edges are the interactions among the users in one sub-website. All members participated in one question page are connected, i.e., the user who wrote the question, users who wrote answers, users who commented on the question or on the answers are all connected to each other. We think of that as a small group of people discussing a topic together. Thus, we see all of the members interacting with each other. We chose to make it undirected because we assumed if a user wrote something on a question page (e.g., an answer or a comment), then it is an indication that this user is interested in this question and all of the discussion around it as a whole. Hence, this user is interacting with the whole users participating in this question for either knowledge sharing or acquisition.
- **Edge timestamps:** Whenever we add an edge to the network, we associate a timestamp with it. This timestamp is simply the time at which the interaction took place. For example, If user A wrote an answer to a question posted by user B at time t_1, then the edge $e = \{A, B\}$ has timestamp t_1, e.g., $\phi(e) = t_1$. If another interaction between A and B appeared again at $t2$ for the same question (some users first comment on a question to clarify it then they write an answer to the question) or for another question, then the timestamp between A and B is updated to t_2, i.e., $\phi(e) = t_2$. Thus we have the most recent interaction timestamp on each edge. The resulted network is G_0, which is a dense network that includes all nodes and edges that we are interested in.
- **Temporal networks:** Now, we divide the observation period **T** (the time between launching a sub-website until its closure for the decayed sub-websites, or until December 2015 for the alive sub-websites) into equal k windows of equal length l, where l is a certain number of days. So, we divide **T**, which is virtually a continuous period of time because users can interact at any time without restriction, into discrete time points $0, t_1 = l, t_2 = 2l, \cdots, t_k = kl$. For each time point t there is an associated network G_t that is constructed as described in Section 7.5.1.

Any node with no edges in any G_t is considered inactive, thus, it is removed.

The choice of each k is a design decision that is based on the longevity of the observation period for the dataset we have. For example, the Economics decayed sub-website has $k = 10$, as longer values of k than 10 resulted in very sparse disconnected graphs. Table 7.6.1 shows a summary of the datasets used, the observation period for the interactions, the number of networks constructed, information about the first and the last network, and the number of extracted cascades. The observation period for the datasets differed according to their active periods; e.g., for the decayed

sub-websites (the first two rows in Table 7.6.1), the last observation day was the last day these sub-websites were active. Conversely, the last three sub-websites are still alive, so the last observation day was the same. Note that the set of nodes V_{G_0} refers to the *core* nodes used for constructing the networks, which means other nodes emerging in-between were ignored. The core nodes were members with a reputation score of at least 500; we tried smaller values than 500, e.g., 100, 200, 300, and 400, for the reputation score and the resulting temporal networks were too sparse, with too many disconnected components, which impedes any subsequent analysis. The reason for this in the context of the StackExchange websites is that there are many users who come only for one question or make only one comment and then do not appear again on the platform. We consider these users as outliers regarding the platform's core activity, e.g., information production; thus, the chosen value, i.e., reputation score $\geq$ 500, is justified from the lower bound side. On the other side, we did not select larger values because there are few users in some communities who have reputation score larger than 500. Thus, the chosen value is justified from the upper bound side.

If a node from the core nodes becomes inactive at some point of time t' and then becomes active again at t'' where $t'' > t'$, then we ignore that, i.e., we assumed that the inactivity of a node is permanent once occurred.

It is evident in Table 7.6.1 that the alive sub-websites *Latex* and *Statistics*, which are considered very active, succeeded in keeping nearly 10% of the core nodes in the last network, whereas this percentage is almost *zero* in the other sub-websites. We found this 10% of the members to be users with a very high overall reputation score. For instance, user number 5001[7] was active in all of the networks used overtime for the *Latex* sub-website, and he/she is in the top 0.09% among all the users of StackExchange and has reputation score 303 thousand. The same behavior was found on the *Statistics* sub-website for user 805[8] who is in the top 0.02% among the StackExchange users and has reputation score 222 thousand. We noticed that these two users were active mainly on the corresponding sub-website, i.e., *Latex* and *Statistics*, respectively. For the *Music* sub-website, the situation is different. The number of members retained from by core nodes was only two users, which is very similar to the decayed sub-websites. Moreover, those two users were mainly active on other sub-websites; for example, user 932[9] was found in all of the networks of the *Music* dataset, but his main activity was on the *Stack Overflow* sub-websites. For the decayed websites, it was hard to get information about the users retained by the core users because no user information was available.

[7] https://tex.stackexchange.com/users/5001/mico
[8] https://stats.stackexchange.com/users/805/glen-b
[9] https://music.stackexchange.com/users/932/leftaroundabout

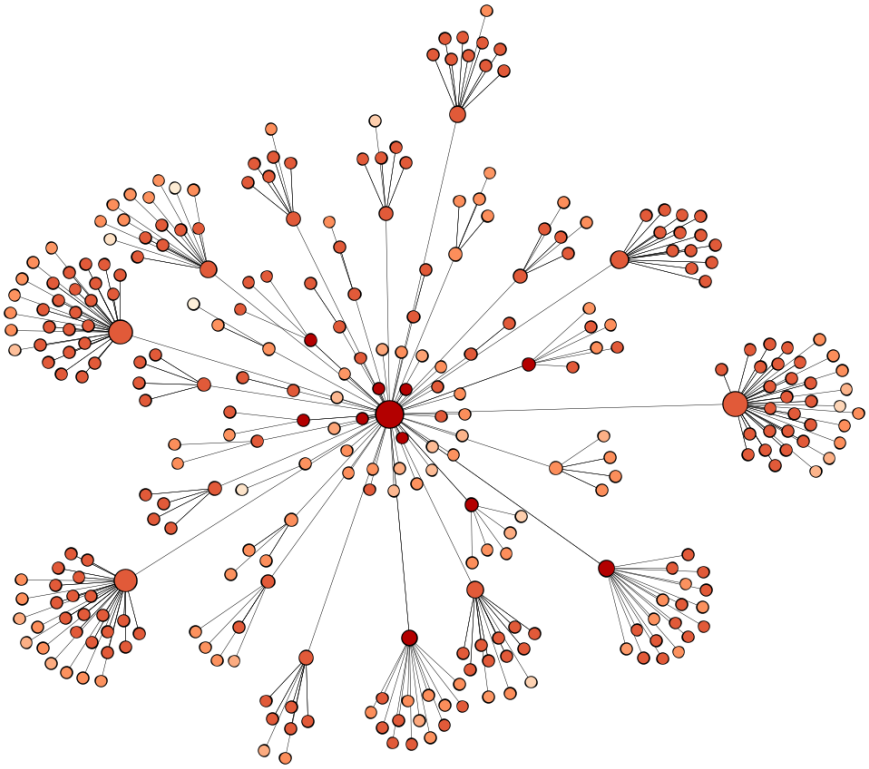
Dataset	Observation period	k	$\lvert V_{G_o}\rvert$	$\lvert E_{G_o}\rvert$	$\lvert V_{G_{k-1}}\rvert$	$\lvert E_{G_{k-1}}\rvert$	$\lvert\mathbf{I}\rvert$
Startups(Decayed)	10.2009 - 09.2013	32	702	9080	2	1	309
Economics(Decayed)	10.2011 - 03.2012	10	33	67	3	2	17
Latex (Alive)	07.2010 - 12.2015	33	498	4823	53	87	169
Statistics (Alive)	07.2010 - 12.2015	32	419	4795	36	37	141
Music (Alive)	04.2011 - 12.2015	38	293	1303	2	1	48

Table 7.6.1: Description of the datasets used and the k networks constructed over the given period. The initial network is G_o and the last observed network is G_{k-1}. The set V_{G_o} contains the core nodes. The number of extracted cascades is $|\mathbf{I}|$.

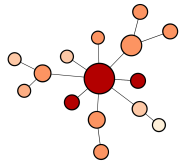
7.7 Results and Discussions

7.7.1 Analysis and modeling results

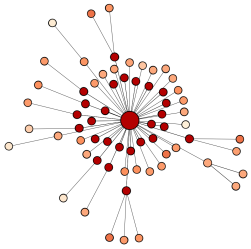
Here, we will first present the results of the analysis by providing information about the largest cascades extracted from the datasets. Figure 7.7.1 shows an arbitrary cascade of the largest cascades (those with the largest number of nodes) of the sub-websites *Startups*, *Economics*, *Statistics*, *Latex*, and *Music*. We observe that the cascades of the decayed sub-websites, such as *Startups* and *Economics*, contain a larger fraction of nodes from the initial network G_o than what we observe in the alive sub-websites. The fraction of nodes in the shown cascades, considering the initial network, are 0.44, 0.45, 0.15, 0.21 and 0.09 for the sub-websites *Startups*, *Economics*, *Statistics*, *Latex*, and *Music*, respectively.



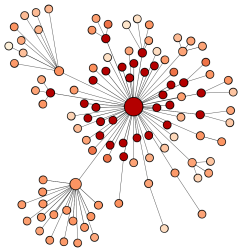
(a) Startups



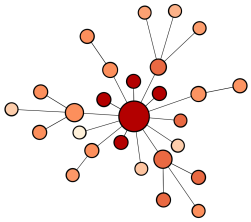
(b) Economics



(c) Statistics



(d) Latex



(e) Music

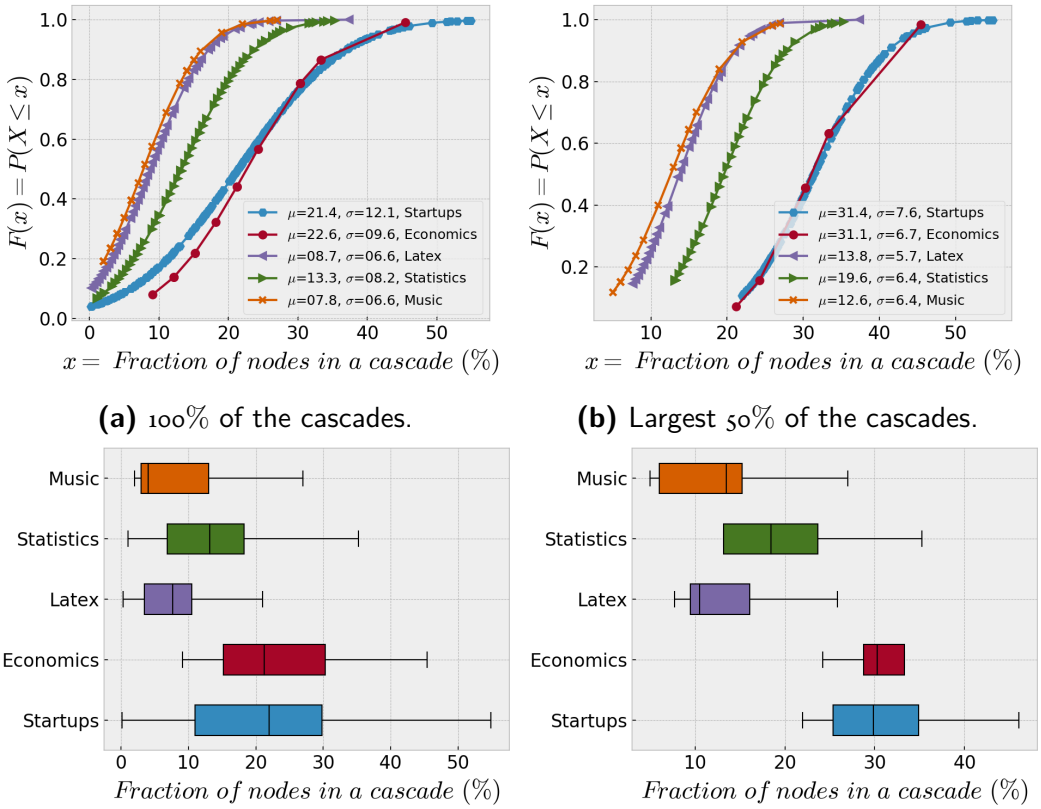
Figure 7.7.1: The largest cascades extracted from the datasets. The color of the node is inversely proportional to the time at which the node became inactive (i.e., the darker the node, the earlier it became inactive), and the size of a node is directly proportional to its degree in the cascade.

The figure also shows that for the decayed sub-websites, the color of the nodes is very close to each other. This suggests that the duration of the decayed sub-websites was short compared to the duration of the alive sub-websites because the colors of the nodes in the alive sub-websites are lighter at the nodes close to the leaves. This will be statistically supported in Section 7.7.1.

The goal of the following sections is to investigate the cascade properties (cascade size, virality, maximum degree, duration, coreness, and similarity) of the decay cascades we extracted. We want to see whether there are universal decay patterns (using CDFs and statistics divergence tests described in 7.5.6) that exist in decayed sub-websites. Also, we want to compare the patterns found in both decayed and alive sub-websites so that we understand how the decay of these sub-websites took place, and which of the studied properties are a good explainer of the decay patterns.

Cascade size

In this section, we investigate the size of a cascade (which is the number of nodes in a cascade) as cascade measure. Figure 7.7.2 shows the results obtained from different sub-websites. We can observe in the figure that all datasets contain cascades that have at least 28% of the nodes from the nodes of the initial network G_0. This percentage is even higher in decayed communities; it reaches 55% on the Startups sub-website and nearly 45% in the Economics sub-website.



(a) 100% of the cascades. **(b)** Largest 50% of the cascades.

(c) 100% of the cascades. **(d)** Largest 50% of the cascades.

Figure 7.7.2: The figure shows the fraction of nodes (of the initial network G_o) in the extracted cascades as CDF (Panels 7.7.2a and 7.7.2b) and as box-plots (Panels 7.7.2c and 7.7.2d).

Claim 1: Different inactivity cascade size patterns exist in alive and decayed sub-websites.

Figure 7.7.2 shows that the cascade size patterns appear visually different. The difference is more explicit in Figures 7.7.2b and 7.7.2d, where the cascades in the decayed communities contain a lot more nodes. To get statistical significance concerning this phenomenon, we used the KS-test described in Section 7.5.6. We found that there is a statistically significant difference between the decayed and the alive sub-websites. We found that the probability distributions of cascade size are the same (e.g., appears to be drawn from the same distribution) in the alive sub-websites ($p \approx 0.12$), are the same for the decayed sub-websites ($p \approx 0.7$), and are different when testing an alive and a decayed website ($p \ll 10^{-6}$).

The only exception to this occurred when testing the statistical significance between the *Statistics* and the *Latex* sub-websites; although both are still alive, the cascade sizes were statistically different ($p \ll 10^{-6}$).

The size of the cascades extracted from different sub-websites shows that inactivity dynamics is common in both alive and decayed sub-websites of StackExchange. However, the size of the cascades in the decayed sub-websites was significantly larger than the size of the inactivity cascades

found in the alive ones. Based on Figure 7.7.2, the smallest cascade in the largest 50% of the cascades contains more than 20% of the nodes from the initial network of the decayed sub-websites (for Startups), compared to nearly 5% for the alive ones (for Music). Our interpretation of this is that there are members of the alive sub-websites who are maintaining the aliveness of these communities and continuously provide content (in terms of, for example, answers to questions), which keeps the platform active. This can be seen in Table 7.6.1, where the number of nodes found in the last observed network of the alive sub-websites is very much higher than that of the nodes found in the decayed sub-websites (except for the Music sub-website). It seems that these members are experts whose existence is vital for sustaining these communities. An investigation of the profiles of some of these members (see Section 7.6) supports our interpretation.

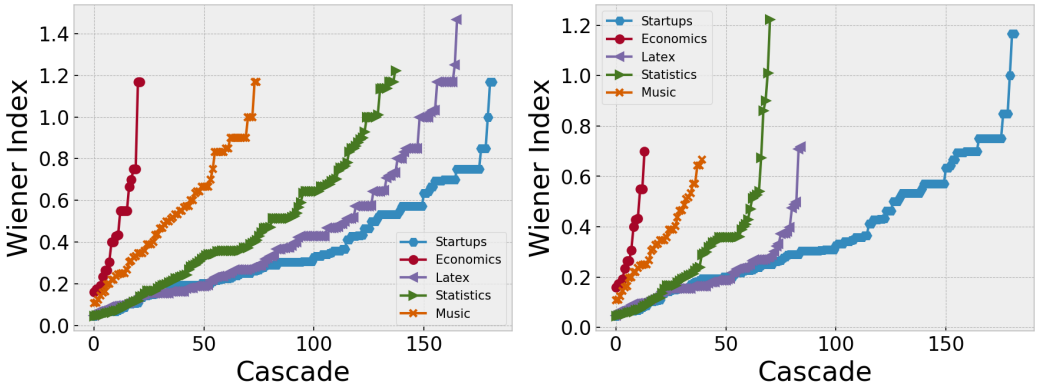
Cascade virality

Here, we want to investigate the decay patterns by looking into the virality of the cascades in both decayed and alive sub-websites. First, let us see what the relation between cascade virality and cascade size is. Figure 7.7.3 shows sorted values of the Wiener Index for the cascades extracted from the used datasets (the whole set of cascades is in Panel 7.7.3a and the largest 50% of these cascades are in Panel 7.7.3b). From this figure, Figure 7.7.3, we see that the largest cascades are not necessarily the most viral ones. For example, we see that the most viral Latex cascade (cascade with the largest Wiener Index) in Figure 7.7.3b is less viral than the most viral cascade in Figure 7.7.3a. This behavior is also evident for the Music and Economics cascades as well. Additionally, we investigated the correlation between the virality and the size of the cascades. In Figure 7.7.3c, we see that there are many cascades with a very close number of nodes, yet with very different virality values. Also, there are some cascades of the same virality but with different cascade size. So, it is not evident that there is a correlation between cascade size and cascade virality values. Thus, we have the following claim.

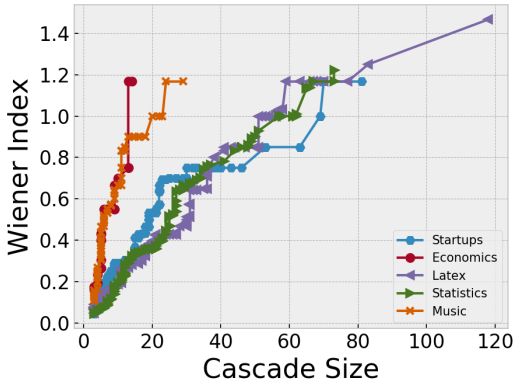
Claim 2: The virality of decay cascades is indistinguishable in alive and decayed sub-website.

To further understand the virality decay patterns, we performed statistical tests to see if these virality patterns are different in decayed and alive sub-websites. Figure 7.7.4 shows the Wiener Index of the cascades extracted from different sub-websites as CDF plots. Generally, the patterns of virality across different sub-websites are statistically the same ($p > 0.1$). The only exception is the *Economics* sub-website, where the virality patterns are statistically different with $p \ll 3 \times 10^{-5}$. This peculiar behavior of the *Economics* sub-website is ascribed to it being a small dataset with only 17 cascades. Surprisingly, the figure shows that the decayed sub-website *Startups* has fewer viral cascades, with a mean of 0.29.

Having the same virality patterns for the decayed and the alive sub-websites suggests that there should be another cascade property affecting the decay of decayed sub-websites. In the following section, we will discuss this in more detail.

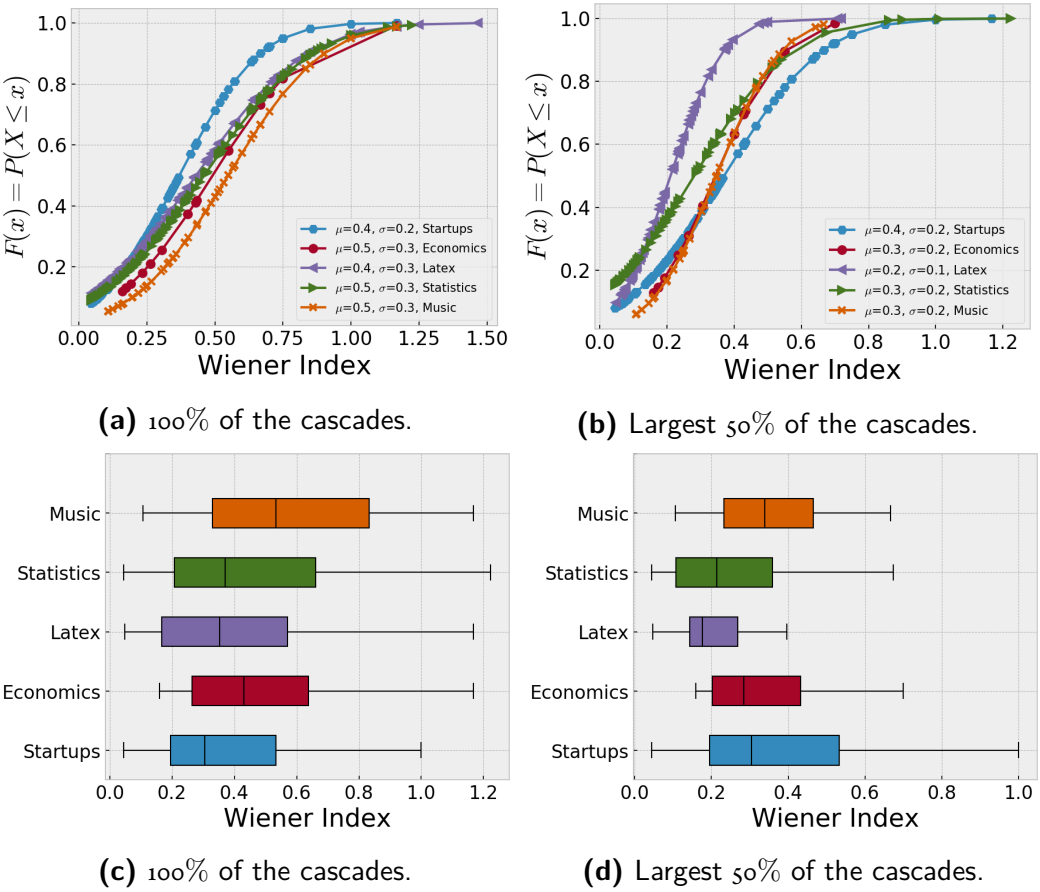


(a) 100% of the cascades. **(b)** Largest 50% of the cascades.



(c) Cascade Size vs Wiener Index.

Figure 7.7.3: The figure shows the real values of Wiener Index of the extracted cascades for the whole cascade set (in Panel a) and the largest 50% of the cascades in Panel b. Panel c shows the correlation between cascade size and virality.



(a) 100% of the cascades.

(b) Largest 50% of the cascades.

(c) 100% of the cascades.

(d) Largest 50% of the cascades.

Figure 7.7.4: The figure shows the Wiener Index of the extracted cascades as CDF (Panels 7.7.4a and 7.7.4b) and as box-plots (Panels 7.7.4c and 7.7.4d).

Maximum degree of cascade

Another pattern that we looked at is the maximum degree in a cascade. Figure 7.7.5 shows the normalized maximum degree in a cascade for different sub-websites. The normalization is done by dividing the maximum degree of a node in a cascade by $n = |V_{G_0}|$ of the corresponding dataset. The normalization is required because the networks used to extract the cascades have a different number of nodes. Thus, a normalization is required to compare the patterns for all datasets. The visualization suggests that the decayed sub-websites *Startups* and *Economics*, contain cascades of nodes with larger degrees than the alive sub-websites.

Claim 3: Inactivity decay is well-described by cascade's node degrees.

The statistical analysis shows that the decayed sub-websites have a very similar distribution of the maximum degree in a cascade with $p > 0.13$. The decayed and the alive sub-websites are statistically different with $p \ll 10^{-8}$. Once again, the Statistics sub-website shows a different pattern: It is neither similar to any of the decayed sub-websites nor to any of the alive sub-websites, with

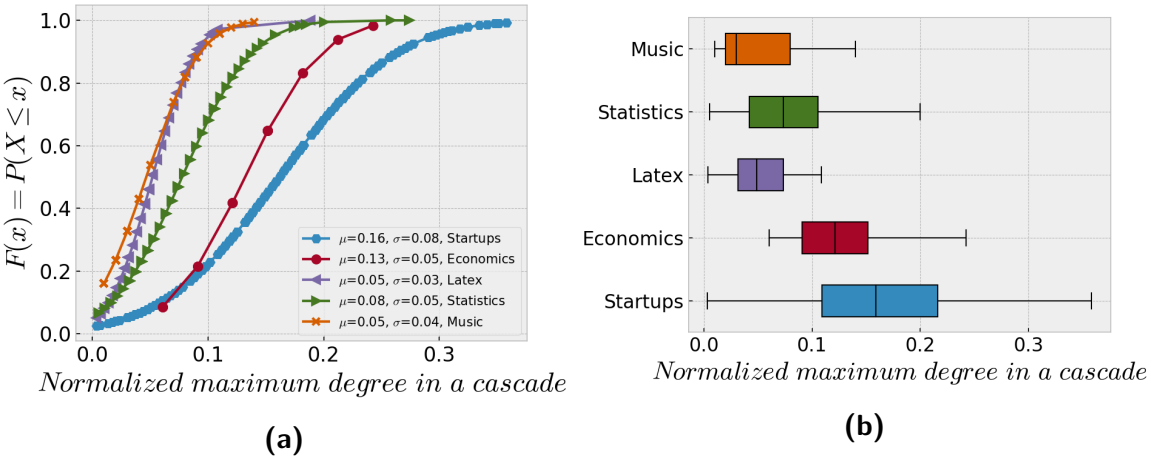
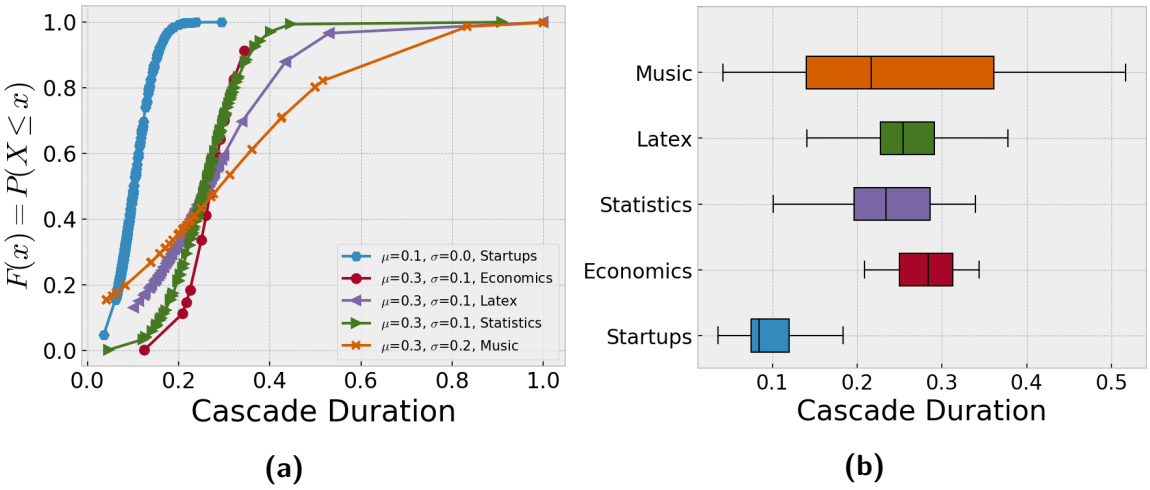


Figure 7.7.5: The figure shows the normalized maximum degree of a node in the extracted cascades as CDF (Panels 7.7.5a) and as box-plots (Panels 7.7.5b).

$p \ll 10^{-8}$. Unexpectedly, the decayed sub-websites we examined had fewer viral cascades than the alive sub-websites. This led us to investigate the micro-properties of the cascades rather than relying only on the macro-properties. We found that the cascades in the decayed sub-websites are less viral, but their nodes have larger degrees compared to those in the alive sub-websites. Additionally, we discovered that cascade initiators in decayed sub-websites have larger degrees in the cascade trees than non-initiators. This indicates that the expert members (who have larger degrees due to their activity and contribution) started the inactivity process, followed by non-expert members. One possible reason for the closure of decayed sub-websites is the lack of activity from those members who should have sustained the community and kept it going until it reached the public version. On the other hand, the more viral cascades in the alive sub-websites, which also have a smaller number of nodes and contain nodes with smaller degrees than the decayed sub-websites, indicate that the effect of inactivity is limited. The reason for this is that the size of the cascades in the alive sub-websites is small, with initiators having smaller degrees, compared to the decayed sub-websites. We conclude that expert members in the alive sub-websites act as obstruction points in the cascade trees, stopping the effect of inactivity cascades from being very disruptive.

Cascade duration

Here, we provide the results for the analysis of cascade duration defined earlier in Section 7.5.1, Equation 7.1. Figure 7.7.6 shows the cascade duration of different sub-websites. The x-axis reflects how long the cascade takes to be completed, i.e., until the formation of the cascade is finished. The figure shows that the cascades in the decayed sub-website *Startups* took noticeably less time to be completed, i.e., it had faster cascades. This is also clearly visible in Figure 7.7.6a. The statistical analysis of cascade duration shows that the other sub-website have their own characteristics, with



(a) **(b)**

Figure 7.7.6: The figure shows cascade duration as CDF (Panel 7.7.6a) and as box-plots (Panel 7.7.6b). Cascade duration is normalized based on the number of networks available for each sub-website.

no common pattern identified ($p < 5^{-10}$).

Claim 4: Statistics alive sub-website *may* go to decay process.

Although the *Statistics* sub-website is alive and falls into the category of alive sub-websites, based on the results described in Section 7.7.1, we discovered that the *Statistics* sub-website inactivity patterns are closer to the patterns found in the decayed sub-websites than to those of the other alive sub-websites. Using D_{JS} described in Equation 7.8, we found, strangely, that the Statistics sub-website is closer to the decayed sub-websites in terms of cascade size, virality, maximum degree in a cascade, and cascade duration. We investigated this behavior and found that the *Statistics* sub-website is *the second least* active sub-website among all StackExchange sub-websites with the fewest answered questions. I.e., only 63% of the questions were answered[10], whereas, on other sub-websites, the answer rate is much higher, for example reaching 93% and 97% on the *Latex* and *Music* sub-websites, respectively. We think that as Statistics is a very interdisciplinary field, many questions are domain-specific that require particular statistics background to be answered. For example, Statistics questions are not like Latex questions, where any member of a certain level of latex experience can answer latex related questions. This behavior, which was caught by our result, supports the effectiveness of the method we used. We think that the *Statistics* sub-website *may* fall into a decay process if its activity level remains as low as it is.

[10]https://stackexchange.com/sites, last check was in May-2018

Cascade coreness

Here, we will examine the coreness of the nodes in a cascade as a microscopic property of a cascade. We start by examining the *coreness* of an initiator. Figure 7.7.7a shows a comparison between the coreness of all non-initiator nodes in network G_o and the coreness of the initiators from all sub-websites as CCDF. The figure shows that the probability of having a coreness of, say x, in the initiators is a bit larger than what is found for all nodes (that is why we used CCDF not CDF). This suggests that the coreness of the initiators is larger than that of the other nodes in the initial network G_o. This was also statistically confirmed with $p \ll 10^{-6}$. However, further examination provided different insights and patterns. We performed the same analysis for each of the sub-websites. For example, in Figure 7.7.7b, there was no different pattern for the sub-website *Startups*, where the initiators have higher coreness for the coreness values in the interval $[1, 12]$, but less coreness for the coreness values in the interval $[13, 29]$. For the other sub-websites in Figures 7.7.7c, 7.7.7e, and 7.7.7d, the initiators have a clear pattern: They have more coreness than the other nodes in the corresponding G_o. An opposite pattern was found in the sub-website *Music* (cf. Figure 7.7.7f).

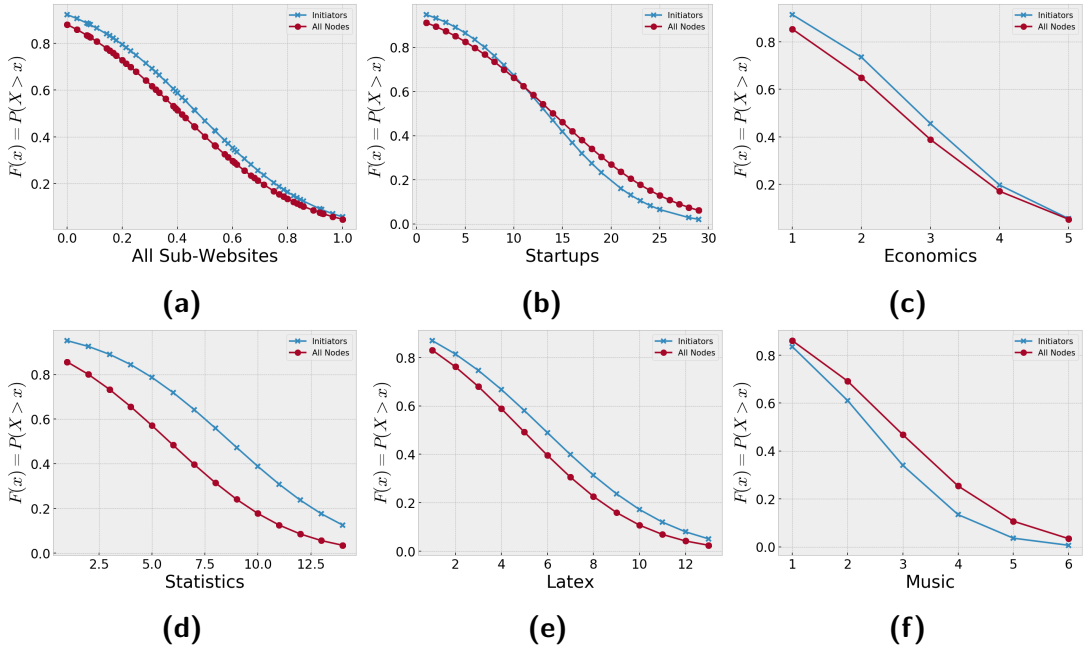


Figure 7.7.7: The figure shows the CCDF of the probability distribution of the coreness of all nodes in the network G_o compared to the coreness of the cascade initiators for all sub-websites combined (Figure 7.7.7a). Besides, the other panels 7.7.7b to 7.7.7f show the CCDF for each sub-website alone. Note that the x-axis scale in panel 7.7.7a is scaled to allow for a proper comparison, as network sizes are different across different sub-websites.

The previous analysis only refers to the initiators. To understand coreness in the temporal context, we define the following: A *cascade path* P is a connected directed subgraph of a cascade $\mathcal{I}$, where the maximum degree for each node of P is 2, obviously, with no cycles. The **coreness mono-**

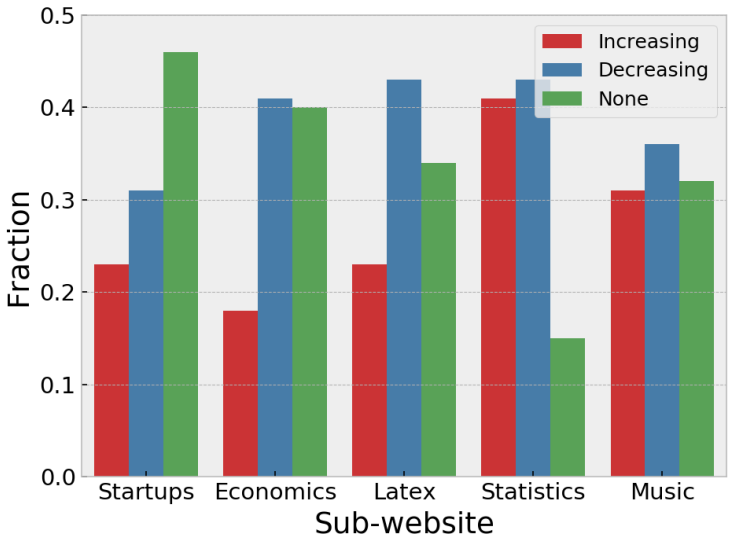


Figure 7.7.8: The coreness monotonicity of all cascade paths extracted from all cascade trees originating from the cascade initiators. On the x-axis, the different sub-websites are shown, and on the y-axis, the fraction of paths that are monotonically increasing, monotonically decreasing, or non-monotone.

tonicity of a cascade path P is defined as follows $\forall\, e = (u, v) \in E_P$:

1 *Non-monotone* if all nodes in a cascade path have the same coreness

2 Otherwise, we have case distinction:

   (a) *increasing* if $core(v) \geq core(u)$

   (b) *decreasing* if $core(v) \leq core(u)$

   (c) *non-monotone* otherwise

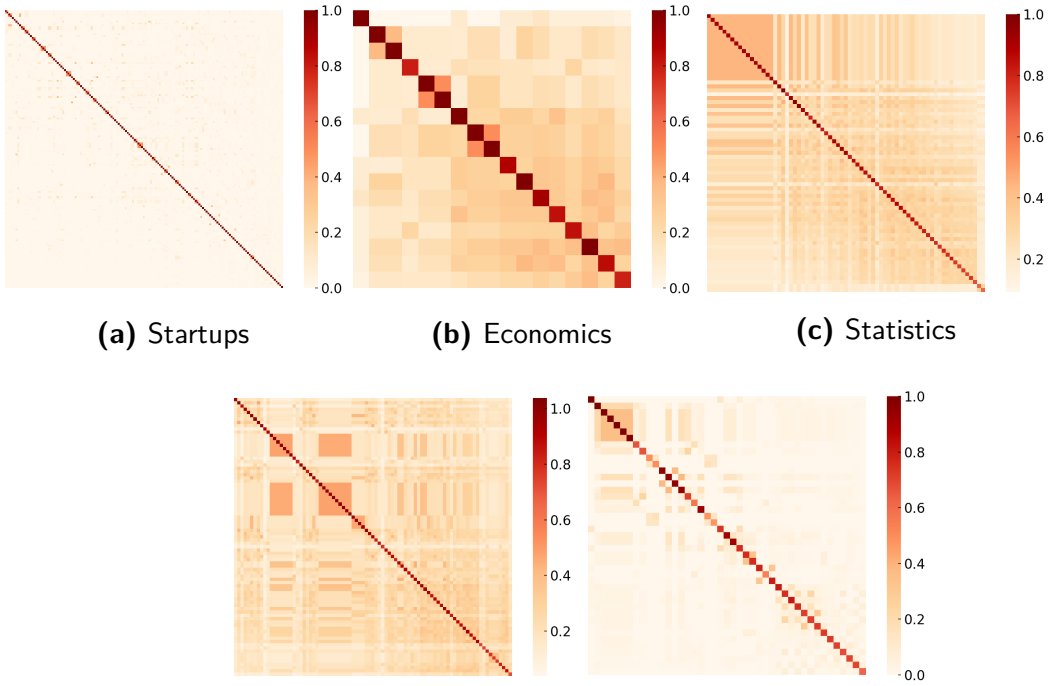
All coreness values are calculated in the initial network G_0. Based on that, we extracted cascade paths from all cascade trees where the first node in a path is the initiator of this cascade tree. Then we examined the coreness monotonicity of these paths. The results are shown in Figure 7.7.8 and indicate that the coreness of the cascade paths is clearly different across different sub-websites. Moreover, the fraction of monotonically increasing and monotonically decreasing paths was nearly identical in some cases (see, for example, the *Statistics* and *Music* sub-websites). Also, in the case of decayed sub-websites (see the *Startups* sub-website), the fraction of non-monotone paths was larger than for any of the other two types.

Claim 5: Coreness (and generally speaking, any single measure) alone does not describe sufficiently well inactivity cascades.

In their work, Garcia et al. [GMS13] posed the question of whether the decay starts from the *interiors* (nodes with high coreness) or from *exteriors* (nodes with low coreness). In their work, they argued that the decay of the Friendster social network started from exterior nodes. Later, Seki and Nakamura [SN17] presented a counter-argument, showing that the decay started from the interiors, and provided a model for understanding the decay process. Here, we argue that the answer to the question *Does the decay start from the interior or the exterior nodes?* is: *Neither*. The results of this chapter show no uniform pattern across different sub-websites that would correlate with the direction and coreness of the decay (cf. Figure 7.7.8). Furthermore, we argue that the question contains an implicit unsupported assumption, namely that it is only the coreness that controls the decay. We strongly believe that coreness alone can not be used to understand the direction of decay dynamics if the direction matters. In Section 7.7.1, we provided a formal framework defining the direction of the decay considering temporal decay, so that we can explicitly tell whether coreness alone can be used as an indicator for the direction of the decay. We found that the initiators of cascades exhibit opposing patterns in terms of whether their coreness is higher or smaller than the coreness of non-initiators. Additionally, we analyzed the coreness of the nodes in the cascade paths (coreness monotonicity) and found evidence that coreness is not correlated with the direction of the decay. Moreover, we performed an analysis using different measures, such as degree and betweenness. We conclude that it is tough to describe the decay process using only one measure. This is also clearly visible in the prediction results (as we will see in the next Section, Figure 7.7.12) where the importance of the features used for predicting cascade size and virality was close. To further support our argument, we predicted cascade size and virality using only one feature. In no case were the results better than when we predicted them using multiple features. We found the results of prediction using one feature to be very close to the baseline predictor; for example, the MAE (Mean Absolute Error) was 0.23, 0.23, 0.22, and 0.22 for predicting cascade virality using betweenness, degree, coreness, and min. cut, respectively. To sum up this point, we think that inactivity decay may be caused by network-independent factors, such as privacy issues, variation in competence between social network providers, and/or content quality. If any of these factors manifest itself, it renders the network measures unusable for describing inactivity decay.

Cascade similarity

Using the similarity measure defined in Equation 7.3, we calculated the similarity of each pair of cascades. Figure 7.7.9 shows a heat map for the similarity of the cascades for different sub-websites. Figure 7.7.9a clearly shows less similarity between the cascades of the *Startups* sub-website, unlike the other panels in Figure 7.7.9. It can also be observed that cascades with a smaller number of nodes seem to be more similar than those with a large number of nodes. An exception is the *Economics* sub-website, where cascades with larger nodes are more similar than those with fewer nodes. To get



(a) Startups **(b)** Economics **(c)** Statistics

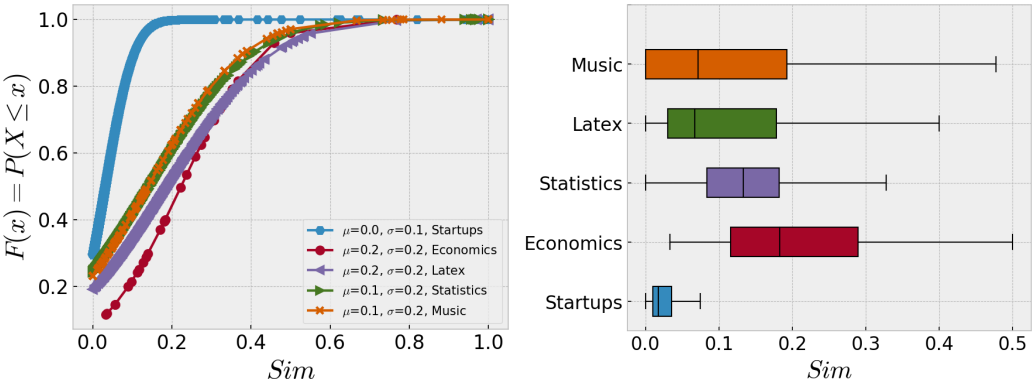
(d) Latex **(e)** Music

Figure 7.7.9: The figure shows the similarity of each pair of cascades for different sub-websites. The cascades were ranked in ascending order based on the number of nodes they have. The darkness of the color is directly proportional to the similarity.

statistical confidence regarding the comparison, we used the statistics described in Section 7.5.6. We found that although all of the sub-websites exhibit different similarity patterns ($p \ll 10^{-8}$), the decayed sub-website Startups has the smallest average similarity with a value of 0.03, compared to 0.21, 0.16, 0.17, and 0.11 for the other sub-websites. This difference can easily be seen in Figure 7.7.10.

Claim 6: Cascade similarity reflects how resilient a network was while it evolved.

The method we described for the extracted cascades in Section 7.5.1 allows for extracting cascades with the same nodes and/or edges. This means that we can measure the similarity of two cascades. Basically, if there are many similar cascades in a sub-website, this means that there are fewer paths on which the inactivity cascade took place than if there are fewer similar cascades. This means that, for cascades with less similarity, many decay propagation paths are susceptible to inactivity, and conversely, for cascades with high similarity, there exist fewer decay propagation paths that are susceptible to inactivity. Thus, cascade similarity can be seen as a measure for the resilience (or vulnerability) of a community for any future model or simulation of inactivity decay. Based on the results described in Section 7.7.1, it is apparent that the decayed sub-websites contain more nodes that are susceptible to inactivity than the alive sub-websites. The similarity of the cascades in



(a) **(b)**

Figure 7.7.10: The figure shows the similarity of each pair of cascades as CDF (Panels 7.7.10a) and as box plots (Panels 7.7.10b). The similarity is defined as described in Equation 7.3.

the alive sub-websites is high, suggesting a lower number of cascade paths.

7.7.2 Prediction results

In this section, we present a prediction framework we designed for predicting some cascade features. We formalize the prediction problem as follows. Given a training set $Z = \{(X_1, y_1), \cdots, (X_n, y_n)\}$, where $X_i = \{x_1, \cdots, x_m\}$ is the set of input features of length m, y_i is the target value to be predicted, and n is the number of data points in the training set. The prediction problem is defined as estimating a function $f(X) = \bar{y}$, where $\bar{y}$ is the predicted target value that is being compared to the real target value y. Thus, the optimization problem is generally defined as *minimize* $\sum \mathcal{L}(f(X), y)$, where $\mathcal{L}$ is an arbitrary cost function. In this chapter, we used the Mean Absolute Error cost function which is defined as $MAE = \frac{1}{n}\sum_{i=1}^{n} |y_i - \bar{y}_i|$.

Features for predicting cascades properties

Here, we describe the data preparation used for predicting cascade size and virality. Algorithm 7.2 shows how the features are used to construct a features data model (FDM) for training and testing. The target values that we will predict, the cascade size and virality, are added to the features data model as well. That is because we do not want to recalculate the whole FDM again. The algorithm used the set of features **f** that are described in Table 7.7.1, and the algorithm constructs the features data model for one data set at a time. Thus, we constructed multiple FDMs, one for each sub-website, and then we combine them into one data set. Therefor, the results in this section are based on the combined FDM. From now on, FDM refers to the combined FDM. It is clear that we used only features from the network G_o and did not use any of the temporal features to make the prediction more realistic, as temporal features of a network exhibit proxies for the predicted values,

which weakens the applicability of the method.

Algorithm 7.2: Node-based transformation algorithm of a graph G_o using the set of node-related **f**. The Algorithm is used to generate the features data model for predicting cascade size and virality.

Input: $G_o = (V, E), \mathbf{f} = (D(v), B(v), C(v), Core(v), E(v), \mathcal{MC}(v), Evce(v), B(e), D(\Gamma(v)))$

Init: $FDM = \emptyset$

for $v \in V$ **do**
  $values = ()$
  for $f \in \mathbf{f}$ **do**
    $values = values \oplus f(v)$ // add the value of each $f \in$ f to the vector features
  if $v \in L_o$ **then**
    // if v is an initiator, add cascade virality and cascade size of the cascade $\mathcal{I}$ initiated by v
    $values = values \oplus (|\mathcal{I}_v|, v_{\mathcal{I}})$
  else
    // else, add zero for both size and virality
    $values = values \oplus (0, 0)$
  $FDM = FDM \cup \{values\}$

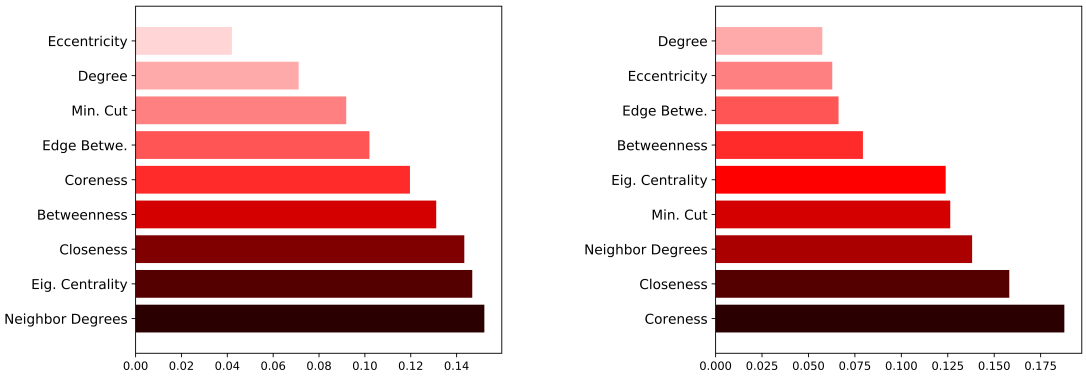
Output: FDM

Measure	Description
$D(v)$	The *degree* of a node v, $D(v) = \|\Gamma(v)\|$, is the cardinality of the set of neighbors $\Gamma(v)$.
$B(v)$	The *betweenness* of a node v is defined as: $B(v) = \sum_{s \in V(G)} \sum_{t \in V(G)} \frac{\sigma_{st}(v)}{\sigma_{st}}$, where $\sigma_{st}(v)$ is the number of the shortest paths between nodes s and t that include the node v and σ_{st} is the number of all the shortest paths between nodes s and t.
$C(v)$	The *closeness* of a node v is defined as: $C(v) = (\sum_{w \in V(G)} d(v, w))^{-1}$, where $d(v, w)$ is the distance between nodes v and w.
$Core(v)$	A k-core subgraph of a graph G is the maximal subgraph such that each node has a degree at least k. The *coreness* [BZ11] of a node $Core(v) = k$ if node v is in the k-core subgraph and not in the $k+1$-core subgraph.
$E(v)$	The *eccentricity* of a node v, $E(v)$, is the maximum distance between node v and node u.
$\mathcal{MC}(v)$	The *minimum cut* of two nodes u, v, $MinCut(u, v)$ is the minimum number of edges that are required to be removed in order to separate the two nodes. The averaged minimum cut of a node v is defined as: $\mathcal{MC}(v) = \frac{1}{n} \sum_{u \in E, u \neq v} MinCut(u, v)$, where n is the number of nodes in a graph.
$Evce(v)$	*The eigenvector centrality* of a node is defined as $Evec(x_i) = \frac{1}{\lambda} \sum_{j \in V_G} a_{ij} x_j$, where λ is a constant and a_{ij} is a location defined by i, j in the adjacency matrix . The measure can be written in matrix form as $\lambda x = A \cdot x$.
$B(e)$	*Edge betweenness* measures the number of times an edge e appears in the shortest path between any two nodes in a graph. It is defined as: $B'(e) = \sum_{v,u \in V_G} \frac{\sigma_{uv}(e)}{\sigma_{u,v}}$. *The incident edge betweenness* of a node is defined as the average edge betweenness for all edges incident to a node v.
$D(\Gamma(v))$	The average degree of the neighbors of a node v.

Table 7.7.1: Definitions of the network-based measures used in this chapter.

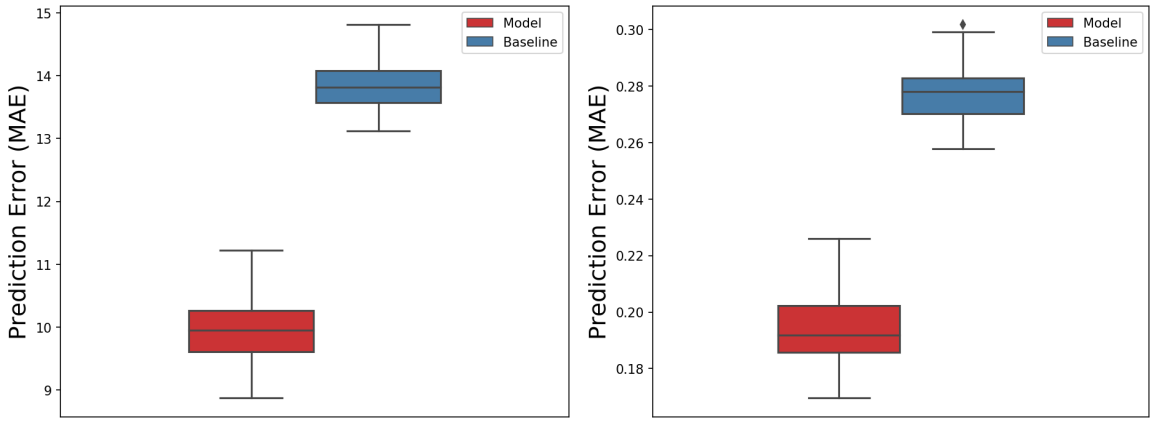
We split the data points of the FDM into two disjoint sets one for training and the other for testing with percentages 75% and 25%, respectively. We shuffled the data before the split so that training and testing will not be biased to one data set. As the FDM contains two target values, the size and virality of a cascade, we used only one target value in the training and testing depending on what we are predicting. I.e., if the experiment is about predicting cascade size, then the used features are those in the set **f** of Algorithm 7.2 and the target is cascade size; thus, we ignore cascade virality. The same logic is applied for predicting cascade virality. To evaluate the performance of the model, we used data points that had not been used during training and then evaluated them using the cost function with the true values of the target. The regression algorithm used was *Gradient Boosting Regression (GBR)* [Fri02], which is basically a decision tree with simple rules that are used for M iterations, where in each iteration a new decision tree is used to predict the previous prediction residual[11]. We used the scikit-learn [PVG+11] Python library implementation of the *GBR*, version 0.20.0. We used the default parameters of the algorithm except for *loss* we used *huber* parameter. The features described in Table 7.7.1 have different effects on the prediction; thus, we performed feature ranking in order to get insights regarding which features are more important during the prediction. Figure 7.7.11 shows the feature ranking for predicting cascade size and cascade virality. Panels 7.7.11a and 7.7.11b shows that the importance of the features is different; for predicting cascade size, the average of *Neighbors Degrees* was the most important one, whereas the feature *Coreness* was the most important one for predicting cascade virality. In both cases, the features Degree and Eccentricity were the least important ones in the set of features. Based on that, we used the five best features from each ranked set. Other combinations of features resulted in lower, but very close, prediction performance. We used the *MAE* as a prediction accuracy measure. As splitting the dataset into training was done randomly, we ran the prediction experiment 100 times to get statistical significance regarding the results, each run with a different random seed. Additionally, we compared the results to a baseline predictor that uses naive rules, such as taking the mean, the median, or a constant value for the predicted target. We compared the prediction results to the best baseline we got, which was the mean baseline. The prediction accuracy of *Cascade Size* in terms of the MAE was 9.9, which is 35% better than the baseline predictor. The prediction results mean that, on average, the predicted cascade size contains ± 10 nodes. The prediction accuracy of *Cascade Virality* in terms of the MAE was 0.194, which is more than 25% better than the baseline predictor. Figure 7.7.12 shows the results of the prediction for the 100 runs we performed for predicting both cascade size and cascade virality in panels 7.7.12a and 7.7.12b, respectively. The figure shows that there is a clear significance in favor of the *GBR* algorithm over the baseline predictor.

[11] The GBR outperformed other algorithms and techniques that we tested, such as Logistic Regression and classical Decision Trees. The technical details of the GBR algorithm can be found in [Fri02].



(a) Feature ranking for Cascade Size **(b)** Feature ranking for Cascade Virality

Figure 7.7.11: The figure shows feature ranking such that $\sum_i w(i) = 1$, where $w(i)$ is the feature rank for the feature i. The method used for generating the importance is Random Forests, where the importance of a feature increases whenever a split in the tree using that feature minimizes the prediction error [LWSG13]. We used the scikit-learn [PVG+11] Python library implementation of the Random Forests regressor, version 0.20.0. We used the default parameters of the algorithm except for the *number of estimators*; we used 300 estimators.



(a) Results for predicting Cascade Size **(b)** Results for predicting Cascade Virality

Figure 7.7.12: The figure shows the prediction performance results for 100 runs for the prediction of cascade size (Panel 7.7.12a) and cascade virality (Panel 7.7.12b). The figure compares the results of the *GBR* prediction algorithm and the results obtained from a baseline predictor.

Claim 7: For temporal networks, the structure of the early network encompasses sufficient information to predict the properties of its potential decay cascades.

It was surprising that using only network features from the network G_o resulted in a satisfactory prediction of a cascade's virality and size. These results suggest that the early structure of an evolving network dictates its future. The prediction model described and evaluated in Section 7.7.2, which used no temporal information at all, indicates that the (in)activity dynamics of social networks is governed by the topological structure of the network itself.

7.8 Closing thoughts

Although the method used in this chapter is reliable and the results have been validated, the work in this chapter is subject to certain limitations, which will be discussed in the following. In order to make sure that the networks we used represent real temporal interaction among the users, we used different time frames to take a snapshot of each sub-website. The reason for this is that each sub-website has a different timespan; for example, the alive sub-websites are still active, unlike the decayed sub-websites, which have a significantly shorter lifespan. We do not believe that our design decisions for selecting the time frames had a significant effect on the results and the conclusions. Also, the results and conclusions in this chapter are valid for the StackExchange sub-websites. We did not check other types of social networks or aimed at generalizing the results to any type of social network.

8

Predicting Interaction Decay Patterns in Online Social Communities

8.1 Synopsis

In this chapter[1], a model and a machine-learning-based framework will be presented in order to identify the inactivity patterns that accompany decay processes in online social networks. The model and the framework use the topological network structure as features for learning these patterns. Predictions and analyses performed on the decayed and alive StackExchange sub-websites reveal insights into the correlation between decay dynamics and network-based features.

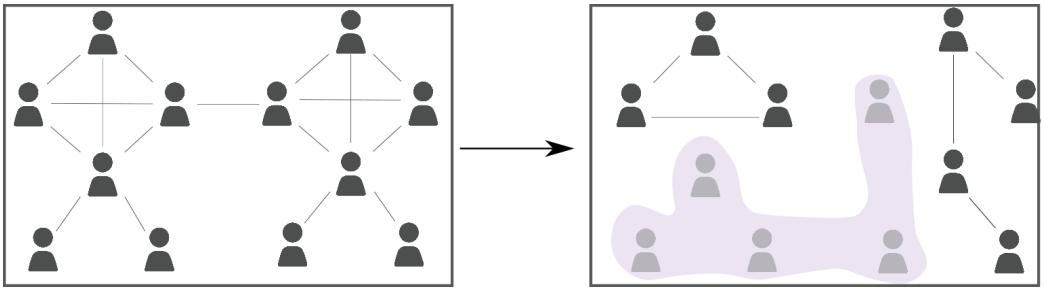


Figure 8.1.1: The goal of this chapter is to predict the members who become inactive.

[1]This chapter is based on the work [Abu18a].

8.2 Introduction

Since the seminal works by Barabási and Albert [BA99] and by Watts and Strogatz [WS98], the field of *network science* has witnessed a tremendous amount of research being performed on the dynamics of systems represented as networks. Social networks have been studied as an example of networks that contain a lot of dynamics overtime. While a lot of social networks have been successful in sustaining their aliveness and growth dynamics, many others have experienced *decay* dynamics. Online social platforms such as MySpace and Friendster are now out of service due to the huge decay they have experienced, causing a massive decrease in their market value.

In this chapter, we are interested in understanding the decay dynamics of Online Social Communities (OSCs) and the interaction patterns that accompany, or possibly cause, community decay[2]. Gaining insights into the patterns of the interaction decay among members of OSCs will enable us to better understand the decay process and hence help to suggest possible actions for prolonging the life of these communities and supporting their resilience against disruption due to inactivity. More precisely, we can predict which members will depart a network (or become inactive) by using the contributed Simple Threshold Model (STM) and by using a contributed supervised binary classification framework employing network-based and exogenous features.

8.3 Related Work

With the rise of network science, researchers were mainly interested in growth dynamics. For example, Newman et al. [JGN01] studied the growth dynamics of social networks between mutual friends and developed a model that shows similar characteristics as those of these real networks. Growth dynamics was then studied extensively by many researchers for different domains. For example, Newman [New01] studied the growth dynamics, namely clustering and preferential attachment, of scientific collaboration networks in physics and biology. Similarly, Bornholdt et al. [EDB02] presented another model for simulating the growth dynamics of social networks. When online datasets became available, Barabási et al. [BJN+02] conducted an empirical study on the evolution of the collaboration patterns of scientific collaboration networks. Leskovec et al. [LKF05] studied the growth dynamics of networks by observing some repeated patterns, namely densification laws and shrinking diameters. Backstrom et al. [BHKL06] investigated the growth dynamics of group formation and community memberships in online social networks. They provided a model for predicting when a member would join a community in a social network. A preferential attachment growth model was presented by Capocci et al. [CSC+06] to study the growth dynamics of the Wikipedia online encyclopedia. Similarly, Kossinets and Watts [KW06] studied the growth dynamics of a social network of students, faculty, and staff members of a university. They found that

[2]We use the term community as a reference for a social network of members of the same interest, e.g., a network extracted from a StackExchange sub-website.

the evolution of the network was mainly affected by the network structure itself and by some other external organizational structures. Kumar et al. [KNT06] provided a large-scale analysis of the evolution of a social network with five million members and more than ten million relationships. Their analysis revealed some structural properties of the growth process in online social networks. Ahn et al. [AHK+07] studied the growth of MySpace and Orkut before they were permanently closed, as real examples of networks with growth dynamics. They studied the scaling behavior of degree distribution over time for these networks and found that they had different exponents. Mislove et al. [MKG+08] extensively studied the growth of the Flicker online social network and found link formation patterns.

The aforementioned works focused on *growth dynamics*. However, the dynamics of social networks is not limited to growth dynamics, but also includes *decay* dynamics, which may occur in a social network, leading to an inactive (decayed) social network. Social network platforms like Orkut, MySpace, Friendster, and Friendfeed are now out of service after being active and growing for a long time. Dorogovtsev and Mendes [DM00] were among the first who studied the decay dynamics in networked data. They mathematically studied the decay properties of networks and found similar characteristics of preferential attachment as reported earlier by Barabási and Albert [BA99]. Since then, however, there is little research that addresses the problem of inactivity in social networks. For example, Garcia et al. [GMS13] studied the properties of different networks (decayed and active ones) in terms of k-core analysis. Later, Malliaros and Vazirgiannis [MV13] presented a method for quantifying and measuring the *engagement* of the members. Their measures enabled assessing the robustness of a network over time. In a related vein, Wu et al. [WDSF+13] developed a method for understanding the dynamics of the social engagement of the members of the co-authorship social network of the DBLP. They showed that there was a correlation between the actions of the departed members in the studied datasets. They also provided some insights regarding the properties of the members who departed the networks. Cannarella and Spechler provided an epidemic model for predicting the dynamics of the members of Facebook [CS14]. The results indicated that Facebook would lose 80% of its users between 2015 and 2017, which has not happened until now (2018). Karnstedt et al. [KRC+11], Kawale et al. [KPS09], and Wang et al. [WGC16] in a recent work provided prediction models for a user's lifespan in online social settings, which they also called *user churn*.

8.4 Contribution

Compared to the previous related work, our work differs with regards to two perspectives. First, users churn normally has a one-to-many relationship between the members and the service provider. In this scenario, the social interaction among the members, which is our main concern, is very limited. In this chapter, the social interaction between the users is the main focus of our models. Sec-

ond, our main concern is the decay of the social interaction between humans in online social networks, as we want to better understand the decay dynamics in online social networks[3]. The contributions of this chapter can be described as follows:

1. An exploratory data analysis of the decayed StackExchange communities and a comparison with the ones that are alive supported by a *ground truth* decayed networks.

2. A Simple Threshold Model (STM) for predicting social inactivity using network-based measures or members' exogenous information.

3. A machine learning framework for predicting social departure using network-based measures and members' exogenous information

4. Guidelines for feature selection in predicting a member's inactivity.

The results provide insights regarding network-based properties as well as the exogenous features of inactive members that are correlated with social inactivity. These insights may help to prevent decay dynamics, to engineer resilient social networks, and express the aliveness of OSCs.

8.5 Dataset

8.5.1 The StackExchange dataset

StackExchange[4] is a portal that includes many question & answer sub-websites for different specific topics. Any of these sub-websites starts out as a beta community until it shows potential for permanent public access. However, not all of these beta communities succeed in having the required activity and attracting enough experts to sustain growth. In such cases, these beta communities are shut down. There are many examples of closed StackExchange beta communities. The content generated by the users during their beta versions is still available, however. We downloaded, parsed, structured, and analyzed a list of the closed communities in order to understand what is going on during the decay (inactivity) dynamics in social networks.

8.5.2 Network construction

We constructed undirected networks from this dataset as follows for the closed business startups sub-website. Nodes are the users of the sub-websites and an edge (interaction) between two nodes (users) A and B appears if user A commented on a question (or comment) posted by user B. Each edge has a timestamp that reflects its creation time (and generally the last time A and B interacted

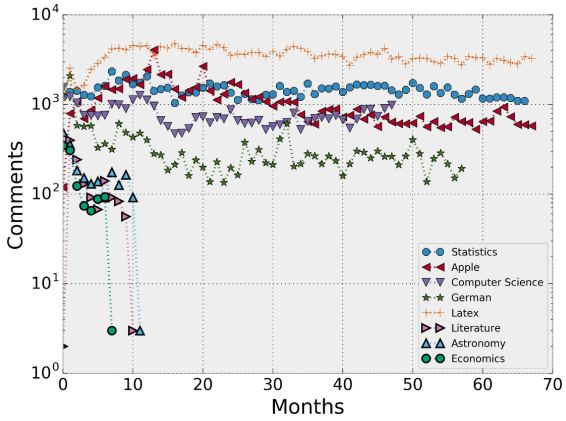
[3]More information about our view of social decay is provided in the previous two chapters.

[4]The dataset used in this chapter is the same as the one used in the previous chapter. The description here is a summarization of the details provided in Section 7.6.1.

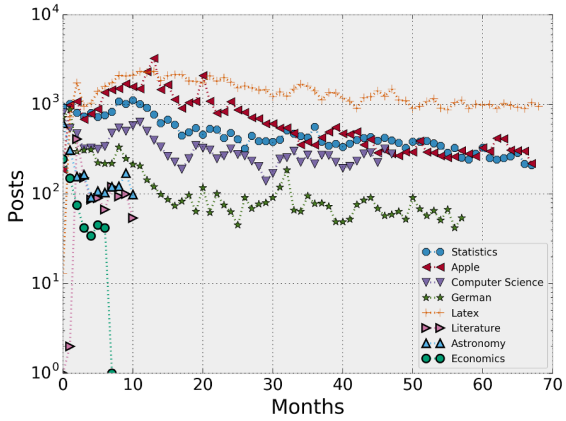
with each other). This network is called G_o. The temporal networks are then extracted from the network G_o such that each network at time t (G_t) contains the edges (with their incident nodes) that have timestamps $\geq t$. We constructed networks from the alive sub-websites and decayed sub-websites exactly as defined in Chapter 7.

8.5.3 Preliminary analysis of activity indicators

In this section, we present some analysis of the activity of the members of the sub-websites of the StackExchange. Figure 8.5.1 shows the activity, in terms of the number of comments (8.5.1a) and the number of posts (8.5.1b), of different sub-websites of the StackExchange over time. The figure illustrates that the activity of the members is almost stable in the alive communities, *Statistics*, *Apple*, *Computer Science*, *German*, and *Latex*, while the activity of the members is decaying in the closed communities, *Economics*, *Literature*, and *Astronomy*.



(a) The number of comments as an indication of activity.



(b) The number of posts as an indication of activity.

Figure 8.5.1: The activity of members of some communities of the StackExchange sub-websites in terms of *comments* and *posts* counted over time. The x-axis represents the number of months since the launch of a website. Communities with bold markers (*Literature*, *Astronomy*, and *Economics*) were closed after the failure of their beta versions.

Figure 8.5.2 depicts the difference between the decayed and the alive communities in terms of active weeks. The decayed communities exhibit significantly fewer active weeks compared to alive communities.

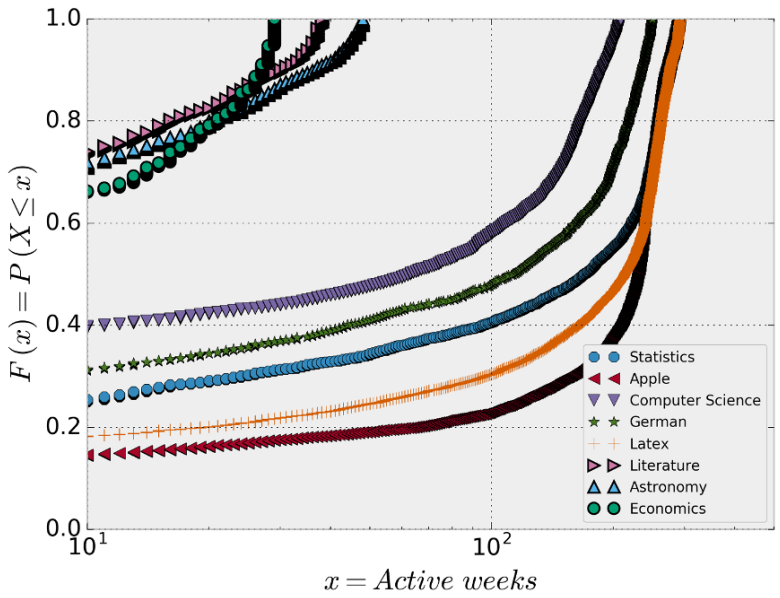


Figure 8.5.2: The figure shows the Cumulative Distribution Function (CDF) of members' active weeks. The number of active weeks is calculated as the difference between the last log-in date and the registration date of a member. The CDF is then calculated as $F(x) = P(X \leq x)$. Note that the x-axis is log-scaled.

8.6 FDM CONSTRUCTION FOR TRAINING AND TESTING

An observed (real) network and a predicted network are defined as $G = (V, E)$ and $G' = (V', E')$, respectively. As we are interested in predicting the departure of nodes, we define false-negative nodes as the nodes in the set $V \setminus V'$, i.e., the set of nodes that exist in the observed network, but whose existence the prediction model missed. Likewise, the set $V' \setminus V$ is the set of false-positive nodes, i.e., the set of nodes that the prediction model predicted, although they are not present in the observed data. Additionally, the set $V \cap V'$ contains the true-positive nodes.

We define the *Members inactivity* problem as follows: Given a network $G_{t_w} = (V_{t_w}, E_{t_w})$ that represents the network at time point t_w, and likewise we define networks G_{t_x} and G_{t_y}, where $t_w < t_x < t_y$. The goal is to predict the departure of the nodes in network G_{t_y} using the information from the networks G_{t_w} and G_{t_x}[5]. For training, we use the networks G_{t_w} and G_{t_x}, where $t_w < t_x$ and t_w and t_x are consecutive, to build an FDM. The FDM captures:

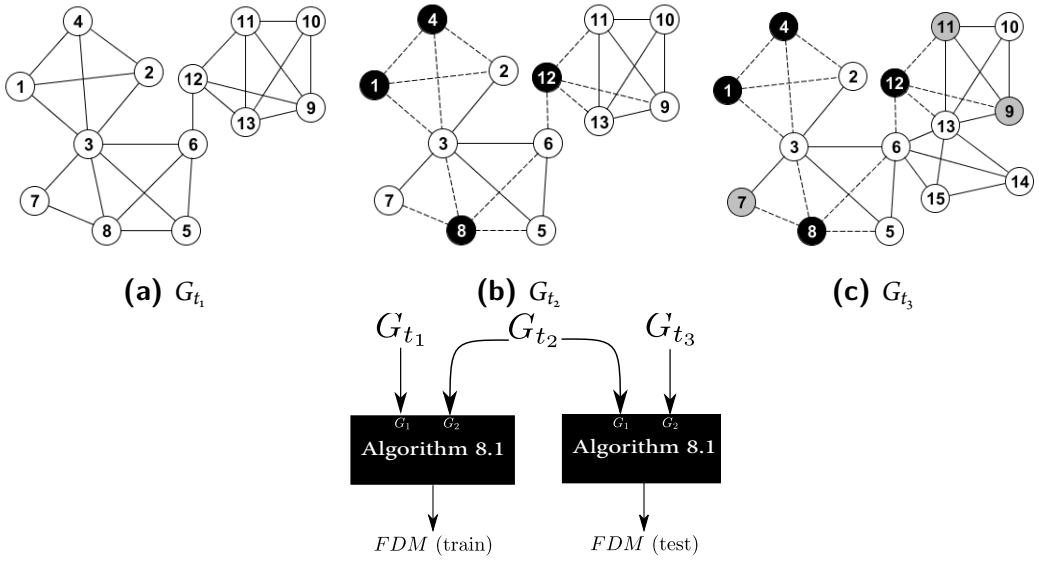
- the properties of the nodes in network G_{t_w}.
- whether they depart or not (using G_{t_x}).

The *inactive* members (members who departed) are the set of nodes $V_{t_w} \setminus V_{t_x}$. This FDM is used to teach a classifier to identify the depart patterns. The set of nodes V_{t_w} is called the *core* nodes, and

[5]Intuitively, we need two networks so that we know the set of nodes that departed.

any node u such that $u \in V_{t_j}$ and $u \notin V_{t_w}$ is ignored for all $t_w < t_j$.

For testing, we construct the FDM in a similar way for any two networks at two consecutive time points such that the network G_{t_w} is *not* one of them. Otherwise, the training and the testing FDMs will be identical. The requirement here is that the two networks used for constructing the training FDM must be consecutive and the two networks used for constructing testing FDM must be consecutive as well. This restriction will make it challenging to learn the depart patterns because the temporal networks we used are decaying networks, which means most nodes will eventually depart. When we say we test on time point t, then we mean that the training was performed on the period $t - 1$, and t. Thus, the networks G_{t-1} and G_t were used to build the FDM for the test set. Figure 8.6.1 shows an example of constructing the training and testing FDMs on exemplar networks. The training and testing in Figure 8.6.1 are performed on three consecutive networks over three consecutive time points. I.e., the middle network is used for training to get the depart label, and it is used as well for testing to capture the properties of the nodes that we are predicting their departure in the network G_{t_3}. However, the training and testing can be performed on any four networks such that the first two are consecutive and the last two are consecutive as well. For example, we can construct the training FDM on networks G_{t_1} and G_{t_2} and then construct the testing FDM on networks G_{t_5} and G_{t_6} for any $t_1 < t_2 < t_5 < t_6$.



(d) Training and testing using Algorithm 8.1

Figure 8.6.1: A schematic illustration shows the different networks, G_{t_1}, G_{t_2}, and G_{t_3}, over time where $t_1 < t_2 < t3$. The nodes in network G_{t_1} are *core* nodes. The training is performed during the period t_1 to t_2, where the black nodes in the network G_{t_2} are the observed nodes that have departed the network. Then we test on network G_{t_3} where we can predict the inactivity of other nodes, e.g., the gray nodes in G_{t_3}, using the properties of the nodes in G_{t_2}. Note that nodes that emerge in the network $G_{t'}$ and are not found in the core node set, e.g., nodes 14 and 15, are ignored. Panel d shows which networks are used for constructing the training set and which networks are used for constructing the testing set.

In the following section, we will give detailed information about the features of the FDM and how exactly it is constructed.

8.7 Method

In this section, we will describe our method, which contains the feature data model we built and used in the prediction in addition to two models for predicting *members departure*.

8.7.1 Features data model (FDM)

We provide the following two types of features for the core nodes along with the depart label:

- *Network-based measures*: These are the values of a node's attributes that are based on the network measures presented in Section 2.3. These measures reflect how a node is connected within the network. For each node $v \in G_{t_1}$, we calculate a set of measures that represents this node's network-based attribute values. These measures, **f**, are: the betweenness centrality ($\mathcal{B}$), the closeness centrality ($\mathcal{C}$), the degree (deg), the minimum cut ($\mathcal{MC}$), and the eccentricity (ecc). Those measures are defined in 2.3.1.

Algorithm 8.1: Node-based transformation of graphs for predicting nodes departure. The network G_1 is observed temporally before network G_2.

Input: $\underbrace{G_1, G_2,}_{\text{two consecutive networks}}$ $\mathbf{f} = (\underbrace{\mathcal{B}, \mathcal{C}, Core, deg, \mathcal{MC}, ecc,}_{\text{node network-based mesures}} \underbrace{Reputation, Views, Upvotes}_{\text{Exogenous attributes}})$

Init: $FDM = \emptyset$

for $v \in V_{G_1}$ **do**
  $values = ()$
  for $f \in \mathbf{f}$ **do**
    // the features are calculated for G_1
    $values = values \oplus f(v)$
  // G_2 is used only to check whether v departed or not
  if $v \in V_{G_2}$ **then**
    $values = values \oplus False$
  else
    $values = values \oplus True$
  $FDM = FDM \cup \{values\}$

Output: FDM

- *Exogenous attributes*: These are the values of a node's non-network attributes of the members. For each $v \in G_{t_1}$, we calculate a set of non-network measures. For the StackExchange dataset, these measures are the *Upvotes* a member received, the profile *View* count (views for short), and the *Reputation* score.

- *Depart label* :In addition to the above two types, we have the *Departure label*, which indicates whether a node $v \in G_{t_1}$ left the network G_2 or not.

We constructed an FDM using using the network-based measures and the exogenous attributes as described in Algorithm 8.1. Later, we will see that the same algorithm is used to construct the FDM using one single feature from features **f** when we compare the STM (which is a single-feature model by design) and the machine learning framework using a single feature. Based on that, we formulate our research questions as follows:

RQ1: *How efficient is it to predict members departing a social community using network-based measures?*

By answering this question, we aim at understanding how efficient it is to use the network topology to understand network decay dynamics.

RQ2: *What are the network-based properties of the members who departed or about to depart a community?*

By answering this question, we want to get insights regarding the properties of the nodes before they depart the network. Thus, networks and community maintainers can initiate counter-actions when a decay process starts to emerge, which makes it possible to sustain resilient networks.

RQ3: *How helpful are exogenous attributes in predicting members departure?*

Obtaining additional information other than the network representation is not always possible. Thus, answering this question will give us more insights into whether the network-based attributes contain sufficient information to predict the departure of members. We will also compare the results of the prediction performed using only the network-based attributes with those of the prediction using only the exogenous attributes.

RQ4: *Do decayed communities embrace inactivity patterns that can be used to study the inactivity of communities that are alive?*

The alive communities of the StackExchange may suffer from member inactivity; however, this inactivity is mitigated by the activity of new members and new discussions that support the aliveness of these communities and make them active until today. Answering this question will give us insights regarding whether or not there are community-independent decay patterns that can be used to track the potential decay of alive communities.

8.7.2 Simple Threshold Model (STM)

We present a simple model for predicting users departure using only one attribute i (either from network-based measures or exogenous attributes). The idea is to find the value for this i for all nodes and sort these values. Afterwards, we find the best threshold value λ that splits the nodes into two disjoint sets such that each element in each set has the same label, either *True* or *False*. The threshold value is chosen such that it maximizes one of the prediction measures provided in Section 4.6. Figure 8.7.1 shows a schematic diagram of this model. More formally, it is a sorted array of the values of an attribute i defined as $values(i)$ and the corresponding departure label array (where the departure label is again taken from the subsequent network). Let f be a function defined as $f : \lambda \rightarrow s$, where s is one of the prediction metrics, then the STM is defined as:

$$\arg\max_{\lambda} f(\lambda) = \{\lambda \mid \lambda \in values(i)\} \tag{8.1}$$

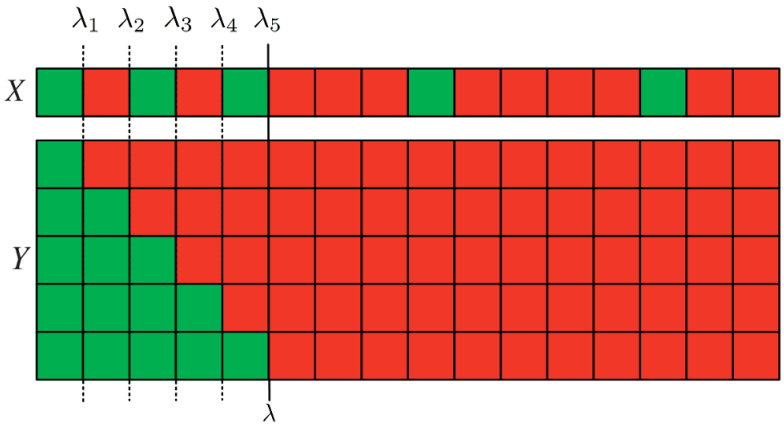


Figure 8.7.1: The diagram shows an example of how the value of λ is computed during the training phase. The set X represents a sorted vector of the values of one attribute, say the *Betweenness* of a node in a network G_{t_1}. For the vector X, green cells mean that the node departed at t_2 and red cells mean that the node did not depart at t_2. The goal is to find a vector Y that is composed of two vectors, each of them having the same value for all of its elements: either True or False (in the figure: green or red). The model aims at finding the best value λ or *Betweenness* such that it maximizes the prediction performance, e.g., the F1-score. That is, the X vector is the actual labels, and the Y vector is the predicted labels. The chosen $\lambda = \lambda_5$ because the values of the F1-score are 0.3, 0.29, 0.5, 0.4, and 0.6 for λ_1,λ_2,λ_3,λ_4, and λ_5, respectively. Later values for λ, i.e., λ_q for $q > 5$, have F1-scores below 0.6.

8.7.3 Machine learning classification

With the STM, we can only benefit from the information provided by one attribute at a time. To incorporate more attributes, we used the whole feature data model (FDM) for training and testing a supervised machine learning binary classifier to predict the *departure label*. For the evaluation, we used the evaluation metrics presented in Section 4.6. For the classification algorithms (and for the entire work in this chapter), we used Support Vector Machines' implementation of *scikit-learn* [PVG+11] with Gaussian kernel and with the default parameters.

8.8 Empirical results

8.8.1 Prediction using one attribute

In this section, we present the results of the community decay prediction using one attribute. We also report the results of the machine learning binary classification using one attribute only. Thus, we can compare the performance of the STM with the machine learning model. Figure 8.8.1 shows the training results of the STM. The STM performs reasonably well for most of the attributes. For example, attributes like *Betweenness*, *Coreness*, *Degree*, and *Views* show an acceptable F1-score. Some other attributes like *Upvotes* and *Eccentricity* contain no significant information that could be used for good prediction as their λ was almost *zero*.

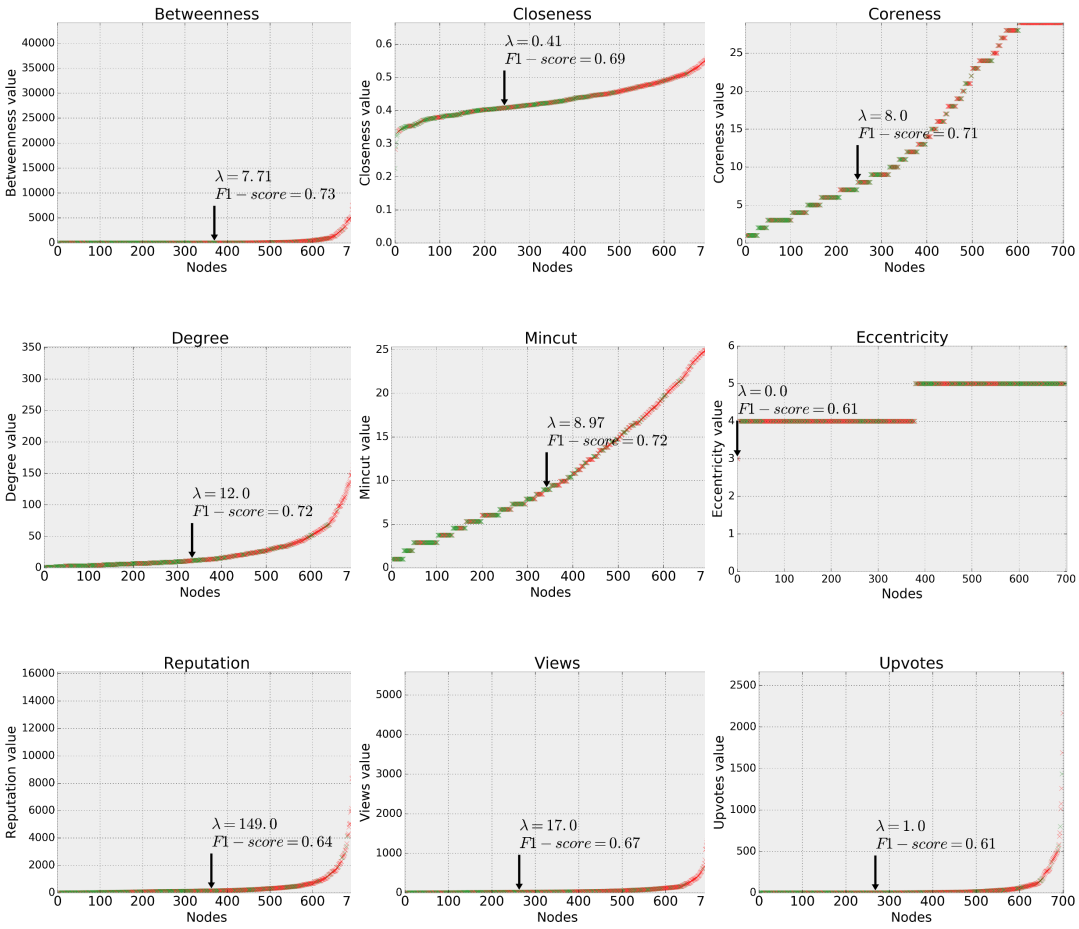


Figure 8.8.1: The figure shows the results of the STM in the training phase. We used the networks of the *Business Startups* decayed dataset. The training period was from *Nov*-2009 to *Jan*-2010 to estimate the best λ, which was then used in the test period from *Jan*-2010 to *Mar*-2010. The best value of the threshold λ is shown in the figure for each attribute associated with the value of the F1-score of the testing phase. The green and red markers indicate whether the node did or did not depart, respectively. The x-axis represents the nodes ranked according to their attribute value.

STM vs. machine learning classifier with one feature

Having trained the STM and obtained the corresponding λ for each attribute, we then predict using the attribute value at λ for different future time points. Figure 8.8.2 shows the prediction results of the STM compared to machine learning model with *one attribute*. Algorithm 8.1 is used also for constructing the training and testing FDMs using one attribute; hence, instead of using the whole set of features, we restricted the algorithm to use only one feature. To our surprise, the performance of the STM was satisfactory. For the attribute *Closeness*, the STM outperformed the machine learning model slightly with an advantage of 0.02 and 0.03 for accuracy and F1-score, respectively, averaged over prediction periods of 2, 4, 12, and 24 months. A similar advantage was found for the attributes

MinCut and *Eccentricity*. On the other hand, there was a slight advantage for the machine learning model over the STM for the attributes *Reputation* score, *Degree*, *Betweenness*, and *Upvotes*. For example, regarding the attribute *Betweenness*, the machine learning model outperformed the STM by only 0.05 and 0.07 in terms of accuracy and F1-score, respectively, averaged over prediction periods of 2, 4, 12, and 24 months. Other attributes such as *Coreness* and *Views* show no difference in the prediction performance when comparing the STM and the machine learning.

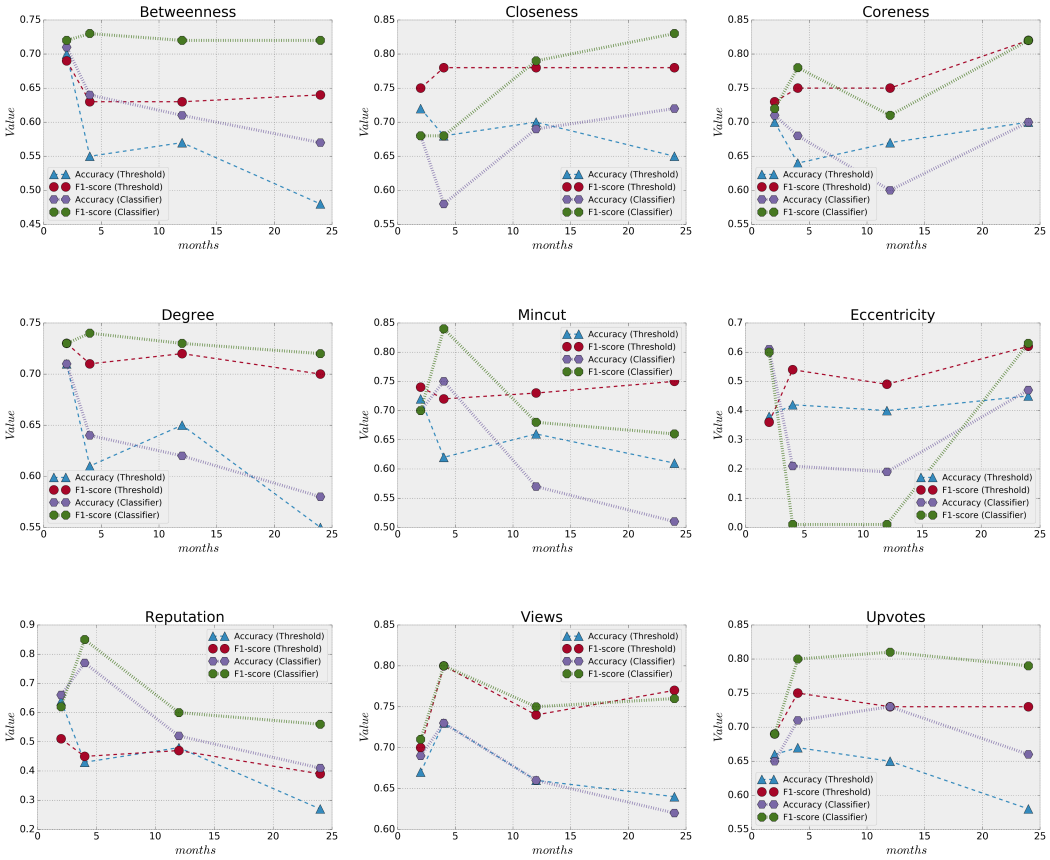


Figure 8.8.2: The figure shows the prediction performance in terms of F1-score and accuracy for the prediction using only one attribute. The figure shows the results for the prediction using the STM and the machine learning classification presented in Sections 8.7.2 and 8.7.3, respectively. We used the networks of the *Business Startups* decayed dataset. The training period was *Nov*-2009 to *Jan*-2010 to estimate the best λ, which was then used in the test periods of 2, 4, 12, and 24 months to get more insights regarding prediction performance. The same period, *Nov*-2009 to *Jan*-2010, was used for constructing the training FDM using one attribute. The testing FDM for machine learning using one attribute as constructed for the test periods 2, 4, 12, and 24 months, as well. Thus, the x-axis represents the prediction time in months, and the y-axis represents the prediction measure values.

It is worth mentioning that the best prediction result of the STM was for the *Coreness* attribute, with prediction performance at 0.83 and 0.7 for F1-score and accuracy, respectively, for the 24-

month prediction period. The machine learning model's best accuracy and F1-score were found for the attribute *Reputation* score, with values of 0.85 and 0.73, respectively.

8.8.2 Prediction using multiple attributes

In this section, we present the prediction results of our machine learning framework. We used machine learning because we may lose much information when limiting our prediction to only one attribute. We emphasize here that all of the experiments performed in this section were performed on two different datasets: one for the training phase, and the other for the testing phase, which supports the validity of our results and conclusions as it eliminates overfitting.

Feature properties

Ranking features, based on their importance, is crucial for selecting the best attributes. Figure 8.8.3 shows the importance of the features used in this chapter.

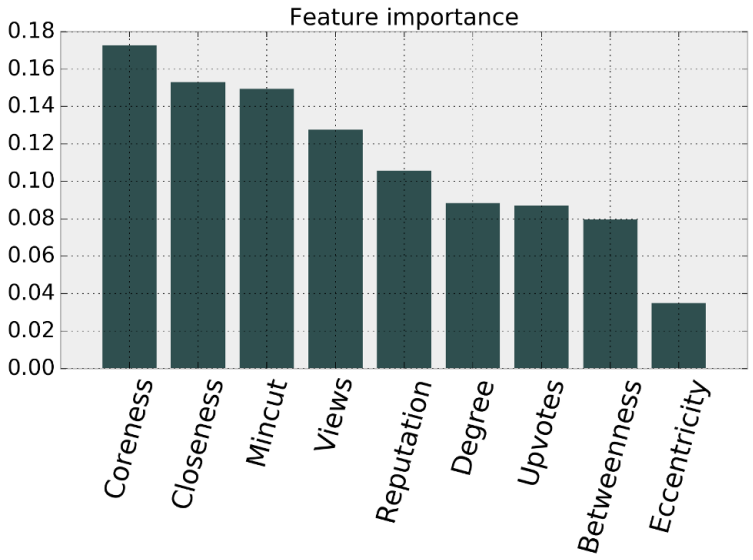


Figure 8.8.3: The figure shows the feature importance such that $\sum_i w(i) = 1$, where $w(i)$ is the feature importance of feature i. The method used for generating the importance is Random Forests, where the importance of a feature increases whenever the split in the tree using that feature minimizes the prediction error [LWSG13].

The figure shows that *Coreness* and *Closeness* are the most important network-based features and that *Views* and *Reputation* score are the most important exogenous information features. The information provided in this figure is valuable for selecting the best set of features. Thus, we report on different training and testing variations of the *FDM* as follows:

- $FDM(\text{all})$, which uses all features.
- $FDM(\text{Best}_4)$, which uses the best 4 features based on Figure 8.8.3.
- $FDM(\text{Best}_1)$, which uses the best one of the network-based features and the best one of the exogenous attributes, together.

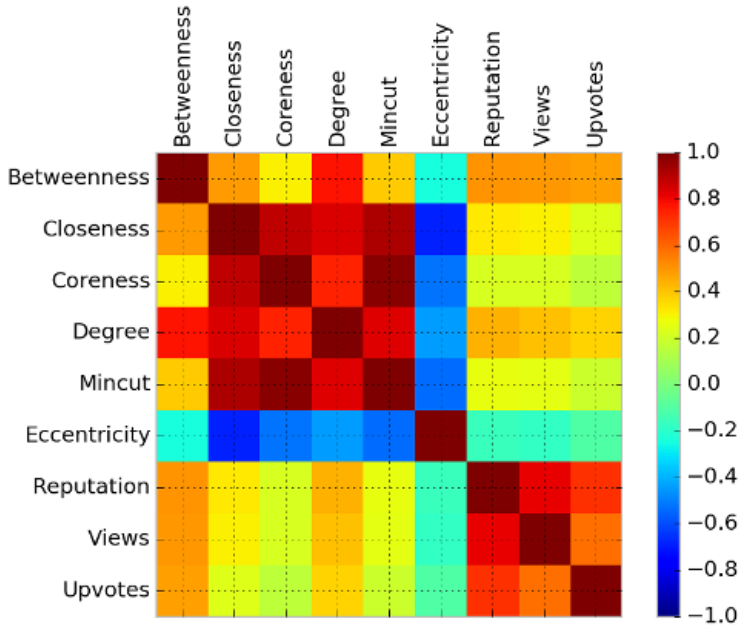


Figure 8.8.4: The figure shows Pearson's correlation coefficient values for the features used, which are defined as: $\rho(f_1, f_2) = \frac{Covariance(f_1, f_2)}{\sqrt{Variance(f_1) \cdot Variance(f_2)}}$, where $f_1, f_2 \in \mathbf{f}$ and $\rho(f_1, f_2) \in [-1, 1]$. The FDM used to generate this figure is the *Business Startups* dataset for the period between *Jan*-2010 and *Mar*-2010.

- $FDM(\text{Best}_2)$, which uses the best two of the network-based features and the best two of the exogenous attributes, together

Using the set of all features is not always the best choice due to some properties of the machine learning classifiers. For example, some classifiers are sensitive to correlated attributes and many classifiers perform poorly with low variance attributes. Thus, we provide additional analysis of the attributes in order to better understand the features. Figure 8.8.4 shows Pearson's correlation coefficient matrix of the attributes. Values close to -1 indicate a negative correlation, while values near 1 indicate a positive correlation. It is preferable to feed machine learning classifiers with as many uncorrelated features as possible. We can see in Figure 8.8.4 that the exogenous features are more correlated to each other. Also, the network-based attributes are more correlated to each other. To see this, Figure 8.8.5 provides more information about the distribution of the attributes along with a one-to-one scatter plot. For example, there is a high correlation between *Coreness* and *Degree* and between *Closeness* and *MinCut*.

Figure 8.8.5 also shows also the distribution of each attribute along the diagonal. The features *Closeness*, *Coreness*, *Degree*, and *MinCut* have more variance than the others, which explains why these attributes have a higher importance in Figure 8.8.3. The data shown in Figure 8.8.1 is clearly non-linearly separable, i.e., there exists no possible threshold that could separate the green and the

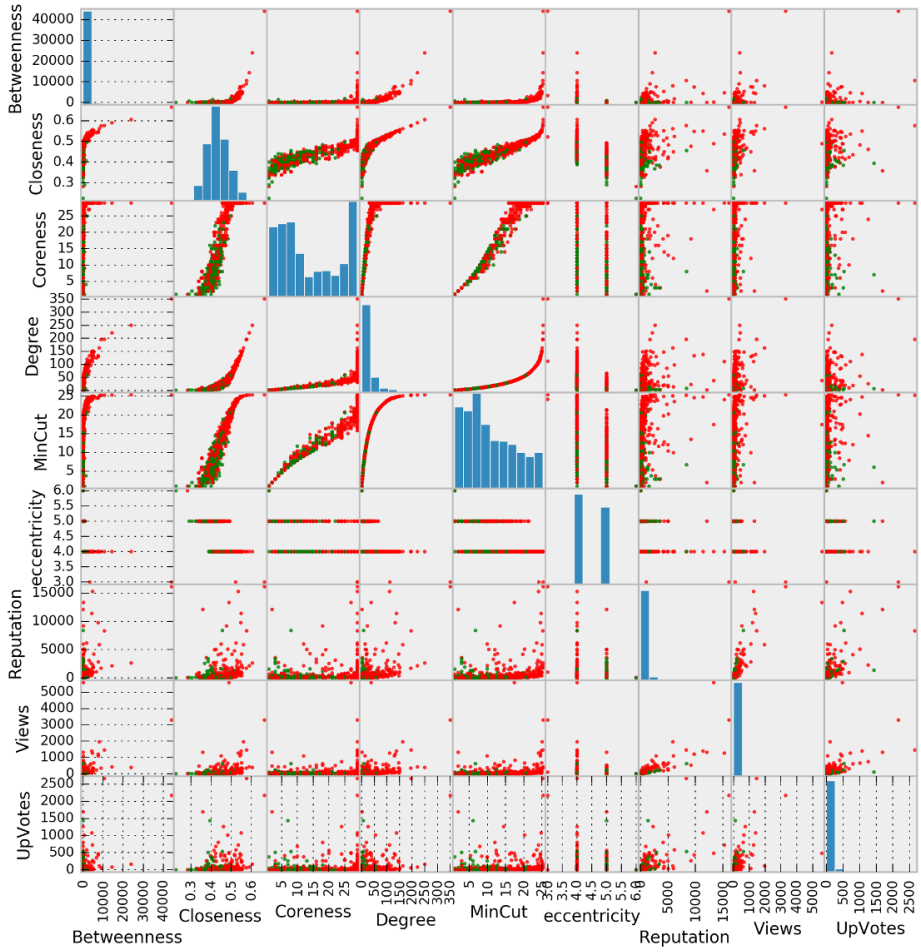


Figure 8.8.5: The figure shows the distribution of each feature (in the diagonal plots) of our model as well as the correlation plot between each pair of two attributes. The green points are nodes that departed the network, and the red ones are nodes that did not depart. The data used to generate this figure is the *Business Startups* dataset for the period between *Jan*-2010 and *Mar*-2010.

red points. The separation of the data becomes even harder when we incorporate more features, like the data points in Figure 8.8.5. In Figure 8.8.6, we show an exemplary prediction on the 2-*D* data of the attributes used. The figure illustrates how classifiers such as Support Vector Machines can provide smooth probabilistic areas for separating points. For example, the blue and the red points of *Coreness* vs. *Closeness* strongly interweave, and the SVM managed to find good separation areas compared to Logistic Regression and Decision Trees. This is why we will basically resort to the SVM in the following results[6].

[6]The technical details of the SVM can be found here [CV95] and are explained in Chapter 2.

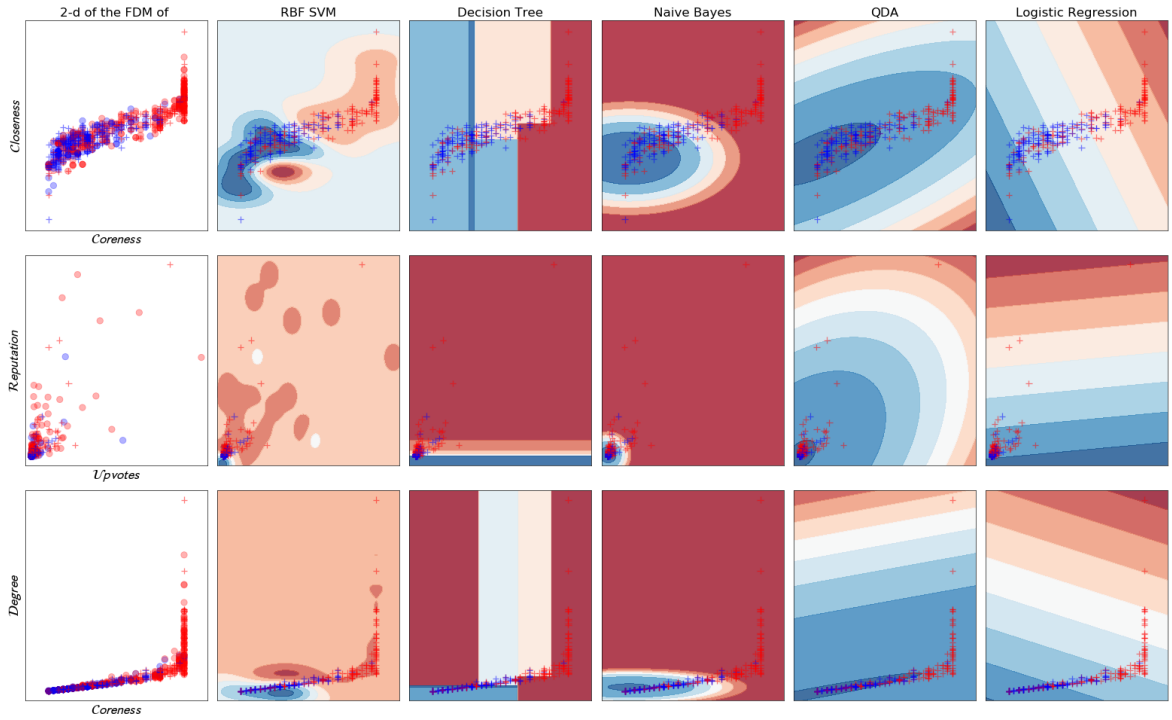


Figure 8.8.6: The figure shows the ability of the machine learning classifiers to segregate non-linearly separable data. We show a 2-D representation of the attributes of the *Business Startups* for the period between *Jan*-2010 and *Mar*-2010. The leftmost panels are the 2-D features before the classification was performed. The red points are False instances, and the blue points are True instances. The red "+" markers and the blue "+" markers are the False and True instances to be classified by the classifier, i.e., the test samples. The other points, none of which are "+" points, are the training points, where the training and the testing points were randomly split with a ratio of 60 to 40 for training and testing, respectively, with Python's random seed *zero*. The color gradient in the prediction plots is the probability of the prediction, i.e., the darker the color, the closer the probability to 1 or 0 (where 1 means the node departed and 0 means it did not depart). The classifiers used are those classifiers that give a probability as a classification result, namely: $\mathcal{SVM}$ with Gaussian kernel [CV95]; $\mathcal{DT}$: decision trees [Qui86]; $\mathcal{NB}$: Naive Bayes [Zha04]; $\mathcal{QDA}$: the Quadratic Discriminant Analysis [Cov65]; $\mathcal{LR}$: Logistic Regression [WD67]. We used the *scikit-learn* package [PVG+11] of Python for the previous algorithms with their default parameter values.

Prediction results

In this section, we present the prediction results of the machine learning prediction framework. Before presenting the results, we will show the baseline results for a sound comparison.

Baseline classifier

There is always the possibility that the classification result is due to chance. To make sure that the results we have are significant, we performed multiple experiments with a baseline classifier. These classifiers are *stratified classification* (which is classification that respects the distribution of the label) and *one class classification* (which is to predict always one class). The first baseline classifier performed better than the other one; thus, we used it as our baseline. In no case were the results of the baseline classifier better than the classifier results of our method. In most cases, the prediction results of the baseline classifier were of accuracy 50%.

Table 8.8.1 shows the prediction results of the *Business Startups* decayed dataset using the *four* variations of the framework and *four* time periods. The overall prediction results are satisfactory. The variation $Best_2$ yielded the best prediction results over the other variations for all prediction periods. One thing to note is that, when the prediction time grew during the prediction period, the prediction performance increased. One interpretation for this is that the machine learning classifiers were able to learn the depart patterns much better than the stay patterns, given that the decayed communities networks ended with a disconnected network, i.e., all of its nodes had already departed the network. Similar results were found for the *Literature* dataset in Table 8.8.2. The prediction results of the *Literature* dataset are higher than those of the *Business Startups* dataset. After investigating the two datasets, we found that the *Business Startups* community went through two phases of decay. After the first decay of the community website, there was another relaunch, which was not successful either and ended with a second phase of decay. This explains its long time span compared to other decayed communities. Thus, there was a fluctuation in the activity of that community, which made it harder for the classifiers to identify the real depart patterns in this community. Table 8.8.2 shows no significant difference between the *four* variations of the model, except for the variation $Best_1$, which shows a slight advantage over the other variations in the 2-months prediction period.

It was tempting to test the alive communities as well. Hence, we used the *Latex* alive community to also predict the departure of their members. The prediction performance for this community was also satisfactory. Again, the variation $Best_1$ showed a slight advantage over the other variations, except for the 36-months period prediction, which is considered a long time span to predict, as shown in Table 8.8.3.

Attributes	2 Months		4 Months		12 Months		24 Months	
	$\mathcal{A}$	F1	$\mathcal{A}$	F1	$\mathcal{A}$	F1	$\mathcal{A}$	F1
FDM(All)	0.72	0.74	0.68	0.79	0.69	0.79	0.76	0.85
$FDM(\text{Best}_4)$	0.73	0.74	0.81	0.72	0.68	0.77	0.72	0.82
$FDM(\text{Best}_1)$	0.74	0.73	0.66	0.73	0.66	0.76	0.7	0.81
$FDM(\text{Best}_2)$	0.72	0.75	0.80	0.87	0.81	0.88	0.78	0.87
Baseline	0.48	0.44	0.46	0.57	0.45	0.53	0.44	0.56

Table 8.8.1: The table shows the prediction results of the machine learning classifier for the networks constructed from the decayed *Business Startups* community dataset. The training period was from *Nov*-2009 to *Jan*-2010. The table shows the prediction for different testing sets, namely after 2, 4, 12, and 24 months. The prediction was done using different variations of the attributes model *FDM* as presented in Section 8.7.1. The table also shows the baseline results obtained as described in Section 8.8.2.

Attributes	2 Months		4 Months		8 Months		12 Months	
	$\mathcal{A}$	F1	$\mathcal{A}$	F1	$\mathcal{A}$	F1	$\mathcal{A}$	F1
FDM(All)	0.68	0.77	0.83	0.91	0.82	0.9	0.88	0.94
$FDM(\text{Best}_4)$	0.7	0.76	0.83	0.91	0.82	0.9	0.88	0.94
$FDM(\text{Best}_1)$	0.82	0.85	0.83	0.91	0.82	0.9	0.88	0.94
$FDM(\text{Best}_2)$	0.68	0.72	0.83	0.91	0.82	0.9	0.88	0.94
Baseline	0.62	0.75	0.50	0.63	0.72	0.83	0.69	0.81

Table 8.8.2: The table shows the prediction results of the machine learning classifier for the networks constructed from the decayed *Literature* community dataset. The training period was from *Aug*-2011 to *Sep*-2011. The table shows the prediction for different testing sets, namely after 2, 4, 12, and 24 months for different variations of the attributes model *FDM* presented in Section 8.7.1. The table also shows the baseline results obtained as described in Section 8.8.2.

Attributes	2 Months		4 Months		12 Months		24 Months		36 Months	
	$\mathcal{A}$	F1	$\mathcal{A}$	F1	$\mathcal{A}$	F1	$\mathcal{A}$	F1	$\mathcal{A}$	F1
FDM(All)	0.71	0.7	0.7	0.71	0.73	0.79	0.65	0.74	0.6	0.73
$FDM(\text{Best}_4)$	0.73	0.73	0.74	0.78	0.75	0.82	0.83	0.9	0.83	0.9
$FDM(\text{Best}_1)$	0.73	0.71	0.74	0.77	0.75	0.82	0.82	0.89	0.81	0.89
$FDM(\text{Best}_2)$	0.71	0.67	0.72	0.80	0.74	0.82	0.81	0.88	0.88	0.94
Baseline	0.52	0.44	0.52	0.51	0.51	0.52	0.47	0.55	0.48	0.58

Table 8.8.3: The table shows the prediction results of the machine learning classifier for the networks constructed from the **alive** "Latex" community dataset. The training period was from *Jun*-2010 to *Sep*-2010. The table shows the prediction for different testing sets, namely after 2, 4, 12, 24, and 36 months, for different variations of the attributes model *FDM* presented in Section 8.7.1. Being alive, the Latex community made it possible to make a prediction using 36 months. The table also shows the baseline results obtained as described in Section 8.8.2.

Datasets		4 Months		8 Months	
Train on (Decayed)	Test on (Alive)	$\mathcal{A}$	F1	$\mathcal{A}$	F1
Business Startups	Latex	0.83	0.89	0.88	0.94
	Statistics	0.79	0.84	0.79	0.81
Literature	Latex	0.72	0.74	0.80	0.89
	Statistics	0.77	0.80	0.72	0.78

Table 8.8.4: Results of cross-dataset prediction where the training was performed on decayed communities and the testing was performed on alive communities. We trained the machine learning classifier on the $FDM(\text{Best}_2)$, which provided the best results.

Then, we predicted the future activity of the members of active communities such as the *Latex* and the *Statistics* communities using the inactivity patterns found in decayed communities such as *Business Startups* and *Literature*. Table 8.8.4 shows the prediction results when the classifiers were trained on the datasets of decayed communities and tested the classifier on the datasets of alive communities. The results suggest better prediction performance than the prediction on the same communities, such as the results in Table 8.8.3. For instance, the F1-score for the time period of 4 months was 0.89 when trained on a decayed dataset compared to 0.80 when trained on the *Latex* dataset itself. The prediction also shows satisfactory results for predicting the departure of members of the *Statistics* community when learning from the decayed communities.

8.9 Discussion

8.9.1 Answering the research questions

Discussion on *RQ1*, *How efficient is it to predict members departing a social community using network-based measures?*: Based on the previous presentation of the models and the results presented in Section 8.8, it is clear that using network-based attributes provides good prediction performance in terms of F1-score and accuracy. The simple prediction model showed acceptable prediction results when only one network-based measure was used. The results were even better and more robust when multiple network-based attributes were used for the machine learning model. However, not all of the attributes were of equal quality for decay prediction. For example, the *Eccentricity* measure was rather useless as it showed bad prediction performance when using the STM. Even worse, this measure is misleading as it showed very high prediction for 24 months when using the STM, which was only the case because its initial λ was calculated as *zero*.

Discussion on *RQ2*, *What are the network-based properties of the members who departed or are about to depart a community?*: Based on Figure 8.8.2, members with less *Betweenness*, less *MinCut*, less *Degree*, less *Closeness*, or less *Coreness* are more susceptible to becoming inactive. This conclusion is also supported by Figure 8.8.5 and by the prediction using machine learning with one attribute and using the STM as shown in Figure 8.8.2. The STM can be utilized as a decay indicator when these attributes reach the corresponding λ of the members of a community.

Discussion on *RQ3*, *How helpful are the exogenous member attributes in predicting members leaving?*: The attributes used, which are based on exogenous information, also showed the potential for providing good prediction results. However, not all of these attributes were helpful. Figure 8.8.3 suggests that the network-based measures were more important than the exogenous attributes when predicting using the machine learning framework.

Discussion on *RQ4*, *Do decayed communities embrace departure patterns that can be used to study the inactivity of communities that are alive?*: Interestingly, the cross-community prediction results shown in Table 8.8.4 suggest that the departure patterns are independent of the community, as we

were able to predict the inactivity of a community from the information of another one. Apparently, the departure patterns are universal across communities when we abstract the interaction as a network.

8.9.2 Threats to validity

Network Quality

The networks used in the experiments were constructed from interactions between the members of the StackExchange sub-website as described in Section 8.5.2. To guarantee good quality of the network, we took the following steps: For each of the communities we used, a link was considered if it appeared at least once during the training period. Other values for link persistence over time yielded sparse networks. The training period was selected depending on the number of months a community survived; for example, for the *Business Startups* community, the training period for constructing the network G_{t_0} was $\delta = 45$ days. We tried different values for δ. For values of $\delta = 45 \pm 5$ days, there was no significant difference in the results. For larger values, e.g., $\delta = 90$ days, we got a few networks that were very dense and were hardly able to capture any meaningful interaction patterns; for smaller values, e.g., $\delta = 5$ days, we got too many very-sparse networks. The same argument was applied to the other communities. We do not expect these design decisions to affect the internal validity of the results.

Training Quality

The networks we used were decaying networks, which means that the nearer the network was to the time at which the community closed, the more inactive member it had. This makes prediction easier for the most recent networks. However, prediction at early time points (e.g., 2 Months in Table 8.8.1 and Table 8.8.2) showed satisfactory results, too, with the ratio between active (did not depart) and inactive (departed) members being 55:45. Additionally, we did not need any validation method (such as percentage split or k-fold cross-validation) as the training and testing sets are disjoint by design.

8.10 Conclusion

Network-based attributes are a good representative of activity behavior in online communities. The STM, which uses only one attribute, can effectively predict users' inactivity. The method we presented for predicting the decay of the members of online social communities provides information about the attributes of members who became inactive. These attributes, the network-based attributes, and some other community-dependent attributes can be used as indicators for the aliveness of an online community. In addition, these attributes can be used to take counteractions when

inactivity behavior is detected. In the context of StackExchange communities, such actions may include new questions and good answer recommendations as well as additional rewards (like badges and points) for the members. One aspect of the methods contributed to this chapter is computational complexity. Some network-based attributes are computationally expensive, especially for large and sparse networks. However, we found that the best results were obtained from attributes that are easy to compute, like *Degree* and *Coreness* [BZ03, BZ11]. We recommend starting with the STM before using the machine learning classifiers, as the machine learning classifiers are computationally expensive for large datasets. The STM provides good indications regarding which attributes to use. The optimization of the STM is computationally cheap for a sorted list; it is $\mathcal{O}(n)$, where n is the number of nodes in the graph.

Overall, the prediction performance of the presented model and framework is satisfactory. For 2-month prediction, the upper bounds are 0.85 and 0.82 for F1-score and accuracy, respectively. For the 4 months, the upper bounds are 0.91 and 0.83 for the F1-score and the accuracy, respectively. The prediction results obtained from time periods that are closer to the shutdown time of a community cannot be generalized as the life times of the decayed communities are not equal.

Part IV

Epilogue

9 Summary

It is well known that people nowadays resort to online social platforms as a basic daily activity for various reasons. This huge, unprecedented step of partly digitalizing our thoughts, actions, and behaviors, has contributed to the noticeable expansion of dozens of online social platforms in many terms, such as their market value, the impact they leave on our lives, the societal change they drive, and many others. Having said that, new challenges and problems have emerged as a result of this change in our daily life that need to be addressed on different levels to keep these changes on the right path.

This thesis has been written in line with this purpose, as its main concern is the dynamics of the interactions between people in online social networks. In this thesis, these interactions are seen from the perspective of both network science and data science, where methods and models have been developed to better understand how people interact in online social networks and also how this interaction decays overtime. In the following sections, a summary of the contributions of this thesis will be presented, followed by our conclusions, the limitations of our work, and future research directions.

9.1 Summary of contributions

During the course of this thesis, a set of contributions has been made to the body of the research and literature. These contributions are mainly related to (1) the dynamics of link formation and tie strength ranking using external information, and (2) the decay of the interaction between people

in online social networks. In the following, the contributions of this thesis will be summarized.

9.1.1 Link prediction, assessment, and ranking

In the first part of this thesis, external information, i.e., external interaction networks, were used to better understand the dynamics of link formation and tie strength ranking in online social networks. Below, we present a list of the contributions related to this part.

- In Chapter 4, a framework for predicting all the links of a social network was presented. The framework utilizes any possible external information about the members of the social network that can be represented as networks. The results of the work presented in Chapter 4 support the claim that the link formation process in social networks is driven not only by the internal structure of the social network but also by the external interactions among the social network's members.

- In Chapter 5, a framework for assessing the links of a social network as well as for ranking the tie strength of these edges was presented. Like the framework presented in Chapter 4, this framework utilize external information about the members of a social network to better assess and rank the links in a social network. The framework uses machine learning classification techniques to assess and rank the edges of a social network. The results of this chapter show that it is possible to satisfactorily assess and rank the links of a social network. The results are supported by a ground truth datasets.

9.1.2 Interaction decay dynamics

In the second part of this thesis, the decay dynamics of the members of a social network as a result of their inactivity was studied. Below, we present a list of the contributions related to this part.

- In Chapter 6, a theoretical model for capturing the mechanics of the decay dynamics in social networks was presented. Additionally, a simulation of the model was provided using data from decayed social networks.

- In Chapter 7, a model for inactivity cascades was defined, and the corresponding cascades were extracted from decayed social networks. Then, an extensive analysis was performed on these cascades, and useful decay patterns and insights into the decay process were identified. The chapter also includes a prediction model for predicting the virality and size of cascades.

- In Chapter 8, a framework and a model for predicting user inactivity in social networks was presented. The framework and the model were tested using decayed and alive social networks. The results of the prediction are satisfactory in terms of the prediction performance measures used. The results also reveal insights into the decay patterns that occur during decay processes in social networks.

9.2 Benefits of the contributions

The contributions discussed above can be beneficial regarding different aspects. Below, we list some of the expected benefits of these contributions, dividing them into two categories: (1) technical-based benefits and (2) application-based benefits.

9.2.1 Technical-based benefits

The contributions provided in Chapters 4 and 5 could not have been realized without the *Network Vectorizing* technique contributed in this thesis (cf. Section 2.4). This transformation is very useful and significantly expands the way we can employ networked data. This transformation with its two types (node-based and edge-based) bridges the gap between the networks with their classical representation and the vector space that is used heavily in machine learning. With the features data model (FDM) built from the networks, link-related problems, such as link prediction, assessment, and ranking, can be handled effectively and powerfully. Additionally, with this FDM, dealing with node-related problems, such as predicting a node's inactivity, becomes viable. Thus, this transformation is provided in this thesis as a tool for the audience that can also be used in contexts other than the topic of this thesis (online social networks).

Another technical contribution is the model presented in Chapter 6. The model and its proven viable optimization guarantees can be used in different directions in social contexts. For example, the model can be used to maximize and accelerate the decay process of unwanted interactions in malicious networks, such as terrorist networks. Conversely, the model can be used to minimize, respectively decelerate, the decay process of interactions that need to be prolonged, such as the decay of customer engagement in an online shop.

9.2.2 Application-based benefits

The contributions of this thesis enable direct applications that can be used in different directions. In the following, we present a list of some of the applications based on the contributions of this thesis.

- Generally speaking, the ultimate goal of a social network is to keep growing with meaningful and purposeful interaction among its members. Link prediction typically contributes directly to the achievement of this goal. In particular, the link prediction contribution presented in Chapter 4 can be useful for better *friend recommendations* on online social platforms because the classical friend recommendations are normally based on the social network itself, without taking into account other information that drives the link formation process in a social network. By utilizing this capability, the quality of the social network can be increased as latent links can be identified that are hard to find without utilizing external information. As a result of having more true links between the members of a social network, all of the services

provided for the members, including better update feeds and targeted ads, can be improved as a result.

- The link assessment and tie strength ranking provided in Chapter 5 have various applications. For example, *fake accounts* can be found through the false-positive links in a social network, and *privacy circles* in online social networks can be created by using the link ranking method provided in the same chapter. The method also allows increasing the quality of the links in a network, which make any subsequent network analysis more reliable. For instance, ruling out the very low-intensity links from a network makes community detection more robust.

- The applications of the work related to interaction decay dynamics presented in Chapters 6, 7, and 8 are manifold. For instance, knowing when members' activity is about to decline (cf. Chapter 8) is valuable information for service providers, enabling them to initiate counteractions and provide incentives for members to *prolong* their activity. Additionally, the cascade analysis and prediction presented in Chapter 7 provide valuable information about *influential members* whose departure may trigger a leave cascade of other members. Moreover, the patterns of member inactivity found in decayed communities in the work provided in Chapters 7 and 8 can be used as an *activity reference model* to alive social networks in the sense of regularly comparing the activity of alive social networks to the reference activity model. This will provide an online indicator about the activity of the social network being monitored.

9.3 Limitations

Throughout the course of this thesis, various challenges arose. Many of them were solved, but some remain as limitations of this work. In the following, some of the limitations of this work will be presented.

Our method for link prediction, assessment, and ranking (cf. Chapters 4 and 5) used data from a social network in addition to external information. Such external information might not always be available, which represents a challenge regarding the application of the method. Having said that, this challenge may not be evident when we talk about the interoperability of systems and services provided online. For example, it is not difficult for a social network platform to gain this external information. For example, location, preferences stored in some browsers' cookies, and sharing information among different kind of accounts from different services are rich sources of external information that can be utilized.

Another issue regards the ground truth used for the tie strength ranking contribution provided in Chapter 5. Ground truth data for tie strength ranking is hard to obtain. Thus, we resorted to a binary ranker as a gold standard measure to evaluate the tie strength ranking provided by the method.

Finally, from the network perspective, decay dynamics can be seen in terms of node inactivity

and/or link removal dynamics. In this thesis, we considered only node inactivity as an indication of a network undergoing a decay process. Although this sounds reasonable, the decay process might not be limited to node inactivity. For example, link removal can also be seen as a form of decay dynamics, but this was beyond the scope of this thesis.

9.4 Future outlook

Over the course of this thesis, a lot of interesting research questions emerged mainly from challenges faced during the work. In the following, we present a list of future directions.

9.4.1 Handling the imbalanced nature of sparse networks

Social networks are naturally sparse. This makes the transformed features data model imbalanced, i.e., the number of vectors representing an edge is much smaller than the number of vectors representing a non-edge. Thus, any edge proximity-based model is inherently imbalanced. Many techniques exist in the literature to avoid the classification limitation in the case of imbalanced datasets [KKP06]. In this thesis, we used SMOTE (Synthetic Minority Over-sampling Technique) [CBHK02], which did not provide any improvement in the prediction performance due to the small datasets we had. The literature contains a lot of techniques that can be used with larger datasets to handle the imbalanced nature of some datasets [KKP06]. Thus, an interesting question in this context is:

> How can we handle the imbalanced nature of sparse networks in the underlying features data model?

One idea that we tried, but did not investigate sufficiently, is the use of random graphs with a fixed degree sequence as data augmentation. The very preliminary results were promising, though the method needs a lot of work and investigation.

9.4.2 Feature selection of networked-data models

The work presented in this thesis builds mainly on calculating topological measures for both edges and nodes. Many of these measures are computationally expensive. Thus, feature selection is one of the challenges we faced during the course of this thesis. To overcome this challenge, we eliminated features that are highly correlated with other features that can be easily computed, such as degree and coreness. An interesting research question in this context is:

> What are the features that are computationally inexpensive and can be used such that an edge or a node is represented well in the features data model?

9.4.3 Network Noisification

In many network analysis applications, the robustness of the contributed method needs to be tested. For example, the work in Chapter 5 contributes a framework for identifying noisy edges. In order to test this framework, we injected a random number of edges into the network in an attempt to find out whether the framework would be able to find these links or not. Now the injected edges are not necessarily noise; an injected edge may be a latent edge that was simply not observed in the dataset. Thus, a robust method for adding noisy edges into a network is needed. Hence, an interesting research question in this context is:

How can a network be efficiently noisified?

9.5 Final words

In this thesis, we presented our work on understanding how links are formed in social networks and how these links can be assessed and ranked using external information. In addition to that, we investigated the decay dynamics of the interaction among the members of a social network. These two categories are intended as a step towards understanding how people behave in online social networks. They are also meant as a step towards coping with the changes imposed by these online social networks on our lives. Understanding the dynamics of the members of online social networks remains an active research area, and this work constitutes as an attempt to answer some of the questions in that respect.

Bibliography

[AA03] Lada A Adamic and Eytan Adar. Friends and neighbors on the web. *Social networks*, 25(3):211–230, 2003.

[AA17] M. Abufouda and Hadil Abukwaik. On using network science in mining developers collaboration in software engineering: A systematic literature review. *International Journal of Data Mining and Knowledge Management Process (IJDKP)*, 7(7,5/6):1–20, 2017.

[ABS+12] Luca Maria Aiello, Alain Barrat, Rossano Schifanella, Ciro Cattuto, Benjamin Markines, and Filippo Menczer. Friendship prediction and homophily in social media. *ACM Transactions on the Web (TWEB)*, 6(2):9, 2012.

[ABSC10] Muhammad Aurangzeb Ahmad, Zoheb Borbora, Jaideep Srivastava, and Noshir Contractor. Link prediction across multiple social networks. In *Data Mining Workshops (ICDMW), 2010 IEEE International Conference on*, pages 911–918. IEEE, 2010.

[Abu18a] Mohammed Abufouda. Community aliveness: Discovering interaction decay patterns in online social communities. In Reda Alhajj, H. Ulrich Hoppe, Tobias Hecking, Piotr Bródka, and Przemyslaw Kazienko, editors, *Network Intelligence Meets User Centered Social Media Networks*, pages 97–118. Springer International Publishing, 2018.

[Abu18b] Mohammed Abufouda. Postmortem analysis of decayed online social communities: Cascade pattern analysis and prediction. *Complexity*, 2018(3873601):1–17, 2018.

[AHK+07] Yong-Yeol Ahn, Seungyeop Han, Haewoon Kwak, Sue Moon, and Hawoong Jeong. Analysis of topological characteristics of huge online social networking services. In *Proceedings of the 16th international conference on World Wide Web*, pages 835–844. ACM, 2007.

[AHSW11] Sitaram Asur, Bernardo A Huberman, Gabor Szabo, and Chunyan Wang. Trends in social media: Persistence and decay. In *Fifth International AAAI Conference on Weblogs and Social Media*, 2011.

[AHZ11] Mohammad Al Hasan and Mohammed J Zaki. A survey of link prediction in social networks. In *Social network data analytics*, pages 243–275. Springer, 2011.

[AJB99] Réka Albert, Hawoong Jeong, and Albert-László Barabási. Internet: Diameter of the world-wide web. *nature*, 401(6749):130, 1999.

[Alt92] Naomi S Altman. An introduction to kernel and nearest-neighbor nonparametric regression. *The American Statistician*, 46(3):175–185, 1992.

[AMMIL12] Yaser S Abu-Mostafa, Malik Magdon-Ismail, and Hsuan-Tien Lin. *Learning from data*, volume 4. AMLBook New York, NY, USA:, 2012.

[AMS09] Sinan Aral, Lev Muchnik, and Arun Sundararajan. Distinguishing influence-based contagion from homophily-driven diffusion in dynamic networks. *Proceedings of the National Academy of Sciences*, 106(51):21544–21549, 2009.

[AS64] Milton Abramowitz and Irene A Stegun. *Handbook of mathematical functions: with formulas, graphs, and mathematical tables*, volume 55. Courier Corporation, 1964.

[ASBS00] Luıs A Nunes Amaral, Antonio Scala, Marc Barthelemy, and H Eugene Stanley. Classes of small-world networks. *Proceedings of the national academy of sciences*, 97(21):11149–11152, 2000.

[AZ14] M. Abufouda and K. A. Zweig. Interactions around social networks matter: Predicting the social network from associated interaction networks. In *2014 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM 2014)*, pages 142–145, Aug 2014.

[AZ15] Mohammed Abufouda and Katharina A. Zweig. Are we really friends?: Link assessment in social networks using multiple associated interaction networks. In *Proceedings of the 24th International Conference on World Wide Web*, WWW '15 Companion, pages 771–776, New York, NY, USA, 2015. ACM.

[AZ17] Mohammed Abufouda and Katharina A. Zweig. Stochastic modeling of the decay dynamics of online social networks. In Bruno Gonçalves, Ronaldo Menezes, Roberta Sinatra, and Vinko Zlatic, editors, *Complex Networks VIII*, pages 119–131. Springer International Publishing, 2017.

[AZ18a] Mohammed Abufouda and Katharina A Zweig. A theoretical model for understanding the dynamics of online social networks decay. In *arXiv preprint arXiv:1610.01538*, 2018.

[AZ18b] Mohammed Abufouda and Katharina Anna Zweig. Link classification and tie strength ranking in online social networks with exogenous interaction networks. In *Behavioral Analytics in Social and Ubiquitous Environments*, pages 1–27. Springer, 2018.

[BA99] Albert-László Barabási and Réka Albert. Emergence of scaling in random networks. *American Association for the Advancement of Science*, 286(5439):509–512, 1999.

[BBR+12] Lars Backstrom, Paolo Boldi, Marco Rosa, Johan Ugander, and Sebastiano Vigna. Four degrees of separation. In *Proceedings of the 4th Annual ACM Web Science Conference*, pages 33–42. ACM, 2012.

[Bea14] S. Boccaletti and et. al. The structure and dynamics of multilayer networks. *Physics Reports*, 544(1):1 – 122, 2014.

[Ber95] DP Bertsekas. *Nonlinear Programming*. Athena Scientific, 1995.

[BGoo] Venkatesh Bala and Sanjeev Goyal. A noncooperative model of network formation. *Econometrica*, 68(5):1181–1229, 2000.

[BHKL06] Lars Backstrom, Dan Huttenlocher, Jon Kleinberg, and Xiangyang Lan. Group formation in large social networks: membership, growth, and evolution. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 44–54. ACM, 2006.

[BJN+02] Albert-Laszlo Barabási, Hawoong Jeong, Zoltan Néda, Erzsebet Ravasz, Andras Schubert, and Tamas Vicsek. Evolution of the social network of scientific collaborations. *Physica A: Statistical mechanics and its applications*, 311(3):590–614, 2002.

[BK14] Christian Bauckhage and Kristian Kersting. Strong regularities in growth and decline of popularity of social media services. *arXiv preprint arXiv:1406.6529*, 2014.

[BKL+15] Kshipra Bhawalkar, Jon Kleinberg, Kevin Lewi, Tim Roughgarden, and Aneesh Sharma. Preventing unraveling in social networks: the anchored k-core problem. *SIAM Journal on Discrete Mathematics*, 29(3):1452–1475, 2015.

[BKR10] N. Benchettara, R. Kanawati, and C. Rouveirol. Supervised machine learning applied to link prediction in bipartite social networks. In *Advances in Social Networks Analysis and Mining (ASONAM), 2010 International Conference on*, pages 326–330, Aug 2010.

[BL11] Lars Backstrom and Jure Leskovec. Supervised random walks: predicting and recommending links in social networks. In *Proceedings of the fourth ACM international conference on Web search and data mining*, pages 635–644. ACM, 2011.

[BME+14] Philippe Bardou, Jérôme Mariette, Frédéric Escudié, Christophe Djemiel, and Christophe Klopp. jvenn: an interactive venn diagram viewer. *BMC bioinformatics*, 15(1):293, 2014.

[BP14] George A Barnett and Han Woo Park. Examining the international internet using multiple measures: new methods for measuring the communication base of globalized cyberspace. *Quality & Quantity*, 48(1):563–575, 2014.

[Buroo] Ronald S Burt. Decay functions. *Social networks*, 22(1):1–28, 2000.

[BV04] Stephen Boyd and Lieven Vandenberghe. *Convex optimization*. Cambridge university press, 2004.

[BWoo] Alain Barrat and Martin Weigt. On the properties of small-world network models. *The European Physical Journal B-Condensed Matter and Complex Systems*, 13(3):547–560, 2000.

[BZ03] Vladimir Batagelj and Matjaz Zaversnik. An o(m) algorithm for cores decomposition of networks. *arXiv preprint cs/0310049*, 2003.

[BZ11] Vladimir Batagelj and Matjaž Zaveršnik. Fast algorithms for determining (generalized) core groups in social networks. *Advances in Data Analysis and Classification*, 5(2):129–145, Jul 2011.

[BZT13] Zhifeng Bao, Yong Zeng, and YC Tay. Sonlp: Social network link prediction by principal component regression. In *Proceedings of the 2013 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining*, pages 364–371. ACM, 2013.

[CALR13] Carlo Vittorio Cannistraci, Gregorio Alanis-Lobato, and Timothy Ravasi. From link-prediction in brain connectomes and protein interactomes to the local-community-paradigm in complex networks. *Scientific reports*, 3, 2013.

[CAU47] Augustin CAUCHY. Methode generale pour la resolution des systemes d'equations simultanees. *Comptes Rendus*, 25:536–538, 1847.

[CBHK02] Nitesh V Chawla, Kevin W Bowyer, Lawrence O Hall, and W Philip Kegelmeyer. Smote: synthetic minority over-sampling technique. *Journal of artificial intelligence research*, 16:321–357, 2002.

[CBS14] Simrat Singh Chhabra, Ajit Brundavanam, and Saswata Shannigrahi. An alternative explanation for the rise and fall of myspace. *arXiv preprint arXiv:1403.5617*, 2014.

[CHLN04] Jin Chen, Wynne Hsu, Mong Li Lee, and See-Kiong Ng. Systematic assessment of high-throughput experimental data for reliable protein interactions using network topology. In *Proceedings of the 16th IEEE International Conference on Tools with Artificial Intelligence*, pages 368–372. IEEE Computer Society, 2004.

[CJ04] Robin Cowan and Nicolas Jonard. Network structure and the diffusion of knowledge. *Journal of economic Dynamics and Control*, 28(8):1557–1575, 2004.

[CML11] Eunjoon Cho, Seth A. Myers, and Jure Leskovec. Friendship and mobility: User movement in location-based social networks. In *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '11, pages 1082–1090. ACM, 2011.

[Cov65] Thomas M Cover. Geometrical and statistical properties of systems of linear inequalities with applications in pattern recognition. *IEEE transactions on electronic computers*, EC-14(3):326–334, 1965.

[CS14] John Cannarella and Joshua A Spechler. Epidemiological modeling of online social network dynamics. *arXiv preprint arXiv:1401.4208*, 2014.

[CSC+06] Andrea Capocci, Vito DP Servedio, Francesca Colaiori, Luciana S Buriol, Debora Donato, Stefano Leonardi, and Guido Caldarelli. Preferential attachment in the growth of social networks: The internet encyclopedia wikipedia. *Physical Review E*, 74(3):036116, 2006.

[CT17] Emily S. C. Ching and H. C. Tam. Reconstructing links in directed networks from noisy dynamics. *Phys. Rev. E*, 95:010301, Jan 2017.

[Cun85] William H. Cunningham. On submodular function minimization. *Combinatorica*, 5(3):185–192, Sep 1985.

[CV95] Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine learning*, 20(3):273–297, 1995.

[DC11] Munmun De Choudhury. Tie formation on Twitter: Homophily and structure of egocentric networks. In *Privacy, Security, Risk and Trust (PASSAT) and 2011 IEEE Third Inernational Conference on Social Computing (SocialCom), 2011 IEEE Third International Conference on*, pages 465–470. IEEE, 2011.

[DC12] Bruce A Desmarais and Skyler J Cranmer. Statistical mechanics of networks: Estimation and uncertainty. *Physica A: Statistical Mechanics and its Applications*, 391(4):1865–1876, 2012.

[Dea02] Charlotte M Deane and et. al. Protein interactions: two methods for assessment of the reliability of high throughput observations. *Molecular & Cellular Proteomics*, 1(5):349–356, 2002.

[DGH+18] Himel Dev, Chase Geigle, Qingtao Hu, Jiahui Zheng, and Hari Sundaram. The size conundrum: Why online knowledge markets can fail at scale. In *Proceedings of the 2018 World Wide Web Conference on World Wide Web*, pages 65–75. International World Wide Web Conferences Steering Committee, 2018.

[Dic45] Lee R Dice. Measures of the amount of ecologic association between species. *Ecology*, 26(3):297–302, 1945.

[DLC11] D. Davis, R. Lichtenwalter, and N. V. Chawla. Multi-relational link prediction in heterogeneous information networks. In *2011 International Conference on Advances in Social Networks Analysis and Mining*, pages 281–288, July 2011.

[DM00] Sergey N Dorogovtsev and José Fernando F Mendes. Scaling behaviour of developing and decaying networks. *EPL (Europhysics Letters)*, 52(1):33, 2000.

[DSP11] Hially Rodrigues De Sá and Ricardo BC Prudêncio. Supervised link prediction in weighted networks. In *Neural Networks (IJCNN), The 2011 International Joint Conference on*, pages 2281–2288. IEEE, 2011.

[DYB03] Gerald F Davis, Mina Yoo, and Wayne E Baker. The small world of the american corporate elite, 1982-2001. *Strategic organization*, 1(3):301–326, 2003.

[EDB02] Holger Ebel, Jörn Davidsen, and Stefan Bornholdt. Dynamics of social networks. *Complexity*, 8(2):24–27, 2002.

[ER59] Paul Erdős and Alfréd Rényi. On random graphs i. *Publicationes Mathematicae (Debrecen)*, 6:290–297, 1959 1959.

[Eve12] Sean F. Everton. *The Noordin Top Terrorist Network*. Cambridge University Press, 2012.

[FFF99] Michalis Faloutsos, Petros Faloutsos, and Christos Faloutsos. On power-law relationships of the internet topology. In *Proceedings of the Conference on Applications, Technologies, Architectures, and Protocols for Computer Communication*, SIGCOMM '99, pages 251–262, New York, NY, USA, 1999. ACM.

[FHT01] Jerome Friedman, Trevor Hastie, and Robert Tibshirani. *The elements of statistical learning*, volume 1. Springer series in statistics New York, NY, USA:, 2001.

[FLL06] Trevor Fenner, Mark Levene, and George Loizou. A stochastic model for the evolution of the web allowing link deletion. *ACM Transactions on Internet Technology (TOIT)*, 6(2):117–130, 2006.

[For10] Santo Fortunato. Community detection in graphs. *Physics reports*, 486(3-5):75–174, 2010.

[Fre77] Linton C Freeman. A set of measures of centrality based on betweenness. *Sociometry*, pages 35–41, 1977.

[Fri02] Jerome H Friedman. Stochastic gradient boosting. *Computational Statistics & Data Analysis*, 38(4):367–378, 2002.

[FTL+11] Michael Fire, Lena Tenenboim, Ofrit Lesser, Rami Puzis, Lior Rokach, and Yuval Elovici. Link prediction in social networks using computationally efficient topological features. In *Privacy, Security, Risk and Trust (PASSAT) and 2011 IEEE Third Inernational Conference on Social Computing (SocialCom), 2011 IEEE Third International Conference*, pages 73–80. IEEE, 2011.

[GAHW15] Sharad Goel, Ashton Anderson, Jake Hofman, and Duncan J Watts. The structural virality of online diffusion. *Management Science*, 62(1):180–196, 2015.

[GER12] Mangesh Gupte and Tina Eliassi-Rad. Measuring tie strength in implicit social networks. In *Proceedings of the 4th Annual ACM Web Science Conference*, WebSci '12, pages 109–118, New York, NY, USA, 2012. ACM.

[GFKT02] Lisa Getoor, Nir Friedman, Daphne Koller, and Benjamin Taskar. Learning probabilistic models of link structure. *Journal of Machine Learning Research*, 3(Dec):679–707, 2002.

[GFT01] Lise Getoor, Nir Friedman, and Benjamin Taskar. Probabilistic models of relational structure. In *In Proceedings of the 18th International Conference on Machine Learning*. Citeseer, 2001.

[GHFX07] Fan Guo, Steve Hanneke, Wenjie Fu, and Eric P Xing. Recovering temporally rewiring networks: A model-based approach. In *Proceedings of the 24th international conference on Machine learning*, pages 321–328. ACM, 2007.

[Gil12] Eric Gilbert. Predicting tie strength in a new medium. In *Proceedings of the ACM 2012 Conference on Computer Supported Cooperative Work*, CSCW '12, pages 1047–1056, New York, NY, USA, 2012. ACM.

[GK09] Eric Gilbert and Karrie Karahalios. Predicting tie strength with social media. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 211–220. ACM, 2009.

[GMS13] David Garcia, Pavlin Mavrodiev, and Frank Schweitzer. Social resilience in online communities: The autopsy of friendster. In *Proceedings of the first ACM conference on Online social networks*, pages 39–50. ACM, 2013.

[GN02] M. Girvan and M. E. J. Newman. Community structure in social and biological networks. *Proceedings of the National Academy of Sciences*, 99(12):7821–7826, 2002.

[Gou13] Georgios Gousios. The ghtorrent dataset and tool suite. In *Proceedings of the 10th Working Conference on Mining Software Repositories*, MSR'13, pages 233–236, 2013.

[GR03] Debra S. Goldberg and Frederick P. Roth. Assessing experimentally derived interactions in a small world. *Proceedings of the National Academy of Sciences*, 100(8):4372–4376, 2003.

[Gra77] Mark S Granovetter. The strength of weak ties. In *Social networks*, pages 347–367. Elsevier, 1977.

[HA99] Bernardo A Huberman and Lada A Adamic. Internet: growth dynamics of the world-wide web. *Nature*, 401(6749):131, 1999.

[HA04] Bernardo A Huberman and Lada A Adamic. Information dynamics in the networked world. In *Complex networks*, pages 371–398. Springer, 2004.

[HCSZ06] Mohammad Al Hasan, Vineet Chaoji, Saeed Salem, and Mohammed Zaki. Link prediction using supervised learning. In *In Proc. of SDM 06 workshop on Link Analysis, Counterterrorism and Security*, 2006.

[HHHZ12] Emöke-Agnes Horvat, Michael Hanselmann, Fred A. Hamprecht, and Katharina A. Zweig. One plus one makes three (for social networks). *PLOS ONE*, 7(4):1–8, 04 2012.

[HL81] Paul W Holland and Samuel Leinhardt. An exponential family of probability distributions for directed graphs. *Journal of the american Statistical association*, 76(373):33–50, 1981.

[HM82] James A Hanley and Barbara J McNeil. The meaning and use of the area under a receiver operating characteristic (roc) curve. *Radiology*, 143(1):29–36, 1982.

[Hoe63] Wassily Hoeffding. Probability inequalities for sums of bounded random variables. *Journal of the American statistical association*, 58(301):13–30, 1963.

[IFF01] Satoru Iwata, Lisa Fleischer, and Satoru Fujishige. A combinatorial strongly polynomial algorithm for minimizing submodular functions. *Journal of the ACM (JACM)*, 48(4):761–777, 2001.

[Isi25] Ernst Ising. Beitrag zur theorie des ferromagnetismus. *Zeitschrift für Physik*, 31(1):253–258, 1925.

[Jac03] Matthew O Jackson. A survey of network formation models: stability and efficiency. *Group Formation in Economics: Networks, Clubs, and Coalitions*, pages 11–49, 2003.

[JGN01] Emily M Jin, Michelle Girvan, and Mark EJ Newman. Structure of growing social networks. *Physical review E*, 64(4):046132, 2001.

[JLDB17] T. Jaya Lakshmi and S. Durga Bhavani. *Link Prediction in Temporal Heterogeneous Networks*, pages 83–98. Springer International Publishing, 2017.

[JMBO01] Hawoong Jeong, Sean P Mason, A-L Barabási, and Zoltan N Oltvai. Lethality and centrality in protein networks. *Nature*, 411(6833):41, 2001.

[JSB+13] Jason J. Jones, Jaime E. Settle, Robert M. Bond, Christopher J. Fariss, Cameron Marlow, and James H. Fowler. Inferring tie strength from online directed behavior. *PLOS ONE*, 8(1):1–6, 01 2013.

[JTA+00] Hawoong Jeong, Bálint Tombor, Réka Albert, Zoltan N Oltvai, and A-L Barabási. The large-scale organization of metabolic networks. *Nature*, 407(6804):651, 2000.

[JW02] Matthew O Jackson and Alison Watts. On the formation of interaction networks in social coordination games. *Games and Economic Behavior*, 41(2):265–291, 2002.

[JWHT13] Gareth James, Daniela Witten, Trevor Hastie, and Robert Tibshirani. *An introduction to statistical learning*, volume 112. Springer, 2013.

[KA06] Hisashi Kashima and Naoki Abe. A parameterized probabilistic model of network evolution for supervised link prediction. In *Sixth International Conference on Data Mining (ICDM'06)*, pages 340–349. IEEE, 2006.

[Kan63] K.J. Kansky. *Structure of Transportation Networks: Relationships Between Network Geometry and Regional Characteristics*. Number no. 84 in 23cm. thesis-university of chicago. University of Chicago., 1963.

[Kee05] Matt Keeling. The implications of network structure for epidemic dynamics. *Theoretical population biology*, 67(1):1–8, 2005.

[KJB+90] Peter D Killworth, Eugene C Johnsen, H Russell Bernard, Gene Ann Shelley, and Christopher McCarty. Estimating the size of personal networks. *Social Networks*, 12(4):289–312, 1990.

[KKP06] Sotiris Kotsiantis, Dimitris Kanellopoulos, and Panayiotis Pintelas. Handling imbalanced datasets: A review. *GESTS International Transactions on Computer Science and Engineering*, 30(1):25–36, 2006.

[KL51] Solomon Kullback and Richard A Leibler. On information and sufficiency. *The annals of mathematical statistics*, 22(1):79–86, 1951.

[Kle98] Jon M Kleinberg. Authoritative sources in a hyperlinked environment. In *In Proceedings of the ACM-SIAM Symposium on Discrete Algorithms*. Citeseer, 1998.

[KLP+05] Dirk Koschützki, Katharina Anna Lehmann, Leon Peeters, Stefan Richter, Dagmar Tenfelde-Podehl, and Oliver Zlotowski. Centrality indices. In *Network analysis*, pages 16–61. Springer, 2005.

[KLSS+15] Arash Abolghasemi Kordestani, Moez Limayem, Esmail Salehi-Sangari, Henrik Blomgren, and Afshin Afsharipour. Why a few social networking sites succeed while many fail. In *The Sustainable Global Marketplace*, pages 283–285. Springer, 2015.

[KM70] Charles Korte and Stanley Milgram. Acquaintance networks between racial groups: Application of the small world method. *Journal of Personality and social Psychology*, 15(2):101, 1970.

[KNT06] Ravi Kumar, Jasmine Novak, and Andrew Tomkins. Structure and evolution of online social networks. In *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '06, pages 611–617, New York, NY, USA, 2006. ACM.

[KPS09] J. Kawale, A. Pal, and J. Srivastava. Churn prediction in mmorpgs: A social influence based approach. In *Computational Science and Engineering, 2009. CSE '09. International Conference on*, volume 4, pages 423–428, Aug 2009.

[KRC+11] Marcel Karnstedt, Matthew Rowe, Jeffrey Chan, Harith Alani, and Conor Hayes. The effect of user features on churn in social networks. In *Proceedings of the 3rd International Web Science Conference*, page 23. ACM, 2011.

[KSSF16] S. Kumar, F. Spezzano, V. S. Subrahmanian, and C. Faloutsos. Edge weight prediction in weighted signed networks. In *2016 IEEE 16th International Conference on Data Mining (ICDM)*, pages 221–230, Dec 2016.

[KW06] Gueorgi Kossinets and Duncan J Watts. Empirical analysis of an evolving social network. *science*, 311(5757):88–90, 2006.

[KWL12] Sanjay Ram Kairam, Dan J Wang, and Jure Leskovec. The life and death of online groups: Predicting group growth and longevity. In *Proceedings of the fifth ACM international conference on Web search and data mining*, pages 673–682. ACM, 2012.

[Laz12] Emmanuel Lazega. *The Collegial Phenomenon: The Social Mechanisms of Cooperation among Peers in a Corporate Law Partnership*. Oxford: Oxford University Press, 2012.

[LBKT08] Jure Leskovec, Lars Backstrom, Ravi Kumar, and Andrew Tomkins. Microscopic evolution of social networks. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 462–470. ACM, 2008.

[Lin91] Jianhua Lin. Divergence measures based on the shannon entropy. *IEEE Transactions on Information theory*, 37(1):145–151, 1991.

[LKF05] Jure Leskovec, Jon Kleinberg, and Christos Faloutsos. Graphs over time: densification laws, shrinking diameters and possible explanations. In *Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining*, pages 177–187. ACM, 2005.

[LM54] Paul F Lazarsfeld and Robert K Merton. Friendship as a social process: A substantive and methodological analysis. *Freedom and control in modern society*, 18(1):18–66, 1954.

[LNK03] David Liben-Nowell and Jon Kleinberg. The link prediction problem for social networks. In *Proceedings of the 12th Intern. Conference on Information and Knowledge Management*, pages 556–559. ACM, 2003.

[Lov83] László Lovász. Submodular functions and convexity. In *Mathematical Programming The State of the Art*, pages 235–257. Springer, 1983.

[LPP11] Joseph T Lizier, Siddharth Pritam, and Mikhail Prokopenko. Information dynamics in small-world boolean networks. *Artificial life*, 17(4):293–314, 2011.

[LSTD10] Zhengdong Lu, Berkant Savas, Wei Tang, and Inderjit S Dhillon. Supervised link prediction using multiple sources. In *Data Mining (ICDM), 2010 IEEE 10th International Conference on*, pages 923–928. IEEE, 2010.

[LWSG13] Gilles Louppe, Louis Wehenkel, Antonio Sutera, and Pierre Geurts. Understanding variable importances in forests of randomized trees. In *Advances in neural information processing systems*, pages 431–439, 2013.

[LZ11] Linyuan Lü and Tao Zhou. Link prediction in complex networks: A survey. *Physica A: statistical mechanics and its applications*, 390(6):1150–1170, 2011.

[MBC16] Víctor Martínez, Fernando Berzal, and Juan-Carlos Cubero. A survey of link prediction in complex networks. *ACM Computing Surveys (CSUR)*, 49(4):69, 2016.

[MCC13] Jeffrey McGee, James Caverlee, and Zhiyuan Cheng. Location prediction in social media based on tie strength. In *Proceedings of the 22Nd ACM International Conference on Information & Knowledge Management*, CIKM '13, pages 459–468, New York, NY, USA, 2013. ACM.

[MDCT13] Ole J Mengshoel, Raj Desai, Andrew Chen, and Brian Tran. Will we connect again? machine learning for link prediction in mobile social networks. In *11th Workshop on Mining and Learning with Graphs*, 2013.

[Mil67] Stanley Milgram. The small-world problem. *Psychology Today*, 1(1), 1967.

[MJ38] Jacob L Moreno and Helen H Jennings. Statistics of social configurations. *Sociometry*, pages 342–374, 1938.

[MKG+08] Alan Mislove, Hema Swetha Koppula, Krishna P Gummadi, Peter Druschel, and Bobby Bhattacharjee. Growth of the flickr social network. In *Proceedings of the first workshop on Online social networks*, pages 25–30. ACM, 2008.

[MLC10] Ian McCulloh, Joshua Lospinoso, and Kathleen M Carley. The link probability model: A network simulation alternative to the exponential random graph model. *Available at SSRN 2729285*, 2010.

[MMR13] Matteo Magnani, Barbora Micenková, and Luca Rossi. Combinatorial analysis of multiple networks. *The Computing Research Repository (CoRR)*, abs/1303.4986, 2013.

[Mor53] Jacob Levy Moreno. *Who shall survive? Foundations of sociometry, group psychotherapy and socio-drama*. Beacon House, 1953.

[MRT18] Mehryar Mohri, Afshin Rostamizadeh, and Ameet Talwalkar. *Foundations of machine learning*. MIT press, 2018.

[MSLC01] Miller McPherson, Lynn Smith-Lovin, and James M Cook. Birds of a feather: Homophily in social networks. *Annual review of sociology*, 27(1):415–444, 2001.

[MV13] Fragkiskos D Malliaros and Michalis Vazirgiannis. To stay or not to stay: modeling engagement dynamics in social graphs. In *Proceedings of the 22nd ACM international conference on Information & Knowledge Management*, pages 469–478. ACM, 2013.

[NC16] Sumit Negi and Santanu Chaudhury. Link prediction in heterogeneous social networks. In *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*, CIKM '16, pages 609–617, New York, NY, USA, 2016. ACM.

[New01] Mark EJ Newman. Clustering and preferential attachment in growing networks. *Physical Review E*, 64(2):025102, 2001.

[New18] MEJ Newman. Network structure from rich but noisy data. *Nature Physics*, 14(6):542, 2018.

[NWF78] George L Nemhauser, Laurence A Wolsey, and Marshall L Fisher. An analysis of approximations for maximizing submodular set functions-i. *Mathematical Programming*, 14(1):265–294, 1978.

[OJT16] Bo Ouyang, Lurong Jiang, and Zhaosheng Teng. A noise-filtering method for link prediction in complex networks. *PLOS ONE*, 11(1):1–12, 01 2016.

[PK16] Pratima and R. Kaushal. Tie strength prediction in OSN. In *2016 3rd International Conference on Computing for Sustainable Global Development (INDIACom)*, pages 841–844, March 2016.

[PLG13] Akshay Patil, Juan Liu, and Jie Gao. Predicting group stability in online social networks. In *Proceedings of the 22nd international conference on World Wide Web*, pages 1021–1030. ACM, 2013.

[Pri65] Derek J De Solla Price. Networks of scientific papers. *Science*, pages 510–515, 1965.

[PRP12] Luca Pappalardo, Giulio Rossetti, and Dino Pedreschi. How well do we know each other? detecting tie strength in multidimensional social networks. In *Advances in Social Networks Analysis and Mining (ASONAM), 2012 IEEE/ACM International Conference on*, pages 1040–1045. IEEE, 2012.

[PU03] Alexandrin Popescul and Lyle H Ungar. Statistical relational learning for link prediction. In *IJCAI workshop on learning statistical models from relational data*, volume 2003. Citeseer, 2003.

[PVG+11] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[Qui86] J. Ross Quinlan. Induction of decision trees. *Machine learning*, 1(1):81–106, 1986.

[Rap53] Anatol Rapoport. Spread of information through a population with socio-structural bias: I. assumption of transitivity. *The bulletin of mathematical biophysics*, 15(4):523–533, Dec 1953.

[Rib14] Bruno Ribeiro. Modeling and predicting the growth and death of membership-based websites. In *Proceedings of the 23rd international conference on World Wide Web*, pages 653–664. ACM, 2014.

[RKKS17] Rahmtin Rotabi, Krishna Kamath, Jon Kleinberg, and Aneesh Sharma. Detecting strong ties using network motifs. In *Proceedings of the 26th International Conference on World Wide Web Companion*, WWW '17 Companion, pages 983–992. International World Wide Web Conferences Steering Committee, 2017.

[RM51] Herbert Robbins and Sutton Monro. A stochastic approximation method. *The annals of mathematical statistics*, pages 400–407, 1951.

[RM03] Ray Reagans and Bill McEvily. Network structure and knowledge transfer: The effects of cohesion and range. *Administrative Science Quarterly*, 48(2):240–267, 2003.

[RSM+02] Erzsébet Ravasz, Anna Lisa Somera, Dale A Mongru, Zoltán N Oltvai, and A-L Barabási. Hierarchical organization of modularity in metabolic networks. *science*, 297(5586):1551–1555, 2002.

[Ruh00] Britta Ruhnau. Eigenvector-centrality—a node-centrality? *Social networks*, 22(4):357–365, 2000.

[SBBV13] Stefan Stieger, Christoph Burger, Manuel Bohn, and Martin Voracek. Who commits virtual identity suicide? differences in privacy concerns, internet addiction, and personality between facebook users and quitters. *Cyberpsychology, Behavior, and Social Networking*, 16(9):629–634, 2013.

[SDC+03] Parongama Sen, Subinay Dasgupta, Arnab Chatterjee, PA Sreeram, G Mukherjee, and SS Manna. Small-world properties of the indian railway network. *Physical Review E*, 67(3):036106, 2003.

[Sei83] Stephen B. Seidman. Network structure and minimum degree. *Social Networks*, 5(3):269 – 287, 1983.

[SGS+16] Andreas Spitz, Anna Gimmler, Thorsten Stoeck, Katharina Anna Zweig, and Emőke-Agnes Horvat. Assessing low-intensity relationships in complex networks. *PLOS ONE*, 11(4):1–17, 04 2016.

[Sha01] Claude Elwood Shannon. A mathematical theory of communication. *ACM SIGMOBILE Mobile Computing and Communications Review*, 5(1):3–55, 2001.

[Shi53] Alfonso Shimbel. Structural parameters of communication networks. *The bulletin of mathematical biophysics*, 15(4):501–507, Dec 1953.

[Sib14] C. Sibona. Unfriending on facebook: Context collapse and unfriending behaviors. In *System Sciences (HICSS), 47th Hawaii International Conference on*, pages 1676–1685, Jan 2014.

[Sim08] Georg Simmel. Sociology: Investigations on the forms of sociation. *Duncker & Humblot, Berlin Germany*, 1908.

[Sin10] Param Vir Singh. The small-world effect: The influence of macro-level properties of developer collaboration networks on open-source project success. *ACM Trans. Softw. Eng. Methodol.*, 20(2):6:1–6:27, September 2010.

[SM86] Gerard Salton and Michael J. McGill. *Introduction to Modern Information Retrieval*. McGraw-Hill, Inc., 1986.

[SN17] Kazunori Seki and Masataka Nakamura. The mechanism of collapse of the friendster network: What can we learn from the core structure of friendster? *Social Network Analysis and Mining*, 7(1):10, 2017.

[SPO14] Tasos Spiliotopoulos, Diogo Pereira, and Ian Oakley. Predicting tie strength with the facebook api. In *Proceedings of the 18th Panhellenic Conference on Informatics*, PCI '14, pages 9:1–9:5, New York, NY, USA, 2014. ACM.

[SR51] Ray Solomonoff and Anatol Rapoport. Connectivity of random nets. *The bulletin of mathematical biophysics*, 13(2):107–117, 1951.

[ST11] Srinivas Gorur Shandilya and Marc Timme. Inferring network topology from complex dynamics. *New Journal of Physics*, 13(1):013004, 2011.

[TCC13] Chengyi Tu, Yuhua Cheng, and Kai Chen. Estimating the varying topology of discrete-time dynamical networks with noise. *Central European Journal of Physics*, 11(8):1045–1055, 2013.

[TCL18] HC Tam, Emily SC Ching, and Pik-Yin Lai. Reconstructing networks from dynamics with correlated noise. *Physica A: Statistical Mechanics and its Applications*, 502:106–122, 2018.

[Thel09] Mike Thelwall. Homophily in myspace. *Journal of the American Society for Information Science and Technology*, 60(2):219–231, 2009.

[TM77] Jeffrey Travers and Stanley Milgram. An experimental study of the small world problem. In *Social Networks*, pages 179–197. Elsevier, 1977.

[TRW09] Mojtaba Torkjazi, Reza Rejaie, and Walter Willinger. Hot today, gone tomorrow: on the migration of myspace users. In *Proceedings of the 2nd ACM workshop on Online social networks*, pages 43–48. ACM, 2009.

[TWAK04] Ben Taskar, Ming-Fai Wong, Pieter Abbeel, and Daphne Koller. Link prediction in relational data. In *Advances in neural information processing systems*, pages 659–666, 2004.

[WD67] Strother H Walker and David B Duncan. Estimation of the probability of an event as a function of several independent variables. *Biometrika*, 54(1-2):167–179, 1967.

[WDSF+13] Shaomei Wu, Atish Das Sarma, Alex Fabrikant, Silvio Lattanzi, and Andrew Tomkins. Arrival and departure dynamics in social networks. In *Proceedings of the Sixth ACM International Conference on Web Search and Data Mining*, WSDM '13, pages 233–242, New York, NY, USA, 2013. ACM.

[WGC16] Y. Wang, Y. Guo, and Y. Chen. Accurate and early prediction of user lifespan in an online video-on-demand system. In *IEEE 13th International Conference on Signal Processing (ICSP)*, pages 969–974, 2016.

[Wie47] Harry Wiener. Structural determination of paraffin boiling points. *Journal of the American Chemical Society*, 69(1):17–20, 1947.

[WLE+16] Xin Wang, Wei Lu, Martin Ester, Can Wang, and Chun Chen. Social recommendation with strong and weak ties. In *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*, CIKM '16, pages 5–14, New York, NY, USA, 2016. ACM.

[WS98] Duncan J Watts and Steven H Strogatz. Collective dynamics of 'small-world' networks. *nature*, 393(6684):440, 1998.

[WS14] Xi Wang and Gita Sukthankar. Link prediction in heterogeneous collaboration networks. In *Social network analysis, community detection and evolution*, pages 165–192. Springer, 2014.

[WSP07] Chao Wang, Venu Satuluri, and Srinivasan Parthasarathy. Local probabilistic models for link prediction. In *Seventh IEEE international conference on data mining (ICDM 2007)*, pages 322–331. IEEE, 2007.

[WXWZ15] Peng Wang, BaoWen Xu, YuRong Wu, and XiaoYu Zhou. Link prediction in social networks: the state-of-the-art. *Science China Information Sciences*, 58(1):1–38, 2015.

[XLZ+12] Wei Xie, Cheng Li, Feida Zhu, Ee-Peng Lim, and Xueqing Gong. When a friend in twitter is a friend in life. In *Proceedings of the 4th Annual ACM Web Science Conference*, pages 344–347. ACM, 2012.

[XNR10] Rongjing Xiang, Jennifer Neville, and Monica Rogati. Modeling relationship strength in online social networks. In *Proceedings of the 19th International Conference on World Wide Web*, WWW '10, pages 981–990, New York, NY, USA, 2010. ACM.

[YCSH12] Yang Yang, Nitesh V Chawla, Yizhou Sun, and Jiawei Han. Link prediction in heterogeneous networks: Influence and time matters. In *Proceedings of The 12th IEEE International Conference on Data Mining, Brussels, Belgium*, 2012.

[YLC15] Yang Yang, Ryan N Lichtenwalter, and Nitesh V Chawla. Evaluating link prediction methods. *Knowledge and Information Systems*, 45(3):751–782, 2015.

[Zac77] Wayne W Zachary. An information flow model for conflict and fission in small groups. *Journal of anthropological research*, 33(4):452–473, 1977.

[ZCH17] Chaoyang Zhang, Yang Chen, and Gang Hu. Inference of targeted interactions of networks with data of driving and driven nodes only by applying fast-varying noise signals. *Physics Letters A*, 381(31):2502–2509, 2017.

[Zea12] Xiaojian Zhao and et. al. Relationship strength estimation for online social networks with the study on facebook. *Neurocomputing*, 95:89–97, 2012.

[Zha04] Harry Zhang. The optimality of naive bayes. *A A*, 1(2):3, 2004.

[ZLZ09] Tao Zhou, Linyuan Lü, and Yi-Cheng Zhang. Predicting missing links via local information. *The European Physical Journal B*, 71(4):623–630, 2009.

[ZLZ+18] Xiu-Xiu Zhan, Chuang Liu, Ge Zhou, Zi-Ke Zhang, Gui-Quan Sun, Jonathan J.H. Zhu, and Zhen Jin. Coupling dynamics of epidemic spreading and information diffusion on complex networks. *Applied Mathematics and Computation*, 332:437 – 448, 2018.

[Zwe14] Katharina Anna Zweig. *Network Analysis Literacy: A Practical Approach to Network Analysis Project Design*. Springer Publishing Company, Incorporated, 2014.

[ZZQ+17] Fan Zhang, Ying Zhang, Lu Qin, Wenjie Zhang, and Xuemin Lin. Finding critical users for social network engagement: The collapsed k-core problem. In *Thirty-First AAAI Conference on Artificial Intelligence*, pages 245–251, 2017.

Curriculum Vitae

EDUCATION

April 2013 – September 2018
Computer Science, Doctoral studies, TU-Kaiserslautern, Germany

October 2010 – January 2013
Computer Science, Master studies, TU-Kaiserslautern, Germany

October 2001 –June 2005
Computer Engineering, Bachelor studies, Islamic university of Gaza, Palestine

EXPERIENCE

October 2018 - today
Artificial Intelligence Systems Creator (Manager), BASF, LUDWIGSHAFEN, GERMANY

April 2013 – September 2018
Research Fellow (Wissenschaftlicher Mitarbeiter), TU-Kaiserslautern, Germany

October 2010 – May 2012
Research Assistant, DFKI and Fraunhofer IESE, Kaiserslautern Germany

February 2006 – October 2010
Software Engineer, Ministry of Interior, Gaza Palestine