Informatics

Effiziente Detektierung einflussreicher Benutzer in Social Recommender Systems

DIPLOMARBEIT

zur Erlangung des akademischen Grades

Diplom-Ingenieur

im Rahmen des Studiums

Business Informatics

eingereicht von

Ing. Paul Erich Stelzhammer, BSc
Matrikelnummer 01426219

an der Fakultät für Informatik

der Technischen Universität Wien

Betreuung: Univ.Prof. Dipl.-Ing. Dr.rer.nat. Radu Grosu
Mitwirkung: Projektass. Hamidreza Mahyar, PhD

Wien, 8. Juli 2020

Paul Erich Stelzhammer

Radu Grosu

Informatics

Efficient Detection of Influential Users in Social Recommender Systems

DIPLOMA THESIS

submitted in partial fulfillment of the requirements for the degree of

Diplom-Ingenieur

in

Business Informatics

by

Ing. Paul Erich Stelzhammer, BSc
Registration Number 01426219

to the Faculty of Informatics

at the TU Wien

Advisor: Univ.Prof. Dipl.-Ing. Dr.rer.nat. Radu Grosu
Assistance: Projektass. Hamidreza Mahyar, PhD

Vienna, 8th July, 2020

Paul Erich Stelzhammer

Radu Grosu

Erklärung zur Verfassung der Arbeit

Ing. Paul Erich Stelzhammer, BSc

Hiermit erkläre ich, dass ich diese Arbeit selbständig verfasst habe, dass ich die verwendeten Quellen und Hilfsmittel vollständig angegeben habe und dass ich die Stellen der Arbeit – einschließlich Tabellen, Karten und Abbildungen –, die anderen Werken oder dem Internet im Wortlaut oder dem Sinn nach entnommen sind, auf jeden Fall unter Angabe der Quelle als Entlehnung kenntlich gemacht habe.

Wien, 8. Juli 2020

Paul Erich Stelzhammer

Danksagung

Ich möchte Prof. Radu Grosu dafür danken, dass ich meine Masterarbeit bei der Cyber-Physical Systems Group der Technischen Universität Wien machen darf. Durch seine fundierten Fachkenntnisse und seiner offenen Haltung konnte ich sehr von seinem Wissen und seiner Erfahrung profitieren. Ich möchte meinem Betreuer Dr. Hamid Mahyar aufrichtig danken. Ohne ihn wäre diese Arbeit nicht möglich gewesen. Er half beim Einstieg in das Thema, unterstützte mich zu jeder Zeit und trieb mich voran. Er unterstützte mich dabei meine Grenzen auszuloten und würdigte mein Engagement mit positiver Kritik. Obwohl er stets einen vollen Terminkalender hat, nahm er sich Zeit für regelmäßige Besprechungstermine, unabhängig von kurzen oder langen geografischen Distanzen oder Zeitverschiebungen. Wir stellten fest, dass durch konsequente Arbeit die Zusammenarbeit sehr gut klappt, obwohl ich manchmal nicht meinen vollen Fokus zeigen konnte. Ich möchte auch Prof. Walter Schwaiger und Prof. Christian Huemer erwähnen, die das Studium der Wirtschaftsinformatik mit ihrem Engagement zu dem, was es heute ist, gemacht haben. Ich hatte die Ehre, an vielen ihrer Vorträge teilzunehmen und lernte die Art und Weise zu schätzen, wie sie sich unermüdlich um das Bildungsangebot im wissenschaftlichen Umfeld kümmern. In diesem Abschnitt meiner akademischen Karriere konnte ich sehr viele Verknüpfungen vergangener Erkenntnisse entwickeln und verbinden, die ohne sie nicht möglich gewesen wären.

In meinem Berufsleben im Bereich der Informatik hat Klemens Schachinger, mein Vorgesetzter und Förderer bei EV Group, meine Zukunft außerordentlich geprägt. Die technische und persönliche Erfahrung, die ich durch dort sammeln konnte, ist immens. Auch Christian Knieling von der Mondi AG half mir, einen erweiterten Horizont zu betrachten, und forderte mich vor allem in Belangen meiner Effektivität.

Weiters bedanke ich mich bei allen, die mich beim Korrekturlesen und bei den Sprachformulierungen unterstützt haben.

Es war eine lange Reise, seit ich 2017 direkt nach Abschluss meines Bachelors mit der Diplomarbeit begonnen habe. Jana unterstützte mich unermüdlich meine Konzentration hoch zu halten und ertrug meine Grantigkeit, die der Fokus mit sich bringt, auch in Zeiten der globalen Pandemie. Sie lässt meine erste Liebe, den Computer, nicht die letzte sein. Wir schaffen uns unsere Perspektive, eine eigene Familie - in einem unterstützenden Umfeld aus Freunden, Fußball und Reisen - als endlose Inspiration - zu gründen. Nicht zuletzt danke ich meiner Familie, die mich bei allem unterstützt, was ich erreichen möchte.

Acknowledgements

I want to thank Prof. Radu Grosu, who allowed me to do my Master thesis with the Cyber-Physical Systems Group at Vienna University of Technology. With his in-depth expertise and open attitude, I could profit very much from his knowledge and experience. I sincerely want to thank my advisor Dr. Hamid Mahyar. Without him, this work would not have been possible. He introduced me to the topic, always supported me and pushed me forward. He was eager that I reach my limits as well as appreciated my commitment with praise. Even though he has a busy schedule, he took time for regular meetings, no matter of short and long geographic distance. It turned out that it was a good match for consistent work even though at some times I could not bring my full focus. Also I would like to mention Prof. Walter Schwaiger and Prof. Christian Huemer, who, with their commitment, made the studies of Business Informatics to what it is today. I had the honor to attend many of their lectures and appreciate the way they take care of the offer of education in the scientific environment. In this chapter of my academic career, I could very much develop and connect links of past experiences that would not have been possible without them.

In my professional life in the field of informatics Klemens Schachinger, my supervisor and supporter at EV Group, exceptionally shaped my future. The technical and life experience I could gain through him is immense. Also, Christian Knieling at Mondi helped me to look at a larger picture and challenged me in my effectiveness in operations.

I also want to thank everyone who supported me in proofreading and language formulations. It was a long journey since I started with the Thesis in 2017 right after I finished my Bachelor. Jana tirelessly supported me in my focus and bore my grumpiness and focus even in times of the global pandemic. She let my first love computer not be the last one and helps me to create our own family in a supportive environment of friends, football and travel as endless inspiration. Last but not least, thanks to my family who supports me in whatever I want to do.

Kurzfassung

Soziale Netzwerke sind allgegenwärtig in unserem täglichen Leben und es ist wichtig die einflussreichsten Benutzer darin erkennen und einordnen zu können. Durchschnitts- oder Klassifizierungsansätze können zur Kategorisierung angewendet werden, aber dies ist möglicherweise nicht ganz richtig. Die bestehenden Indikatoren sind nicht spezifisch genug. Daher möchten wir in dieser wissenschaftlichen Arbeit die Erkennung der einflussreichsten Benutzer in einem Social Recommender-System verbessern, indem wir ein Maß zur Zentralitätsbestimmung für einen bipartiten Graphen entwickeln. Die Definition was ein einflussreicher Knoten ist hat eine entscheidende Bedeutung. Der Fokus liegt auf den strukturellen Attributen der einzelnen Knotenpunkte. Bestehende Messmethoden werden erweitern und kombiniert und wir setzten voraus, dass diese mit Datensätzen aus der Praxis arbeiten können. Damit können Empfehlungen aus der Analyse von Netzwerken aussagekräftiger getroffen werden. Wir definieren effiziente Parameter für die Methode und testen ihre Grenzen indem wir sie mit anderen Ansätzen vergleichen. Wir folgen dem Design Research-Ansatz mit dem ein neues Artefakt erstellt wird. Sehr wichtig ist dabei die Evaluierung, die experimentell durchgeführt wird. Die Metrik wird in Programmiercode implementiert und mit existierenden Netzwerken ausgeführt. Wir vergleichen Datensätze die die Wirklichkeit modellieren mit anderen wissenschaftlichen Praktiken. Einen Einblick was mit der von uns erstellten Grundlage möglich ist geben passende Anwendungen in den Bereichen Knotenreihung und Verbindungserkennung.

Abstract

Social networks surround us in our everyday life and we want to improve finding the most influential users in it. Therefore we could take the average measure or clustering approaches, but this might not be entirely accurate. Given indicators are not specific enough. Therefore in the Thesis, we want to improve to detect the influential users in a Social Recommender System by designing a centrality measure for a bipartite graph. A definition of what are influential nodes is vital. The focus is on the structural attributes of the individual nodes. It supposes to work for real-world data and outperforms other standard measures to make recommendations in the networks more meaningful. We define the properties of the concept so that it is most effective and test the limits by comparing it to other approaches. The Design Research approach is the method to be applied, we are defining a new artifact accordingly. The measure should be implemented with a program code to compute it with real-world networks. Most important is the evaluation, which will be done experimentally with comparing the real data results to other common indices and try to find out boundaries. With our foundation, we perform prominent applications in Node Ranking and Link Prediction, which can provide an outlook for further potential.

Contents

CHAPTER 1

Problem Definition

Since the beginning of the 21st century, the rise of Online Social Networks was tremendous. In every aspect of our life, we can contribute to the connected world and share our perspective. The main driver in this development was the hardware development concerning size and computational power, bringing smart devices to our tools and pockets.

In the Online Social Network, the society is abstracted in a digital model around the properties of the people and their related objects. The interaction possibilities are endless and suggested key figures and recommendations can help to get a feeling about the environment. Information is available, and it is a matter of analysis of the valuable data to which we want to contribute with our work. To know about the network is crucial in case of practical implementation for drawing implications leading to decisions. We want to find out which challenges in the complexity and speed of technological possibilities there are and how this can be approved.

For analysis, we look into tools in network science, which enable us to declare statements of the given connections and interpret the results. For a concrete definition, we look into how a mathematical graph is sufficient to model a lot of real-world scenarios.

After giving the first introduction about the contribution, challenges and implementation further definitions in scientific literature are required.

One topic in media research is the classification of social network sites. Kitezmann et. all in [KHMS11] see sites, blogs, networks and wikis to create, modify, share and discuss content. They distinguish through the level of self-presentation and social presence. Classical social media sites like Facebook show medium social presence and high self-presentation. Virtual game worlds like world of warcraft show a low self-presentation level and a high social presence. This socioecological categorization draws a link to the potential of the content of analysis.

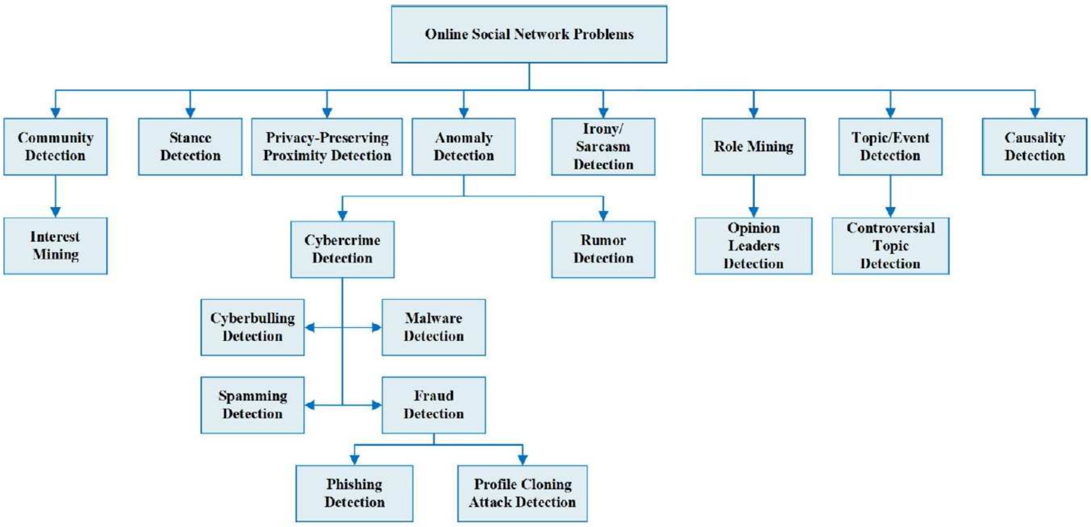


Figure 1.1: Overview of Online Social Network Problems as by Can and Alatas in [CA19]

Due to the significant progress in the last years, internet technologies emerged to a further level and online social networks have become widespread worldwide. The infrastructure and the products evolved such that everyone can be part of it.

Complex networks describe a various number of advanced systems, whether they are social, biological or technical. Understanding the structure may bring various benefits. To detect the most influential parts with centrality measures is very fundamental in network science. It appears that when it comes to the point of detecting various characteristics of users, being something the most (maximization) is very crucial. The focus on the peaks has the drawback of losing the majority of the system. We take a look into a more structural review of the users connections, to detect some meaningful declaration of the users. Also the definition of influential is very broad, some effort is needed to sort different results of representative users.

In the following sections, we discuss the analysis of Online Social Networks and their related problems, which are a big topic in scientific research. Additionally, we specify our problem definition and the area in which we are moving. We further introduce Research Methods in Computer Science, followed by our problem statement and our Design Science approach. We elaborate on the paradigms in the related research fields and further describe the way of implementation and evaluation. Afterwards, we conclude the first chapter with an outline of the further chapters.

1.1 Online Social Network Problems

Can and Alatas in [CA19] created a great overview of social network analysis problems and applications. They survey that amongst others community detection, anomaly detection and role mining are very relevant problems, as we can see in Figure 1.1. Role mining and community detection play a role in various areas other than social networks like web graphics or transportation networks. Classification or abnormality detection can help to detect particular categories and treat them accordingly. The discovery of abnormalities in technological networks like IP-traces can help to detect security issues. Nodes that do not correlate to the standard pattern may be a threat. In social networks, advertising by roles is the main application. The messages to the users can be tailored to the preferences increasing the demand. When we are looking if a user is influential or representative, we find ourselves in the fields of opinion leader detection and interest mining. In opinion leader detection, prominent users that affect other users are influential. In interest mining, the related properties of users are tried to find out.

According to Rossi and Ahmed in [RA14] and shown in Figure 1.2 Role Mining is done in Graph-based, Feature-based or Hybrid approaches. Looking to the graph obviously network structural properties are more in focus. While on feature-based approaches the individual node's attributes are considered. Graph-based methods are calculating directly with the graph representation. Feature-based strategies are working on a transformation based on the designed feature activation. For Graph-based approaches blockmodels and row/column similarity of Adjacency Matrix are methods that can be used. Structural, Automophoric, Regular or Stochastic Equivalence are distinguished.

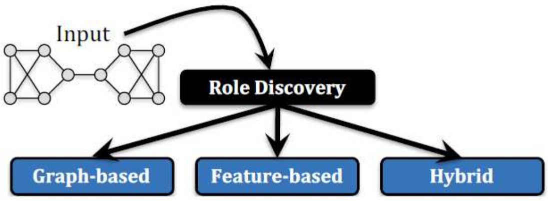


Figure 1.2: Role Discovery Methods Taxonomy as by Rossi and Ahmed in [RA14]

In the field of role mining, opinion leader detection is one crucial application. Opinion leaders are individuals who have a significant impact on others. They can shape others' ideas or are in a position to influence others' behavior. It can play a huge role in advertising or in political spheres. Recent approaches for this are seen from Agarwal et. al. in [ALTP12] and Aghdam and Navimipour in [AN16]. Also Moldovan in [MMRYT17] discuss the application of Opinion leadership in small groups. In this artificial context, it is required to mention that any influence manipulation in any case must follow the basic rules of ethics. The ethical responsibilities are covered in the statement of the Association for Computing Machinery in [Cou17].

A way to analyze a network is in an abstracted graph model. This is used to number the structure and find notable nodes. Prominent nodes are usually highly connected. Important nodes are primarily detected using centrality measures as further discussed in that paper. A node could typically be a user who is connected to others. The degree is the count of connected links of a node to another node. With this, the first basic measure is introduced. Opsahl in [OAS10] surveys that degree centrality, betweenness centrality and closeness centrality are typical measures to identify the importance based on the linkage to other nodes.

Computer support reached a level where systems are able to suggest guidance and offer advice with the data they observed. Other's experiences may benefit to better detect similarities to our own interest and need. The bipartite setup - which is lacking for a lot of existing measures - fits very well to the user item architecture of social recommender. In the application of giving recommendations to a user, a simple graph does not have a complete data structure. With a simple graph, we are just modeling users, which we can relate to other users. We are limited to relate to any other object. To give more meaningful recommendations, we are going to relate a user to an object. The objects are the second type of nodes in a graph. This extension of different kinds of nodes perfectly reflects the relation of two nodesets in a bipartite network. A graph represents a Social Network. A bipartite graph represents a Social Recommender. In a bipartite graph user-item, user-event, user-article, user-movie and many more relations can be modeled very nicely. The meaning of social in this context is always that we always relate to users. The conclusion here is that the bipartite network perfectly fits to the relation that is important for a social recommender system.

A social bipartite network where nodes are distinguished in two sets is capable of abstracting numerous real world scenarios as Guillaume and Lapaty summarize in [GL06]. Additionally finding bipartite properties may allow analysis from a different perspective. For example people buying products are two entities which can be modeled. Most of the standard methodologies are not available for the special case of a bipartite network where we will take a deeper look into.

Following the information of our example of people are buying products and those people are somehow connected, Social Recommender System may benefit the task of product suggestions for potential buyers. Collaborative filtering according to Breese et. al. in [BHK98] is based on information of the user. What will they favor in comparison to other users is suggested by the similarity to other users. As Arggarwal states in [A$^+$16] content Based filtering uses the description of items and users' preferences based on article information. The properties of the objects are in focus. Hybrid models combine the most common aspects of content based and collaborative filtering. With Machine Learning methodologies like Bayes Classifier, Cluster Analysis or decision trees, the most common recommendation can be found.

According to Martinez et. al. in [MBC16] in the field of network mining, a lot of related problems are currently studied. Community detection, structural network analysis or network visualization are a few. One of the most interesting problems is link prediction. New connections or an unknown association are to be detected. Based on the existing links and the attributes of the nodes a prediction should be done. Successful appliances are done in biological networks with protean relations. Also collaboration prediction in co-authorship data of scientific work is one application. Research fields can be discovered and elaborated on how future collaboration is possible. In Social network analysis,

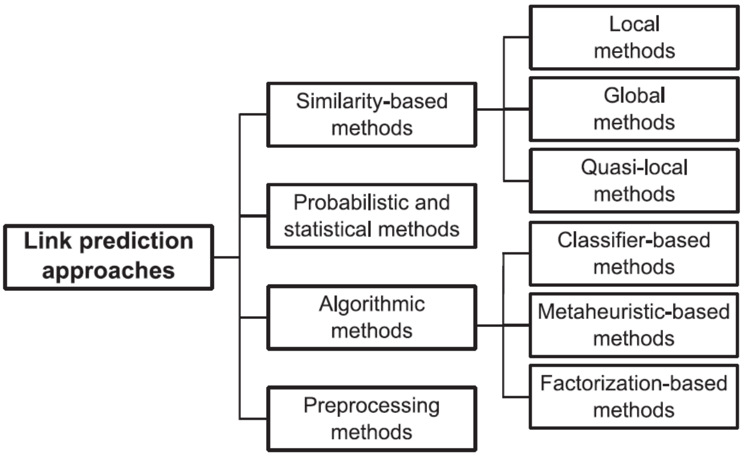


Figure 1.3: Link Prediction Techniques taxonomy as by Martinez et. al. in [MBC16]

the focus on the structure of criminal networks in order to anticipate action is worth mentioning. Referring back to what we saw earlier in Figure 1.1 according to Can and Alatas in [CA19], link prediction can be found in the problem field of community detection and more specifically interest mining. When studying the different approaches and applications, they are most often similarity-based. Those approaches assume that they are similar if they are connected to the same nodes. Further there is distinguished between local and global methods. Link prediction techniques are categorized in the taxonomy of Figure 1.3. Wu et. al. in [WSZ$^+$19] show a very similar overview. Their primary focus is on detecting influential nodes.

But what if one wants to find out the most common nodes rather than the most central ones? The most general node could speak as a characteristic for the whole network. With gaining a behavioral representation, we cover the interests of the critical mass. The analysis can help to identify any significant, representative user kind and in this sense determines the very influential users. Those influential users lay the foundation of the system. Those particular nodes draw connections on how to reach other nodes. Its properties may show on how to get to other nodes which are also essential and strengthen the whole network. It is a kind of averaging like it is done with a list of numbers. Several approaches from sampling, clustering, algorithms to centrality measures need to be reviewed. Often some kind of ranking needs to be interpreted. One of the main challenges is to find some categorization of all the existing methodologies based on the applied data and on the target outcome. Just then we will be able to compare the results.

In interest mining, the user context can also be reviewed in a broader context. Either user can be associated with other users. Also a user of different networks that are not connected can be compared. Or additional user information in a sense of inter-social influence can be considered. Trust between users is a meaningful element. In our definition, we will review the trust of users in regards to influence similarity. Trust information may be additionally available to rating information. The strength of the impact of social neighbors and trust relationships is to be distinguished. With this thought, we can combine information out of to separate networks, the user-object relation is combined with some user-user information.

Having those facts by hand, in a bipartite environment we have the challenge of finding those common connections of nodes. As unipartite methodologies are mostly not available for bipartite graphs, we have the possibility to either rewrite the network or redefine the existing centrality measures. Existing centrality measures can just be used or be adapted to be feasible and meaningful for bipartite networks. Applying those measures will get an entirely different conclusion if the data is separated. If the methodologies are adapted slightly, the data can interpret a whole new dimension of information. Banerjee et. al. in [BJP17] take a look in the properties of a projected bipartite network. In this case, just one nodeset is extracted while the other edges are abstracted. The result is a new transformed graph. The weighting of edges can be adapted according to the abstraction. The bipartite network became a unimode network and common centralities measures can be used.

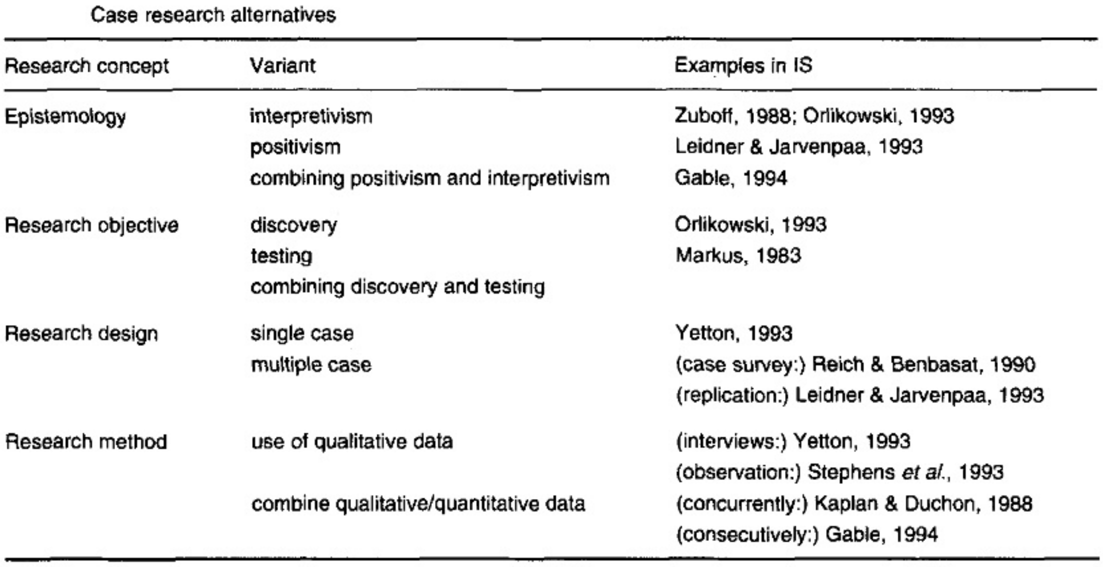
Strogatz in [Str01] is exploring amongst other things confusing effects of projections. We see the disadvantage that the projection of network is available from bipartite to unipartite network but the other way around it is more difficult. As projection always comes with information loss, it is not the preferred option. Also centrality measures most commonly find the most central rather than the most influential nodes. Therefore we see a need for more advanced approach using a structural measure to identify influential nodes in a bipartite graph.

To solve the problem of finding the most important nodes in a bipartite network we introduce JS-BiRank in this work. We calculate the least deviation to other nodes by comparing the distribution of a node's neighbor structure. The information gain by considering our particular common neighbor structure is unique. Nodes, connected as likely to unimportant nodes than to important ones, are dominant. The result is used to rank the nodes from most to least influential.

At this point, we once more expand our view to start with some basics in Research in Computer Science before later focusing more on the details.

1.2 Research Methods in Computer Science

When discussing whether the field of Informatics is Science or not we need to take a look at the terminology. In German Languages there is no literal translation of Computer Science, Informatics is used. Though the English term seems pretty accurate. As a basis structural sciences. The Business in the extended form of Business information brings several implications. The connections brings the classical Business Administration as well as the macroeconomics combined with the technical part. The technical part focuses on software engineering and working with information systems. Taking attributes of both



Case research alternatives

Research concept	Variant	Examples in IS
Epistemology	interpretivism	Zuboff, 1988; Orlikowski, 1993
	positivism	Leidner & Jarvenpaa, 1993
	combining positivism and interpretivism	Gable, 1994
Research objective	discovery	Orlikowski, 1993
	testing	Markus, 1983
	combining discovery and testing	
Research design	single case	Yetton, 1993
	multiple case	(case survey:) Reich & Benbasat, 1990
		(replication:) Leidner & Jarvenpaa, 1993
Research method	use of qualitative data	(interviews:) Yetton, 1993
		(observation:) Stephens *et al.*, 1993
	combine qualitative/quantitative data	(concurrently:) Kaplan & Duchon, 1988
		(consecutively:) Gable, 1994

Figure 1.4: Case Research

worlds additionally the field of modeling is created. It is just a matter of the point of view. As we heard in the Seminar talks in the part of Business Informatics, there are two main approaches in the world of Computer Science. We are comparing behavior and constructive research. Behaviour research tries to describe the existing world. While constructive research tries to build new parts of in the world. Those parts are called artefacts. So taking an behavioral approach could be called science/theory building. An constructive research comes with the direction to engineering which can be a method in science.

One should be careful to use the term Methodology in a correct manner. A method describes the instruction and leads to a result. On the other hand Methodology is the discipline of different research methods. Cavaye in [Cav96] discusses case study research in its possibilities. The paper is very precise about terminology: "Following Galliers (1992) and Weick (1984), this paper distinguishes between the terms 'research approach (or strategy)' and 'research method'. Research strategy is defined as 'a way of going about one's research, embodying a particular style and employing different methods'. Research method is defined as 'a way to systemise observation, describing ways of collecting evidence and indicating the type of tools and techniques to be used during data collection'." Alternatives in case research are listed as in Figure 1.4 making a wide picture of research possibilities. "Constructivist epistemology is an epistemological perspective in philosophy about the nature of scientific knowledge." [1] as in [Rou00].

Constructivists and positivists claim whether reality needs to be constructed or already

[1] https://psychology.wikia.org/wiki/Constructivist_epistemology

exists and need to be described. An objective is very basic and targets the existence and connections. Design a method or use a method are apparent approaches. An instruction of a procedure can solve a problem. Also adjusting existing methods with parameters of the case data can be evaluated. This also holds in a broader picture not limited to cases.

Ramesh et.al. in [RGV04] that in computer science around 80 percent work with a formulative research approach and almost all paper use the method of mathematical conceptual analysis. Mostly computing elements or abstract concepts are developed. Another overview by Baskerville and Wood-Harper in [BWH98] shows characteristics of journals in information systems. Action research evaluates activities after diagnosis in a practical setup.

In the following sections will define the problem statement and discuss Design Science and the Evaluation. Also special characteristics in Social Network Analysis and Network Theory will be taken a look at.

1.3 Thesis Problem Statement

To formulate our problem statement we are following a template for design problems. It starts with the problem context which is treated with a designed artifact. The artifact's requirement are specified and the stakeholder's goals defined. This template helps us to later define the problem statement:

Improve to detect influential users in a Social Recommender System
<*problem context*>

by designing a centrality measure for a bipartite graph
<*treating it with a designed artifact*>

that works for real world data and outperforms other common measures
<*artifact requirement*>

in order to make recommendation in the network more meaningful
<*stakeholder goals*>.

This formulation was one of the sections which needed the most reworks, and about every word we thought about a lot. For example, to justify why to use "data" instead of "network" in the context of real-world is just not to use the word "network" repeatedly.

A different and extended formulations of the problem statement can be found below:
We have a social recommender system and we are interested in finding the most influential (representative) users in the network. We have social data and we want to improve to find the most influential users. We have data (bipartite graph) and want to the most representative node. We could take averaging or clustering measures, but this

might not be perfectly accurate. Given measures are not sophisticated and specific enough.

The layers of abstractions need to be followed in order to be precise of what to tackle. The application layer is more on a Social Network or Social Recommender level where a user is part of the network and which is the data we work with. On the implementation level we need to speak about graphs and the according nodes and the related centrality measures.

The formulation of the research question is as following:

RQ1) What is an appropriate matter to find the most influential users in a social network?

As a sidenote, *matter* is used in the formulation as *means* could be confused with the arithmetic meaning. We mentioned, we are weighing every word in this formulation very much. For example, most "influential" can be discussed in wording versus representative and average. When discussing the topic on an more technical layer, the abstraction of "user in a social network" could be replaced by "node in a bipartite graph".

Several applications like ranking, link prediction and trust based recommendation are framed in the overall topic. Those could be formulated as fractional research questions. RQ1.1) What is an appropriate matter to rank the most influential users in a social network? RQ1.2) What is an appropriate means to predict links between the most representative users in a social network? RQ1.3) What is an appropriate matter to use implicit trust data in a social recommendation system?

Stakeholders of the process are interested in recommendations in a social network. The approach is to find the representative user. Representative nodes are depending on the definition seen as ordinary. Literature shows that there are measures to find important users, but the definition of important is not clear. In order to use automatic recommendation, the result will differ. Our approach to find the average node is to compare the similarity of nodes, and the biggest similarity sum may is our result. Different methods to calculate the similarity of nodes need to be evaluated. The expected result is to be able to look at graphs in a different manner with the knowledge of certain representative nodes and have additional information about the attributes than with standard centrality measures.

It is worth to mention that we are trying to find several users as in plural, even though some formulations can not make it precise and are just speaking of a user or the user. Here we are also in line with the argumentation of finding a majority. The users to find are the most relevant concerning to our investigation. Single users are just taken as examples to elaborate on the principle.
When writing this, the distinction between 'a' and 'the' in the English language was challenging. While 'a' can be any, 'the' is a specific object. The usage in the scientific language can differ or just a German bias lets us use them intuitively differently.

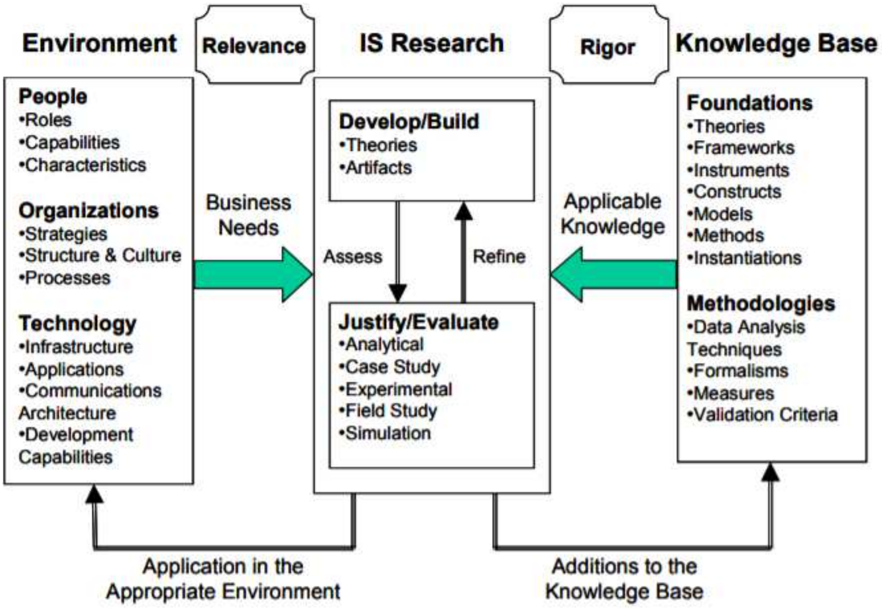


Figure 1.5: Information Systems Research Framework as by Hevner et.al. in [HMPR04]

When using the abstraction of data, it can be referred to in more detail by graph, network or recommender. Data is just the usage for data "set".
Different levels are discussed. Social network, social recommender and user are on the operational level, the social level. Network or graph and user are on the concrete layer are the execution level. A graph is the representation of a network, while a bipartie graph is the representation of a recommender.

As clearly an artifact is designed to improve to detect a certain behavior the problem is treated as a design problem.

1.3.1 Design Science

Referring to the Introduction section, in our case we change things as they are defined. Artifacts are built by taken a look at the reality's problems. Theory designed in behavioural research can be used as a foundation for the construction. The solution leads back to the reality as a different angle of view.

Analyzing the problem with reading appropriate scientific literature is inevitable. When designing the artefact analytical reasoning needs to be done. To deduce the foundation logics and math are used in a formal reasoning. Also possible boundaries may be evaluated.

Figure 1.5 shows the Information Systems Research Framework taken from Hevner et.al. in [HMPR04]. The relevance of research must be given and the scientific rigor is guaranteed by the foundation and methodologies. „Evaluation is what puts the ‚Science‘ in ‚Design Science‘“, as in Hevner.

In many occurrences we talk about Information Systems in the field of Computer Science. In this work we would see it as the informational part as the dataset used and the information which can be gained out of it.

The interesting part is the evaluation using analytical, simulation, instantiation and experimentation. A form of instantiation is implementing the idea in form of a program to compute and execute it. In our case the measure should be implemented with a programming language to compute it with real datasets. The datesets are publicly available. Datasets need to be selected based on their attributes, to have a wide range of different properties. The Koblenz network connection as described by Kunegis in [Kun13] holds currently more than 260 networks and some of them show bipartite properties.

1.3.2 Evaluation

Evaluation is then done experimental, because the mathematical challenge of a formal proof is too big in our case. It may include verification, validation and demonstration of the artifact. While verification checks if the defined requirements are met, validation is to proof that the expected result is hold. Demonstration is closely related to implementing and showing a result.

In our case experimental evaluation is foreseen. To compare the results in different datasets with other common measures should lead to improvement for certain properties. The boundaries of computational costs as well as parametric plausibility need to be tested.

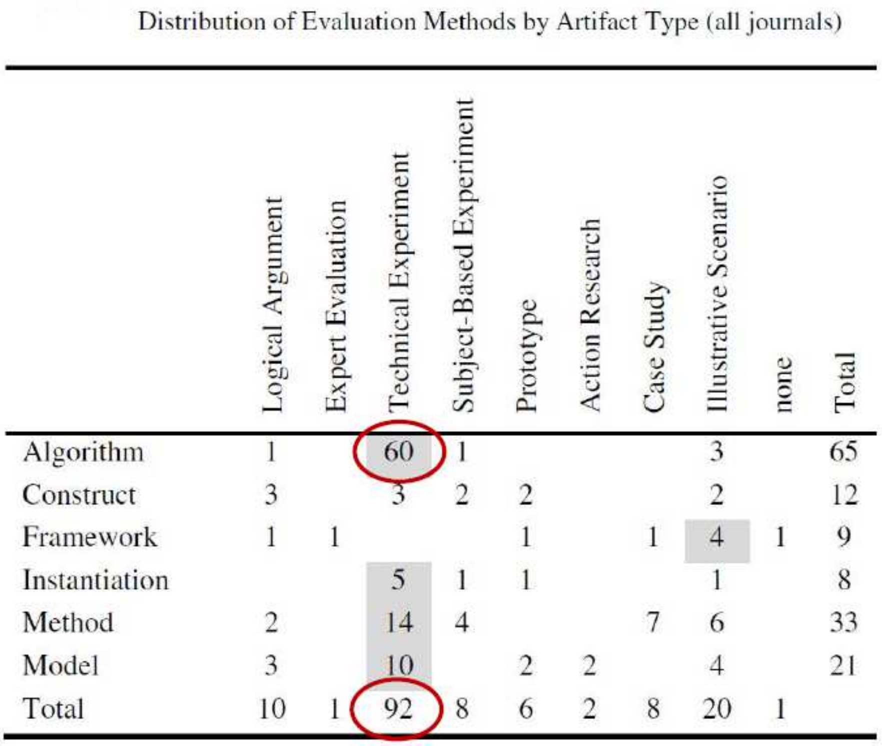
Kaplan and Maxwell in [KM05] argue in the perspective of their role of theory that an Evaluator must lie the foundation of a work always on theory. The theoretical construct can lead the foundation of gaining data but should not constrain or limit the question to answer. Pfeffers et.al. in [PRTV12] surveys evaluation methods used by scientific articles in the field of Information Systems, Computer Science and Computer Engineering. Focusing on Design Science approaches Figure 1.6 shows that most commonly technical experiments are used as evaluation technique.

1.4 Social Network Analysis

In the field of Social Network Analysis also some specialities of the field appear.

The network term as a structural mathematical graph is seen strictly formal. This contradicts with the social part of a Social Network. Building any social connection of a community in a graph is a form of abstraction. In the thesis the definition of influential, representative or average node is crucial.

According to Flick in [Fli10] in research we distinguish in qualitative and quantitative methods. Quantitative work with common statistics and know data. More interesting is



Distribution of Evaluation Methods by Artifact Type (all journals)

	Logical Argument	Expert Evaluation	Technical Experiment	Subject-Based Experiment	Prototype	Action Research	Case Study	Illustrative Scenario	none	Total
Algorithm	1		60	1				3		65
Construct	3		3	2	2			2		12
Framework	1	1			1		1	4	1	9
Instantiation			5	1	1			1		8
Method	2		14	4			7	6		33
Model	3		10		2	2		4		21
Total	10	1	92	8	6	2	8	20	1	

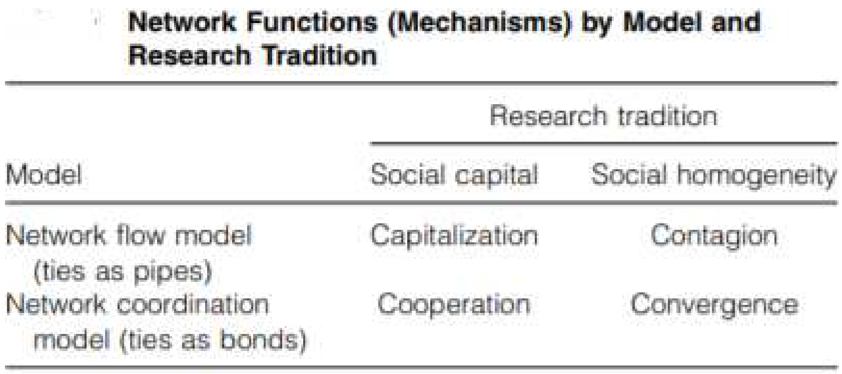
Figure 1.6: Evaluation Methods

qualitative work. Here the data is not standardised and the measures not defined. It brings the flexibility to define new aspects. As we will use given datasets, it is a form of quantitative.

Jannach et.al. in [JZGG12] give an overview about research with recommender systems. Collaborative filtering and movie and video recommendation was most common when publishing the work in 2012. Techniques used are mostly presented in a computer science setup rather than in information systems paper. Offline experiments and user studies are common while just a few publications worked with a formal proof or a case study.

1.4.1 Network Theory

Borgatti and Halgin in [BH11] discuss and characterize network theory. They argue that two models, the flow model and the bond model, are commonly used. In the flow model the distance to other nodes is relevant. In the bond model the connection to other nodes is relevant. Both model have different applications. In Figure 1.7 different conclusions are listed. For contagion functions characteristics are taken by other nodes. At convergence mechanisms two independent nodes have the same attributes and show same the characteristics. In capitalization position brings abilities while for cooperation mechs nodes group together. Those are applications of network information.



Network Functions (Mechanisms) by Model and Research Tradition

	Research tradition	
Model	Social capital	Social homogeneity
Network flow model (ties as pipes)	Capitalization	Contagion
Network coordination model (ties as bonds)	Cooperation	Convergence

Figure 1.7: Network Functions

Omta et.al. in [OTB01] sees research in Network Science beneficial for Simulations and Case Studies in various fields like business economics and organizational theory.

1.5 Outline

Can and Alatas in [CA19] very nice summarize Online Social Network Problems. From community/event/topic detection over Anomaly and Role mining to Causality reasoning, they give hints on how they could be implementing. Network Theory often lies the foundation.

We discussed some terminologies in Computer Science to relate to the common methodologies. With defining the thesis' problem statement we could relate to Design Science, it's attributes and focus on evaluation. Hevner's work brings a lot of input to dig into more in detail. For uses of research method surveys more recent data could be searched. The specialities in Social Network Analysis and Network Theory were just a sneak preview of what is there.

In the following chapters, we first take a look at existing approaches and discover related work with views on the topic from different perspectives. With this knowledge, we introduce our methodology to find out the most influential node. We extensively evaluate the measure and apply it to different datasets.

CHAPTER 2

State of the Art

In this chapter we will report how we proceeded in the literature review. Then some introduction to behavior representative users and their potential in bipartite networks is given. The foundation of influence detection will be discussed by showing which measures are already available and how recommendations are done nowadays.

A systematic literature review is defined by Kitchenham and Charters in [KC07]. Prof. Stefan Schulte presented systematic literature review in our Research methods seminar as seen in Figure 2.1. Using a systematic way helps to find a structured way of having a fair picture of the research field. Another main focus is to inspect the methodologies used in order to be reliable and transparent. The goal is to discover the gaps in the current research in which to contribute with your work. To reduce the number of available papers by each step helps to just go into details for some preselected, promising content.

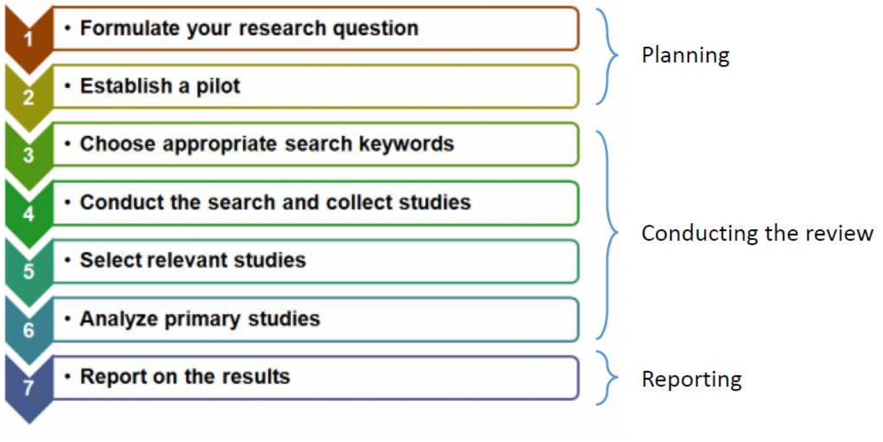

1
• Formulate your research question
2
• Establish a pilot
Planning
3
• Choose appropriate search keywords
4
• Conduct the search and collect studies
5
• Select relevant studies
6
• Analyze primary studies
Conducting the review
7
• Report on the results
Reporting

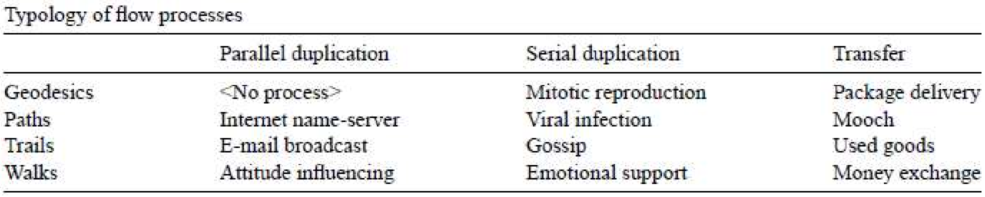
Figure 2.1: Systematic Literature Review as by Kitchenham and Charters in [KC07]

First we describe how we approached and what was discovered in our literature review. This was adapted of the systematic literature review process. In a stage were we already had a basic idea and a first model prototype, keywords for further research where selected. The main keywords were "influential", "user", "bipartite" in several combinations as well as "social network analysis", "recommender application" and "link prediction". When scrolling through the work we found, some preselection was done. Often literature is too specific for a special case or uses different methodologies. A lot of algorithm-based models or hybrid implementation needed to be distinguished from our graph-based centrality approach. Also looking how often the defined keywords in the end appear in the document can hint about the relevance. This was done when looking for applications with the words "social", "Centrality", "Algorithm", "Spreading", "SIR", "shell" and "Kendall". Out of this we discovered the area of Link Prediction to be relevant for our work. Spreading and time-based models could be concluded to be not that relevant. K-shell appeared to be prominently mentioned and some algorithm-based methods could not be directly related to our work.

Accessing the literature is an issue, though most were available through the TU Wien account. Springer Link and IEEE Explore were the leading platforms found through Google Scholar.

Landauer in [Lan88] discussed research methods in human-computer interaction when both fields were just about to emerge to be combined on a large scale in the late 80s. Humans can be seen from a socioecological or psychological perspective. Their behavior vastly varies and they have fallible intuition. Computers are mainly based on hardware, software and the relating required input. Some need for operation is a necessity to create value for the human, otherwise one would simply not operate. Several methods like the invention of systems or features, specifications of it and design orientation are targeted and discussed. This strategy is similar to the Design Science approach of Hevner as discussed in Chapter 1. Further Landauer suggests that statistical analysis may be performed in human-computer interaction. The statistically average performance may be taken for the behavioral representation. Robustness is essential to tackle variation and to settle outliners. In our context it is more about the average human in an environment of a social abstraction model operated in a computational setting, rather than a human directly giving input to a system. This indirectness emerged through the modernity and abstraction of the real world in the virtual world.

In literature, behavioral representative or average user are often mentioned. Social influence analysis and the problem of important users in social networks is viewed for two cases. Behavioral representative users detection methods are one topic. Identifying influential users is the other. While representative focuses on the average users, influential users have a maximization of influence. The different categories of methods are centrality metrics, sampling and clustering. Various measures of influence like similarity, diversity and a learning model are compared to centrality which is the main focus. Common data which is used is users' reply relationship, conversation content and response immediacy, joint influence powers and user trusted networks.



Typology of flow processes

	Parallel duplication	Serial duplication	Transfer
Geodesics	<No process>	Mitotic reproduction	Package delivery
Paths	Internet name-server	Viral infection	Mooch
Trails	E-mail broadcast	Gossip	Used goods
Walks	Attitude influencing	Emotional support	Money exchange

Figure 2.2: Typology of Flow Processes

When looking for average nodes in a graph we found no promising association. It seems that the term "average" is generally not related directly to "node".

Terzian et.al. in [Ter17] evaluates arithmetic mean definitions. He is nicely describing harmonic averages were big differences result in a lower number. The opposite are contraharmonic means were big differences result in a higher number. The example he brings refers to a "social gap", which is a nice relation to our work. He further introduces a social distance metric (no centrality) in which the neighbors' position is crucial. In our case of centrality relation of mean is when looking for a score. The median is related when getting a ranking. Tightly above average - above depends on the perspective- nodes might be more interesting than a bit below average. Because if below there is no high value with any measure. This is also the argument by Sikic et. al. in [ŠLAFŠ13], that rules of spreading before the outbreak of an epidemic are not known and therefore not just the central nodes instead also the average to central nodes are very important to display. Bridging nodes, on the other hand, might not be that important. The conclusion is that the underestimation of non-hub nodes has valid drawbacks. In spreading examples, the average is discussed in a different context than influential. Those models are time related and the average means how much time it takes in usually that half are affected, in other words till a rumor or virus is spread.

In "Centrality and network flow" [Bor05], Stephen P.Borgatti created one another great fundamental work of him. Borgatti distinguishes typologies of flow processes: used goods, money, gossip, e-mail, attitudes, infection, packages and their ways of effecting each other through geodesic, ways, paths or trails as in Figure 2.2. With simulations for the different applications it is checked whether the common centrality measures lead to the expected result. For betweenness it was checked how the traffic is increased when some mooching effect is simulated. This means that not the shortest path is taken anymore, here side effects are visible of nodes which are often used as bridges but not most likely the shortest path. For the money exchange process simulation shows that even nodes with very low centrality values are active and participating the network. This paper is facing the criticism that networks change over time and centrality theory does not consider the facts in a very abstracted picture. It is important to think about the usage when selecting a measure. The paper makes the point to view centrality as an expected value rather than node participation for the specific case.

Strolling around common search engines and encyclopedias we made observations about search counts and definitions. To get an idea about the relevance of wording "influential" versus "representative" we stress Google Scholar results since 2016 (executed February 2020). We find 43 700 results for "influential node" and 107 000 results for "representative node". This shows that both terms are fundamentally used.

Following definition of node influence metrics are very nice. "Node influence metrics are measures that rank or quantify the influence of every node (also called vertex or seed) within a graph."[1] ! We suspect a research group created the Wikipedia entry by themselves in order to cite their work and make it publicly available to just in the scientific community. The definition of "influential" is very nice and directing also to our understanding. The line of argumentation is very similar to ours. We can conclude that influential must not mean most informative. The average is closer to the vast majority and further we discover the cited work we go into as below.

Borgatti and Everett in [BE06] discuss that the accuracy of centrality is highly dependent on the network topology. Additionally they discuss that cohesion and application have a crucial role in choosing a metric. Very dense networks will appear more systematic which makes networks with low coherence harder to analyze.

Most approaches are based on common centrality measures and follow the same thoughts of connection, distance or bridging. This is also confirmed by Borgatti and Everett in [BE06]. Anyhow to get a whole picture, all information needs to be combined. They name it Total Involvement = Radiality + Mediality . Radiality means central, mediality mediate or connect via a bridge. Accessibility and expected force dependent on probability and spreading. The approach looks like closeness centrality. It is introduced by Lawyer in [Law14], Travencolo and da Costa in [TC08], Viana and da Costa in [VBC12]. They also show nice references but no hint for implementation.
In the approach of Mao et.al. in [MX18] Degree, Closeness and Betweenness Centrality are combined to one algorithm to find out influential nodes. Their definition of influential says that rumors or viruses are spread from one most influential node. Their simple evaluation just considers average precision (AP) and Kendall rank correlation coefficient (KRCC) compared with the initial three centrality measures.
On the other hand Zhao et.al. in [ZHT$^+$15] try to find a graph coloring solution for multiple spreaders. In a clustered nodeset the highest centrality is taken to get multiple independent spreaders.
Wei in [WPH$^+$18] considers that influential users are part in more communities. The bridging nodes are most relevant as they are part of more communities. Scatterplots to see the correlation of measures and Kendal Tau Rank correlation are used for evaluation.

When looking in Google Scholar for ”social distance metric” interesting concepts are discovered. Tang et. al in [TMML09] develop a metric to measure the change of a graph over time. The temporal distance uses the shortest path in all combined ways over time. Crucial to understand is, that the connection over time needs the correct order to

[1] https://en.wikipedia.org/wiki/Node_influence_metric

get from start to end. It is used in social network analysis for measuring information spreading. Parkinson et.al. in [PLW14] discuss a neurosciencial approach on what spatial and temporal distance mean in our brain for a social relationship. Interesting is that this is approached with machine learning for brain pattern analyzing and further clustering to gain an interpretation. Wu et.al in [WHZH11] use similarity to be able to recommend. Similarity is used to relate to an input. The input of the user is used for the recommendation of a same (similar) user. To use distance metrics learning to choose the distance metrics brings additional complexity. The application is on social images. Watts et.al. in [WDN02] discusses why social networks are searchable. When searching we want to get a recommendation for the best answer.

We conclude what we found out in the literature review so far. Behavioral representatives are prominently discussed. Averaging behavior is a crucial measure and measures may vary a lot depending on the network characteristics. Even not that densely connected nodes can have significant effects on the systems, and may represent a majority of nodes. There are different types of flow processes modeled by a graph. Measures are often a combination of different methods. Specifically social distance metrics often compare similarity of users.

Now it is clear what our requirements for influential users need to look like. In the following sections, we further work on the definition of *influential*. We discuss how to detect those users in order to determine them. To model our specifications, we require to look into network science and what centrality measures exist. Our relations to similarity concludes the chapter.

2.1 Finding Influential Users

We now want to find a definition for the kind of user we aim to identify. First we want to give an intuition by natural language without further scientific implications. The wording we use in our title is "detecting influential user". Influencing someone else means that the behavior of another individual changes based on the presence of one other. Influential is always in the meaning of importance. Being most prominent implies a kind of maximization of attention. But what tells us that other nodes are also substantial? Even the majority of nodes are not close to powerful but may use their quantitative power to affect others. The strength or presence of typical user count as a symbol to exemplify the larger group, community or network. It expresses the attributes of the majority of other nodes. And summing up the similar characteristics of this majority, the sum is more significant than the sum of the different features of the most famous nodes. The behavioral representative tells that one can be taken for a whole group or a community.

With the simple example of a list of numbers, we want to discuss the average and the implications to the social network case. The average value, no matter if it is the arithmetic mean or the median, is a representing value of the whole list. With just seeing the average value and not having any further information, an essential statement can still be made.

It is information about the center of the list. It is a standard value, the - with the highest probability - expected value. Similar in those key indicators are boundaries like minimum and maximum framing the range. Also outliners are prominently discussed. They give a lot of information, but are not our first priority. In bipartite social networks, some possible key indicators are calculated with centralities. From a structural perspective, we use the connections and interactions for similarity comparison of nodes. In a network the average node in a network is harder to find as the system is more complex than a list. Nevertheless parallelity can be drawn. The average is an intuition for what we are aiming to find the most representative nodes, or in other words, the most influential. Having this simple introduction in mind, we further elaborate on the definitions with the help of the state of art literature in this chapter.

Below there are some phrases we found in abstracts in which fields influential nodes are important and where it is esssential to know about them: fast information, news spreading; distribution of ideas, opinion, thoughts, experiences; understand hierarchical structure; poll analysis; provide connection between people; influence; control spread of information (accelerate); viral marketing by sending messages to this set and reach maximum attention; safe operation of network; optimizing network structure; disease detection and virus control (hinder)

Further approaches in literature can give additional input on how the detection can be performed. Crandall et. al. in [CCH$^+$08] note the connection between similarity and centrality. There is an interplay between similarity and connection because people are similar in their behaviors to there neighbors. As an example, Wikipedia is used. Silva et.al. in [SGMJZ13] develop ProfileRank which finds relevant content and influential users based on information diffusion. Leskovec and Faloutsos in [LF06] derive a representative graph from specific data. This is also called sampling. When choosing data sampling might help to scale down or convert data. Zhu et. al. in [ZGVGA07] introduce a ranking algorithm. Ranking of the result is one method of quantifying a relation to other nodes just by this order. Another aspect which is raised at that point is to give diversity a focus. Characterizing other in other words clustering and identifying with an algorithm is done by Maia et. al. in [MAA08]. Also this approach is algorithm-based. Clustering is a methodology to group data. The reference data in this example is user behavior in YouTube. Users are grouped and have attributes to be featured with. Depending on the available information on a topic the result may come out differently. The sampling-based algorithm of Papagelis et al. in [PDK13] to efficiently explore a user's ego network and to quickly approximate quantities of interest. The ego perspective is one certain interesting aspect, as information can look different from another angle. In a social structure building Recommender Systems my influence the user and their behavior. This can even build a bubble.

Maximization is key when it comes to the most influence of a user. Li et. al. in [LZLC14] uses a finite difference equation, which is an iterative method with incrementing steps, to establish a certain rank. A random Walk Process is used to resolve the score of every node and indicates influence. This LeaderRank performs better than PageRank in several disciplines. Users' reply relationship, conversation content and response immediacy is considered by Tang and Yang in [TY12].

According to Sumith et.al. in [SAB18] influence maximization is a thriving topic for example concerning information diffusion. They survey different approaches. They define influence maximization in a graph by a set of users, for which the information spread is getting maximized when going through. Most prominently used are independent cascade models (ICM). ICM main idea is to measure the probability of a node u influencing another node v. They state for diffusion models, that bipartite influence model may fulfill directed and weighted properties but no time aspect or multiple activation properties. Another common information propagation model is SIR. It considers time or at least iteration for spreading. Many references in literature work consider this SIR model. In any case influence maximization is a NP hard problem, as not all possibilities can be calculated. Another very interesting survey of different methods by Namtirtha et.al. in [NDD20]. Yet another survey we found of Bamakan et. al. in [BNQ19].

2.2 Determining Behavioural Representative Nodes

Now we move a step further in how to detect behavioral representative users, how it can be implemented with the help of network science and executed.

2.2.1 Focus on Bipartite Networks

Liebig and Rao in [LR14] are identifying influential nodes in bipartite networks using the clustering coefficient. A certain pattern of nodes and connections are defined and an inference to this pattern is made. There are certain cycle finding algorithms (also in Python library NetworkX which we will use later) to implement the coefficients. Rather than transforming a one-mode network into a bipartite network, using four-way and six-way cluster is considered.

2.3 Bipartite Networks

Borgatti and Everett in [BE97] discuss network analysis of two-mode data. A bipartite graph $G = \{V_1, V_2, E\}$ has two different node sets V_1 and V_2 and link set E. The nodes are distinguished by two node sets and nodes are just linked from one set to the other. The graph in Figure 2.3 has node sets $V_1 = \{A, B, C, D\}$ and $V_2 = \{1, 2, 3, 4, 5, 6, 7\}$ and is used for basic examples.

2.4 Centrality Measures

Lapaty in [LMDV08] defines neighborhood as following. In a classical graph a neighbor of a node v is denoted as $N(v) = u \in V_1|V_2, (u, v) \in E$. The neighborhood is defined by the elements of $N(v)$. The degree of v is the number of nodes in $N(v)$ as $d(v) = |N(v)|$.

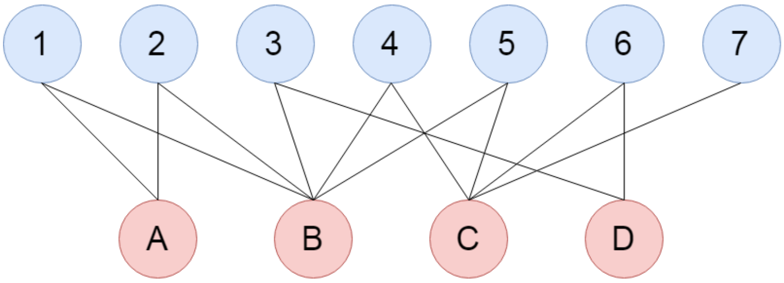


Figure 2.3: A simple bipartite graph with node sets $V_1 = \{A, B, C, D\}$ and $V_2 = \{1, 2, 3, 4, 5, 6, 7\}$

In our simple graph of Figure 2.3 A has degree 2, B has degree 4, C has degree 4 and D has degree 2.

Common measures in a graph are size, number of links and average degree. Also density, the number of existing links divided by the number of possible links, describes properties of a graph.

2.4.1 Degree Centrality

As in Borgatti and Everett in [BE97] degree centrality d^* of a node i is defined as the degree of the node d divided by the number of nodes of the according set n:

$$d_i^* = \frac{d_i}{n_2}, d_j^* = \frac{d_j}{n_1}; i \in V_1, j \in V_2$$

In our simple graph of Figure 2.3 A has degree centrality 0.29, B has 0.57, C has 0.57 and D has 0.29.

2.4.2 Closeness Centrality

As in Borgatti and Everett in [BE97] closeness centrality is defined. The intuition behind is to find how many steps it takes to get to all other nodes which are summed up. The steps measure the distance. The number of edges of the shortest path connecting two nodes is called the distance. Bouttier et. al in [BDFG03] calls this the geodesic distance. Additionally for a bipartite network normalization by dividing by c is required. c is the sum of shortest connections required to get to any other node:

$$c_i^* = \frac{n_2 + 2(n_1 - 1)}{c_i}, i \in V_1$$
$$c_j^* = \frac{n_1 + 2(n_2 - 1)}{c_j}, j \in V_2$$

In our simple graph of Figure 2.3 A has closeness centrality 0.42, B has 0.76, C has 0.62 and D has 0.52. By this measure B is the most central node as it is very densely connected.

2.4.3 Betweenness Centrality

First defined by Freeman in [Fre77] betweenness centrality of node v is defined as:

$$c_b(v) = \sum_{v \neq s \neq t} \frac{\sigma_{st}(v)}{\sigma_{st}}$$

Where $\sigma_{st}(v)$ is the number of shortest connection of two nodes s and t via v. σ_{st} is the number of shortest connection of two nodes s and t. The intuition is how many times a node acts as a bridge in a shortest paths of a connection of two other nodes. For the bipartite network normalization by the maximum possible value considering the relative size of the two node sets is done according to Borgatti and Haglin in [BH11] as below.

$$b_{max}(V_1) = \frac{1}{2}[n_2^2(s+1)^2 + n_2(s+1)(2t-s-1) - t(2s-t+3)]$$

where $s = (n_1 - 1) \div n_2$ and $t = (n_1 - 1) \mod n_2$. For $b_{max}(V_2)$ take $n_2 = n_1, s = p, t = r$. In our simple graph of Figure 2.3 A has betweenness centrality 0.01, B has 0.63, C has 0.36 and D has 0.07. Again also by this measure d is detected as the most central node. This is not very meaningful as this is just caused by the plenty of connected edges.

The primary authors building the foundation of a lot of research are worth to mention. From the University of Kentucky, Stephen P. Borgatti, in collaboration with Martin G. Everett from the University of Manchester, built the foundation for bipartite centrality measures. Also, Linton C. Freeman from the University of California remarkably contributed by the fundamental work introducing betweenness centrality. They led the path to further modern approaches, and the core of this work is based on their early ideas.

2.4.4 Advanced Centrality Measures

Based on the basic measures previously introduced, more complex metrics will now follow.

Page et.al in [PBMW99] with PageRank citation ranking founded a new era by revolutionizing digital search engines and leading the digital revolution to transform societies, as any available information can be found just on demand on any time. Also the reliability reached a new stage. Page Rank is based on Eigenvector Centrality. Also Katz Centrality works similar.

Taheri et.al. in [TMF$^+$17a] establish a HellRank method. Where the number probability distribution of the number a node's neighbor is taken to be compared by the Hellinger distance. The node with the least divergence to all other node is ranked as very representative.

Carmi et. al. in [CHK$^+$07] introduce the method of k-shell decomposition. First step is to delete all nodes in a graph with degree one. These nodes $k_s = 1$. Then this is repeated with degree two nodes and $k_s = 2$ is assigned. This is repeated n times till no nodes remain in the graph. According to Sheikhahmadi et.al. in [SNZ17] in Figure 2.4 we the different kshell decomposition layers. Layers are named as in the algorithm

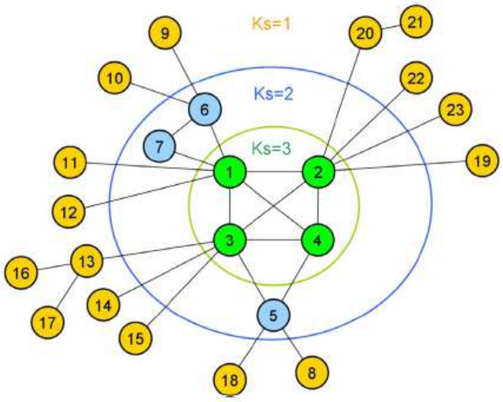


Figure 2.4: The k-shell layers according to Sheikhahmadi et.al. in [SNZ17]

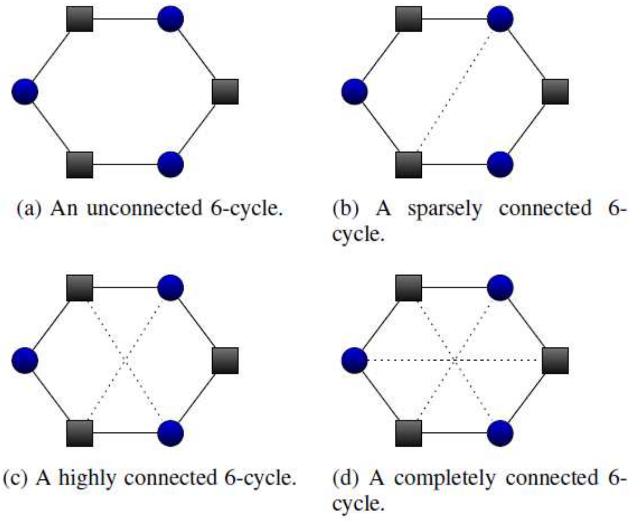
name "shell". The method seems very similar than an inverse of the degree. But taking a deeper thought we see that everytime we delete a node, another nodes degree is also deleted and may be also deleted with the same iteration (see node 13,16 and 17) . With the next iteration the weakly connected nodes are not even considered anymore and so the stronger bonded nodes get more weight (see node 6 and 7). Thinking about it, there a challenge. We get a lot of same values in dense network! This is clearly visible already with simple graphs. Most of the times the outer shell is removed by degree one. Continuing soon most pattern get drained. High degree distribution standard deviation is required to get nicely spreaded results. In our simple graph just node 7 has $k_s = 1$, all other nodes have $k_s = 2$.

2.4.5 Cluster-based Measures

Looking into cluster-based measures forms the idea to look at specific patterns. No matter if they are called cycle, butterfly or common neighbor, implications are drawn to the surrounding and even the whole graph when finding those kinds of objects in at a particular count. It always comes with a form of grouping. Additionally bipartite graphs just allow particular patterns and allow to mark the nodesets as an attribute of the pattern. Clustering also helps for the segregation of meaningful node connections of different communities.

Liebig and Rao in [LR14] follow the idea of detecting a 4-path structure as in Figure 2.5. A 4-path in a bipartite network includes 4 nodes and 4 connections. Finding out cycles and different structures compared to each other lead to clustering coefficients of a node. This idea is based on Opsahl et.al. in [Ops13]. Square clustering is defined by Lind et.al. in [LGH05].

Similar to 4-path motifs, Sanei-Maheri et.al. define butterfly finding algorithms in [SMST18]. Butterflies are the smallest unit of cohesion in a bipartite network. Sariyuce and Pinar use this structure further in [SP18] for dense subgraph discovery within a


(a) An unconnected 6-cycle.

(b) A sparsely connected 6-cycle.

(c) A highly connected 6-cycle.

(d) A completely connected 6-cycle.

Figure 2.5: All possible structures that a bipartite 6-cycle may have according to Liebig and Rao in [LR14]

bipartite network. Daminelli et. al. in [DTDC15] adapt a common neighbor index to bipartite networks to get a conclusion of the likelihood of the interaction of two nodes.

By Singh et.al. in [SSKB19] is looking for pattern in a Twitter Follower Network considering different type of interactions: Retweet, Reply and Mentioning. All is represented in a adjacency matrix. When comparing pattern recognition with clustering the easiest difference in prerequisites is that pattern itself need to be predefined. On the other hand clustering methods needs no configuration on what to find.

Lapaty Clustering Coefficient

Lapaty in [LMDV08] introduces the idea of common neighbors with other nodes of the same node set. Therefore the neighbor of neighbor N(N(u)) is crucial. These are also called second-order neighbors and u is not excluded. It is processed based on clustering coefficients in several variations. The baseline equation is defined as

$$c_u = \frac{\sum_{v \in N(N(v))} c_{uv}}{|N(N(u))|}.$$

Defining in a measure of how many of the neighbors are connected to another node. This is divided by a definition of a bigger neighbor structure depending on the mode. Resulting in a coefficient between 0 and 1. For pairwise calculation for node u and v following modes are defined:

dot:

$$c_{uv} = \frac{|N(u) \cap N(v)|}{|N(u) \cup N(v)|}$$

min:

$$c_{uv} = \frac{|N(u) \cap N(v)|}{min(|N(u)|, |N(v)|)}$$

max:

$$c_{uv} = \frac{|N(u) \cap N(v)|}{max(|N(u)|, |N(v)|)}$$

So whether the common neighbor count is divided by the sum neighbor each node has or divided by the minimum or maximum count one node has. In our simple graph of Figure 2.3 A has clustering coefficient mode dot 0.4, B has 0.28, C has 0.24 and D has 0.18. A seems to have most neighbors with its second level neighbor in common. A has two out of five neighbors in common with its only neighbor of neighbor B. But these values do not give a hint about the connection to C and D. Therefore we try to find a better way of comparing nodes in the form of distribution.

2.5 Similarity

Getting more into the network, not looking at all the nodes the comparison measure to associate two nodes is similarity. The opposite or in other words the inverse is distance. Distance measures dissimilarity. The more two objects are away from each other, the more distance they have. The more distance they have the more dissimilar they are. The more distance they have the less similar they are. Therefore the distance and similarity proportion is inverse.

2.5.1 Similarity Scores

In this section similarity scores are listed. When looking at them we are expecting to compare similarity in order to get an idea of the difference between two nodes.

The number of common neighbors of sets $N(u)$ and $N(v)$ is defined as

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

For two neighbor sets $N(u)$ and $N(v)$ the Jaccard coefficient is defined as

$$JC(u, v) = \frac{|N(u) \cap N(v)|}{|(N(u) \cup N(v)|}.$$

The intuition is, if there are a lot of common neighbors, than there is a high Jaccard coefficent. It is a coefficent because the number is always between 0 and 1.

A high score means high similarity of nodes can lead to a link prediction. A link prediction just needs to be considered if the nodes are not already connected.

Node centrality can be reached by sum of coefficent to all other nodes divided by the count of nodes.

The consequences for the bipartite case are 1) nodes of different sets will never have common neighbors, 2) same set cannot have a connection already, this makes mapping possible.

Those implications can not be generalized for all similarity measures. It is just relevant if they are neighborhood based like the Jaccard Coefficent. Neighborhood based are local measures and often seen as a "triangle closer".

Other similarity scores which are frequently used are Adamic-Adar index and Preferential Attachment score.

For two neighbor sets $N(u)$ and $N(v)$ the Adamic-Adar index is defined as

$$A(u,v) = \sum_{x \in N(u) \cap N(v)} \frac{1}{\log |N(x)|}.$$

For two neighbor sets $N(u)$ and $N(v)$ the Preferential Attachment score is defined as

$$PA(u,v) = |N(u)||N(v)|.$$

as by Liben-Nowell and Kleinberg in [LNK04].

More global approaches are found in Katz Index and Random Walks. In a random walk, we stroll around in the network from a starting node to another node until we find a satisfactory result. Katz Index is a generalization of the Eigenvector centrality. The node's neighbors are checked with the Adjacency Matrix of every other node connected.

The common on application of similarity measures is link prediction. This assumes that similar nodes are likely to have a connection.

For our application we need to look into structural similarity rather than direct analogies.

2.6 Applications of Jensen-Shannon Divergence

To compare distributions evaluation of different divergences is done by Arjovsky in [ACB17]. Here Wasserstein distance is used for unsupervised learning with Generative Adversarial Networks.

Classical applications of Jensen-Shannon Divergence are discussed in the following subsection. Several applications can also be found in quantum physics.

2.6.1 From Estimation to Classification

Aslam and Pavlu in [AP07] use Jensen Shannon-Divergence to query hardness estimation. Active learning for probability estimation is used by Melville et.al in [MYSTM05]. Spatial robot position calculation is taking advantage in Martin et. al. recent work [MCM$^+$17].

2.6.2 Word and Speech Recognition

Word co-occurrence probability is calculated by Dagan et. al. in [DLP99]. Statistical Language Analysis of words and it's nearest neighbors is analyzed by Lee in [Lee01]. The probability of the word "acquire" to occur with "company" is given by the number of "company" and "acquire" occurring together divided by the number of occurrence of just "company". Therefore first the data is trained and tested afterward.

2.7 Social Recommendation and Trust

Recommendation Systems became very famous since Online Social Network sites like Facebook are presenting friend suggestions, video streaming platforms like Youtube are suggesting videos which are in you are interested. Music streaming platforms like Spotify are suggesting music you might like based on what you heard. And even the placement of content is based on personal attributes. Travel search engines recommend hotels which are based on earlier preferences and resulted booking. Movie ratings identify users interested in specific genres in order to suggest similar. Also rating of items in e-commerce platforms does similar things. In 2020 the possibilities and touchpoints in our every day life are countless. All is based on your individual user information. This user centric approach is the social component and in literature often called Social Recommendation System. We could further distinguish between content recommendation and people recommendation.

As introduced in Chapter 1, recommender systems can be done in several ways, there are content-based, knowledge-based, demographic-based, collaborative filtering, popularity based and hybrid approaches. Ricci et.al. in [RRS11] created a nice handbook introducing techniques and evaluation.

Human computer interaction may bring information to track in various forms. The classic mouseclick as input triggers many actions. But starting an operation is just one explicit manner. With response marking obligations, signalizing consent or rejection is possible. Even only viewing content can bring implicit feedback through scrolling or observation measurement.

In all consequence, it needs to be considered that an algorithm or system never must be superficial. It is always developed in an interest and standard ethic rules are relevant. Biased results and discrimination through minority data or different treatment of input is still an issue. A prominent example is face recognition of black-skinned people, which may work worse because of missing contrast in the picture. Also simple languages by smaller complexity of characters may enhance the possibilities of the usage of enhancements through tools. General Data Protection Regulation need to be complied with at all time when using person-related data.

Li and Chen in [LC13] look into the relation of recommendation based on bipartite graphs and link prediction. They distinguish recommendation algorithms in collective local and graph related features. Again heuristic methods are reviewed with learning-based approaches. Heuristic means randomness is used somehow and therefor it is not guaranteed to find the correct answer. Most available work was either using local features for learning-based methods or were graph related heuristic models. Also they confirm that most link prediction research focus on unipartite network, however the recommendation problem is bipartite.

The user to object relation is perfectly represented in a bipartite network. The main application of a recommender is to suggest the best fitting items for the user.

For collaborative based filtering, user feedback is used for recommendation. While for content based suggestions, the item's attributes are relevant. Collaborative filtering can further can be grouped into model based and neighborhood based methods. Matrix factorization is the most common content based filtering approach. And for matrix factorization the most common approach is single value decomposition (SVD):

$$\hat{R} = \hat{U}\,\hat{\Sigma}\,\hat{V}^T$$

n are the number of users. m are the number of objects. d are the number of features to represent the users taste and items. They are also called latent or hidden. $\hat{R}$ is the expected rating matrix in the dimension of $n \times m$. $\hat{U}$ is the users taste matrix in the dimension of $n \times d$. $\hat{\Sigma}$ is the significance of feature matrix in the dimension of $d \times d$. $\hat{U}$ is the object description matrix in the dimension of $d \times m$.

The strength of a concept is found in the diagonal. The rating matrix is ahead in equation. wording: concept vs feature As in naming "decomposition" is a kind of split information to individual attributes which expresses pattern.

To the "social" as in social Recommender System - social relationship as trust or social interest is meant. Due to the connection and experience in the relation, recommendation from friends are more appreciated than of acquaintance.

The idea is to improve Recommender System based on Social Network information of who trusts whom. Trust is typical as application.

Guo et. al. in [GZYS15] makes the observation that trust information can be additionally used to rating information. User similarity like social neighbors just have a weak positive correlation in rating. On the other hand trust based relations ship have a strong positive correlation in ratings. We want to close this gap between user similarity and trust concepts in Recommender Systems.

Conservative models use just rating data for predictions. The main work is to extend the model with social data in order to improve recommendation. The baseline is the SVD++ model as Koren in [Kor08].

Li and Chen in [LC13] confirm Koren's SVD++ as a local feature collective and learning based approach.

Compared to the idea of matrix factorization, SVD++ extends the concept. SVD++ adds biases: An overall one for user specifics and one for item specifics. Also implicit feedback is additionally considered in the equation:

$$\hat{r}_{uv} = \mu + b_u + b_v + q_i^T(p_u + |N(u)|^{-\frac{1}{2}} + \sum_{n \in N(u)} y_i^T)$$

$\mu + b_u + b_v$ is the baseline estimate. q_i^T is the item model. $(p_u + |N(u)|^{-\frac{1}{2}} + \sum_{n \in N(u)} y_i^T)$ is the user model. $|N(u)|^{-\frac{1}{2}}$ is just a normalizing factor for the latetend features.

The model is executed as a minimization problem in several iterations of a machine learning task.

Taheri et. al. in [TMF+17b] extracts implicit social relation for Social Recommendation Techniques in User Rating Prediction. The main idea is to use the user similarity to conclude their trust relation, This is referred to as implicit social relation. Taheri's HellTrustSVD is based on and extending the TrustSVD - Recommender of Guo. Guo is the baseline where the trust matrix is already defined. The trust term T_v^+ in Guo paper equation (1) is the relevant term to modify.

This combination of traditional rating recommendation additionally sourced with trust data, results in a improved model in order to make better recommendations.

2.8 Expected Result

With the knowledge out of the reviewed literature we are able to frame our investigation. A lot of measures and algorithms available and there are stochastic (based on probabilities) as well as deterministic (fixed input leads to fixed output) approaches. The given methods lack of a structural analysis of the connections to others. Finding some representative user instead of maximization of given properties is usable. Also there is a potential of the bipartite setup to be used for recommendation purposes. We are expecting to receive a unique result fitting to our the definition of influential.

As our structural approach of finding some behavioural representative users is unique no correlation to other metrics is expected. Through the result scores mapping of one bipartite nodeset to relate to an unipartite graph supposed to be possible. When trying link prediction precision as the vast majority structure is the base for prediction, a good level of precision should be reached. As links are possible from each node to any other node any hit out of the top few ranked predictions are meaningful. The bipartite setup models a user to item relation very nicely and therefore will be elaborated as an input for a social recommendation system.

CHAPTER 3

Methodology and Approach

In our approach we need to be carefully about what we indicate as Method and what is the Methodology. Methodology is the term to use for a combination of methods. The methodology discuses the theoretical analysis and the systematic procedure of the methods applied.

Similar to that we are also sensitive to the perception in the difference of measure and metric. Measure is a count represented as a number as a result from a measurement. For example, body weight or room temperature are measures. In opposition when being precise, a metric is derived by taking a calculation between various measures. Further synonyms which could be used are indicators, indices or simply concept.

In this work, we try to find an influential user, which is representative for the network. Therefore we want to find out the similarity of nodes. Because of this, we calculate the difference to other nodes. Finding the lowest summed distance defines the most structural average node. The idea of common neighbors is very structural. It can be very nice represented as a distribution. To compare two distribution some divergence matter is required. The requirements need to be even strengthened because all distance measures are divergence measures but not vice versa. We first need to find a well-defined distance metric.

Our line of argument is similar as in the HellRank-paper by Taheri et.al. in [TMF$^+$17a].

3.1 Well-defined Distance Metrics

We first define what is a well-defined distance metric as defined by Hunter in [Hun12]. Basic algebra and analysis methods are taken out of Drmota et.al. in [DGKP14].

Definition - Notation of the distance function is d(x,y) between every pair of points x, y of a metric space set X. For a well defined distance metric the following properties hold.

Positive Definiteness $d(x, y) \geq 0$ and $d(x, y) = 0$ if and only if $x = y$

Symmetry $d(x, y) = d(y, x)$

Inequality $d(x, y) \leq= d(x, z) + d(z, y)$

A distance metric is well defined if the defined properties hold. Positive definiteness says that no negative values are allowed and the distance can just be zero if x and y are in the same location. Symmetry says that the way from x to y is the same as the way from y to x. Inequality is the triangly property the direct way is shorter than the way through another point. The intuition behind those properties are easy to imagine in a two dimensional space.

3.1.1 Jensen-Shannon Divergence

Now we need to select a divergence measure. The well known KL divergence introduced by Kullback and Leibler in [KL51] is not feasible as it is not symmetric. Therefore we take a look at Jensen-Shannon Divergence. An intuition is to calculate the total divergence to the average. It is based on Jensen's inequality and the Shannon entropy.

Suppose a discrete probability distribution $P = (p_1, ..., p_m)$ and $Q = (q_1, ..., q_m)$. The length of the vectors is defined as m. A continuous notation is not needed for the application of graphs. The discrete notation is used as defined by Cha in [Cha07]:

$$d_{JS}(P||Q) = \frac{1}{2}(\textstyle\sum_{i=1}^{m} P_i ln(\frac{2P_i}{P_i + Q_i}) + \textstyle\sum_{i=1}^{m} Q_i ln(\frac{2Q_i}{P_i + Q_i}))$$

It is important to note that the square root of the Jensen Shannon divergence is the Jensen Shannon distance.

The definition of Jensen-Shannon divergence based on KL divergence as in Arjovsky et. al. in [ACB17] is

$$d_{JS}(P||Q) = KL(P, (P + Q)/2) + KL(Q, (P + Q)/2).$$

Lemma 1 - Jensen Shannon divergence for all positive real vectors is a well-defined distance metric function.

Proof - Based on the true metric properties with probability vector P and Q.

$$d_{JS}(P||Q) \geq 0$$

$$d_{JS}(P||Q) = 0 \Leftrightarrow p = q$$

$$d_{JS}(P||Q) = d_{JS}(q, p)$$

The triangle inequality is also mentioned in [CA10] and [Lee99] but without a proof. Though a proof for the square root can be found in [LMB$^+$08].

3.1.2 JS Divergence combined with our Common-Neighbor Approach

In this section, with introducing JS divergence with our common neighbor approach we calculate the structural distance rather than a geodesic distance.

A common neighborhood is meant in a way of a mutual connection. The relationship is shared through a note which is joined together through an individual.

Nodes cannot be directly compared as they may have different numbers of neighbors which is not represented by the distribution. The distance measure does not give any information about the similarity of two nodes. Though the structure of neighbors is described.

We want to claim that by the degree of the comparing nodes we can tell which limit values the node will be above (and below). The problem we want to solve is how to calculate upper and lower bounds, having the degrees of the nodes to compare given. This should show that the count of neighbors influences the bounds.

We focus on one side of a bipartite network. Suppose a bipartite network G with sets V_1 and V_2. x is a node in network G in which it's neighborhood is N(x). δ is the maximum count of common neighbors which is connected to x through another node.

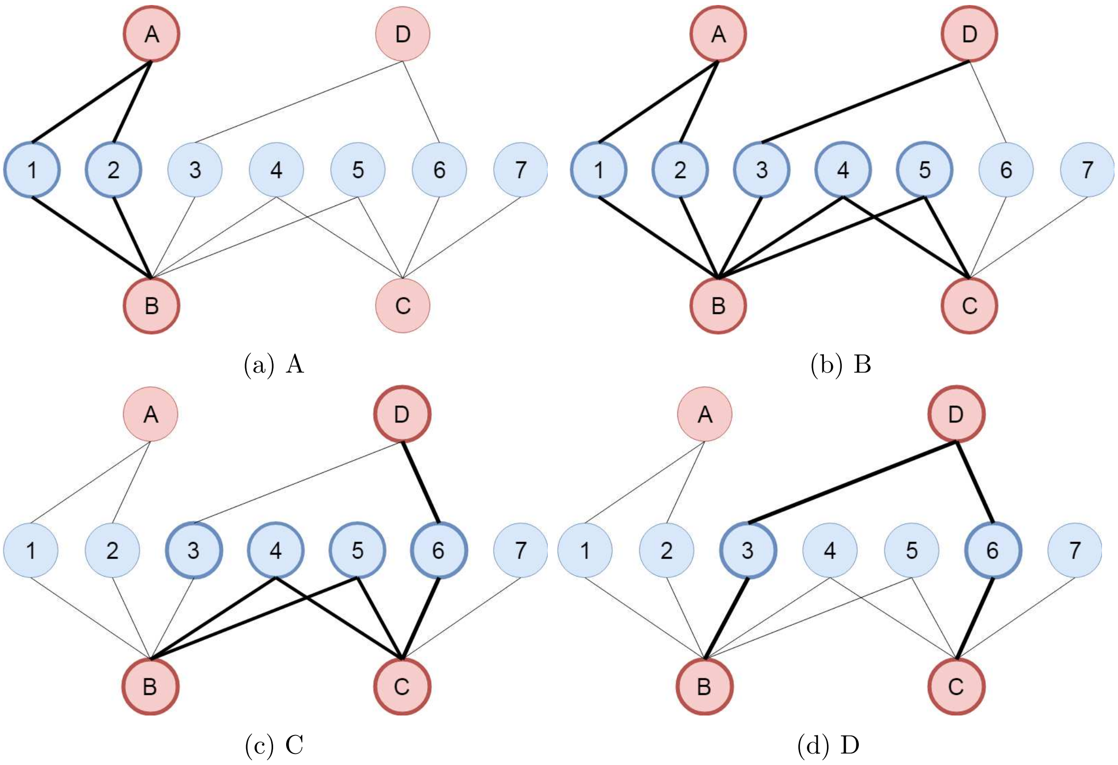
Let l_i be the number of nodes of the set V_1 which are connected to x through i single nodes. Apparently those connections nodes are common neighbors and are part of set V_2. Suppose the vector $L_x = (l_i, .., l_\delta)$ be the normalized distribution of l_i for all neighbor of neighbor of x. Now we introduce the JS divergence between two nodes x and y on one side of the bipartite network:

$$d(x, y) = d_{JS}(L_x||L_y)$$

The function $d(x, y)$ represents the difference in the distribution of L_x and L_y. To the best of our knowledge, this approach of combining divergence with the clustering based common neighbor idea is novel.

3.1.3 Apply to the Bipartite Network

Now, JS divergence is applied to calculate the similarity of nodes. Referring again to our Simple Graph in Figure 2.3 we can see that node A and node B have two neighbors in common. Node A has one connection to it's nodeset V_1 with two connections just through one node of set V_2 each. The distribution of common neighbors shows with how many neighbors of neighbors does it have a certain number of common neighbors. For example node B has one neighbor of neighbor with one common neighbor (D) and two neighbor of neighbors with two common neighbors (A, C). The distribution function vectors of node A is (0,1), of B is (1,2), of C is (1,1) and of D is (2,0). The probability



(a) A

(b) B

(c) C

(d) D

Figure 3.1: The common neighbor approach is shown. The connections counting for building the distribution vector are highlighted.

distribution function vector of node A is (0,1), of B is (0.33,0.66) of C is (0.5,0.5) and of D is (1,0).

In Figure 3.1 the common neighbor approach is shown. The connections counting for building the distribution vector are highlighted.

3.1.4 Method-flow Summary

Figure 3.2 shows a summary of the line of argumentation of the method. It is important to understand to which extend the relation is drawn. A graph has many nodes but the different steps report the relevance of specific multiplicity. We narrate to the operation "of two nodes" or "of a node". The divergence we discuss is in a structural matter and used to establish our distance metric. We calculated the well defined distance in order to get the similarity of two nodes. The smaller the distance, the more similar the two nodes are. When getting a result relating to just one node, the similarity to each other node is considered. As applications link prediction and node ranking utilize at different stages.

To receive nice values, some kind of normalization may happen at each step. Dividing by the minimum or maximum number of a series leads not normalized values.

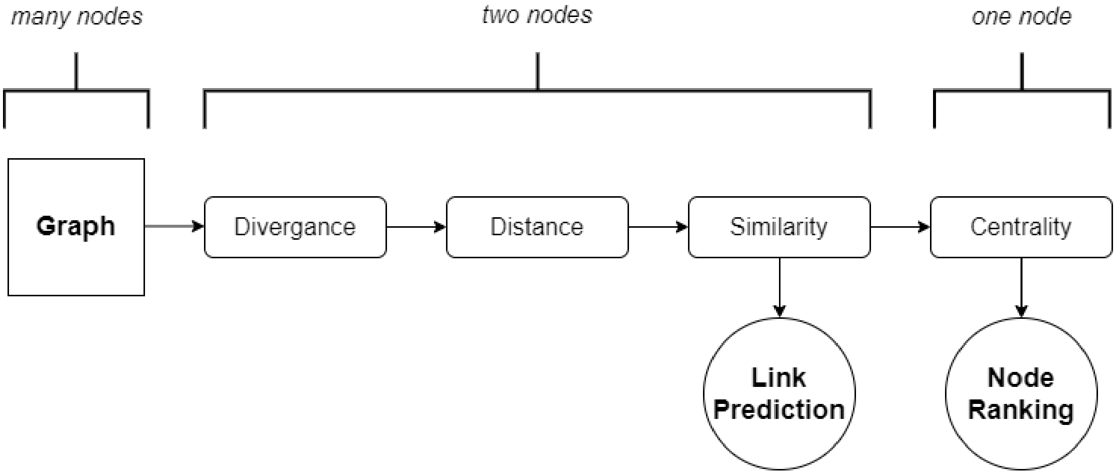


Figure 3.2: Combining our different methods in flow a flow as a line of argument for our measure

Nodes that are most similar in its neighbor's specific structure are most influential. We evaluate the influential nodes in a real bipartite network to conclude if our assumptions were correct. In the application, we draw implications to the individual nodes of the dataset. The evaluation helps to detect strength as well as limits. For specific attributes like density or formation, the setup may benefit the significance or show tipping points of the system. A comparison with other approaches helps to have a baseline. This association helps to look into the boundaries and outlining nodes.

3.2 Evaluation Setup

In order to proof the feasibility and correctness we need to evaluate our approach. To do so we will implement our metric to further apply it with input. With checking the outcome we can validate the method and verify if the expected result is met. We follow an experimental evaluation approach.

3.2.1 Development Environment

Before we take look into the options we have to compute our implementation, we define the requirements. As soft requirements it should be very generally spoken be easy to use, straight forward and lightweight. Below we further assess challenges in our implementation.

Our computing is strictly deterministic and procedural. We do not need a special sequence or objects. So a scripting like languages is sufficient, no object orientation not required.

The main field is in the applications of graphs, so a supporting library dedicated especially on graphs is appreciated.

It is an advantage to run platform independent. Our approach is very mathematical so no framework for application building is required.

Common tools used in lectures are R or MathLab. Those have limited features.

Object oriented software development is often done in Java. It has the big advantage of being freely available and a lot of Framework and tool support. The community makes it very independent. Large scale projects are possible

In Data Science and Security lectures Python is very popular. Machine learning tasks benefit from the available tools. Security simulations count on the reliability. The stability combined with the flexibility of extensions makes it very attractive for all kind of projects. It is already one of the most used programming languages. NumPy and SciPy for scientific computing with high performance requirements and Pandas for general data analyis and visualization are one of many available libraries. [1]. NumPy, mathplotlib, SciPy lib are part of SciPy. SciPy holds the implementation for our required correlation metrics.

As for our method the graph support of a tool is most crucial we need to take a further look in library support. In Java there is JGraphT and the commercial BFO - Big Faceless Graph Library. In Python there is NetworkX kite refers to networkX [2] Other than NetworkX there are also other Pyhton network library implementation. [3] IGraph seems to be one of the most relevant. [4] IGraph is also available for R. [5] [6] is mentioned very commonly. In JavaScript there is GraphLib available. [7] C# has NodeXL available.

For our approach no specific sophistic datastorage like an object relational database is required. Neo4J is a specialized database for graphs. A dataset is represented as node, edges the resulting adjacency matrix. Therefore a textfile is sufficent. Also no history as a memory is involved.

Threading and multi process might become an issue if the processed data amount is big.

For a more sophisticated event based architecture, data stream processing might become relevant. This could especially be the case when dealing with a huge amout of unstructured data. Collected data is sorted by operators in a stream before processing. Here one implementation Apache Flink supports bipartite graphs.[8] Library assistence for Java,

[1] https://steelkiwi.com/blog/python-for-ai-and-machine-learning/

[2] https://kite.com/python/docs/networkx.bipartite

[3] https://wiki.python.org/moin/PythonGraphApi

[4] https://igraph.org/python/doc/igraph.Graph-class.html

[5] https://rstudio-pubs-static.s3.amazonaws.com/214011_2e5e0489035742209fb1880d3f91b581.html

[6] https://rpubs.com/cjsegneri/bipartite

[7] https://stackoverflow.com/questions/14483473/is-there-any-javascript-libraries-for-graph-operations-and-algorithms?rq=1

[8] https://ci.apache.org/projects/flink/flink-docs-stable/dev/libs/gelly/bipartite_graph.html

Python and others exists. This again shows that graph implementation are widely supported.

In any case basic syntax is easy to use, and special cases need to be developed. Semantic results have to be verified afterwards. No matter which tool is used the implementation result has to be reliable and reproducible.

When looking through all option, also other possibilities would be valid, but Python and networkX setup is sufficient and was chosen. Python seems to be most straight forward, reliable and simple to handle. NetworkX benefits from it's large community.

The implementation was done in Phython using the library NetworkX as defined by Hagberg et.al. in [HSSC08]. We are using it in version 2.2.

To get a feeling which topics are currently trending, we reviewed the release notes of networkX. The release cycle is twice a year and the last version was published in October 2019. Specificly we look for the centrality and biparite related features. In 2.4 minimum weight bipartite matching, group centrality measures and incremental closeness centrality was implemented. In 2.2 percolation centrality was implemented. In 2.0 centrality algorithms were harmonized with respect to the default behavior of the weight parameter. NetworkX also provides load, harmonic, percolation and second order centrality capabilites, tough they might not work for a bipartite setup.

As a supporting tool for code development JetBrains PyCharm was selected. The graphic user interface is very user friendly and it brings a few useful features. Of course runtime and code highlighting is cruical for every development environment compared to code in plain files and command line. Additionally library management and debugging abilites are very supporting.

In order to keep track of any changes done in the implementation it is very common to use a git repository as a open source and free version control system [9]. Also for as a preperation for project scaling and collaboration git is helpful.

Visualization of graphs is another topic. Render and design effects trigger different requirements. This is why this topic should be viewed separately. d3js is a popular JavaScript Library for Visualisation. Gephi is a standalone application for network analysis and application. Those are the tools which are considered if any representation is required and then selected on the use case which to use. For simple graphs or flow diagrams in this documentation draw.io[10] was used.

When visualizing a bipartite graph usually the two nodesets are seperated. Another form is to see layers of nodesets where the connections are just allowed from one layer to another. [11]

For Machine Learning tensorflow is a very common library used in Python. In order to find an implementation of TrustSVD several approaches were used.

[9] https://git-scm.com/

[10] http://www.draw.io

[11] https://stackoverflow.com/questions/3399340/how-do-i-implement-a-bipartite-graph-in-java

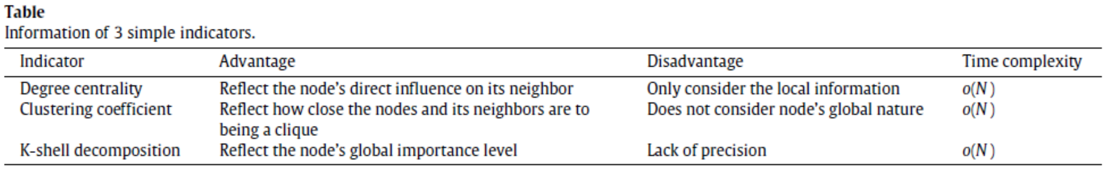


Table
Information of 3 simple indicators.

Indicator	Advantage	Disadvantage	Time complexity
Degree centrality	Reflect the node's direct influence on its neighbor	Only consider the local information	$o(N)$
Clustering coefficient	Reflect how close the nodes and its neighbors are to being a clique	Does not consider node's global nature	$o(N)$
K-shell decomposition	Reflect the node's global importance level	Lack of precision	$o(N)$

Figure 3.3: Influential Node Simple Indicator's Complexity

We are facing issues trying existing TrustSVD implementations, we found several as below though could not run any of them successfully: [12] (based on tensorflow) [13] (just python 3, but I do not get tutorial results) [14] (problem with 3rd party library)

Most likely I have library dependency issues. I tried with python 2.7, 3.6 and 3.7. Also I could not run tensorflow. Other issues which I suspect causing me troubles are spaces in the directory path or missing authorization in directories like root. A linux based environment may work better.

Further tools which are elaborated are graph lab (python) and LibRec (https://www.librec.net/). LibRec is written in Java, which due to compatiblitiy it was not reviewed in the first place and just got focus later. Though there is a so called demo mode in version 1.3 which allows to easily run different rating prediction in the commanline. Datasets are configurable and parameter adjustable as well. The API documentation can be found on their website. Apparently LibRec was developed by Guo et.al who were also publishing [GZYS15]. As all LibRec supported datassets are mentionend in the paper. Datasets and the according calculations can be easiely be converted in python to textfiles or gpickle format. It turned out that LibRec is ready for straight forward usage and could be used for our experiments.

3.2.2 Complexity

For most metrics every node has to be compared with every other node. The encountered complexity of $\mathcal{O}(n^2)$ per node leads to the possibility to calculate data sets of some 10.000 nodes within hours on a common computer. Afterwards the computation runs soon into limits of resources. The square complexity leads to four times the runtime when doubling the data.

It appears that because all local information of all nodes need to be saved for calculations exhausting the memory and space complexity more than the action computational time complexity.

According to Wen et.al. in [WTWJ18] in Figure 3.3 we that common methods all run with linear complexity. Figure 3.4 shows the complex algorithms used in the same paper.

[12] https://github.com/gtshs2/TrustSVD

[13] https://orange3-recommendation.readthedocs.io/en/latest/

[14] https://github.com/hongleizhang/RSAlgorithms

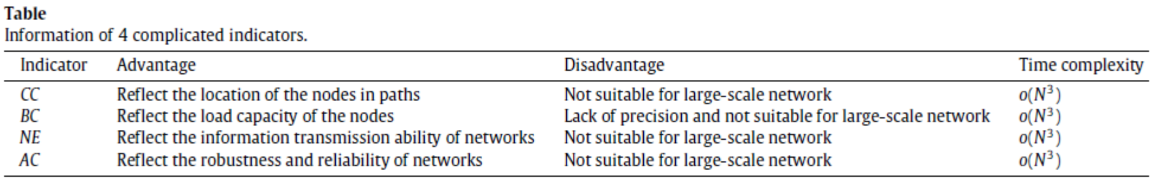


Table
Information of 4 complicated indicators.

Indicator	Advantage	Disadvantage	Time complexity
CC	Reflect the location of the nodes in paths	Not suitable for large-scale network	$o(N^3)$
BC	Reflect the load capacity of the nodes	Lack of precision and not suitable for large-scale network	$o(N^3)$
NE	Reflect the information transmission ability of networks	Not suitable for large-scale network	$o(N^3)$
AC	Reflect the robustness and reliability of networks	Not suitable for large-scale network	$o(N^3)$

Figure 3.4: Influential Node Indicator's Complexity

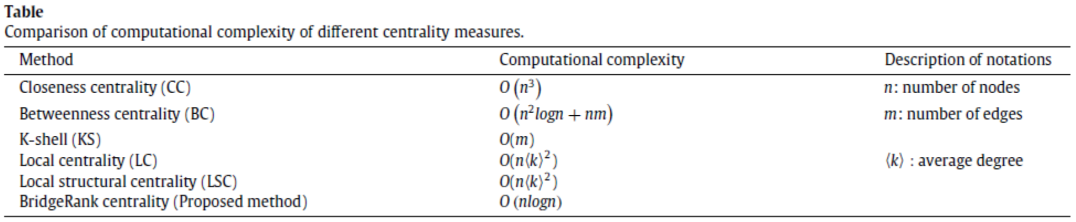


Table
Comparison of computational complexity of different centrality measures.

Method	Computational complexity	Description of notations
Closeness centrality (CC)	$O\,(n^3)$	n: number of nodes
Betweenness centrality (BC)	$O\,(n^2 logn + nm)$	m: number of edges
K-shell (KS)	$O(m)$	
Local centrality (LC)	$O(n\langle k\rangle^2)$	$\langle k\rangle$: average degree
Local structural centrality (LSC)	$O(n\langle k\rangle^2)$	
BridgeRank centrality (Proposed method)	$O\,(nlogn)$	

Figure 3.5: Complexity of advanced centrality measures

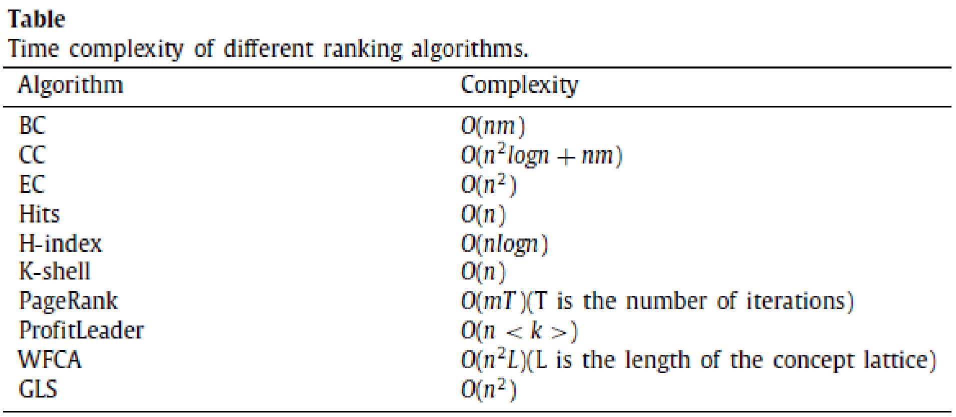


Table
Time complexity of different ranking algorithms.

Algorithm	Complexity
BC	$O(nm)$
CC	$O(n^2logn + nm)$
EC	$O(n^2)$
Hits	$O(n)$
H-index	$O(nlogn)$
K-shell	$O(n)$
PageRank	$O(mT)$(T is the number of iterations)
ProfitLeader	$O(n < k >)$
WFCA	$O(n^2L)$(L is the length of the concept lattice)
GLS	$O(n^2)$

Figure 3.6: Complexity of Ranking algorithms

The presentation is very simplified, as other papers show Betweenness, Closeness, PageRank Centralities have even higher complexity as walk throughs are required.

According to Salavati et.al. in [SAM18] in Figure 3.5 we see that Closeness and Betweenness centrality computation are very time consuming as the paths from each node to every other in each possible way needs to be found.

Sheng et.al. in [SDW+20] in Figure 3.6 additionally distinguishes between n as the nodes complexity and as the m neighbor nodes (in other words edges). So they abstract local and global influences.

Wang et.al. in [WZXD16] and Zhang et.al. in [ZYD+19] also consider complexity analysis of related methodologies.

Measure	max(Complexity)
Degree Centrality	$\mathcal{O}(n)$
Closeness Centrality	$\mathcal{O}(n^3)$
Betweenness Centrality	$\mathcal{O}(n^3)$
PageRank	$\mathcal{O}(n)$
Clustering	$\mathcal{O}(n)$
k-shell	$\mathcal{O}(n)$
JSBiRank	$\mathcal{O}(n^2)$

Table 3.1: time complexity

Summarizing of the below result with the for us relevant measures can be found in table 3.1. Just the for us relevant measures are considered. Big O Notation shows time complexity. For simplification we just consider node number and no edge number. Closeness and betweenness centrality are very extensive as so many shortest paths and the according saving of information is required. PageRank is surprisingly cheap, as it just considers the neighbors importance but still looks at local attributes. For a clustering approach based on local metrics each node just needs to be visited once. The maximal complexity according to definitions found in literature are shown.

The intuition for closeness and betweenness centrality is that calculation needs to be done from each node to every other node for every possible path. For JSBiRank the calculation is just required for each node to every other node.

In order to optimize our program code we node that we are limited by time complexity of the algorithms. A clear focus on effective implementation is required. Each calculation just needs to be done one, buffered to be reused and taking care of the sequence to not run in to any unnecessary iteration. Of course redundant code needs to be avoided. With this in mind we are able to work memory and time efficient.

3.3 Implementation of JS-BiRank

We now want to get a score result for each node to be able to compare them. With calculating the distance $d(x, y)$ between each node with every node of the same nodeset we get a n times n JS-matrix. For our simple graph this leads into the following result.

$$JS-Matrix(G) = \begin{array}{c c} & \begin{array}{cccc} A & B & C & D \end{array} \\ \begin{array}{c} A \\ B \\ C \\ D \end{array} & \begin{pmatrix} 0 & 0.132 & 0.216 & 0.693 \\ 0.132 & 0 & 0.014 & 0.318 \\ 0.216 & 0.014 & 0 & 0.216 \\ 0.693 & 0.318 & 0.216 & 0 \end{pmatrix} \end{array}$$

To get a score value per node the node set count is divided by the sum of distances with all nodes:

$$JSBiRank - Score(x) = \frac{n_1}{\sum_{y \in V_1} d(x,y)}$$

To normalize the score we multiply by the minimum score and divide by the maximum score. Then a result between 0 and 1 is reached. Normalized JSBiRank-Score of node A is 0.428. Normalized JSBiRank-Score of node B is 0.959. Normalized JSBiRank-Score of node C is 1.0. Normalized JSBiRank-Score of node D: 0.363. This result is clearly valid, as for the common neighborhood criteria, B and C have the most common structure and the average most little deviation to the other nodes.

3.4 Experimental Evaluation

3.4.1 Used Datasets

When looking for datasets, several sources where explored. Kunegis in Konect (Koblenz Network Collection) [Kun13] and Leskovac and Krevl with SNAP (Stanford Large Network Dataset Collection) [LK14] are noteable collections A very large bipartite dataset provided by Rossi and Ahmed in [RA15] with around 896.3k edges. The internet movie data base (IMDB) network is frequently used in scientific machine learning experiments.

A selection of datasets with different attributes was selected to ellaborate on the topic.

The famous Davis' women dataset [DGG09] shows 18 women participating in 14 events. In Figure 3.7 the calculated JSBiRank Scores can be seen. All scores are normalized by the maximum value to be able to compare. Myra, Helen and Charlotte are the most influential user. Their friends' connection is equally distributed. They are not highly connected with a core group, nor just connected to single people. Ladies who are highly connected over people to a person and ones which just have a connection over a person over single or dual strings are ranked low. Flora and Olivia are clearly the biggest outliners as they are just participating at two events where additionally a lot of other people take place.

The Facebook-like forum dataset Opsahl provides in [Ops13] is a two-mode dataset of a forum where user contribute on a certain thread. There are 1421 nodes in form of 899 users and 522 topics and 7029 edges. The weighting information about how many characters the users contributed is not used.

An artificially generated graph with 100 nodes per set and high average degree is used to discover limits. To see the behaviour with low density the net2m dataset is used. Norwegian boards of publicly listed companies are tracked for changes on a monthly basis following Seierstad and Opsahl in [SO11].

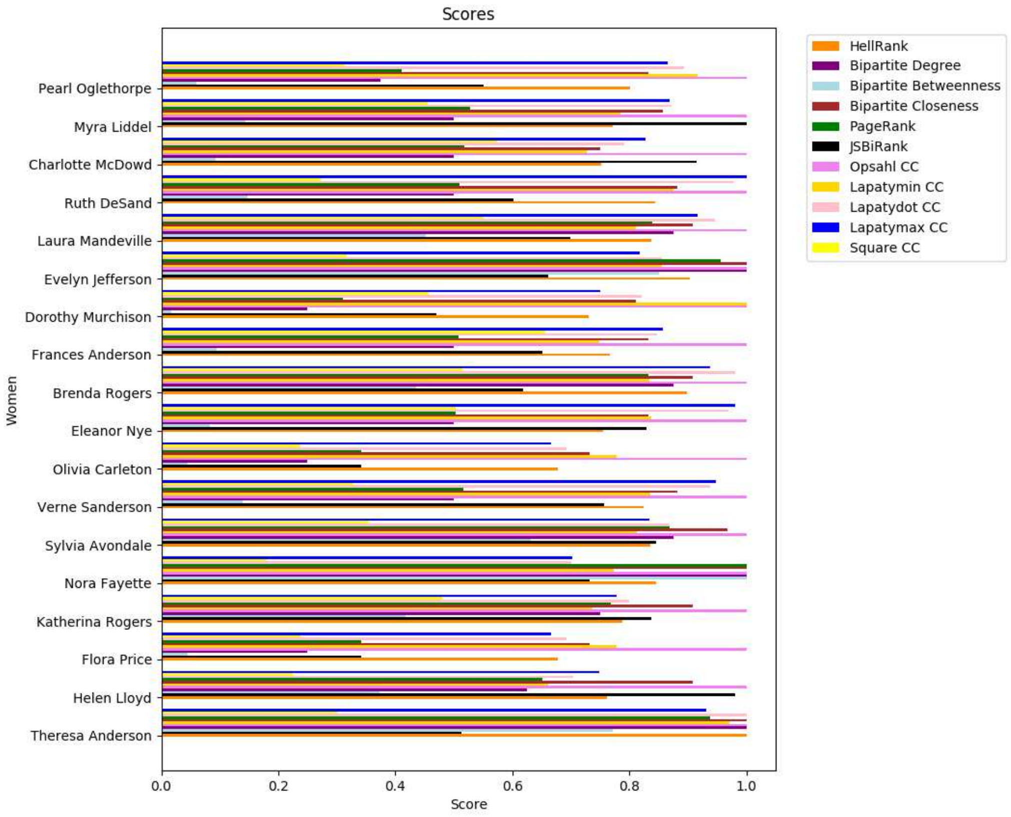


Figure 3.7: Scores

Table 3.2 shows an overview of the considered datasets. The number of nodes and edges are described. As well there is information about the minimum and maximum degree (d), the density and the clustering coefficient (cc).

All datasets used fulfill the bipartite properties in which every node needs to be assigned to one of the two nodesets.

In Figure 3.8 the differnt degree distributions of the used datasets are visualized. Additionally a graph inset is shown, which also gives a nice idea of the common proberties. Islands that are not connected parts and not connected outliners are visible. For dense graphs black edges predominate the picture.

Out of the graph we see that preprocessing of data required is not obligatory. Just if their are serious drawbacks in the result visible by very untypical set of data, the according preparation step need to be taken care of at this point.

Network Graph	|V|	|E|	min(d)	max(d)	Density	cc
Southern Woman	18+14	89	4	7	0.353	0.33
Facebook-like forum (OF)	1,421	7,089	6	10	0.015	0.08
Artifical	100+100	2,987	25	33	0.298	0.18
Norwegian boards (net2m)	1,187	1,130	1	2	0.006	0.71

Table 3.2: Used Dataset's Attributes

3.4.2 Evaluation Metrics

Experiments with Pearson correlation for data correlation are made. Kendall based on Abdi in [Abd07] could be used and Spearman Metrics (Kokoska and Zwillinger in [KZ00]) is used for Rank correlation.

The prominent Pearson correlation coefficient is often simply referred to as "correlation coefficient". It is a measure of linear associations between two normal distributed variables. The result, a value in the interval [-1,1] represents the relationship, where 1 is a perfect correlation and -1 is an inverse connection. 0 indicates no connection between the two values.

The Pearson correlation coefficient is probably the most widely used measure for linear relationships between two random variables (X, Y) and thus often just called "correlation coefficient". Usually, the Pearson coefficient is obtained via a Least-Squares fit and a value of 1 represents a perfect positive relation-ship, -1 a perfect negative relationship, and 0 indicates the absence of a relationship between variables. The correlation coefficient is defined as

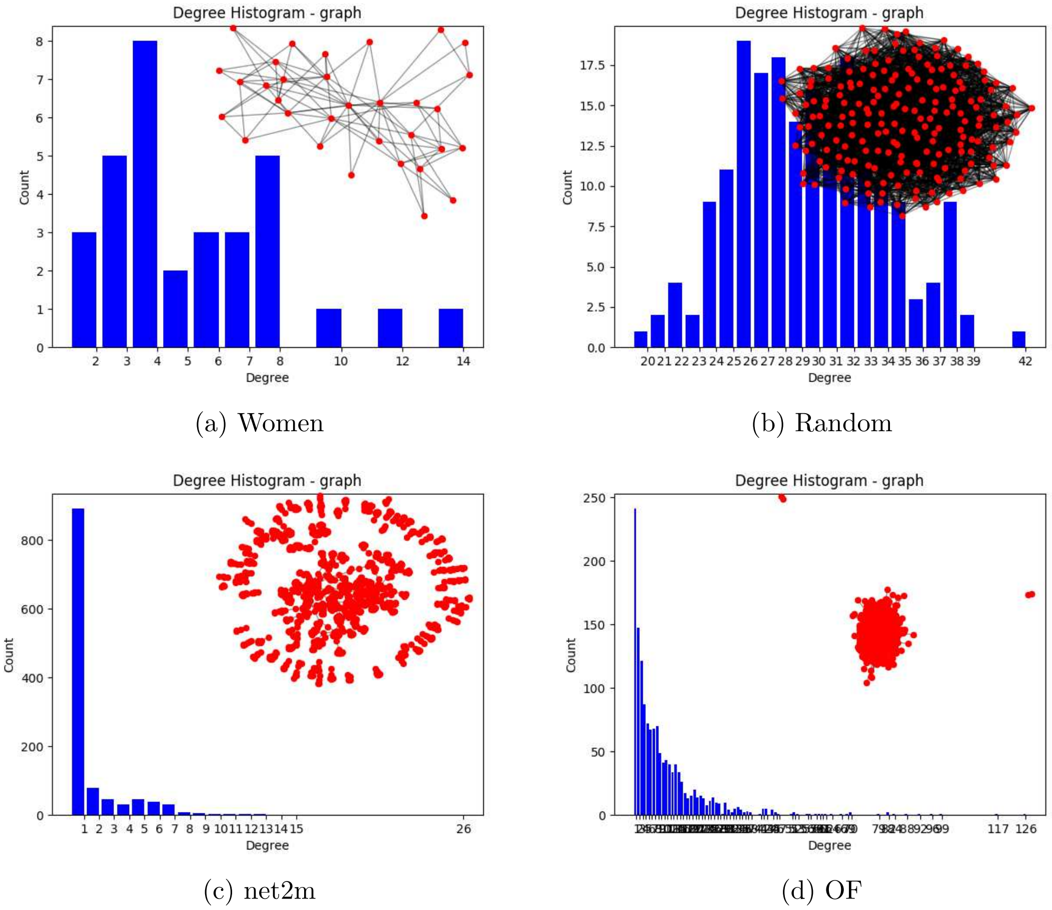
$$\rho_{X,Y} = \frac{\mathrm{cov}(X,Y)}{\sigma_x \sigma_y}.$$

where σ_X is the standard deviation of X and σ_Y is the standard deviation of Y.

Similar like Pearson correlation coefficent, Kendall's Tau measures the correlation specifically of the ranking of values. The ranking is an ordering of the values and does not consider the absolute value. The rankings are represented by the variables X and Y, which can hold continuous as well as ordinal data. The result represents the dependency or the relations of the two variables. Tau is defined as

$$\tau_{X,Y} = \frac{P - Q}{\sqrt{(P + Q + T_X) * (P + Q + T_Y)}}.$$

where P is the number of concordant pairs. Q the number of discordant pairs. T_X is the number of ties in X. T_Y is the number of ties in Y. If a there is a tie for the same pair in X and Y it is not considered. Kendall's original work in [Ken45] discusses in more details how to deal with ties in ranking and suggested two more Tau Coefficients.



(a) Women

(b) Random

(c) net2m

(d) OF

Figure 3.8: Degree distribution with graph inset of used datasets. The blue bars show the count of a certain degree. In the inset nodes are in red, and edges in black.

Another rank based correlation measure is defined by Spearman as

$$\rho_{X,Y} = 1 - \frac{6Q}{n(n^2-1)}$$

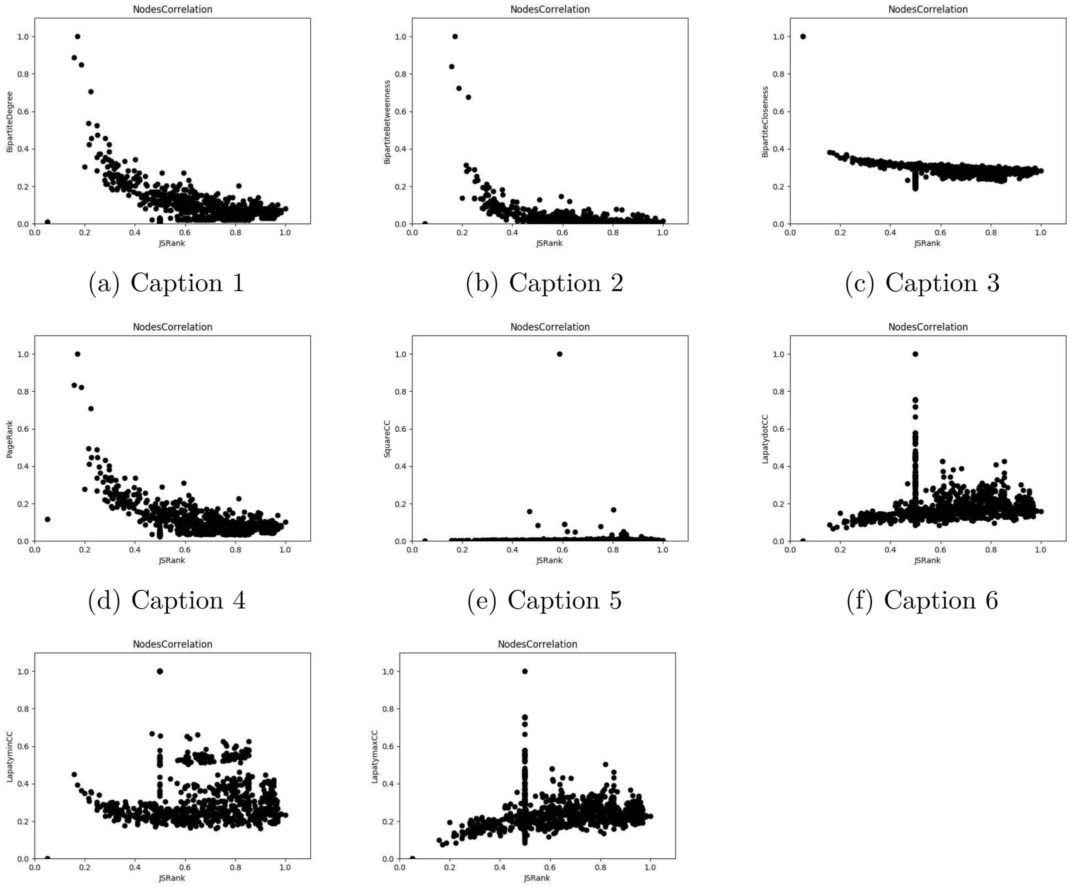
where

$$Q = \sum_{i=i}^{n} (R(x_i) - R(y_i))^2.$$

Q is the pairwise distances of the ranks of the variables X and Y and n is the number of samples.

3.4.3 Correlation to other Metrics

In Figure 3.9 correlation of JSBiRank scores in the OF dataset with common measures is shown. All scores are normalized by the maximum value to be able to compare. The



(a) Caption 1

(b) Caption 2

(c) Caption 3

(d) Caption 4

(e) Caption 5

(f) Caption 6

(g) Caption 7

(h) Caption 8

Figure 3.9: Scatterplots of JSBiRank Score with common measures

nodes are very well distributed horizontally, bringing a high diversity and additional information out of same results of the compared measure. With this certain data, it seems that just a few nodes reach high scores with the common measures and the mass, therefore, reaches scores around 0.2. High JSBiRank Scores often are in the typical range of the average score.

Also outliners can be very nicely explained. One pattern visible for JSBiRank normalized score of 0.5 (real score 14.037) shows all the nodes with who just have common neighbors of 1. This distribution in the common neighbor vector is curcial. Most common nodes have just slightly a few more than one common neighbor, real score of 25 leads to 0.95 distributed at one neighbor. 0.5 distributed at one neighbor leads to score 10. High degree nodes tend to have more common neighbors and therefore are often ranked low, i.e. node 100 with a degree of 88. The most least score of 33 results when a node has no common neighbor, then it is just connected to one node. A workaround vector (1,0,0) is

Measure	Degree Centrality	Betweenness Centrality	Closeness Centrality	PageRank
Correlation	-0.08	-0.07	-0.08	-0.08
Measure	Lapaty dot CC	Lapaty min CC	Lapaty max CC	Square CC
Correlation	0.01	-0.02	0.02	-0.03

Table 3.3: Pearson Score Correlation

used for this case, representing a hundred percent of zero common neighbors. Just two nodes in the whole dataset have just one connection (nodes 33 and 574). These nodes could be removed during preprocessing.

The dataset requires to be highly connected. The metric works better the denser the graph is, as more common neighbor patterns are established. The maximum degree and the according outliners have no effect as this node will then be not influential and ranked low.

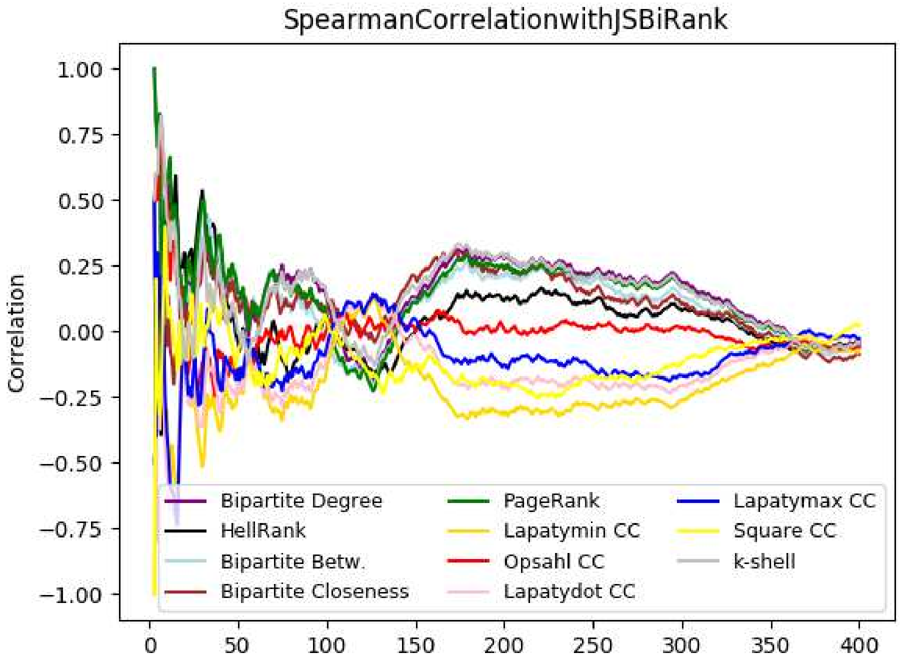


Figure 3.10: Spearman Rank Correlation

In Table 3.3 Pearson Correlation to JSBiRank Scores is shown. No correlation is visible which shows that the method is novel and not related to any other. In Figure 3.10 Spearman correlation to JSBiRank of top k=400 nodes is shown. Despite the fact that we are not looking for most central users for which the common methodologies are defined still certain correlation can be found. Especially local neighbor measures like

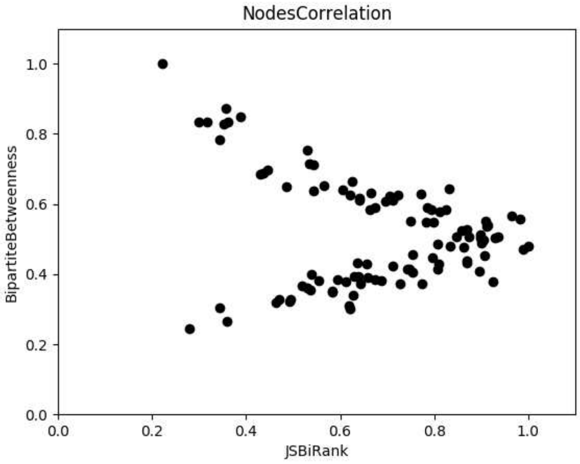


Figure 3.11: JSBiRank Scatterplot for Correlation with common measure Bipartite Betweeness

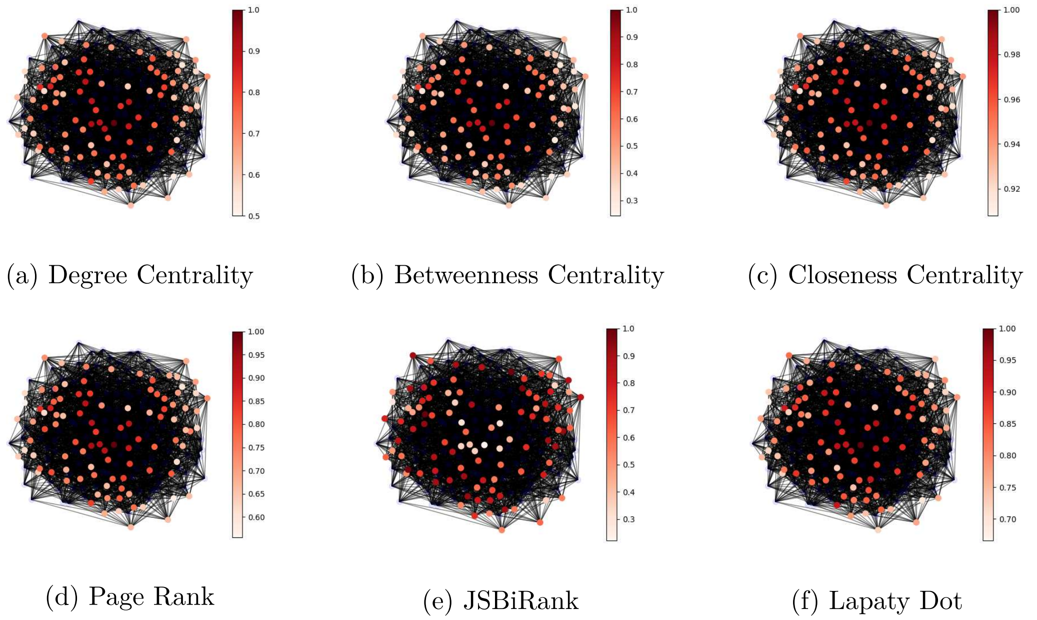
Degree Centrality, k-shell and PageRank show correlation. Also global measures like Closeness and Betweenness Centrality show affects. Clustering-based approaches seem to be inversely related. Though no clear interpretation is possible and the little correlation is more or less random and will look different for other datasets.

3.4.4 Favored Features and Drawbacks

Depending on the attributes of a dataset different results are to expect. The average node is just found with graphs of certain properties. Especially the density and the resulting average degree show big effects on the outcome.

Figure 3.11 shows that with artificial dense dataset some kind of clustering is visible. The scores are nicely distributed throughout the whole range from zero to one. Betweenness Centrality as a representative for common measures is just used to visualize a comparison. Also average values in other measures reach high scores with JSBiRank bringing a nice argument to find "average" nodes. It looks split because values with low and high common centralities generate low JSBiRank values, while the average generate high values. It seems that finding influential nodes works best with very dense datasets with mostly high degree nodes. Unfortunately no considerable real world dataset could be found for further tests.

A drawback is discovered with the net2m dataset which has low density. Equal JSBiRank scores are seen quite often. This is the case if there is a certain common structure. Like most nodes have none or one common neighbor. This nodes then reach the same score as they show same divergence to other nodes. This property could be used to find outliners, who don't have this common structure, as the have another score.



(a) Degree Centrality (b) Betweenness Centrality (c) Closeness Centrality

(d) Page Rank (e) JSBiRank (f) Lapaty Dot

Figure 3.12: Colormaps show scores and identity of different measures (30% connected, 200 nodes)

With taking a look at different dense datasets, we found out that our measure works as expected with datasets of high average degree. When using sparse datasets, the method can be used to find outliners.

3.4.5 Finding the Identity of Influential Nodes

In our case we see the identity of a node defined as which node it is and which attributes it has. As attributes we soley care about the degree.

For further evaluation and checking where this highly rated nodes lie in the dataset we draw a colormap. We expect the position of highly ranked nodes to hint us if there is a specific role of the highly ranked nodes. Closeness centrality for example should show nodes in the center with a very high score as average distance to other nodes is then likely to be short. Betweeness score should highlight briding nodes and degree centrality nodes with a lot of connections. Zhou and Hansen in [ZH15] extensivly survey the visiualization form of colourmaps and mention graph representation as a form of application.

In Figure 3.12 we see the different centrality score by the intensity of the color red. To focus just on one of the two bipartite sets the other one is very transparent blue to not interfere. As the graph is very dense the edges in black create the background.

With our artificial dense graph we see almost no difference between Degree, Closeness and Betweenness centrality as the graph is so highly connected. The plot shows us that it is also the same node who scale about the same. JSBiRank identity and attributes, are exactly the ones which are "average" by other measures.

For Degree, Closeness and Betweenness centrality (Figure 3.12a - 3.12c) our positioning algorithm shows the high rated nodes in the very center. There is some indirect bias in our positioning algorithm as it place most connected nodes in the center. More interesting is that with JSBiRank the vast majority has high ratings in an extend circle surrounding the center as we can see in Figure 3.12e. No clear pattern is visible for Lapaty in the Dot version of 3.12f.

Our setup extreme setup of the dense random graph is parameterized with percentage of connections to other graphs, therefore if the node count increase the absolute value of average degree of the nodes also increases. This leads to the correlation effect, that with a high number of nodes, everything is connected very likely in a short path. This result is seen with percentage of connection of 30 and node numbers of a bipartite nodeset from 10 to 100. With a lower density graph like our simple graph of 2.3 we expect Degree centrality and Closeness centrality showing very different results and seems to correlate in this case. With correlate we mean that the score are different but if we scale or normalize them it is very similar. This holds even for betweenness centrality but is not visiable as soon in terms of density as for closeness. For not so dense graph it is difficult to see something. A small graph would be required but if it is small it has to less attributes to provide a meaninful result in calcultation. The bigger and the denser a graph is the clearer the characteristics are visible. This also holds for PageRank and Lapaty Dot.

3.5 Application of Link Prediction

In the bipartite setup nodes of a set are not connected to each other. We now discuss how we use similarity scores to predict links of two nodes of the same nodeset. The implementation will visualize the outcome and draw implications to the certain nodes.

An important question is which similarity score is valid to determine the connection. We see two options for the parametrization: 1) Take top-n highest values to predict links 2) Use values above a certain threshold to tell ties Hence we need to configure either count of links or the limit score. The score number is used as a weighting factor if some additional complexity in more information content is required. In terms of information content, we can also speak of entropy. The weight tells how bonded two nodes are, or in other words, how strong the relationship is.

Figure 3.13 shows the women out of the Davis dataset with predicted links to each other. The colored nodes represent the n woman. The links are calculated with JSBiRank Similarity. The picture is drawn with distance values instead of similarity scores because zero distance does not result in a valid score. A new mapped unipartite graph results out of the original bipartite graph. In the original graph no direct connections in nodesets existed. In our case we are interested in the social part therefore just the nodeset of women is displayed. The graph was parameterized to see a minimal figure, where each

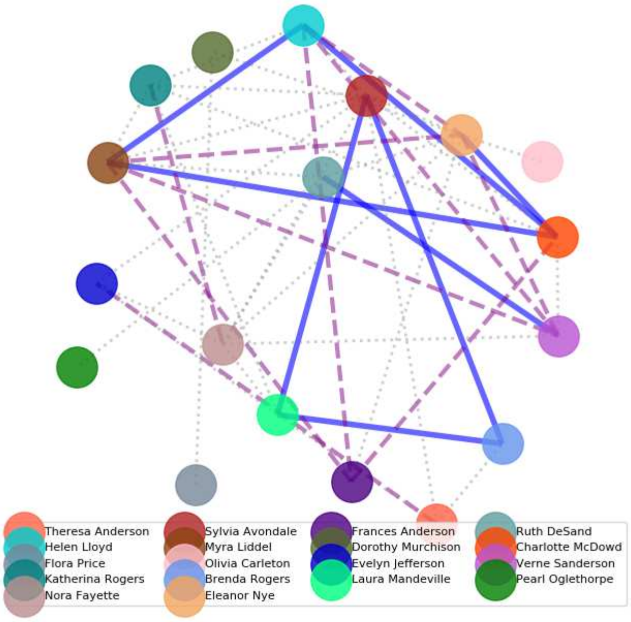


Figure 3.13: The new JSBiRank Similarity mapped graph of the women dataset. The blue line shows strong connection, medium connection dashed purple line, weak connection.

woman has at least one connection. The connection weight limit was set to the value to receive the minimum corrected graph for n = number of women. The blue lines show a strong relationship. A triangle connection pattern of Sylvia, Laura and Brenda are discovered. Another one connects Charlotte, Myra and Helen. The representation highlights also the strong sole connection between Elanor and Charlotte, as well as Ruth and Frances.

What is presented here is different to the node scores overview diagram. The similarity score is always related to a node pair. This pair adds another dimension, which is why it is no longer feasible to compare it to other measures or do correlation investigations at this stage.

For evaluation of our link prediction we want to use common precision determination metrics. Yang et.al in [YLLL16] surveys different evaluation methods. One simple common methods is

$$Precision@N = \frac{N_r}{N}$$

telling the rate of true (right, correct) predictions N_r compared to all predictions N. In order to be able to compare to data, we randomly take out $x\%$ of the links of our truth data and see if they get predicted or not.

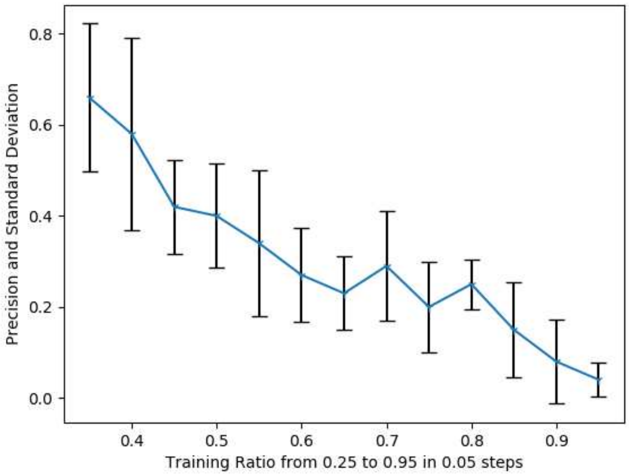


Figure 3.14: Link prediction precision shown in the error plot for the Davis Women Dataset - true prediction of top n=20 result, 5 seeds used to average seeds and standard deviation is shown vertically

Kunegis et.al. in [KDLA] discuss the very different case of link prediction in bipartite networks. Evaluation is difficult because therefore, links need to be removed to check how good they are detected. There are different representations of a bipartite graph: 1) whole bipartite graph 2) projection of one set 3) mapped graph by similarity - if this is checked if it looks the same after removing of the original graph it would more be a measure of robustness

Single edges can not be removed as link prediction just works for same nodeset. This is why we need to remove two links of the opposite nodeset, in other words remove a (second order-) neighborhood. This is kind of working with the projection without losing information. A projection is seen as removing the other nodeset but keep all connections. Afterwards we check how many of the removed links were predicted. We will visualize it as the precision over training ratio. A third dimension is the parameterization of link prediction, which also can be fixed to the best bringing result parameter. In Figure 3.14 we apply our link prediction scenario to the Davis Women dataset and n is fixed to 20 as it works best. n=10 and n=30 showed an increased standard deviation.

A more structured way to select training and test data splits could be a k-fold cross-validation approach. Though it is not easily feasible to split the dataset, hence more different seeds could still be used and the result averaged.

Typically standard deviation is taken to be displayed in the error plot in the vertical lines. The variance would be just the squared version of standard deviation. The square implies that if standard deviaton values are lower than 1 then the variance is even smaller than the standard deviation.

The decreasing trend is natural, as more connections already exist in a setup where just a low number of links is removed. As links are potentially given from each node to any other node any new prediction hit out of the top few ranked is meaningful.

The experiment to check the effect of sparsity for the precision of link prediction is done as a "sole experiment". It is not meaningful to compare to existing link prediction methods. Those link prediction methods require the graph to be projected where information is lost, at least in the most simple setup. For our experiment we do not project the graph, but in order to pay attention to the bipartite setup, just consider the correct projection if two connections, the one to the other nodeset and the one back to the considering nodeset, are predicted correctly. So two edges need to be predicted correctly to predict connection via a path of length two, of two nodes of the same nodeset.

The comparison of Link Prediction with JSBiRank Similarity to Jaccard Coefficient, Adamic Adar Index and Preferential Attachment Score is difficult. We chose the coefficients to sourced with a projected graph, tough in a bipartite setup the prediction just counts if both required edges of a connection of two nodes of a nodeset are established. With JSBirank similarity we look at structural properties globally for the whole graph rather than rely on direct connections in a local consideration.

Other options than Precision@N to evaluate the result could be counting the absolute number of true predictions. Just looking at top n results is a form of parameterization of the prediction. Further possibilities of evaluation link prediction could be using ROC or F1-score.

In the application of link prediction we showed that the JSBiRank Similarity Scores are valid to use for mapping the bipartite graph to a unipartite form. The precision of the calculation was evaluated by taking out some links and check how accurately they will be predicted.

3.6 Application of a Trust-based Social Recommender System

For recommender systems, its performance is an issue, as well as data sparsity. A further common issue of investigation is the cold start behaviour. We look into the challenges in regards to source a recommender with additional input based on JSBiRank scores. The datasets used for evaluation are shown in Table 3.4.

Taheri et. al. in [TMF$^+$17b] with the TrustSVD implementation grants rating prediction. For the trust source to matrix factorization a threshold for tie of scores is required, normal distribution is taken.

Implicit social relation is calculated out of common ratings user have given to movies. A movie which is rated by two users is seen as a connection without any weight. Those connections are used for calculating scores through JSBiRank similarity.

data	users	items (movies)	ratings	trust data available
FilmTrust	1,508	2,071	35,497	yes
MovieLens(100k)	943	1,682	100,000	no

Table 3.4: Recommender data sets for movie rating prediction

The first kind of datasets (FilmTrust, CiaoDVD, Epinons according to Kunegis in [Kun13]) have user - item rating information available, as well as user - user trust information. TrustSVD is computed with the given explicit trust data as well as the implicit trust data. With this the first experiment is performed to prove that implicit trust data works as good as explicit trust data.

We take a look into datasets where no explicit data is available. The second experiment shows that recommendation with data without trust information available improve when using implicit trust. Important to understand is that JSBiRank is used to calculate implicit user - user trust values out of the user - item ranking.

For evaluation purposes the mean average error (MAE) and the root mean square error (RMSE) are used. The overall goal is to minimize the errors. The intuition is to sum the difference of the rating prediction with actual value:

$$MAE = \frac{\sum_{i,j} \hat{r}_{u,j} - r_{u,j}}{N}$$

$$RMSE = \sqrt{\frac{\sum_{i,j} (\hat{r}_{u,j} - r_{u,j})^2}{N}}$$

HellTrustSVD gets implicit trust feedback from the datasets. This is favourable as the bipartite structure fits to the user-item relation. If two users rated one item the user are connected. Trust data is user-user.

We repeated the trust recommender experiment (TrustSVD) out of the Guo paper with their LibRec library (this was the reference for Taheri). Taheri's reports his results have almost the same MAE/RMSE value than Guo showing his method to be valid. In the experiment we use dataset Filmtrust which contains user, films and an according rating. Additionally social trust between users is given.

To calculate the implicit user relation JSBiRank Scores are used. More specifically the similarity (not distance) scores are used, which means the higher the score to 1 the most similar the node is to the majority of the others. The score threshold > 0.9 is taken. We want to find out if implicit user data can compare to explicit user data in the trust recommender.

In Table 3.5 we see the TrustSVD runs parametarizations of d $= 10$ and $\lambda = 0.5$. A bigger number of latent features increases the informations and improves the result just slightly.

Trust Source	d	λ	MAE	RMSE
explicit	10	0.5	0.626	0.826
implicit	10	0.5	0.627	0.820

Table 3.5: Performance comparison of the trust based recommender TrustSVD with source of real and implicit social relations - performed on the FilmTrust dataset

Recommender	Threshold	d	λ	MAE	RMSE
SVD++		10		0.718 (0.001)	0.913 (0.001)
JSBiTrustSVD	> 0.97	10	1.0	0.717 (0.001)	0.910 (0.002)

Table 3.6: Performance comparison when using implicit social data performed on the MovieLens (100k) dataset - standard error is reported in parthesis

λ is a weighting factor to optimize the result and prevent overfitting or underfitting. A value was chosen with which reasonable good results are found. The performance comparison of the errors shows that explicit user data performs as good as implicit data. So it is valid to use implicit user data.

The model is not deterministic, so it does not always produce the same output given input and fixed parameters. Several runs were performed and the best result was taken. The stochastic model is an approximation, in our case in different iterations, which gives estimations in a probability distribution of expected outcome. More complicated models are able to be computed without running into complexity limitations.

In the experiment we predict the rating, other calculations could also recommend an item based on ratings.

Not all datasets have trust information available, for example our MovieLens (100k). In our next experiment we source TrustSVD with implicit trust data instead. The implicit social data is calculated with JSBiRank Similarity Score, therefore we call the recommendation engine JSBiTrustSVD. The configuration taken is num.factors=10 (d), num.max.iter=200 and learn.rate=0.001 .

In Table 3.6 we see that JSBiTrustSVD outperforms SVD++. Even if it is just a slight improvement, Koren in [Kor10] points out that even small accuracy enhancement lead to a notable better result. Our experiments showed that runs with a lower number of latent features ($d = 5$) are not very different, which proved our result to be reliable in respect to the feature dimension. A threshold was chosen to deliver meaningful results. There is a tendency of improved results if the count of social input data decreases.

The key outcome of this experiment is that it is valid to use our JSBiRank Similarity metric as an input to a Social Recommender System. With JSBiTrustSVD, we can use matrix factorization implementation TrustSVD even if no social input data is available. The additional trust information gives additional information to the latent feature vector.

We obtain human relations out of the ratings given. The behavioral representation between users helps to examine social associations.

A user influence metric is used to feature the user's trust in each other. We saw that this explicit trust data performs similarly to the implicit user data. Often there are no trust scores available, and with our approach, we can resolve this problem by using assumed trust cores and combine them feeding the recommender.

CHAPTER 4

Conclusion

4.1 Reflection and Relations to the Studies

The discussed topics matches subjects concerned in the several courses lived through the academic career at TU Wien. The according curricula combine topics of economics and computer science and elaborate their common synergy[1]. The corresponding Bachelor curricular prepares with the basics. Basic network and graph theory is introduced in "Algebra and Discrete Mathematics". "Foundations of Program Construction" and "Fundamentals of Business Management" teaches basic programming and helps with basic concepts to implement models in script languages. Out of "Algorithms and Datastructures 1" runtimes of algorithms and efficient processing of large-scale data can be applied. "Statistics and Probability Theory" introduces basic statistical methods for evaluation while "Data analysis" brings sophisticated approaches including clustering. The curricula of the Master studies Business Informatics at the TU Wien in the version of October 2018 brings foundation content of the fields of Information Systems Engineering, Enterprise Engineering, Data Analytics, Management Science and Economic Modeling. Most crucial for the work is the thriving part of Data Analytics. In "Business Intelligence" Preprocessing and Big Data Technologies are part of the content. With Machine Learning methods which are categorizing data a big parallel is seen. "Model-based Decision Support" shows applications of model driven approaches. In the field of Enterprise Engineering and Economical Modeling "E-Commerce" shows a theoretical introduction to the foundations of network theory and it's implementation in state of art platforms. "Recommender Systems" shows advanced use as application for example in E-Commerce systems. "Computational Social Simulation" shows models with regards to the social aspect. Nevertheless, social skills are very much enforces in various lectures and groupwork and very beneficial to exchange experiences within colleagues. The approach of most

[1] TU Wien Faculty of Informatics Curricula - http://www.informatik.tuwien.ac.at/studium/angebot/studienplaene [Accessed 2020]

lectures in the Master program to work with scientific literature drives the scientific approach and helps to get a feeling of being precise, transparent in referencing and formalize every definition.

I understood for myself that both - education and knowledge - is necessary to become an expert in a particular field. At some point it gets very specific and the formal knowledge base shows formalities and is an acknowledgment of how far you can manage. It also helps to go into the width and depth of the topics. Education and specific training help in a purpose-oriented manner in activities that can be used to earn money in business. The basis for this is definitely in education. Human and economic aspects cannot and do not necessarily have to be part of university teaching. The very abstract presentation conducts a special way of thinking, it is more knowledge than education. It can serve as preparation to find your way in the future. My personal experience with my studies has shown that I learn above all the art to learn. On the other hand, the few attempts at grooming resulted in having to be ready to the point. Nevertheless, one may find a compromise that 50% is sufficient for a positive grade. But building half a program or half a car wouldn't work in the economy. In the end, however, I still often only learned for the exam, but getting through in my studies is proof of performance and durability. So somehow you learn for life!

For someone, it might be hard to tell what was the reason for one to join a specific field or why they ended up researching a specific topic. The progressive approach of technology is very fascinating. It is a mixture of creating something in the old sense of building, but thinking in a paradigm of a whole system, creating an overview and seeing the interplay of complex attributes deliver a very special impulse of being challenged by and sophisticated environment. Another satisfaction is to be capable of managing it. Once a system is stable, it is very satisfying not to change anything, though tipping points and resilience appear not as apparent as they might be. Automation and it's continuous improvement help humankind to connect information and support daily life in forms of robots and enable simplification and abstraction of global connections. With all that development, still every moment in time will have its challenges but efficiency in the end improves. This big vision is motivation and inspiration to me to proceed with what I am doing.

4.2 Summary

Our approach is a global structure analysis of a social network in order to find a ranking for behavioural representative user. The well-defined distance metric Jensen-Shannon Divergence is used to calculate the Centrality of nodes in a graph. The nodes were ranked and correlations to other methods are investigated but just very limited available. We have a unique approach. The Similarity scores are further used to enable mapping of a bipartite nodeset to find the connections in one nodeset. Evaluation showed good precision of the prediction. The bipartite setup enables an application for trust-based recommendation. We can extract implicit social data in order to source trust based

Recommendation system. This enables those kind of recommendations if no trust data is available.

When looking for for the average instead of a maximum in a bipartite network we faced several issues. With the influential node definition of being most similar to all others, there are still several possibilities on comparing for similarity. Also the attributes of a network influence the result massively. Not just looking at the node itself rather than to certain aspects of its neighbors brings the result of finding nodes which speak for the network and show the typical node as defined by our JSBiRank approach. The fact that just few correlation is visible to other metrics can conclude that further connection to existing methodology could be evaluated. Especially the field of bipartite networks brings a lot of potential for common metrics to be introduced.

The main contribution is found in the centrality measure for bipartite graphs. For the bipartite setup just very few metrics are available. The evaluation result of no correlation to other metrics show the unique approach. Jennsen Shannon is commonly used in other disciplines and now was combined with the basics in network theory. The centrality measure brings the capability for a novel bipartite to uniparite graph mapping approach. Also it was confirmed that social information can be extracted from rating information and be sourced to a trust-based recommendation system.

Also we are planning to submit the paper to Physica A: Statistical Mechanics and its Applications. The title of the paper is "JS-BiRank: Ranking Influential Nodes in Bipartite Networks via Jensen-Shannon Divergence" and it covers the unique method to gain the most representative users by the calculation with JSBiRank.

4.3 Future Work

In the professional computer technology journal c't [2] we found a nice visualization as in Figure 4.1 which is a bipartite graph. It summaries some finding in the cybercrime underground. The discovered languages hint to very specific malicious applications used. Different IoT-related topics on the right sight seem to be differently popular in the communities on the left side. Russia has strong tendencies for malicious applications. The interesting thing related to our work is that it is represented in a bipartite graph and further analysis with our centrality measure could be thought of.

Other applications often displayed in a bipartite matter are voting trend analysis performed after a selection. The representation of voters moving from one party to another is often visualized in a so called Sankey Digramm. This is a bipartite graph with the same nodesets on both sides connected by weighted edges. We currently find no solution to apply our method to those kind of graphs because common neighbor distribution is build on full common neighbor count and can not be split. Otherwise the distribution will look completely different, distribution of behind decimal point/fractions not possible.

[2] https://documents.trendmicro.com/assets/white_papers/wp-the-internet-of-things-in-the-cybercrime-underground.pdf

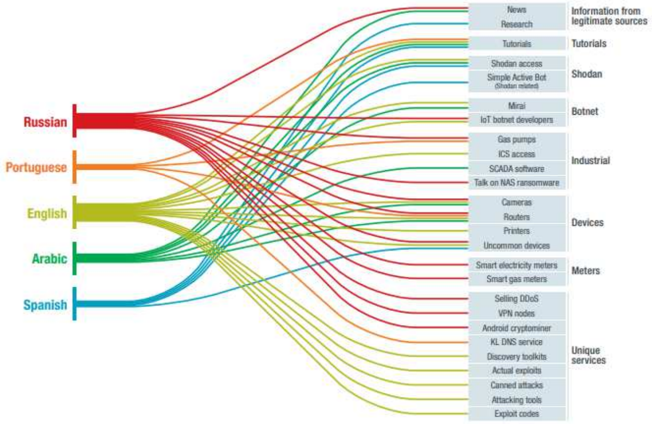


Figure 4.1: A bipartite-like representation of malicious topics discussion in five underground communities

Implementation in Python is possible with matplotlib.sankey. chord . The related graph required could be a similar, cyclic presentation. A version considering weighted edges could bring even more applications. Also some transformation rules to get from a weighted graph to a not weighted graph could be a first step.

An improvement in evaluation could be reached if bigger datasets are used. Improving performance or multi-threaded implementations could lead to further implications and more details in the results. Also memory optimizations in the used datastructures could benefit to deal with space complexity.
Also to try further applications with (trust-based) recommender system is a potential future task. The prominent tensorflow framework should bring sophisticated options.

As a future task investigating whether the result may be used for community detection in Group Recommender Systems by Mahyar et.al. in [MGKM+17]. Also different approaches with divergence measures like the mentioned Wasserstein Distance of Arjovsky et. al. in [ACB17] is worth further experiments.

Learning (as well as algorithms) could combine different methods. As Wen et.al. in [WTWJ18] distinguish between simple and complicated node indicators.

Algorithmics view has nice attributes for bipartite graphs: Vertex Cover, Dominating Set. It is to find minimal graphs out of existing ones. Either one node per edge or one node as neighbor. Those approaches would bring specialities in a bipartite graph. Also to build a tree out of a network could be interesting. So called hypergraphes connect multi nodes per edge.
Also trees are kind of bipartite graphs, every layer is a nodeset.

4.3.1 Further related Literature

We searching for ”representative influential node” we find some nice books which could be further concluded:

Plemenos and Miaoulis in [PM12] published a book existing of papers. Especially the paper by Doulamis et.al. is interesting as it discusses Inter-Social Influence following the concept of treating more than one network at once. So the network becomes a whole additional dimension of more networks where information can be concluded out of.

Tang and Li in [TL15] discuss social tie analysis and the Influence Maximization Selection working with user interaction. "Social Influence analysis" is tackled by topical Affinity propagation - a Clustering based learning algorithm using a "message passing" concept. Topical is meant as in topic and "hidden observation" are crucial for the algorithm.

Gunopulos et.al. in [GHMV11] in his book published a paper by Dan He which shows a simple, weighted and clustering based additive Influence Models comparing Authors in Industry and Academia. It is clustering based like Jaccard Similarity Index.

Cherifi et.al. in [CGM$^+$19] discuss classical versus hierarchical approaches and also mentioning inter-networks (additional network dimension). Unfortunately it is algorithm-based but nice figures show densing out the graph and try hierarchy forming.

A network dismantling problem and some random walk - algorithm are topics which we sometimes roam when researching.

Unfortunately some promising work could not be accessed. Peng et.al. survey influence analysis in social networks. Jannach et. al. in [JZFF10] provide extensive introduction to recommender systems in their compendium. Yang et.al. in [YLLL16] discuss Social Collaborative Filtering by Trust. Accessing the literature is usually not that big of an issue. Most were available through the TU Wien account. Springer Link and IEEE Explore were the main platforms which were used and we felt to be capable of having access to the most important part of the research field.

Conclusion reviewing books of various fields show that fields are connected. Complex Networks, Machine Learning, Data Mining and Computer Graphics have overlaps. Fundamental work can be used in various applications. Algorithmics are very strong for which this work can be seen as a baseline. Inter-network or inter influence, no matter on which level you wanna call it "inter" is a thing. it represents to add a dimension of not just discussing one network but take a look at several networks. with it comes to sum up, influence additive. Related to this additive models of summing up influence based on different properties can help to generate a result from different sources.

Our approach of detecting influential users in a social recommender system was extensively review, and strolling around literature showed that we just touched a tiny part of what is possible. However, we find to have contributed with new insights, listing what is also there and concluding with seeing a lot of potential in recommendation systems.

Bibliography

[A$^+$16] Charu C Aggarwal et al. *Recommender systems.* Springer, 2016.

[Abd07] Hervé Abdi. The kendall rank correlation coefficient. *Encyclopedia of Measurement and Statistics. Sage, Thousand Oaks, CA*, pages 508–510, 2007.

[ACB17] Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein gan. *arXiv preprint arXiv:1701.07875*, 2017.

[ALTP12] Nitin Agarwal, Huan Liu, Lei Tang, and S Yu Philip. Modeling blogger influence in a community. *Social Network Analysis and Mining*, 2(2):139–162, 2012.

[AN16] Samad Mohammad Aghdam and Nima Jafari Navimipour. Opinion leaders selection in the social networks based on trust relationships propagation. *Karbala International Journal of Modern Science*, 2(2):88–97, 2016.

[AP07] Javed A Aslam and Virgil Pavlu. Query hardness estimation using jensen-shannon divergence among multiple scoring functions. In *European conference on information retrieval*, pages 198–209. Springer, 2007.

[BDFG03] Jérémie Bouttier, Philippe Di Francesco, and Emmanuel Guitter. Geodesic distance in planar graphs. *Nuclear Physics B*, 663(3):535–567, 2003.

[BE97] Stephen P Borgatti and Martin G Everett. Network analysis of 2-mode data. *Social networks*, 19(3):243–269, 1997.

[BE06] Stephen P Borgatti and Martin G Everett. A graph-theoretic perspective on centrality. *Social networks*, 28(4):466–484, 2006.

[BH11] Stephen P Borgatti and Daniel S Halgin. On network theory. *Organization science*, 22(5):1168–1181, 2011.

[BHK98] John S Breese, David Heckerman, and Carl Kadie. Empirical analysis of predictive algorithms for collaborative filtering. In *Proceedings of the Fourteenth conference on Uncertainty in artificial intelligence*, pages 43–52. Morgan Kaufmann Publishers Inc., 1998.

[BJP17] Suman Banerjee, Mamata Jenamani, and Dilip Kumar Pratihar. Properties of a projected network of a bipartite network. In *Communication and Signal Processing (ICCSP), 2017 International Conference on*, pages 0143–0147. IEEE, 2017.

[BNQ19] Seyed Mojtaba Hosseini Bamakan, Ildar Nurgaliev, and Qiang Qu. Opinion leader detection: A methodological review. *Expert Systems with Applications*, 115:200–222, 2019.

[Bor05] Stephen P Borgatti. Centrality and network flow. *Social networks*, 27(1):55–71, 2005.

[BWH98] Richard Baskerville and A Trevor Wood-Harper. Diversity in information systems action research methods. *European Journal of information systems*, 7(2):90–107, 1998.

[CA10] Andrzej Cichocki and Shun-ichi Amari. Families of alpha-beta-and gamma-divergences: Flexible and robust measures of similarities. *Entropy*, 12(6):1532–1568, 2010.

[CA19] Umit Can and Bilal Alatas. A new direction in social network analysis: Online social network analysis problems and applications. *Physica A: Statistical Mechanics and its Applications*, page 122372, 2019.

[Cav96] Angèle LM Cavaye. Case study research: a multi-faceted research approach for is. *Information systems journal*, 6(3):227–242, 1996.

[CCH$^+$08] David Crandall, Dan Cosley, Daniel Huttenlocher, Jon Kleinberg, and Siddharth Suri. Feedback effects between similarity and social influence in online communities. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 160–168. ACM, 2008.

[CGM$^+$19] Hocine Cherifi, Sabrina Gaito, José Fernendo Mendes, Esteban Moro, and Luis Mateus Rocha. *Complex Networks and Their Applications VIII: Volume 1 Proceedings of the Eighth International Conference on Complex Networks and Their Applications COMPLEX NETWORKS 2019*, volume 881. Springer Nature, 2019.

[Cha07] Sung-Hyuk Cha. Comprehensive survey on distance/similarity measures between probability density functions. *City*, 1(2):1, 2007.

[CHK$^+$07] Shai Carmi, Shlomo Havlin, Scott Kirkpatrick, Yuval Shavitt, and Eran Shir. A model of internet topology using k-shell decomposition. *Proceedings of the National Academy of Sciences*, 104(27):11150–11154, 2007.

[Cou17] ACM US Public Policy Council. Statement on algorithmic transparency and accountability. *Commun. ACM*, 2017.

[DGG09] Allison Davis, Burleigh Bradford Gardner, and Mary R Gardner. *Deep South: A social anthropological study of caste and class*. Univ of South Carolina Press, 2009.

[DGKP14] Drmota, Gittenberger, Karigl, and Panholzer. *Mathematik fur Informatik*. Heldermann, 2014.

[DLP99] Ido Dagan, Lillian Lee, and Fernando CN Pereira. Similarity-based models of word cooccurrence probabilities. *Machine learning*, 34(1-3):43–69, 1999.

[DTDC15] Simone Daminelli, Josephine Maria Thomas, Claudio Durán, and Carlo Vittorio Cannistraci. Common neighbours and the local-community-paradigm for topological link prediction in bipartite networks. *New Journal of Physics*, 17(11):113037, 2015.

[Fli10] Uwe Flick. *Qualitative sozialforschung*. 2010.

[Fre77] Linton C Freeman. A set of measures of centrality based on betweenness. *Sociometry*, pages 35–41, 1977.

[GHMV11] Dimitrios Gunopulos, Thomas Hofmann, Donato Malerba, and Michalis Vazirgiannis. *Machine Learning and Knowledge Discovery in Databases, Part III: European Conference, ECML PKDD 2010, Athens, Greece, September 5-9, 2011, Proceedings*, volume 6913. Springer Science & Business Media, 2011.

[GL06] Jean-Loup Guillaume and Matthieu Latapy. Bipartite graphs as models of complex networks. *Physica A: Statistical Mechanics and its Applications*, 371(2):795–813, 2006.

[GZYS15] Guibing Guo, Jie Zhang, and Neil Yorke-Smith. Trustsvd: Collaborative filtering with both the explicit and implicit influence of user trust and of item ratings. In *Twenty-Ninth AAAI Conference on Artificial Intelligence*, 2015.

[HMPR04] Alan R Hevner, Salvatore T March, Jinsoo Park, and Sudha Ram. Design science in information systems research. *MIS quarterly*, pages 75–105, 2004.

[HSSC08] Aric Hagberg, Pieter Swart, and Daniel S Chult. Exploring network structure, dynamics, and function using networkx. Technical report, Los Alamos National Lab.(LANL), Los Alamos, NM (United States), 2008.

[Hun12] John K Hunter. An introduction to real analysis. *University of California at Davis, California*, 2012.

[JZFF10] Dietmar Jannach, Markus Zanker, Alexander Felfernig, and Gerhard Friedrich. *Recommender systems: an introduction*. Cambridge University Press, 2010.

[JZGG12] Dietmar Jannach, Markus Zanker, Mouzhi Ge, and Marian Gröning. Recommender systems in computer science and information systems–a landscape of research. In *International Conference on Electronic Commerce and Web Technologies*, pages 76–87. Springer, 2012.

[KC07] Barbara Kitchenham and Stuart Charters. Guidelines for performing systematic literature reviews in software engineering. 2007.

[KDLA] Jérôme Kunegis, Ernesto W De Luca, and Sahin Albayrak. The link prediction problem in bipartite networks. *Computational Intelligence for Knowledge-Based Systems Design*, page 380.

[Ken45] Maurice G Kendall. The treatment of ties in ranking problems. *Biometrika*, 33(3):239–251, 1945.

[KHMS11] Jan Kietzmann, Kristopher Hermkens, Ian McCarthy, and Bruno Silvestre. Social media? get serious! understanding the functional building blocks of social media. *Business Horizons*, 54(3):241 – 251, 2011.

[KL51] Solomon Kullback and Richard A Leibler. On information and sufficiency. *The annals of mathematical statistics*, 22(1):79–86, 1951.

[KM05] Bonnie Kaplan and Joseph A Maxwell. Qualitative research methods for evaluating computer information systems. In *Evaluating the organizational impact of healthcare information systems*, pages 30–55. Springer, 2005.

[Kor08] Yehuda Koren. Factorization meets the neighborhood: a multifaceted collaborative filtering model. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 426–434, 2008.

[Kor10] Yehuda Koren. Factor in the neighbors: Scalable and accurate collaborative filtering. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 4(1):1–24, 2010.

[Kun13] Jérôme Kunegis. Konect: the koblenz network collection. In *Proceedings of the 22nd International Conference on World Wide Web*, pages 1343–1350. ACM, 2013.

[KZ00] Stephen Kokoska and Daniel Zwillinger. *CRC standard probability and statistics tables and formulae.* Crc Press, 2000.

[Lan88] Thomas K Landauer. Research methods in human-computer interaction. *Handbook of human-computer interaction*, pages 905–928, 1988.

[Law14] Glenn Lawyer. Understanding the spreading power of all nodes in a network: a continuous-time perspective. *arXiv preprint arXiv:1405.6707*, 2014.

[LC13] Xin Li and Hsinchun Chen. Recommendation as link prediction in bipartite graphs: A graph kernel-based machine learning approach. *Decision Support Systems*, 54(2):880–890, 2013.

[Lee99] Lillian Lee. Measures of distributional similarity. In *Proceedings of the 37th annual meeting of the Association for Computational Linguistics on Computational Linguistics*, pages 25–32. Association for Computational Linguistics, 1999.

[Lee01] Lillian Lee. On the effectiveness of the skew divergence for statistical language analysis. In *AISTATS*. Citeseer, 2001.

[LF06] Jure Leskovec and Christos Faloutsos. Sampling from large graphs. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 631–636. ACM, 2006.

[LGH05] Pedro G Lind, Marta C Gonzalez, and Hans J Herrmann. Cycles and clustering in bipartite networks. *Physical review E*, 72(5):056127, 2005.

[LK14] Jure Leskovec and Andrej Krevl. SNAP Datasets: Stanford large network dataset collection. `http://snap.stanford.edu/data`, June 2014.

[LMB$^+$08] PW Lamberti, AP Majtey, A Borras, Montserrat Casas, and A Plastino. Metric character of the quantum jensen-shannon divergence. *Physical Review A*, 77(5):052311, 2008.

[LMDV08] Matthieu Latapy, Clémence Magnien, and Nathalie Del Vecchio. Basic notions for the analysis of large two-mode networks. *Social networks*, 30(1):31–48, 2008.

[LNK04] D Liben-Nowell and J Kleinberg. The link prediction problem for social networks, cikm 2003, 2004.

[LR14] Jessica Liebig and Asha Rao. Identifying influential nodes in bipartite networks using the clustering coefficient. *arXiv preprint arXiv:1406.5814*, 2014.

[LZLC14] Qian Li, Tao Zhou, Linyuan Lü, and Duanbing Chen. Identifying influential spreaders by weighted leaderrank. *Physica A: Statistical Mechanics and its Applications*, 404:47–55, 2014.

[MAA08] Marcelo Maia, Jussara Almeida, and Virgílio Almeida. Identifying user behavior in online social networks. In *Proceedings of the 1st workshop on Social network systems*, pages 1–6. ACM, 2008.

[MBC16] Víctor Martínez, Fernando Berzal, and Juan-Carlos Cubero. A survey of link prediction in complex networks. *ACM Computing Surveys (CSUR)*, 49(4):1–33, 2016.

[MCM+17] Fernando Martín, Juan Carballeira, Luis Moreno, Santiago Garrido, and Pavel González. Using the jensen-shannon, density power, and itakura-saito divergences to implement an evolutionary-based global localization filter for mobile robots. *IEEE Access*, 5:13922–13940, 2017.

[MGKM+17] Hamidreza Mahyar, Elahe Ghalebi K, S Mojde Morshedi, Saina Khalili, Radu Grosu, and Ali Movaghar. Centrality-based group formation in group recommender systems. In *Proceedings of the 26th International Conference on World Wide Web Companion*, pages 1187–1196. International World Wide Web Conferences Steering Committee, 2017.

[MMRYT17] Sarit Moldovan, Eitan Muller, Yossi Richter, and Elad Yom-Tov. Opinion leadership in small groups. *International Journal of Research in Marketing*, 34(2):536–552, 2017.

[MX18] Chengying Mao and Weisong Xiao. A comprehensive algorithm for evaluating node influences in social networks based on preference analysis and random walk. *Complexity*, 2018, 2018.

[MYSTM05] Prem Melville, Stewart M Yang, Maytal Saar-Tsechansky, and Raymond Mooney. Active learning for probability estimation using jensen-shannon divergence. In *European conference on machine learning*, pages 268–279. Springer, 2005.

[NDD20] Amrita Namtirtha, Animesh Dutta, and Biswanath Dutta. Weighted kshell degree neighborhood: A new method for identifying the influential spreaders from a variety of complex network connectivity structures. *Expert Systems with Applications*, 139:112859, 2020.

[OAS10] Tore Opsahl, Filip Agneessens, and John Skvoretz. Node centrality in weighted networks: Generalizing degree and shortest paths. *Social networks*, 32(3):245–251, 2010.

[Ops13] Tore Opsahl. Triadic closure in two-mode networks: Redefining the global and local clustering coefficients. *Social Networks*, 35(2):159–167, 2013.

[OTB01] SWFO Omta, Jacques Trienekens, and George Beers. Chain and network science: A research framework. *Journal on Chain and Network Science*, 1(1):1–6, 2001.

[PBMW99] Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. The pagerank citation ranking: Bringing order to the web. Technical report, Stanford InfoLab, 1999.

[PDK13] Manos Papagelis, Gautam Das, and Nick Koudas. Sampling online social networks. *IEEE Transactions on knowledge and data engineering*, 25(3):662–676, 2013.

[PLW14] Carolyn Parkinson, Shari Liu, and Thalia Wheatley. A common cortical metric for spatial, temporal, and social distance. *Journal of Neuroscience*, 34(5):1979–1987, 2014.

[PM12] Dimitri Plemenos and Georgios Miaoulis. *Intelligent computer graphics 2011*, volume 441. Springer, 2012.

[PRTV12] Ken Peffers, Marcus Rothenberger, Tuure Tuunanen, and Reza Vaezi. Design science research evaluation. In *International Conference on Design Science Research in Information Systems*, pages 398–410. Springer, 2012.

[RA14] Ryan A Rossi and Nesreen K Ahmed. Role discovery in networks. *IEEE Transactions on Knowledge and Data Engineering*, 27(4):1112–1131, 2014.

[RA15] Ryan A. Rossi and Nesreen K. Ahmed. The network data repository with interactive graph analytics and visualization. In *AAAI*, 2015.

[RGV04] Venkataraman Ramesh, Robert L Glass, and Iris Vessey. Research in computer science: an empirical study. *Journal of systems and software*, 70(1-2):165–176, 2004.

[Rou00] Routledge(Firm). *Concise Routledge Encyclopedia of Philosophy*. Routledge, 2000.

[RRS11] Francesco Ricci, Lior Rokach, and Bracha Shapira. Introduction to recommender systems handbook. In *Recommender systems handbook*, pages 1–35. Springer, 2011.

[SAB18] N Sumith, B Annappa, and Swapan Bhattacharya. Influence maximization in large social networks: Heuristics, models and parameters. *Future Generation Computer Systems*, 89:777–790, 2018.

[SAM18] Chiman Salavati, Alireza Abdollahpouri, and Zhaleh Manbari. Bridgerank: A novel fast centrality measure based on local structure of the network. *Physica A: Statistical Mechanics and its Applications*, 496:635–653, 2018.

[SDW$^+$20] Jinfang Sheng, Jinying Dai, Bin Wang, Guihua Duan, Jun Long, Junkai Zhang, Kerong Guan, Sheng Hu, Long Chen, and Wanghao Guan. Identifying influential nodes in complex networks based on global and local structure. *Physica A: Statistical Mechanics and its Applications*, 541:123262, 2020.

[SGMJZ13] Arlei Silva, Sara Guimarães, Wagner Meira Jr, and Mohammed Zaki. Profilerank: finding relevant content and influential users based on information diffusion. In *Proceedings of the 7th Workshop on Social Network Mining and Analysis*, pages 1–9, 2013.

[ŠLAFŠ13] Mile Šikić, Alen Lančić, Nino Antulov-Fantulin, and Hrvoje Štefančić. Epidemic centrality—is there an underestimated epidemic impact of network peripheral nodes? *The European Physical Journal B*, 86(10):440, 2013.

[SMST18] Seyed-Vahid Sanei-Mehri, Ahmet Erdem Sariyuce, and Srikanta Tirthapura. Butterfly counting in bipartite networks. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 2150–2159. ACM, 2018.

[SNZ17] Amir Sheikhahmadi, Mohammad Ali Nematbakhsh, and Ahmad Zareie. Identification of influential users by neighbors in online social networks. *Physica A: Statistical Mechanics and its Applications*, 486:517–534, 2017.

[SO11] Cathrine Seierstad and Tore Opsahl. For the few not the many? the effects of affirmative action on presence, prominence, and social capital of women directors in norway. *Scandinavian Journal of Management*, 27(1):44–54, 2011.

[SP18] Ahmet Erdem Sarıyüce and Ali Pinar. Peeling bipartite networks for dense subgraph discovery. In *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining*, pages 504–512. ACM, 2018.

[SSKB19] Shashank Sheshar Singh, Kuldeep Singh, Ajay Kumar, and Bhaskar Biswas. Mim2: multiple influence maximization across multiple social networks. *Physica A: Statistical Mechanics and its Applications*, 526:120902, 2019.

[Str01] Steven H Strogatz. Exploring complex networks. *nature*, 410(6825):268, 2001.

[TC08] Bruno Augusto Nassif Travençolo and L da F Costa. Accessibility in complex networks. *Physics Letters A*, 373(1):89–95, 2008.

[Ter17] Vagan Terziyan. Social distance metric: from coordinates to neighborhoods. *International Journal of Geographical Information Science*, 31(12):2401–2426, 2017.

[TL15] Jie Tang and Juanzi Li. Semantic mining of social networks. *Synthesis Lectures on the Semantic Web: Theory and Technology*, 5(2):1–205, 2015.

[TMF$^+$17a] Seyed Mohammad Taheri, Hamidreza Mahyar, Mohammad Firouzi, Elahe Ghalebi, Radu Grosu, and Ali Movaghar. Hellrank: a hellinger-based centrality measure for bipartite social networks. *Social Network Analysis and Mining*, 7(1):22, 2017.

[TMF$^+$17b] Seyed Mohammad Taheri, Hamidreza Mahyar, Mohammad Firouzi, Elahe Ghalebi K, Radu Grosu, and Ali Movaghar. Extracting implicit social relation for social recommendation techniques in user rating prediction.

In *Proceedings of the 26th International Conference on World Wide Web Companion*, pages 1343–1351, 2017.

[TMML09] John Tang, Mirco Musolesi, Cecilia Mascolo, and Vito Latora. Temporal distance metrics for social network analysis. In *Proceedings of the 2nd ACM workshop on Online social networks*, pages 31–36, 2009.

[TY12] Xuning Tang and Christopher C Yang. Ranking user influence in healthcare social media. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 3(4):73, 2012.

[VBC12] Matheus P Viana, Joao LB Batista, and Luciano da F Costa. Effective number of accessed nodes in complex networks. *Physical Review E*, 85(3):036105, 2012.

[WDN02] Duncan J Watts, Peter Sheridan Dodds, and Mark EJ Newman. Identity and search in social networks. *science*, 296(5571):1302–1305, 2002.

[WHZH11] Pengcheng Wu, Steven Chu-Hong Hoi, Peilin Zhao, and Ying He. Mining social images with distance metric learning for automated image tagging. In *Proceedings of the fourth ACM international conference on Web search and data mining*, pages 197–206, 2011.

[WPH$^+$18] Hao Wei, Zhisong Pan, Guyu Hu, Liangliang Zhang, Haimin Yang, Xin Li, and Xingyu Zhou. Identifying influential nodes based on network representation learning in complex networks. *PloS one*, 13(7), 2018.

[WSZ$^+$19] Jiehua Wu, Jing Shen, Bei Zhou, Xiayan Zhang, and Bohuai Huang. General link prediction with influential node identification. *Physica A: Statistical Mechanics and its Applications*, 523:996–1007, 2019.

[WTWJ18] Xiangxi Wen, Congliang Tu, Minggong Wu, and Xurui Jiang. Fast ranking nodes importance in complex networks based on ls-svm method. *Physica A: Statistical Mechanics and its Applications*, 506:11–23, 2018.

[WZXD16] Zhixiao Wang, Ya Zhao, Jingke Xi, and Changjiang Du. Fast ranking influential nodes in complex networks using a k-shell iteration factor. *Physica A: Statistical Mechanics and its Applications*, 461:171–181, 2016.

[YLLL16] Bo Yang, Yu Lei, Jiming Liu, and Wenjie Li. Social collaborative filtering by trust. *IEEE transactions on pattern analysis and machine intelligence*, 39(8):1633–1647, 2016.

[ZGVGA07] Xiaojin Zhu, Andrew B Goldberg, Jurgen Van Gael, and David Andrzejewski. Improving diversity in ranking using absorbing random walks. In *HLT-NAACL*, pages 97–104, 2007.

[ZH15] Liang Zhou and Charles D Hansen. A survey of colormaps in visualization. *IEEE transactions on visualization and computer graphics*, 22(8):2051–2069, 2015.

[ZHT+15] Xiang-Yu Zhao, Bin Huang, Ming Tang, Hai-Feng Zhang, and Duan-Bing Chen. Identifying effective multiple spreaders by coloring complex networks. *EPL (Europhysics Letters)*, 108(6):68005, 2015.

[ZYD+19] Wei Zhang, Jing Yang, Xiao-yu Ding, Xiao-mei Zou, Hong-yu Han, and Qing-chao Zhao. Groups make nodes powerful: Identifying influential nodes in social networks based on social conformity theory and community features. *Expert Systems with Applications*, 125:249–258, 2019.