FACULTY OF ENGINEERING
AND ARCHITECTURE

Opinion Dynamics and Link Prediction in Networks: Models, Data Efficiency, and Robustness

Xi Chen

Doctoral dissertation submitted to obtain the academic degree of
Doctor of Computer Science Engineering

Supervisors

Prof. Tijl De Bie, PhD - Prof. Jefrey Lijffijt, PhD

Department of Electronics and Information Systems
Faculty of Engineering and Architecture, Ghent University

February 2022

GHENT
UNIVERSITY

FACULTY OF ENGINEERING
AND ARCHITECTURE

Opinion Dynamics and Link Prediction in Networks: Models, Data Efficiency, and Robustness

Xi Chen

Doctoral dissertation submitted to obtain the academic degree of
Doctor of Computer Science Engineering

Supervisors
Prof. Tijl De Bie, PhD - Prof. Jefrey Lijffijt, PhD

Department of Electronics and Information Systems
Faculty of Engineering and Architecture, Ghent University

February 2022

GHENT
UNIVERSITY

ISBN 978-94-6355-571-5
NUR 983, 984
Wettelijk depot: D/2022/10.500/12

Members of the Examination Board

Chair

Prof. Gert De Cooman, PhD, Ghent University

Other members entitled to vote

Prof. Thomas Demeester, PhD, Ghent University
Prof. Femke Ongenae, PhD, Ghent University
Prof. Jie Tang, PhD, Tsinghua University, China
Prof. Panayiotis Tsaparas, PhD, University of Ioannina, Greece
Prof. Sofie Van Hoecke, PhD, Ghent University

Supervisors

Prof. Tijl De Bie, PhD, Ghent University
Prof. Jefrey Lijffijt, PhD, Ghent University

Acknowledgements

First of all, I would like to thank my supervisors, Tijl De Bie and Jefrey Lijffijt, for all the help, support, and mentoring throughout my PhD years. Tijl guided me into the world of scientific research with excellent supervision along the way of my exploration. He always asks thought-provoking questions, provides insightful advice, and gives us his timely support. More importantly, the advice I got from him on being patient, relaxing to focus, and valuing all the experiences has benefited me a lot. I still and will always remember Jefrey sharing his view on the goal of doctoral studies during our BrainBuilding in Durbuy, which is to *grow up as a person*. It has been guiding me. And Jefrey's passion for the visualizations will always encourage me to treat every figure carefully. I have learned so much from both of you, and I feel lucky and honored to be one of your students.

Apart from the fact that Bo is an extremely reliable collaborator, the help I received from him is nothing short of life-changing. The books you recommended or sent as gifts, the weightlifting classes you gave, the research articles or news you shared, the meditation approach you introduced, and a lot more have totally changed my lifestyle to be much more research-friendly. Thank you for all the encouragement!

I also want to send warm thanks to the examination committee members for taking their valuable time to actively participate in my graduation procedures, read the thesis, provide helpful comments, and ask insightful questions. Special thanks go to Panayiotis and Jie. It was an absolute pleasure to have collaborated with Panayiotis, and I also thank Jie for offering me the valuable opportunity (during the pandemic) to visit his research group in the summer of 2021. I appreciate the help from Maarten and Jefrey on the Dutch summary of this thesis.

It has been a pleasure to be part of the AIDA group. I want to thank all the group members: Ahmad, Alex, Bo, Dieter, Edith, Lemon, Len, Maarten, Nan, Paolo, Raphaël, Sander, and Yoosof, as well as the support staff in IDLab. Without you, my life in Ghent cannot be so memorable. Special thanks to my old friend Lemon, not only for introducing me to Tijl but also for being a true friend and for those good times since our master years.

I am grateful that my research projects have received funding from the European Research Council under the European Union's Seventh Framework Programme (FP7/2007-2013) (ERC Grant Agreement no. 615517), and under the

European Union's Horizon 2020 research and innovation programme (ERC Grant Agreement no. 963924), from the Flemish Government under the "Onderzoeksprogramma Artificiële Intelligentie (AI) Vlaanderen" programme, from the FWO (project no. G091017N, G0F9816N, 3G042220), from the European Union's Horizon 2020 research and innovation programme and the FWO under the Marie Sklodowska-Curie Grant Agreement no. 665501.

Last but not least, I sincerely thank my parents, my family members, my friends, as well as myself.

Ghent, January 2022
Xi Chen

Samenvatting
– Summary in Dutch –

Naarmate onze wereld meer en meer geconnecteerd en online wordt, worden mensen blootgesteld aan massa-informatie in verschillende vormen, met inbegrip van (maar niet beperkt tot) teksten, afbeeldingen, geluid en video, vanuit verschillende bronnen, zoals telefoons, tv's en laptops. Als gevolg daarvan heeft de gedigitaliseerde omgeving een sterke invloed op de opinie van de mensen op een manier die sterk verschilt van vroeger, toen de verspreiding van informatie traag en geografisch beperkt was. Opinievorming is al lang een onderzoeksonderwerp in de sociale wetenschappen. Het heeft de laatste jaren ook de aandacht gewekt van de computerwetenschappen vanwege de populariteit van (online) sociale netwerken.

De studie van de opinievorming richt zich op het modelleren van hoe individuen hun opinie bijwerken en uiteindelijk vormen als gevolg van interne gedachten en externe invloeden. Sociale media platformen zoals Twitter en Facebook hebben de uitwisseling van standpunten online vergemakkelijkt, waardoor er een toename kwam aan de hoeveelheid en soorten invloeden uit de omgeving op de opinievorming. Controverse, onenigheid, conflict, polarisatie, en opiniedivergentie in sociale netwerken zijn het onderwerp geweest van veel recent onderzoek. Onderzoekers hebben bestudeerd hoe deze concepten kunnen worden gekwantificeerd en vervolgens geoptimaliseerd door het beïnvloeden van de opinies van een klein aantal mensen, bv. *influencers* op sociale media, of het bewerken van de connectiviteit van het netwerk, d.w.z. de blootstelling van mensen aan verschillende informatiebronnen. Dit betekent dat met strategische interventies marketing niet alleen viraal kan gaan, maar ook dat het risico dat de publieke opinie wordt gemanipuleerd ook groter wordt. Deze strategieën zijn niet ongewoon; het is dus belangrijk om een meer expliciet inzicht te hebben van de bijbehorende processen.

Opinievormingsmodellen vormen het fundament van dit onderzoek. Zij definiëren regels voor het modelleren van hoe mensen hun opinies bijwerken door interacties met anderen. Bestaande modellen zijn verouderd, omdat ze niet in staat zijn rekening te houden met sociale verschijnselen in de realiteit en ze hebben problemen die kunnen resulteren in ongewenst gedrag. Er zijn dus inspanningen nodig om de problemen aan te pakken en de modellen up-to-date te houden. Ondertussen is het verkrijgen van de opinie van mensen als reële waarden al lang

een zeer uitdagende taak gebleven. Zonder die opinies te kennen, kunnen we nog steeds begrippen als controverse, conflict, en polarisatie kwantificeren en optimaliseren door alleen gebruik te maken van de kanalen voor informatie-uitwisseling, namelijk de connecties in sociale netwerken? We zullen deze vraag later in deze thesis beantwoorden.

Connecties in sociale netwerken zijn van cruciaal belang voor de opinievorming aangezien zij blootstelling aan informatie vertegenwoordigen, wat betekent dat zij bepalen hoe opinies worden beïnvloed, namelijk door wie en in welke mate. Die connecties zijn echter niet altijd bekend, omdat we in een wereld leven waarin we bijna alles slechts gedeeltelijk waarnemen. Niet alleen in sociale netwerken, maar ook in andere netwerken, zoals proteïne-proteïne interactienetwerken, netwerken van neuronen, en netwerken van consumenten en producten, worden connecties meestal maar gedeeltelijk waargenomen. Verschillende methoden zijn ontworpen om connecties te voorspellen in netwerken die momenteel ontbreken of die zich in de toekomst zullen vormen, wat in wezen een belangrijk netwerkprobleem is genaamd connectievoorspelling. Connectievoorspelling heeft een breder toepassingsgebied buiten de studie van de opinievorming. Veel problemen in de echte wereld kunnen worden geformaliseerd als het voorspellen van connecties in netwerken, bijvoorbeeld vriendschapssuggesties op Facebook, aanbevelingen in e-commerce, en de voorspelling van proteïne-proteïne interacties.

Huidige methoden voor connectievoorspelling gaan ervan uit dat alle connecties bekend zijn en beschouwen alle niet geziene connecties als niet verbonden, terwijl voor veel knoop-paren het niet bekend is of de twee knopen verbonden zijn. Bovendien zijn veel methoden voor connectievoorspelling niet transparant, waardoor hun robuustheid tegen vijandelijke aanvallen de laatste tijd zorgen baart. Daarom is het cruciaal om te focussen op de data-efficiëntie en betrouwbaarheid van de methoden voor connectievoorspelling, wat betekent dat ze competitieve prestaties moeten leveren met minder data, in minder tijd en terwijl ze robuust moeten zijn tegen aanvallen.

Deze thesis bevat vijf bijdragen over opiniedynamica en methoden voor connectievoorspelling om aan de eerder genoemde behoeften te voldoen. De eerste twee hoofdstukken behandelen opinievormingsmodellen. We hebben een bekend opinievormingsmodel uitgebreid uit om rekening te houden met twee sociale fenomenen uit de realiteit in Hoofdstuk 2 en hebben een probleem opgelost dat bestond in een populaire variant in Hoofdstuk 3. In Hoofdstuk 4 kwantificeerden en minimaliseerden we het risico van conflicten in sociale netwerken zonder specifieke opinies te kennen. Dat betekent dat we ons puur richtten op die connecties in het netwerk die bepalen hoe opinies worden bijgewerkt. In de laatste twee hoofdstukken hebben we gewerkt aan het verbeteren van een specifiek type van connectievoorspelling gebaseerd op netwerk inbedding. In het bijzonder hebben we enerzijds de aanpak van connectievoorspelling data-efficiënter gemaakt door gebruik te maken van actief leren in Hoofdstuk 5 en anderzijds betrouwbaarder door de robuustheid tegen aanvallen te onderzoeken in Hoofdstuk 6.

Opinievormingsmodellen. Een opinievormingsmodel definieert de regels voor

het bijwerken van opinies, bijvoorbeeld hoe iemands opinie wiskundig kan worden berekend als een functie van de eigen opinie en die van vrienden. Twee essentiële elementen voor deze modellen zijn: (i) de opinies van de individuen en (ii) de connecties in het sociaal netwerk, die vriendschappen vertegenwoordigen. Opinievormingsmodellen worden bepaald door vele factoren, waaronder de bijwerkingsregels, de soorten opinies, en de voorwaarden voor de interacties. Wellicht één van de meest bekende opinievormingsmodellen is het DeGroot model, dat het gemiddelde blijft nemen van de opinies van individuen met hun vrienden tot een convergentie. Ook de variant van het DeGroot model voorgesteld door Friedkin en Johnsen is populair. Deze maakt een onderscheid tussen interne en geuite opinies en zullen wij verder het FJ-model noemen.

In Hoofdstuk 2 introduceren wij het BEBA model, dat gebaseerd is op het DeGroot model, om twee bekende sociale fenomenen mee te nemen in opinievormingsmodellen. Het eerste fenomeen is het Backfire Effect: de nijging dat iemand zich bij de blootstelling aan een tegenovergestelde mening juist verankert in zijn eigen mening. Wat er voor zorgt dat hun opinie extremer wordt in plaats van modereert. Het tweede is Biased Assimilation: de nijging van individuen om opinies vergelijkbaar met die van henzelf sneller over te nemen. Zover ons bekend is dit het eerste DeGroot-type model dat het Backfire Effect mee neemt. We gaven een grondige theoretische beschrijving en een empirische analyse van het voorgestelde model. Hieruit bleek dat er intuïtieve voorwaarden zijn voor het onstaan van polarisatie en consensus en voor de eigenschappen van de resulterende opinies.

In Hoofdstuk 3 pakken we vervolgens een probleem van het FJ model aan door een normalisatie te introduceren. Gemotiveerd door een observatie in het FJ-model hebben wij een genormaliseerd Friedkin en Johnsen-model voorgesteld, namelijk het NFJ-model. Voor elk individu gaat het FJ-model uit van zowel een onveranderlijke interne opinie als een geuite opinie die kan verschillen van maar meer in overeenstemming is met wat de vrienden zeggen. In zijn elementaire vorm is het FJ-model is niet echt realistisch omdat het aangeeft dat hoe meer vrienden men heeft, hoe minder haar interne opinie van belang is in haar geuite opinie. Om dit probleem aan te pakken, hebben wij een wijziging van het FJ-model voorgesteld, namelijk het NFJ-model, die iemands externe invloed van vrienden normaliseert en het gewicht op haar interne opinie constant houdt. Vervolgens onderzochten we de gevolgen van de normalisatie, zowel theoretisch als empirisch.

Risico van Conflict. Het verkrijgen van de reëel-waardige opinies is al lang een uitdagende taak, en opinies over verschillende onderwerpen komen niet noodzakelijk overeen met verschillende netwerkstructuren (we bespreken namelijk verschillende onderwerpen met dezelfde vrienden). In Hoofdstuk 4 onderzoeken we opinieverschillen in sociale netwerken zonder de eigenlijke opinies te kennen, namelijk door enkel de netwerktopologie te gebruiken. We kwantificeren en optimaliseren het risico van conflict voor zowel de gemiddelde als de worst-case opinievector over alle mogelijke distributies van opinies op het sociale netwerk. Voor sommige maten van conflict (de divergentie van opinies) zijn de optimalisatieproblemen voor de risico's niet-convex, wat resulteert in vele lokale minima. Wij

hebben een theoretische en empirische analyse gemaakt van de aard van sommige van deze lokale minima en toonden aan hoe zij gerelateerd zijn aan bestaande organisatiestructuren. Het risico van conflicten hangt louter af van de banden in sociale netwerken, die bepalen hoe informatie die van invloed is op de opinievorming zich verspreidt.

Connectievoorspelling. De connecties in sociale netwerken, maar ook die in andere netwerken, zoals biologische netwerken, zijn niet altijd bekend en moeten dus worden voorspeld. Naast de eerder genoemde opiniedynamiek, is een ander belangrijk onderwerp van deze thesis connectievoorspelling. Dit probleem wordt steeds vaker opgelost door netwerkinbeddingsmethoden die recent zijn voorgesteld vanwege hun state-of-the-art prestaties. Met de nadruk op deze klasse van methoden voor connectievoorspelling met behulp van netwerkinbedding, bieden we data-efficiënte en robuuste oplossingen. In Hoofdstuk 5 passen we actief leren toe om de data-efficiëntie te verbeteren voor connectievoorspelling op gedeeltelijk geobserveerde netwerken.

We richten ons op het verbeteren van de data-efficiëntie van de methoden voor connectievoorspelling die voor ons van bijzonder belang zijn, namelijk diegene welke gebruik maken van netwerkinbedding, waarmee het probleem kan worden opgelost met alleen de waargenomen netwerkinformatie. Voor het niet-waargenomen deel kan de connectiviteit tussen twee knooppunten vaak worden opgevraagd, hoewel tegen een kostprijs, wat bekend staat als actief leren. Om de gegevensefficiëntie te verbeteren, hebben wij *actief leren* toegepast om het ALPINE (Active Link Prediction usIng Network Embedding) raamwerk te ontwikkelen. ALPINE identificeert de meest informatieve connectie-status die naar schatting de meest significante verbetering van de nauwkeurigheid van connectievoorspelling zal veroorzaken als deze wordt opgevraagd. Wij hebben ook verschillende opvragingsstrategieën voor ALPINE voorgesteld, aangezien de opvragingen moeten worden gedaan met de nodige overweging van de kosten van de opvraging. Onze empirische resultaten op data uit de echte wereld toonden aan dat ALPINE schaalbaar was en de nauwkeurigheid van de connectievoorspelling verbeterde met veel minder opvragingen dan passief leren. We analyseerden ook de relatieve verdiensten van de strategieën, wat inzichtelijke richtlijnen oplevert voor mensen in de praktijk.

In Hoofdstuk 6 bestudeerden we de robuustheid tegen aanvallen voor connectievoorspelling om de betrouwbaarheid van de methode te verbeteren. Hoewel connectievoorspelling met behulp van methoden voor netwerkinbedding state-of-the-art prestaties bereikt, veroorzaakt het gebrek aan transparantie bezorgdheid over de robuustheid van het model tegen aanvallen. Men kan zich afvragen of kleine vijandelijke wijzigingen van het netwerk een significante invloed zullen hebben op de connectievoorspellingen bij het gebruik van een netwerkinbeddingsmodel, wat een onvoldoende onderzocht probleem is inzake robuustheid bij connectievoorspelling. Dit hoofdstuk draagt bij tot het vullen van deze leemte. In het bijzonder, gegeven een probabilistisch netwerkinbeddingsmodel en een netwerk, meten we de gevoeligheid van de connectievoorspellingen van het model voor

kleine verstoringen van de netwerkstructuur. Onze aanpak maakt het dus mogelijk om de meest kwetsbare connecties en non-connecties voor dergelijke verstoringen te identificeren. Wij analyseerden verder de kenmerken van de meest en minst gevoelige verstoringen en bevestigden vervolgens empirisch dat onze aanpak met succes de meest kwetsbare connecties en non-connecties op een tijdsefficiënte manier identificeert dankzij een effectieve benadering.

Hopelijk kunnen onze bijdragen en overeenkomstige resultaten in deze thesis relevant onderzoek helpen, niet alleen voor computationele sociale wetenschappen, maar ook voor machinaal leren en kunstmatige intelligentie. De opinievormingsmodellen van BEBA en NFJ zijn onze pogingen om te zoeken naar redelijke en up-to-date modellering van opiniedynamiek in sociale netwerken. De studie van het risico van conflicten is een voorbeeld van hoe computerwetenschappen kan helpen bij de reële sociale problemen van polarisatie en soortgelijke concepten zoals diversiteit van opinies. We hopen dat dit het licht kan werpen op vele andere mogelijkheden die bijdragen tot een meer diverse maar minder verdeelde wereld. Tenslotte is er nood aan data-efficiënte en robuuste methoden om via machinaal leren verbanden te voorspellen. In het algemeen zou het interessant zijn om te zien of, binnen een paar jaar, dit werk kan bijdragen aan het beantwoorden van de vraag hoe *machines opinies kunnen vormen*, als een verdere stap op het huidige *machinaal leren* en *redeneren* voor complexere kunstmatige intelligentie.

Summary

As our world gets increasingly networked and connected, it exposes people to mass information in various forms, including but not limited to text, image, audio, and video, coming from different sources, such as phones, TVs, and laptops. As a result, the digitized environment heavily influences people's opinions in a way that is very different from the old days when the spread of information was slow and geographically limited. Opinion formation has long been the research subject in social sciences. It has also attracted attention from the computer science community in recent years owing to the popularity of (online) social networks.

The study of opinion formation focuses on modeling how individuals update and eventually form their opinions due to internal thoughts and external influences. Social media platforms like Twitter and Facebook have facilitated the exchange of views online, which increases the amount and the types of environmental impacts on opinion formation. Controversy, disagreement, conflict, polarization, and opinion divergence in social networks have been the subject of much recent research. Researchers have studied how these concepts can be quantified and then optimized by influencing the opinions of a small number of people, e.g., the influencers, or editing the network's connectivity, i.e., people's exposures to different sources of information. This means that with strategic interventions, not only can the marketing get viral, but the risk of having manipulated public opinion also gets higher. These strategies are not uncommon; thus, it is vital to have a more explicit understanding of the corresponding processes.

Opinion formation models serve as the fundamental part of these studies. They define rules for modeling how people update their opinions through interactions with others. However, existing models are outdated as being unable to account for real-world social phenomena, and they suffer from issues that result in undesired behaviors. Thus, effort is required to keep the models up-to-date and address the issues. Meanwhile, obtaining people's opinions as real values has remained for a long time a very challenging task. Without knowing those opinions, can we still quantify and optimize concepts like controversy, conflict, and polarization with only the channels for information exchange, i.e., the links in social networks? We will answer this question later in this thesis.

Connections in social networks are crucial for opinion formation as they represent information exposures, meaning that they control how the opinions are in-

fluenced, i.e., by whom and to what extent. However, those connections are not always known since we live in a world where we observe almost everything only partially. Not only for social networks, links in other networks, e.g., protein-protein interaction networks, networks of neurons, and consumer-product networks, are usually partially observed. Several methods have been designed to predict links in networks that are currently missing or those that will form in the future, which is essentially a vital network problem called link prediction. Link prediction has a broader range of applications outside the study of opinion formation. Many real-world problems can be formalized as predicting links in networks, e.g., Facebook friendship suggestions, e-commerce recommendations, and the prediction of protein-protein interactions.

Current link prediction approaches assume that all connections are known and treat unobserved as unlinked status, while for many node pairs, it is not known if the two nodes are linked. Additionally, many link prediction methods lack transparency, so their robustness against adversarial attacks has been causing concern recently. Therefore, it is crucial to focus on the data efficiency and model reliability of the link prediction methods, which means achieving competitive performance with fewer data and in less time, and being robust against adversarial attacks.

This thesis includes five paper contributions on opinion dynamics and link prediction methods to address previously mentioned needs. The focus of Chapters 2 and 3 is opinion formation models. We extend a classic opinion formation model to account for two real-life social phenomena in Chapter 2 and address an issue of a popular variant of it in Chapter 3. In Chapter 4, we quantify and minimize the risk of conflict in social networks without knowing any specific opinions. That means we focus purely on the links in the network that control how opinions are updated. Then in Chapters 5 and 6, we work on improving a specific type of link prediction approach based on network embedding. More specifically, we propose to make the link prediction approach more data-efficient using active learning in Chapter 5 and more reliable through investigating its adversarial robustness in Chapter 6.

Opinion Formation Models. An opinion formation model defines the rule for opinion updating, e.g., how one's opinion can be computed mathematically as a function of its own and friends' opinions. Two essential elements for these models are: (i) the opinions of the individuals; (ii) the connections in the social network representing friendships. Opinion formation models are distinguished by many factors, including the updating rules, the types of opinions, and conditions on the interactions. Arguably, one of the most well-known opinion formation models is the DeGroot model, which keeps averaging individuals' opinions with their friends until a convergence. The DeGroot model's variant proposed by Friedkin and Johnsen that differentiates the internal and expressed opinions, which we further refer to as the FJ model, is also a popular choice of study.

In order to account for two known social phenomena, we propose a novel opinion formation model called BEBA by extending the DeGroot model in Chapter 2. The first phenomenon is the Backfire Effect: the cognitive bias that an opposite opinion may further entrench people in their stances, making their opinions more

extreme instead of moderating them. The second is Biased Assimilation: the tendency of individuals to adopt other similar opinions to their own. To the best of our knowledge, this is the first DeGroot-type opinion formation model that captures the Backfire Effect. We provide a thorough theoretical and empirical analysis of the proposed model, which reveals intuitive conditions for polarization and consensus to exist, as well as the properties of the resulting opinions.

Then in Chapter 3, we address an issue of the FJ model by introducing a normalization. Motivated by an observation in the FJ model, we propose a normalized Friedkin and Johnsen model, namely the NFJ model. For each individual, the FJ model assumes both an immutable internal opinion and an expressed opinion that may differ but is more in agreement with what the friends say. The FJ model in its elementary form might not be realistic in some scenarios because it indicates that the more friends one has, the less her internal opinion matters in her expressed opinion. To address this issue, we propose a modification of the FJ model, namely the NFJ model, that normalizes one's external influence from friends and keeps the weight on her internal opinion a constant. Then we investigate the consequences of the normalization, both theoretically and empirically.

Risk of Conflict. Getting the real-valued opinions has long been a challenging task, and opinions on different issues do not necessarily correspond to distinct network structures (i.e., we discuss various topics with the same friends). In Chapter 4, we investigate the opinion differences in social networks without knowing the actual opinions, meaning to use only the network topology. We quantify and optimize the risk of conflict for both the average- and the worst-case opinion vector over all possible distributions of opinions on the social network. For some measures of conflict (i.e., the opinion divergence), the optimization problems for the risks are non-convex, thus resulting in many local minima. We provide theoretical and empirical analysis on the nature of some of these local minima and show how they are related to existing organizational structures. The risk of conflict depends purely on the links in social networks, which control how information affecting opinion formation diffuses.

Link Prediction. The links in social networks, as well as those in other networks, such as biological networks, are not always known but need to be predicted. In addition to opinion dynamics, link prediction is the other important subject of this thesis. This problem is solved increasingly often by network embedding methods proposed recently due to their state-of-the-art performance. Focusing on this class of link prediction approaches, we provide data-efficient and robust solutions.

In Chapter 5, we apply active learning to improve data efficiency for link prediction on partially observed networks. We focus on improving the data efficiency of the link prediction methods of our particular interest, namely those using network embedding, which can solve the problem with only the observed network information. For the unobserved part, the connectivity between two nodes can often be queried, although at a cost, which is known as active learning. To improve data efficiency, we apply *active learning* to develop the ALPINE (Active Link Prediction usIng Network Embedding) framework. ALPINE identifies the

most informative link status estimated to cause the most significant improvement of the link prediction accuracy if queried. We also propose several query strategies for ALPINE as the queries must be made with due consideration, owing to the cost. Our empirical results on real-world data show that ALPINE is scalable and boost link prediction accuracy with far fewer queries than passive learning. We also analyze the relative merits of the strategies, providing insightful guidance for practitioners.

Lastly, in Chapter 6, we study the adversarial robustness for link prediction to improve the method's reliability. Although link prediction using network embedding methods achieves state-of-the-art performance, its lack of transparency causes concern about the model robustness against adversarial attacks. One could wonder if minor adversarial modifications of the network will significantly impact the link predictions when using a network embedding model. This is an insufficiently explored problem on robustness concerning link prediction. Our paper contributes to filling this gap. More specifically, given a probabilistic network embedding model and a network, we measure the sensitivity of the model's link predictions to small perturbations on the network structure. Thus, our approach allows one to identify the most vulnerable links and non-links to such perturbations. We further analyze the characteristics of the most and least sensitive perturbations and have empirically confirmed that our approach successfully identified the most vulnerable links and non-links in a time-efficient manner thanks to an effective approximation.

Hopefully, our contributions and corresponding results in this thesis can help relevant research in the community, not only for computational social science but also for machine learning and artificial intelligence. The opinion formation models of BEBA and NFJ are our attempts to seek up-to-date and reasonable modeling of opinion dynamics in social networks. The study on the risk of conflict is an example of how computer science can help with the real social problems of polarization and similar concepts like opinion diversity. We hope it could shed light on many other possibilities contributing to a more diverse but less divided world. Finally, the work on link prediction calls for attention to building data-efficient and robust machine learning methods. More broadly, it would be interesting to see if, in a few years, our work can contribute to *machine opinion formation*, either as a way to help understand human opinion formation or as a further step to the current *machine learning* and *reasoning* for more complex artificial intelligence.

Table of Contents

Acknowledgements **i**

Samenvatting **iii**

Summary **ix**

Acronyms **xvii**

1 Introduction **1**
1.1 Context and Motivation 1
1.1.1 Opinion Formation Models 2
1.1.2 Risk of Conflict in Social Networks 3
1.1.3 Link Prediction 4
1.2 Research Contributions 6
1.3 Publications 11

2 Opinion Dynamics with Backfire Effect and Biased Assimilation **13**
2.1 Introduction 14
2.2 Related Work 15
2.3 Model Definition 16
2.3.1 Preliminaries and Background 16
2.3.2 The BEBA Model 17
2.3.3 Comparison between BEBA and BOF 18
2.4 Theoretical Analysis 21
2.4.1 A Single Agent in a Fixed Environment 21
2.4.2 Polarization and Consensus for All Nodes in a Network . 22
2.5 Experimental Analysis 24
2.5.1 The Influence of the Entrenchment Parameter β 25
2.5.2 The Influence of the Initial Opinions $\mathbf{y}(0)$ 26
2.5.3 The Influence of the Network G 27
2.5.4 Real-world Dataset Analysis 30
2.5.5 Opinion Manipulation under BEBA. 30
2.6 Conclusion and Future work 30
Appendix 2.A Proof of Theorem 1 32
Appendix 2.B Proof of Theorem 2 35
Appendix 2.C Supporting Figure 38

3 The Normalized Friedkin-Johnsen Model **39**
3.1 Introduction and Motivation . . . 40
3.2 The Normalized Friedkin-Johnsen Model, and a Theoretical Analysis 41
3.2.1 The Normalized Friedkin-Johnsen model . . . 41
3.2.2 Implications of the Normalization on the Quantification of Conflict in Networks . . . 43
3.3 Discussion and Experiments . . . 45
3.3.1 Opinion Formation . . . 46
3.3.2 Quantifying Conflict . . . 47

4 Quantifying and Minimizing Risk of Conflict in Social Networks **49**
4.1 Introduction and Motivation . . . 50
4.2 Opinion Formation Models . . . 52
4.3 Conflict and Conflict Risk . . . 53
4.3.1 Conflict Measures . . . 53
4.3.2 A Conservation Law of Conflict . . . 55
4.3.3 Conflict Risk of a Network . . . 57
4.4 Minimizing the Conflict Risk . . . 58
4.4.1 Algorithms . . . 58
4.4.2 Local Optima of the ACR for Different Risk Measures . . 61
4.5 Empirical Evaluation . . . 62
4.5.1 Datasets . . . 62
4.5.2 Experimental Findings . . . 63
4.6 Related Work . . . 67
4.7 Conclusions and Further work . . . 69

5 Active Link Prediction using Network Embedding **71**
5.1 Introduction . . . 72
5.2 Background . . . 75
5.2.1 Active Learning and Experimental Design . . . 76
5.2.2 Network Embedding and Link Prediction . . . 76
5.2.3 Related Work . . . 78
5.3 The ALPINE Framework . . . 79
5.3.1 Network Embedding for Partially Observed Networks . . 79
5.3.2 Active Link Prediction using Network Embedding—The Problem . . . 80
5.3.3 The ALPINE Framework . . . 81
5.4 Query Strategies for ALPINE . . . 82
5.4.1 Heuristics . . . 83
5.4.2 Uncertainty Sampling . . . 83
5.4.3 Variance Reduction . . . 84
5.5 Experiments and Discussion . . . 87
5.5.1 The Benefit of Partial Network Embedding . . . 88
5.5.2 Qualitative Evaluation of ALPINE . . . 90
5.5.3 Quantitative Evaluation of ALPINE . . . 92

5.5.4 Discussion 98
5.6 Conclusions 100
Appendix 5.A The Observed Fisher Information Matrix 101
Appendix 5.B The Prediction Variance 102

6 Adversarial Robustness for Link Prediction 105
6.1 Introduction 106
6.2 Related Work 107
6.3 Preliminaries 109
6.3.1 Link Prediction with Probabilistic Network Embedding . . 109
6.3.2 Virtual Adversarial Attack 109
6.4 Quantifying the Sensitivity to Small Perturbations 110
6.4.1 Problem Statement and Re-Embedding (RE) 111
6.4.2 Incremental Partial Re-Embedding (IPRE) 111
6.4.3 Theoretical Approximation of the KL-Divergence 112
6.5 Experiments 113
6.5.1 Case Studies 114
6.5.2 Quality and Runtime of the Approximations 118
6.6 Conclusion 119

7 Conclusion and Discussion 121
7.1 Conclusion 121
7.2 Directions for Future Work 126

Acronyms

ACR	Average-case Conflict Risk
AI	Artificial Intelligence
ALPINE	Active Link Prediction usIng Network Embedding
AUC	Area Under the ROC Curve
BA	Barabási-Albert (network)
BEBA	Backfire Effect and Biased Assimilation
BOF	Biased Opinion Formation
BQM	Boolean Quadratic Maximization
CNE	Conditional Network Embedding
CNE_K	Conditional Network Embedding for the Knowns
ER	Erdős-Rényi (network)
FJ	Friedkin-Johnsen (model)
GCN	Graph Convolutional Network
GDI	Global Disagreement Index
GNN	Graph Neural Network
HALLP	Hybrid Active Learning approach for Link Prediction
IPRE	Incremental Partial Re-Embedding
KL-divergence	Kullback-Leibler divergence
LDS	Local Distribution Smoothness
LP	Link Prediction
MLE	Maximum Likelihood Estimation (or Estimator)

NDCG Normalized Discounted Cumulative Gain
NDI Network Disagreement Index
NE Network Embedding
NFJ Normalized Friedkin-Johnsen (model)

PON Partially Observed Network

RE Re-Embedding
RWC Random Walk Controversy

SDP Semidefinite Programming
SND Social Network Distance
SVM Support Vector Machine

VAT Virtual Adversarial Training

WCR Worst-case Conflict Risk
WS Watts-Strogatz (network)

1 Introduction

"Life is opinion[1]*."*

–Marcus Aurelius, Meditations

This first chapter briefly introduces the conducted research, including the context, the motivations, the problems investigated, and the resulting findings of the main contributions on opinion dynamics and link prediction in networks. It also contains an overview of the thesis, followed by a publication list.

1.1 Context and Motivation

Before the world started to become digitized, social scientists had been investigating how people form their opinions for decades since the last century [1, 2]. The digital revolution not only brought electronics to the world but also changed information dissemination significantly. As a result, it heavily influences the opinion formation process of people due to the information explosion and their increasing amount of exposure to unchecked 'facts' online, as well as offline. Online social networks, such as those formed on platforms like Facebook and Twitter, have facilitated the opinion exchanges in society. The topics of the opinion exchanges are not restricted, meaning that they can be the promotion of certain products and the propaganda for political purposes, which might be risky. To prevent potential threats of opinion manipulation, we need to study this critical topic of opinion

[1] A personal choice of translation.

dynamics in our current digitized era. In particular, we want to understand, compared to the old days without the internet, how those added information exposures, namely the connections in social networks, affect the process of public opinion formation.

While the connections in social networks are essential, they are not always detected. This corresponds to another significant network problem of link prediction that focuses on predicting links not only for social networks but also for all networks in general, such as biological networks. In this thesis, we study the opinion dynamics on social networks, together with the problem of predicting links of networks in general. More details will follow.

1.1.1 Opinion Formation Models

The formation of opinions is studied not only in the research field of social sciences but also in other disciplines, from psychology to economics and physics [3, 4]. In more recent years, the models for opinion formation and dynamics have also attracted the attention of computer scientists in a relatively new area termed computational social science [5], which focuses on investigating social and behavioral relations and interactions with computational tools. For example, studies have been done to quantify and optimize concepts of polarization, controversy, and conflict on social networks [6, 7] by manipulating a set of individuals' opinions or by changing the network connections via which the opinions are influenced [8–10]. The techniques are investigated further with applications in politics and brand perception due to the observation of polarized views [9, 11, 12]. The models for opinion formation are the fundamental element for relevant research.

People form their opinions through interactions with others, and opinion formation models define how these communications impact their opinions. Several models for opinion formation have been proposed and studied [13–18], which are characterized by their assumptions for interactions to take place, the types of opinions considered (i.e., internal and expressed opinions), as well as the linearity of the model [3, 4, 19]. One of the most well-known models is the DeGroot model [1] that serves as the basis for many of its variants, and one popular choice of study among all its variants is the Friedkin-Johnsen (FJ) Model [2]. Other popular models include the Bounded Confidence Models [18, 20, 21] with which a user is only influenced by other opinions that are within ϵ distance, and the Voter Model [14, 22] in which the individual opinions are adopted from friends. Our main contributions on the topic of opinion dynamics in this thesis are based on the DeGroot model and the Friedkin-Johnsen model. We will give a more detailed introduction to both after denoting some of the notations necessary here in this chapter.

Notation. Let $G = (V, E, w)$ be an undirected network with $V = \{1, \ldots, n\}$

the set of nodes, $E \in V \times V$ the set of $m = |E|$ edges, where $(i, j) \in E$ iff $(j, i) \in E$ and w is a function that maps an edge $(i, j) \in E$ onto its weight $w_{ij} = w_{ji} > 0$. We use $N(i)$ to denote the set of nodes connecting to node i: $N(i) \triangleq \{j \in V | (i, j) \in E\}$.

In the DeGroot model, opinions are formed via iterative averaging of the individual opinion and the opinions of friends until convergence [1]. Opinions in the model are real-valued and dynamic such that they all have time steps. Every person $i \in V$ updates his/her opinion $x_i(t+1)$ at time $t+1$ as the weighted sum of their own opinion weighted w_{ii} and those of the friends j weighted w_{ij} at time t. While w_{ii} represents the node's belief in its own opinion, w_{ij} stands for the strength of the relationship. Given an undirected weighted graph $G = (V, E, w)$, the updating rule is defined as:

$$x_i(t+1) = \frac{w_{ii}x_i(t) + \sum_{j \in N(i)} w_{ij}x_j(t)}{w_{ii} + \sum_{j \in N(i)} w_{ij}}. \tag{1.1}$$

The Friedkin-Johnsen (FJ) Model [2] proposed in 1990 is a popular extension of the DeGroot Model [1], which has been used in many studies [9, 10, 23]. The FJ model assumes that every individual $i \in V$ in the social network has two kinds of opinions: a private and immutable internal opinion s_i, and a publicly expressed opinion z_i that can be different from s_i and more in agreement with what the friends express. It indicates that if an individual i is influenced by no one (i.e., has no friends), or the friends $j \in N(i)$ have their expressed opinions z_j the same as the individual's internal opinion s_i, i expresses the private opinion, i.e., $z_i = s_i$. However, the expressed opinions are usually affected by friends due to a desire for social acceptance, and they can be modeled as the weighted sum of the node's own internal opinion and the expressed opinions of the friends:

$$z_i = \frac{w_{ii}s_i + \sum_{j \in N(i)} w_{ij}z_j}{w_{ii} + \sum_{j \in N(i)} w_{ij}}. \tag{1.2}$$

Same as the DeGroot model, the FJ model forms the opinions through averaging. However, we consider the (expressed) opinions static here because, given the static internal opinions, the vector of all expressed opinion z_i in the social network, i.e., $i \in V$, can be solved and interpreted as the Nash Equilibrium in the opinion formation game [24].

1.1.2 Risk of Conflict in Social Networks

One of the hot discussion topics in recent years is the controversy and polarization on social media platforms [12, 25], such as Facebook and Twitter. These platforms offer people unprecedented access to social interactions and communications, exposing them to publicly expressed opinions on controversial issues. As opposing

views become more accessible than before, differences of opinions become more evident than in the pre-digital era when the world was much less connected. Prior work has focused on political opinions [11,12,26], for example in elections. Later, general opinions divergence independent of any topic are also investigated [10].

The measuring of the opinion differences can help evaluate how controversial specific topics are, which can be done using sentiment analysis tools [27, 28], such as the SentiStrength [29] that can give the sentiment score from -4 to 4 for short text, and also by measuring the differences of known opinion distributions on a given social network [11, 12, 30]. We focus on the latter approach and use it to measure the amount of conflict within social networks. This type of study suggests strategic interventions that will enable us to prevent and mitigate conflict effectively and promote viral marketing campaigns. It has received much attention recently [8, 10, 31], in which two ways to intervene are investigated. The first strategy aims at changing the network connections to affect the opinion formation process, while the second strategy is to influence the values of a set of opinions directly [8–10, 32].

However, getting people's opinions on specific topics is not always easy. For example, it is challenging to measure the two types of opinions in the FJ model. Even though the expressed opinions can be obtained using sentiment analysis tools, they cannot be verified as precise values. Moreover, getting the private internal opinions is beyond reach in practice as people may not always know what they really think of things. Meanwhile, existing attempts to reduce the conflict in social networks usually consider only a single or a specific set of issues, while in the real world, different topics do not necessarily correspond to different social networks. Thus, minimizing the conflict on one issue would potentially increase the conflict on another. That asks for attention on solving the research problem of minimizing the conflict without knowing any set of opinions, i.e., relying purely on the network connections.

1.1.3 Link Prediction

Obtaining all the social network connections that control how neighbors influence individual opinions is not always easy. It is not easy for other types of networks in general, such as neural networks, consumer-product networks, and protein-protein interaction networks. This problem is defined as the vital network task of link prediction. It is a problem of predicting future interactions in temporal networks (e.g., social relations to be formed in a social network) or to infer missing links in static networks (e.g., existing but currently undiscovered protein-protein interactions) [33]. Link prediction has a wide range of applications in the real world, including the prediction of connections on LinkedIn, the recommendation for Netflix, the identification of hidden interactions in a crime network, and more. There-

fore, data-efficient and reliable link prediction approaches will benefit many aspects of our daily life.

Several classical methods for link prediction have been proposed since the problem was formed [34], and they mainly consider the similarity between pairs of nodes, i.e., the more similar the two nodes, the more likely that they are linked. While those approaches remain competitive for now, link prediction based on the state-of-the-art network embedding methods already matches and regularly exceeds them in performance [35]. Thus, we focus on a specific class of link prediction methods using network embedding. We will give a detailed introduction to network embedding and how it can be used for link prediction.

Network Embedding, also known as graph representation learning, learns to embed nodes in networks as real-valued vectors in low-dimensional space, or equivalently, it represents the relational graph data into tabular form [36]. More specifically, given an unweighted network $\mathcal{G} = (V, E)$ with V being the set of nodes and $E \subseteq \binom{V}{2}$ the set of edges, a network embedding model aims at finding a mapping $f : V \to \mathbb{R}^d$ for nodes to be transformed into d-dimensional real vectors. The resulting vectors are denoted as the embeddings of the nodes: $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n]^T \in \mathbb{R}^{n \times d}$. Those node embeddings can be used further for downstream tasks, such as graph visualization, link prediction, node classification. We mainly study the important network task of link prediction.

To do link prediction, a network embedding model needs to find a function $g : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ of $\mathbf{x}_i$ and $\mathbf{x}_j$ such that the probability of nodes i and j being linked is computed. With $\mathbf{X}$, the function g can be found by training a classifier on some of the linked and unlinked node pairs. It can also follow naturally from the network embedding model. We prefer the latter type and choose a method called Conditional Network Embedding (CNE) [37] as our base model for improving the data efficiency and robustness of link prediction methods. CNE is a method that preserves the first-order proximity information between nodes by maximizing the probability of the network conditioned on the embedding. Let $\mathbf{A} \in \{0, 1\}^{n \times n}$ denote the adjacency matrix of the network $\mathcal{G} = (V, E)$, i.e., $a_{ij} = 1$ if $\{i, j\} \in E$ and zero otherwise. CNE aims at finding an optimal embedding, denoted as $\mathbf{X}^*$, that maximizes its objective function below [37]:

$$P(\mathcal{G}|\mathbf{X}) = \prod_{\{i,j\} \in E} P(a_{ij} = 1|\mathbf{X}) \cdot \prod_{\{k,l\} \notin E} P(a_{kl} = 0|\mathbf{X}), \tag{1.3}$$

where $P(a_{ij} = 1|\mathbf{X}) = g(\mathbf{x}_i, \mathbf{x}_j)$ if evaluated at the optimal embedding $\mathbf{X} = \mathbf{X}^*$. It uses two half normal distributions (i.e., $\mathcal{N}_+(d_{ij}|\sigma_1^2)$ and $\mathcal{N}_+(d_{kl}|\sigma_2^2)$) for the distances between linked and unlinked node pairs, such that the parameters $0 < \sigma_1 < \sigma_2$ ensures that connected nodes (i, j) will be embedded closer and the disconnected nodes (k, l) will be farther. $\gamma = \frac{1}{\sigma_1^2} - \frac{1}{\sigma_2^2}$ is a model parameter that will be used often for derivations.

Although link prediction based on network embedding has quite competitive performance and thus has been applied increasingly often, it suffers from some issues. These issues not only prevent it from being applied efficiently but also affect the method's reliability. The link prediction problem itself indicates that networks are usually far from being fully observed, meaning that the network connections used for training the model can be incomplete. Model training with incomplete or wrongly labeled data is both time-consuming and ineffective. Thus there is a need to develop data-efficient methods that use only the verified correct link information, i.e., the observed part of the network. Meanwhile, the network embedding methods are not as transparent as those classical and more straightforward link prediction approaches, so their robustness against adversarial attacks cannot be guaranteed, which also needs investigation.

We have given brief introductions and backgrounds of the two main research topics: opinion dynamics and link prediction. Next, we can go into the specific research problems investigated in each of the five contributions after showing an overview of how they are organized in this thesis.

1.2 Research Contributions

This thesis contains five main research contributions on opinion formation and dynamics and the vital network task of link prediction. We start from the fundamental element for studying opinion dynamics in social networks, i.e., the opinion formation models. In Chapter 2, we extend the DeGroot model to account for newly observed social phenomena [38]. In Chapter 3, we address an issue of the popular FJ model by introducing a normalization [39]. Then in Chapter 4, after realizing the challenges in getting the real-valued opinions and identifying the correlations among the opinions on different issues, we quantify and minimize the conflict in social networks without any opinion information [23]. Our method addresses the two challenges by proposing a novel notion called the risk of conflict, which depends purely on the network connections. Then, departing from the context of opinion dynamics, we turn to a more general network task of link prediction, focusing on improving a specific type of link prediction methods based on network embedding. In Chapter 5, considering an often-ignored fact that networks are usually partially observed, we apply active learning to improve the data efficiency of the link prediction method [40]. The work allows one to train the network embedding model with only the available network information and query the most informative unobserved link status for better performance. Chapter 6 presents a study on the adversarial robustness of probabilistic network embedding for the link prediction task, allowing the identification of any potential adversarial attacks that will significantly influence the link prediction performance when using a network embedding method [41].

Overview. An overview of the contributions included in this thesis is shown in Figure 1.1, illustrating the connections between them. The basic idea is that we start from the opinion formation models *on* social networks in Chapters 2 and 3, turn to the problem of investigating the risk of conflict when given no opinion in Chapter 4, and then focus purely on the links *of* networks in Chapters 5 and 6, which is a more general problem for all networks and broader than the domain of opinion dynamics. What is worth mentioning is that, chronologically, the paper in Chapter 4 was done firstly, and the study in Chapter 3 depends largely on it. However, for the transition from opinion dynamics to link prediction, we put Chapter 4 in the middle of the five contributions. Meanwhile, there is a certain extent of repetition in these two chapters. While the results of Chapter 4 are only summarized in Chapter 3, the relevant details will be referred to for clearer presentation.

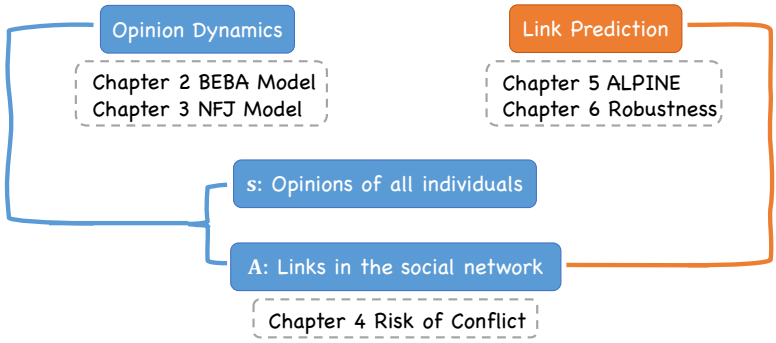


Figure 1.1: Overview of the research contributions in this dissertation.

Chapter 2 Opinion Dynamics with Backfire Effect and Biased Assimilation. The updating rule of the DeGroot model [1] defined in Eq. 1.1 reveals elegantly and intuitively the fact that people form their opinions through social interactions. The model guarantees convergence of opinions to a consensus [3], but it also rejects the possibility of opinion polarization, contradicting some empirical observations [42]. The DeGroot model has a long history, yet it is not up-to-date since it cannot capture some social phenomena, such as biased assimilation. Biased assimilation is called the confirmation bias or myside bias. It refers to the phenomenon where people prefer to be influenced more strongly by the opinions that confirm their own beliefs than those contradicting them [43]. A line of research has tried to incorporate biased assimilation into the opinion formation models, which turned out to lead to polarization [6] and opinion clustering [44].

Another social phenomenon known as the backfire effect in social psychology is an extreme case of biased assimilation [45, 46]. It states that when faced with opinions that contradict one's beliefs strongly, she will not only discredit it but also become more entrenched in her own opinion, thus termed as backfire. It has

not yet attracted much attention and remains overlooked in the study of opinion formation models. However, we show that the backfire effect could help explain polarization that has emerged in our society, which is impossible in the DeGroot model. Our research result is a novel DeGroot-type opinion formation model that simultaneously models the Backfire Effect and Biased Assimilation: the BEBA model. The theoretical analysis of the proposed model shows the conditions we found for opinions in a social network to reach polarization and consensus. With extensive experiments, we further confirm the theoretical results and evaluate the influence of different factors on the opinions dynamics of the BEBA model.

Chapter 3 The Normalized Friedkin-Johnsen Model. Observing the updating rule of the FJ model in Eq. 1.2, we see that if a node i has more friends or the stronger these friendships are, represented by the w_{ij} when $j \in N(i)$, the less important it values its own internal opinion because the ratio of w_{ii} in $w_{ii} + \sum_{j \in N(i)} w_{ij}$ decreases. This observation could be undesirable because people do not have to lose their convictions due to having many friends. Thus, the FJ model in its original form is not always realistic. We propose to normalize the influence one receives from the expressed opinions of friends such that the self-appraisal of w_{ii} [7] remains constant and independent of the number of friends or the strength of the friendships. The authors of the FJ model had indicated our proposed normalization. However, it is often ignored in recent work, which turns out to be necessary, especially for studies on editing the network connections to achieve specific optimization goals, such as influence maximization.

In Chapter 3, we propose the normalized variant of the FJ model, namely the NFJ model, and then investigate how it differs from the vanilla FJ model. Theoretically, we analyze the qualitative differences between the NFJ and the FJ model, focusing on a recently discovered conservation law of conflict for the FJ model [23]. It is not a surprise that the conservation law no longer holds, and we are able to identify a term for conflict elimination. The term has an interesting interpretation that, to reduce conflict, opinionated people should be less influential. Experimental results also show how the two models led to different opinion formation and quantification of conflict. One thing worth mentioning is that the normalized version of the FJ model is also used in [47], which was published after our work had been finished.

Chapter 4 Quantifying and Minimizing Risk of Conflict in Social Networks. One challenge for validating the existing opinion formation models has been that the real-valued opinions are not easily accessible. While the expressed opinions might be obtained by analyzing the sentiment, the internal opinions (i.e., in the FJ model) can be beyond reach in practice. Additionally, the studies along the line of research on minimizing the conflict or controversy could fail the task because they usually focus on a single or a few given issues, ignoring the risk of triggering more significant amounts of conflict on other issues. To overcome these two short-

comings, we depart from considering both the individual opinions and the social network structure to propose the notion of *risk of conflict*, which depends purely on the network. It essentially means how resilient a given network is against conflict on all possible issues. By editing the network structure to reduce the risk of conflict, we can obtain a more robust network topology that is resilient against conflict on all possible topics instead of just a single one.

The *risk of conflict* we propose applies to two cases: the worst and the average case over all possible distributions of opinions on the network, resulting in the *worst-case conflict risk (WCR)* and the *average-case conflict risk (ACR)*. Based on a set of conflict measures we summarized from the literature, we show how WCR and ACR could be reduced by local edits of the network links using two types of algorithms of coordinate descent and conditional gradient descent. The empirical results and theoretical analysis provide insights into the network structure with the slightest risk of conflict. For example, one interesting remark is that the common management structures in companies, i.e., a flat organization as a clique, or a hierarchical organization as a tree, turned out to contain the smallest differences of the expressed opinions over the links in a social network. That might help explain why we have such structures in companies nowadays. The paper might also contribute to inspiring more work in opinion dynamics without knowing the explicit opinions.

Chapter 5 Active Link Prediction usIng Network Embedding. As the problem of link prediction itself indicates, networks are usually partially observed. That is because obtaining the complete information of network connectivity, especially for the large-scale networks with millions of nodes, demands a heavy workload that can be expensive or slow. In practice, networks are almost always only partially observed, so many pairs of nodes still have unknown link status [48]. It is an often-ignored fact affecting several existing link prediction methods due to the lack of differentiation between the known unlinked and unknown link status, similar to using unlabeled data points as negative examples. The influence of ignoring unknown link status is more conspicuous in a specific class of link prediction methods we are interested in than others, i.e., those using network embedding, because the network embedding models try to embed linked nodes closer while unlinked nodes further in the embedding space. Treating the unknown link status as known unlinked would cause less-efficient model training and less-effective performance due to using wrongly labeled data. Thus, we argue that active learning can be applied here, with which we could query the most informative link status that has not yet been observed to improve the link prediction performance with a limited cost for querying.

In Chapter 5, we propose ALPINE (Active Link Prediction usIng Network Embedding), the first method utilizing active learning for link prediction based on network embedding. The ALPINE framework aims to improve the link prediction

performance by identifying and querying the most informative link status that has not yet been observed but is estimated to benefit the embedding algorithm maximally. Then the newly acquired link status can be used as additional information for model training. The informativeness of the candidate link statuses is calculated by the utility functions of the query strategies we develop for ALPINE. Our empirical results show that the differentiation between the known unlinked and unknown link status improves the data efficiency of the network embedding model. The quantitative experiments further confirm that ALPINE boosts link prediction accuracy with far fewer queries than the random strategy. Moreover, they also provide insights and actionable guidance for practitioners to apply our method in real-world scenarios.

Chapter 6 Adversarial Robustness of Probabilistic Network Embedding for Link Prediction. Machine learning methods are not guaranteed to be robust against adversarial attacks [49] due to the lack of transparency. Network Embedding algorithms, such as Graph Neural Networks (GNNs) [50], are no exception. Studies have shown that imperceptible perturbations on the input data (i.e., the network connection or node attribute) fed to GNNs can lead to a dramatic drop in the node classification performance [51,52]. The relevant robustness has to be investigated if we want to apply network embedding models for the link prediction task in a reliable manner. More specifically, we want to know if there exist any minor adversarial modifications to the network topology that will significantly influence the link prediction performance when using a network embedding method. Robustness for link-level tasks turns out to be an insufficiently explored topic because the research attention for the robustness of graph learning methods has been focused mainly on the classification task at either the node or the graph level [51–56]. To fill the gap, we investigate the adversarial robustness of a probabilistic network embedding model, called Conditional Network Embedding (CNE) [37], for link prediction, in Chapter 6.

Given CNE and a network, we measure how sensitive the model is when flipping the link between a single node pair (i.e., existing edge to non-edge, and vice versa). The sensitivity is measured as the change in the link predictions due to that edge flip. An intuitive way for explaining it is we measure the impact of an edge flip, which is considered imperceptible, as the KL-divergence between the link probability distributions before and after flipping. It allows one to identify the links or non-links that are vulnerable to be adversarially perturbed such that if attacked, the link predictions will change significantly. We illustrate with case studies how structural perturbations influence the link predictions and then analyze the characteristics of the perturbations according to their sensitivity. Moreover, to avoid costly re-training of the model, we develop time-efficient approximations based on the gradient information and empirically show that they are also of significantly good quality.

1.3 Publications

Publications in international journals

- **Xi Chen**, Panayiotis Tsaparas, Jefrey Lijffijt and Tijl De Bie. *Opinion Dynamics with Backfire Effect and Biased Assimilation*. PLOS ONE. 2021;16(9). [38] Presented in Chapter 2.

  An early version of this article was also presented and included in non-archived proceedings as:

  - **Xi Chen**, Panayiotis Tsaparas, Jefrey Lijffijt and Tijl De Bie. *Opinion Dynamics with Backfire Effect and Biased Assimilation*. In the 15th International Workshop on Mining and Learning with Graphs (MLG), 2019. [57]

- **Xi Chen**, Bo Kang, Jefrey Lijffijt and Tijl De Bie. *ALPINE : Active Link Prediction usIng Network Embedding*. APPLIED SCIENCES-BASEL. 2021;11(11). [40] Presented in Chapter 5.

Publications in archived proceedings

- **Xi Chen**, Bo Kang, Jefrey Lijffijt and Tijl De Bie. *Adversarial Robustness of Probabilistic Network Embedding for Link Prediction*. Accepted to the 3rd Workshop on Machine Learning for Cybersecurity and to be in proceedings of ECML PKDD 2021 Workshops. 2021. [41] Presented in Chapter 6.

- **Xi Chen**, Jefrey Lijffijt and Tijl De Bie. *Quantifying and Minimizing Risk of Conflict in Social Networks*. In Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 1197- 1205. 2018. [23] Presented in Chapter 4.

Publications in non-archived proceedings

- **Xi Chen**, Jefrey Lijffijt and Tijl De Bie. *The Normalized Friedkin-Johnsen Model (a Work-in-progress Report)*. Presented at the PhD Forum of ECML PKDD, 2018. Available at UGent biblio: https://biblio.ugent.be/publication/8574755. [39] Presented in Chapter 3.

2

Opinion Dynamics with Backfire Effect and Biased Assimilation

Abstract The democratization of AI tools for content generation, combined with unrestricted access to mass media for all (e.g. through microblogging and social media), makes it increasingly hard for people to distinguish fact from fiction. This raises the question of how individual opinions evolve in such a networked environment without grounding in a known reality. The dominant approach to studying this problem uses simple models from the social sciences on how individuals change their opinions when exposed to their social neighborhood, and applies them on large social networks.

We propose a novel model that incorporates two known social phenomena: (i) *Biased Assimilation*: the tendency of individuals to adopt other opinions if they are similar to their own; (ii) *Backfire Effect*: the fact that an opposite opinion may further entrench people in their stances, making their opinions more extreme instead of moderating them. To the best of our knowledge, this is the first DeGroot-type opinion formation model that captures the Backfire Effect. A thorough theoretical and empirical analysis of the proposed model reveals intuitive conditions for polarization and consensus to exist, as well as the properties of the resulting opinions.

2.1 Introduction

Recent years have seen an increasing amount of attention from the computational social science in the study of opinion formation and polarization over social networks, with applications ranging from politics to brand perception [9, 11, 12]. Much of this research leverages pre-existing opinion formation models that have been studied for decades [3, 4]. These models formalize the fact that people form their opinions through interactions with others. One of the best-known models is the DeGroot model [1], which considers an individual's opinion as dynamic and updates it iteratively as the weighted average of the individual's current opinion and those of her social neighbors. The weights represent the strength of the social connections.

The DeGroot model is elegant and intuitive, and it guarantees that the opinions converge towards a consensus [1, 3]. However, opinions formed with it cannot polarize, which contradicts empirical observations [42, 58]. Variants of the DeGroot model have been proposed to incorporate *biased assimilation* [6, 44], which is also known as *confirmation bias* or *myside bias* and refers to the phenomenon where information that corroborates someone's beliefs affects those beliefs more strongly than information that contradicts them [43]. Incorporating biased assimilation has been shown to potentially lead to polarization [6] or opinion clustering [44].

An extreme manifestation of confirmation bias is a behavior known in social psychology as the *Backfire Effect* [45, 46]. It refers to the fact that, when an individual is faced with information that contradicts her opinion, she will not only tend to discredit it, but will also become more entrenched and thus extreme in her own opinion. The backfire effect may help explain the emergence of polarization. Yet, it has so far been overlooked by existing opinion formation models.

Motivated by these observations, we propose a novel opinion formation model that simultaneously models the Backfire Effect and Biased Assimilation - the BEBA model. BEBA depends on a single—intuitive, node-dependent—parameter β_i, which we call the *entrenchment* of node i. The parameter captures both the tendency of node i to become more entrenched by opposing opinions and the bias towards assimilating opinions favorable to its own. Our main contributions are:

- We propose the BEBA model of opinion formation, which accounts for both the Backfire Effect and Biased Assimilation. To the best of our knowledge BEBA is the first DeGroot-type opinion formation model that incorporates the Backfire Effect.
- We theoretically analyze the BEBA model, studying conditions for reaching consensus or polarization.
- We empirically evaluate, on real and synthetic data, the influence of the entrenchment parameter, the initial opinions, and the network topology, on

the opinion dynamics of BEBA.

2.2 Related Work

Opinion formation has been studied in diverse research fields, from psychology and social sciences to economics and physics [3, 4]. The former mostly use empirical methods to understand the factors that affect opinion formation, while the latter mostly aim to understand emergent behavior implied by these theories.

Two observations from psychology and social sciences relating to our work are the biased assimilation and backfire effect [59, 60], which state that individuals are more inclined to accept opinions closer to their own [43], and that, when exposed to the opposite opinions, individuals entrench themselves in their own opinions [45, 61, 62], respectively. The existence of the backfire effect is controversial. It is observed in many studies, but there are also failures to find the evidence of it [63, 64]. For example, it is reported negligible on Reddit in a recent study [65]. However, the result may not be robust because the expressed opinions gathered on Reddit are not necessarily consistent with people's *intrinsic* opinions [2]. The backfire effect remains to be further investigated on improved measures and experimental designs [64], and our modeling of it serves that purpose.

We study the common setting where opinions are formalized as real values, formed through social interactions (see [3] and [4] for surveys). Existing opinion formation models can be described as linear or nonlinear depending on how the opinions are represented [19]. The most popular models include the Voter model [14, 22], the DeGroot model [1], and the Friedkin-Johnsen model [2]. Yet, none of these account for the biased assimilation or backfire effect.

There is work on modeling the fact that users are more influenced by opinions closer to their own. The bounded confidence models [18, 20, 21] assume that a user is influenced only by opinions that are within ϵ of its own. With rewiring and the relaxation of the bound, the variations of the bounded confidence model are used to further model confirmation bias and polarization in the formation of public opinion [66]. The work of Kempe et al. [67] assumes that there are different types of opinions and users are influenced by opinions of similar types. Das et al. [68] consider a biased version of the voter model that biases individuals to adopt similar opinions.

The work most closely related to ours is that of Dandekar et al. [6], who propose a variant of the DeGroot model to capture the biased assimilation effect. Their model is called the Biased Opinion Formation (BOF) model, and we treat it as our baseline because both ours and the BOF model are DeGroot-type. In the BOF model, the importance that a node attaches to the opinion of a neighbor depends on their agreement. However, it cannot model the backfire effect and introduces cognitive irrationality. We will contrast and highlight the differences between the

two models with an illustrative example after formally introducing our model. Before that, the detailed definition of the baseline BOF model, together with that of the vanilla DeGroot model will be introduced in the following section as background of our work.

2.3 Model Definition

In this section, we first describe the notations and two existing models that are most relevant to our work (i.e., the DeGroot and the BOF model), then we formally introduce our nonlinear opinion formation model - BEBA, which generalizes the DeGroot model and accounts for both backfire effect and biased assimilation. Finally, we provide a comparison between BEBA and the BOF model on an illustrative example, to contrast and highlight their differences.

2.3.1 Preliminaries and Background

Notation. Let $G = (V, E)$ denote a connected undirected network, with $V = \{1, ..., n\}$ the set of nodes, and $E \subseteq V \times V$ the set of $m = |E|$ edges, where $(i, j) \in E$ iff $(j, i) \in E$. When the network is weighted, $w_{ij} = w_{ji}$ represents the weight of edge (i, j). We use $N(i)$ to denote the set of neighbors of node i: $N(i) \triangleq \{j \in V | (i, j) \in E\}$.

All models we include in this work can be defined as dynamical systems, where opinions are real numbers updated iteratively within a fixed interval of $[0, 1]$ or $[-1, 1]$. To discriminate between the two intervals, we use x for opinions in $[0, 1]$ and y for opinions in $[-1, 1]$. We use $x_i(t)$ (resp. $y_i(t)$) to denote the opinion of node i at iteration/time $t = 0, 1, 2, \ldots$; $\mathbf{x}(t)$ (resp. $\mathbf{y}(t)$) to denote the opinion vector of the network at time t; x_i (resp. y_i) to denote the opinion of node i after convergence for $t \to \infty$ (if that limit exists); and $\mathbf{x}$ (resp. $\mathbf{y}$) to denote the corresponding vector.

The DeGroot Model. This model [1] is an averaging opinion formation model, where the individual's opinion is determined by the average of her own opinion and that of her neighbors. More specifically, the updating rule is:

$$x_i(t+1) = \frac{w_{ii}x_i(t) + \sum_{j \in N(i)} w_{ij}x_j(t)}{w_{ii} + \sum_{j \in N(i)} w_{ij}} \tag{2.1}$$

where w_{ii} represents the extent to which the node values her own opinion, and w_{ij} is the strength of the connection/friendship between node i and j. Iterative opinion updates will converge to a stationary state, where every node has the same opinion $x_i = x^*$ [3]. Therefore, the model always reaches consensus, and never polarizes.

Biased Opinion Formation - BOF. The BOF model [6] generalizes the DeGroot model to incorporate *biased assimilation*. Given a weighted undirected graph $G =$

(V, E, w), every node $i \in V$ is assigned a bias parameter $b_i \geq 0$. Higher values of b_i means that node i is more biased towards her own opinion. The opinion value $x_i(t) \in [0, 1]$ is interpreted as the degree of support for opinion position 1 (i.e., the highest possible opinion value), while $1 - x_i(t)$ is the support for 0. BOF is defined by

$$x_i(t+1) = \frac{w_{ii}x_i(t) + (x_i(t))^{b_i}s_i(t)}{w_{ii} + (x_i(t))^{b_i}s_i(t) + (1 - x_i(t))^{b_i}(d_i - s_i(t))} \tag{2.2}$$

where $s_i(t) \triangleq \sum_{j \in N(i)} w_{ij}x_j(t)$ is the weighted sum of i's neighbouring opinions, and $d_i \triangleq \sum_{j \in N(i)} w_{ij}$ is the weighted degree of node i. During the updating process, node i weighs confirming and disconfirming evidence in a biased way: weighing the neighboring support for opinion 1 by $(x_i(t))^{b_i}$, and that for opinion 0 by $(1 - x_i(t))^{b_i}$. When $b_i = 0$, the BOF model is identical to the DeGroot model. However, when $b_i \neq 0$, this model introduces cognitive irrationality since an individual's opinion will change even when the neighboring opinion is the same to its own. We will show that our model does not suffer from this problem.

2.3.2 The BEBA Model

We now define the BEBA model, which also generalizes the DeGroot model to incorporate not only biased assimilation but also the backfire effect. To capture both phenomena, we adapt the DeGroot model by dynamically setting the edge weights. For BEBA, the opinion vector at time t is $\mathbf{y}(t)$, with $y_i(t) \in [-1, 1]$. Rather than using fixed weights as in the DeGroot model, we propose to let the weights be determined by the opinions. Specifically, for an edge $(i, j) \in E$ we define the edge weight $w_{ij}(t)$ at time t as

$$w_{ij}(t) = \beta_i y_i(t) y_j(t) + 1. \tag{2.3}$$

The product $y_i(t)y_j(t)$ captures the degree of (dis)agreement between the opinions of node pair (i, j). The parameter $\beta_i > 0$, which we call the *entrenchment parameter* of node i, determines the level of the influence caused by that (dis)agreement with node j on i's updating with $w_{ij}(t)$: the larger, the stronger the biased assimilation and backfire effect.

Given the weights $w_{ij}(t)$, the opinions in the BEBA model are updated similarly to the DeGroot model:

$$y_i(t+1) = \frac{w_{ii}y_i(t) + \sum_{j \in N(i)} w_{ij}(t)y_j(t)}{w_{ii} + \sum_{j \in N(i)} w_{ij}(t)} \tag{2.4}$$

Note that when $\beta_i = 0$, the BEBA updating rule is identical to that of the DeGroot model (Eq. (2.1)) for unweighted networks. When $\beta_i \neq 0$, we discriminate two cases depending on $w_{ij}(t)$:

1. **Backfire Effect is modeled when $w_{ij}(t) < 0$.** Negative weight means $\beta_i y_i(t) y_j(t) < -1$. Since $\beta_i > 0$, $y_i(t) y_j(t) < 0$, that is, nodes i and j hold opposing views. Multiplying $y_j(t)$ with this negative weight $w_{ij}(t)$ in the summation in the numerator leads to a contribution of the same sign as $y_i(t)$, while adding the negative weight to the denominator reduces it, inflating the resulting quotient. The combination of these two effects models the backfire effect.

2. **Biased Assimilation is modeled when $w_{ij}(t) > 0$.** We consider two cases:

   (a) $-1 < \beta_i y_i(t) y_j(t) < 0$: Here nodes i and j hold opposing but not too different opinions. Node i critically evaluates the conflicting opinion of node j, but still assimilates it to a reduced extent.

   (b) $0 < \beta_i y_i(t) y_j(t)$: Since $\beta_i > 0$, node i and j have both positive or negative opinions here, resulting in an increased weight $w_{ij}(t)$. In this case, node i assimilates the opinion of neighbor j more strongly if the extent of their agreement is stronger.

Note that the denominator in Eq. (2.4) can become 0 resulting in a diverging opinion, or negative causing an unnatural opinion reversal. We consider this situation to be beyond the model's validity region, and thus we refine the BEBA updating rule as follows:

$$y_i(t+1) = \begin{cases} \operatorname{sgn}(y_i(t)) & \text{if } w_{ii} + \sum_{j \in N(i)} w_{ij}(t) \leq 0, \\ \frac{w_{ii} y_i(t) + \sum_{j \in N(i)} w_{ij}(t) y_j(t)}{w_{ii} + \sum_{j \in N(i)} w_{ij}(t)} & \text{otherwise.} \end{cases} \tag{2.5}$$

Moreover, for a small denominator, the resulting opinions may fall outside the range $[-1, 1]$. To address this, we additionally clip negative and positive values at -1 and 1.

2.3.3 Comparison between BEBA and BOF

There is a similarity between the BOF and our BEBA model, in that both alter the weights in the DeGroot model. Comparing to the linear DeGroot model, both BEBA and BOF are nonlinear. Now we study how the two nonlinear models differ with an illustrative example. Using a star graph consisting of five nodes as illustrated in Fig 2.1, we update the opinion of the center node (i.e., node 1) with both models for one iteration and observe how the resulting opinions for the two models differ.

First, we deal with the fact that BOF assumes only positive opinion values, while our model assumes opinions being both positive and negative. Note that the value range of opinions is important in both models, since the BOF model weights the opinion values, while our model exploits the disagreement in the sign. To

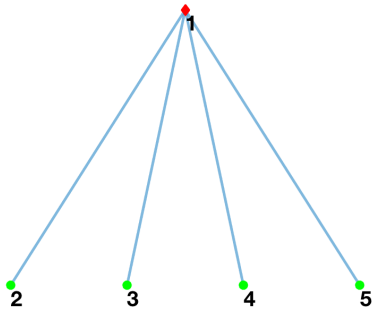


Figure 2.1: **The star graph of five nodes.**

compare the models, we assume positive opinion $x_i(t) \in [0, 1]$ on all nodes for the update of BOF; and we transform them to $[-1, 1]$ by setting $y_i(t) = 2x_i(t) - 1$ for BEBA. After computing $y_1(t + 1)$ with BEBA, we rescale the opinions back to $[0, 1]$.

In this experiment we assume $x_i(t)$ identical for nodes $i = 2, 3, 4, 5$, and $x_i(t) \in [0, 1]$ for all nodes. We set $w_{11} = 1$ for both models, $b_1 = 1$ for BOF, and consider the values of 1 and 2.5 for β_1 in BEBA. The opinion value $x_1(t + 1)$ updated with both models, as a function of $x_{2,3,4,5}(t)$ and $x_1(t)$ is shown in Fig 2.2. The difference between the two models becomes clearer when $x_1(t)$ takes extreme values (i.e., 0 or 1), and we study this below.

Fig 2.3(a) shows the curves for the two models when $x_1(t) = 0$. In BOF, the opinion $x_1(t+1)$ remains unchanged at value 0. This is true regardless of the value of b_1. Thus, extreme nodes never change their opinions, even a little, even when they are not biased at all. However, according to biased assimilation, unbiased individuals are influenced by similar opinions, and even extreme nodes assimilate opinions that are close to their own. In contrast, our model better captures the biased assimilation in this case. In Fig 2.3(a), for $\beta_1 = 1$, which corresponds to a mildly biased node, the opinion of node 1 can be moderated by that of her neighbors to different extents, while $x_1(t + 1)$ never exceeds 0.5. Therefore, extreme nodes are not stuck in the extremes.

To further highlight the difference between the two models and better understand the backfire effect, we increase β_1 to 2.5, and set $x_1(t) = 0.25$ as shown in Fig 2.3(b). In BOF, $x_1(t + 1)$ becomes smaller than $x_1(t) = 0.25$ even when all neighbors are holding the same opinion $x_{2,3,4,5}(t) = 0.25$, which does not make sense according to [46]. But in BEBA, we make sure that node 1 does not react to persuasion that coincides with its own current opinion, see point $(0.25, 0.25)$. Meanwhile, we observe the backfire effect with BEBA that when the disagreement between node 1 and her neighbors becomes large (i.e., when $x_{2,3,4,5}(t) > 0.9$), $x_1(t + 1)$ drops under 0.25, until it takes the extreme at opinion 0.

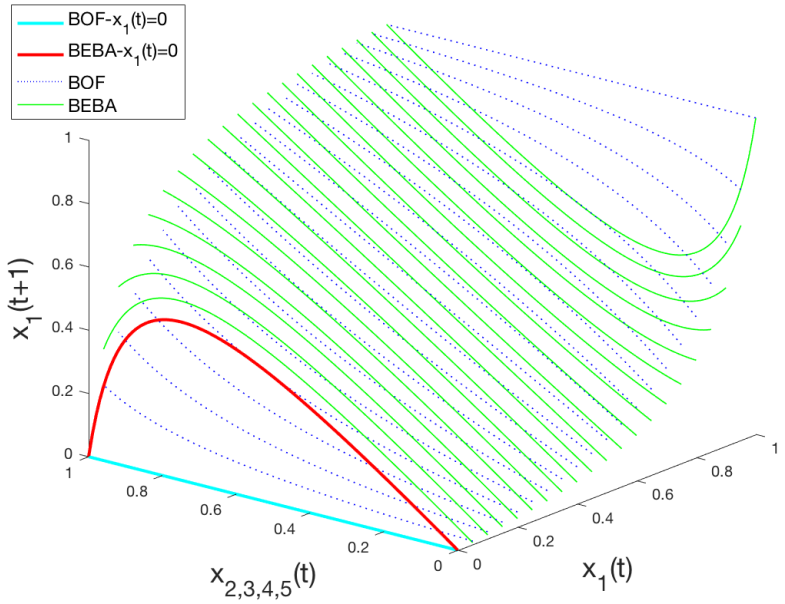


Figure 2.2: **Opinion formation on the star graph.**

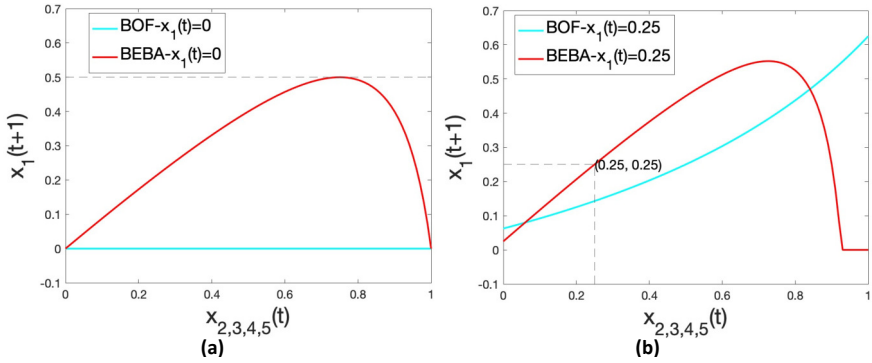


Figure 2.3: $x_1(t+1)$ **as a function of** $x_{2,3,4,5}(t)$**.** *(a)* $\beta_1 = 1$, $b_1 = 1$, $x_1(t) = 0$*; (b)* $\beta_1 = 2.5$, $b_1 = 1$, $x_1(t) = 0.25$.

From the plots in Fig 2.3 we also observe that for the different combinations of β_1 and $x_1(t)$, there exists a value of the neighboring opinions that causes the largest change in $x_1(t+1)$. For example, when $\beta_1 = 1$ and $x_1(t) = 0$, neighboring opinion of around 0.75 is the most influential as shown in Fig 2.3(a); for $\beta_1 = 2.5$ and $x_1(t) = 0.25$, opinion around 0.7 is the most influential according to Fig 2.3(b). This provides insight on influence maximization and misinformation correction that a moderate opinion could be more effective than an extreme one.

2.4 Theoretical Analysis

This section contains theoretical analysis of the BEBA model for two settings. First we investigate the dynamics of opinions for a single agent in a fixed environment, and secondly we study the dynamics of polarization for all nodes in a connected social network.

2.4.1 A Single Agent in a Fixed Environment

Here we theoretically analyze the limit behavior of a single agent's opinion in an environment with a fixed opinion. An analysis of this type has been done for the BOF model. The setup is admittedly somewhat artificial but helps to gain a better understanding of BEBA. It has been deemed realistic in cases where the fixed environment consists of the news media, billboards, etc [6]. It also models the situation where the single agent is connected to a network that is large enough such that adding that agent will not meaningfully affect the network.

For the agent i, we denote $y(t) \in [-1, 1]$ its opinion at time t, $\beta > 0$ its entrenchment parameter, and y its converged opinion - $\lim_{t\to\infty} y(t)$. We assume the agent weighs its own opinion with $w_{ii} = w$. For simplicity, we only consider the situation where the environment contains one node, but it should be noted that the analysis below can be easily generalized to several nodes (see Appendix 2.A). Let $p \in [-1, 1]$ be the fixed environmental opinion. Then, according to BEBA, the agent updates its opinion as:

$$y(t+1) = \begin{cases} \operatorname{sgn}(y(t)) & \text{if } w + \beta p y(t) + 1 \leq 0, \\ \frac{wy(t)+\beta p^2 y(t)+p}{w+\beta p y(t)+1} & \text{otherwise.} \end{cases} \tag{2.6}$$

Before stating a theorem that quantitatively characterizes the limit y, we consider the behavior in two cases. The first case is for a sufficiently small β (i.e., not biased), while the second is for a sufficiently large β (i.e., biased). In the first case, the fixed environment's opinion p will be sufficiently attracting such that $y = p$ regardless of $y(t)$. The same is true when $p = 0$: the neutral opinion is never polarizing and thus always attracting. The second case can further be divided into three sub-cases as the limit y will depend on the similarity between $y(t)$ and the environment's opinion p: (a) if $y(t)$ is similar to p, p should have an attracting effect on $y(t)$ such that $y = p$; (b) if $y(t)$ is very different from p, however, the backfire effect will cause the agent's opinion to diverge from p, such that $y = \operatorname{sgn}(y(t))$; (c) between the former two sub-cases there will be a 'sweet spot' where $y(t)$ is neither sufficiently similar to p for $y(t)$ to converge to p, nor sufficiently different for it to diverge to $\operatorname{sgn}(y(t))$ - this is an unstable equilibrium where $y(t)$ remains constant through time, i.e., $y = y(t)$.

This intuition is formalized in the following theorem (proofs in Appendix 2.A).

For conciseness and transparency, we state it for the situation where $p \leq 0$ as it is trivial to adapt the theorem for $p \geq 0$.

Theorem 1. *For a single agent with opinion $y(t)$ and entrenchment parameter β in a fixed environment represented by opinion p, depending on the value of β relative to p:*

Case 1: *When $p = 0$ or $\beta < -1/p$, the agent's opinion always converges to p: $y = p$.*

Case 2: *When $p < 0$ and $\beta \geq -1/p$, there are three possibilities depending on how similar $y(t)$ is to p, as illustrated in Fig 2.4.*

a: *If $y(t) < -\frac{1}{\beta p}$, $y(t)$ will be sufficiently attracted to p such that $y = p$.*

b: *If $y(t) > -\frac{1}{\beta p}$, $y(t)$ will diverge away from p such that $y = \text{sgn}(y(t)) = 1$.*

c: *If $y(t) = -\frac{1}{\beta p}$, $y(t)$ will remain constant through time such that $y = -\frac{1}{\beta p}$.*

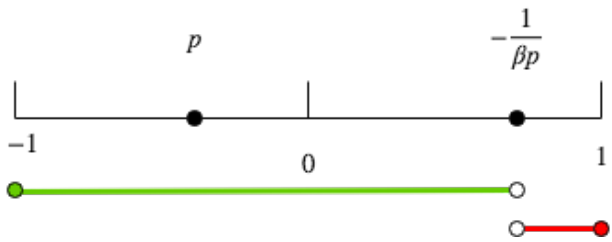


Figure 2.4: **Graphical illustration of Case 2 from Theorem 1 (i.e. $p < 0$ and $\beta \geq -1/p$).** *(a) For values of $y(t)$ in the green range, $y(t)$ will converge to $y = p$. (b) For values of $y(t)$ in the red range, $y(t)$ will diverge to $y = 1$. (c) For $y(t) = -\frac{1}{\beta p}$, $y(t)$ will not change such that $y = -\frac{1}{\beta p}$.*

Theorem 1 already suggests that opinions under the BEBA model evolve to one of three possible states: consensus as in Case 1 and Case 2(a), polarization as in Case 2(b), and an unstable state of persistent disagreement as in Case 2(c).

2.4.2 Polarization and Consensus for All Nodes in a Network

We now extend our analysis from the single agent to a group of individuals that can update their opinions at any time step t. The dynamics of polarization are investigated theoretically with respect to different values of the entrenchment parameter β. It was argued by the authors of the BOF model that homophily alone, without biased assimilation was not sufficient for polarization in the DeGroot model [6].

As for BEBA, the backfire effect and biased assimilation are sufficient to lead to polarization or consensus, depending on the parameters and the initial opinions, even when there is no homophily. The theorem below (proofs in Appendix 2.B) makes this clear, by providing easy-to-realize sufficient conditions for polarization or consensus to occur.

Theorem 2. *Let $G = (V, E)$ be a connected unweighted undirected network. For all $i \in V$, let $y_i(t) \in [-1, 0) \cup (0, 1]$ be the opinion of node i at time t, $w_{ii} = 1$ and $\beta_i = \beta > 0$ for all $i \in V$. Denote $\mathbf{y}(t)$ the opinion vector of G at time t, $|\mathbf{y}(t)|$ the vector with the absolute values of all opinions. Then at convergence the BEBA model can lead to the following states:*

1. *Polarization: When $\beta > \frac{1}{[\min|\mathbf{y}(0)|]^2}$, $\forall i \in V$, $|y_i| = 1$ and there exist both opinion -1 and 1.*
2. *Consensus: When $\beta < \frac{1}{[\max|\mathbf{y}(0)|]^2}$, there exists a unique $y^* \in [-\max|\mathbf{y}(0)|, \max|\mathbf{y}(0)|]$ such that $\forall i \in V$, $y_i = y^*$.*

A special case of particular theoretical interest is when $\min|\mathbf{y}(0)| = \max|\mathbf{y}(0)|$. Then there are only two different opinions in the network, with the same absolute value but opposite signs (i.e. they could represent 'for' and 'against' an issue of interest). In this case, a borderline situation emerges to which we refer as *persistent disagreement*. It can be proved concisely by relying on Theorem 2, and thus we state it as a Corollary:

Corollary 1. *Let $G = (V, E)$ be a connected unweighted undirected network where $V = V_1 \cup V_2$, $V_1 \cap V_2 = \emptyset$. For all $i \in V$, let $w_{ii} = 1$ and $\beta_i = \beta > 0$. Assume for all $i \in V_1$, $y_i(0) = y_0$ and for all $i \in V_2$, $y_i(0) = -y_0$ for some $0 < y_0 < 1$. Then the BEBA model can result in the following states:*

1. *Polarization: When $\beta > \frac{1}{y_0^2}$, $\forall i \in V$, $|y_i| = 1$ and there exist both opinion -1 and 1.*
2. *Persistent disagreement: When $\beta = \frac{1}{y_0^2}$, $\forall i \in V_1$, $y_i(t') = y_0$ and $\forall i \in V_2$, $y_i(t') = -y_0$, for all $t' \geq 0$.*
3. *Consensus: When $\beta < \frac{1}{y_0^2}$, there exists a unique $y^* \in (-y_0, y_0)$ such that $\forall i \in V$, $y_i = y^*$.*

Intriguingly, these conditions in Theorem 2 and Corollary 1 are independent of the network structure and depend only on the entrenchment parameter β and the opinion vector at time 0. Yet, it should be noted that the value of the consensus and the eventual polarized state do depend on the network structure. Moreover, the network structure, and the distribution of the opinions over it, do determine whether polarization or consensus will arise when neither of the sufficient conditions of Theorem 2 are satisfied. These claims are confirmed in experiments in the next section.

2.5 Experimental Analysis

In the previous section, we provided sufficient conditions for our model to reach consensus or polarization. We now perform an experimental analysis of how these two phenomena manifest themselves on real and synthetic data. Our goal is to answer the following questions:

- In the case when consensus is reached, what is the value of the consensus opinion, and how does the entrenchment parameter β, the initial opinions $\mathbf{y}(0)$, and the network structure affect this value?
- In the case when the opinions polarize, what is the state of the polarization, and how is it affected by the entrenchment parameter β, the initial opinions $\mathbf{y}(0)$, and the network structure?

We use both real-world and synthetic data in our experiments. The real datasets include Zachary's Karate Club network [69] where we use synthetic opinion vectors, and six Twitter networks gathered with real opinions (computed using sentiment analysis) for different events ranging from political elections to sports [70, 71]. To fit our setting, we process the Twitter networks to make sure that their adjacency matrices are symmetric. See Table 2.1 for network statistics. Meanwhile, following the way Abebe et al. used for processing the real opinions [47], we normalize the first set of opinions for each event into range $[0, 1]$. After that, we transform the opinions to $[-1, 1]$ for BEBA.

Table 2.1: **Real-world Network Summary**

Network	$\|V\|$	$\|E\|$	Event
Karate	34	78	Friendship among members of a university karate club.
Tw:Club	703	3322	Barcelona getting the 1st place in La-liga 2016.
Tw:Sport	703	3322	Champions League final 2015, Juventus vs Real Madrid.
Tw:US	533	13564	US Presidential Election 2016.
Tw:UK	231	905	British Election 2015.
Tw:Delhi	548	3638	Delhi Assembly Election 2013.
Tw:GoT	947	7922	The promotion on "Games of Thrones" 2015.

The synthetic networks, with which we use randomly generated opinions, are:

- Erdős-Rényi (ER) networks $G(n, \rho)$ with binomial degree distributions, where ρ is the edge probability [72];
- Watts-Strogatz (WS) networks $G(n, K, \sigma)$ that have the small world property [73] - with K being the average degree and σ the rewiring probability;
- Barabási-Albert (BA) networks $G(n, M_0, M)$ that are scale-free, where M_0 is the number of initial nodes and M the number of nodes a new node is connected to [74].

2.5.1 The Influence of the Entrenchment Parameter β

From Theorem 2, we know the stationary opinion vector $\mathbf{y}$ of our model polarizes when $\beta > \frac{1}{[\min|\mathbf{y}(0)|]^2}$, and reaches consensus when $\beta < \frac{1}{[\max|\mathbf{y}(0)|]^2}$. These thresholds are far away apart. In practice, the transition between consensus and polarization occurs at a value much lower than $\frac{1}{[\min|\mathbf{y}(0)|]^2}$ and higher than $\frac{1}{[\max|\mathbf{y}(0)|]^2}$. We now take the Karate network as an example and examine the relation between β and polarization experimentally using random initial opinion vectors.

Let β^P denote the threshold between consensus and polarization for an opinion vector - the smallest β that results in polarization. More specifically, what we observe is that consensus is reached when $\beta < \beta^P$ and the stationary opinions polarize when $\beta \geq \beta^P$. Since we do not restrict opinions to be only $-y_0$ and y_0 as in Corollary 1, there is no persistent disagreement observed in our experiments. Also, note that even though we assume the identical entrenchment parameter for all nodes in a network both in the theoretical and experimental analysis, the chances are people will have different levels of entrenchment in the real world. Fig 2.5(a) shows the distribution of the empirical β^P values for 10,000 different random opinion vectors, where each opinion is uniformly sampled between $[-1, 1]$. The value of β^P for each random opinion vector is found by grid search from 0 to 10 at a step size of 0.1. We observe that the threshold for polarization - β^P is much smaller than the theoretical value, which should be around 10^4 according to the sampled opinions having the minimum value around $1e-4$. However, on the Karate network, the empirical value of β^P is below 5 for most of the random $\mathbf{y}(0)$, and never exceeds 7.

We further study the opinion dynamics for one individual opinion vector from the 10,000 samples. Fig 2.5(b) shows the variance of its stationary opinions as a function of β. We observe that as β increases, the opinion vector converges from consensus to polarized states. The variance stays zero if there is consensus, while when the variance is greater than zero, polarization is obtained (i.e., different variances correspond to different polarized states). For this $\mathbf{y}(0)$, the transition from consensus to polarization happens at $\beta^P = 2.2$ and no persistent disagreement was observed.

When consensus is reached, Fig 2.5(c) shows that the consensus value becomes less neutral as β increases. This is true for 78.74% of the 10,000 random opinion vectors on the Karate network. Meanwhile, different values of β do not necessarily result in the same polarized state. The heatmap Fig 2.5(d) shows different polarized states with different values of β for this $\mathbf{y}(0)$, where each column corresponds to a specific value of β and each row to a specific node. The color indicates the node opinions with the dark blue being -1 and yellow being 1.

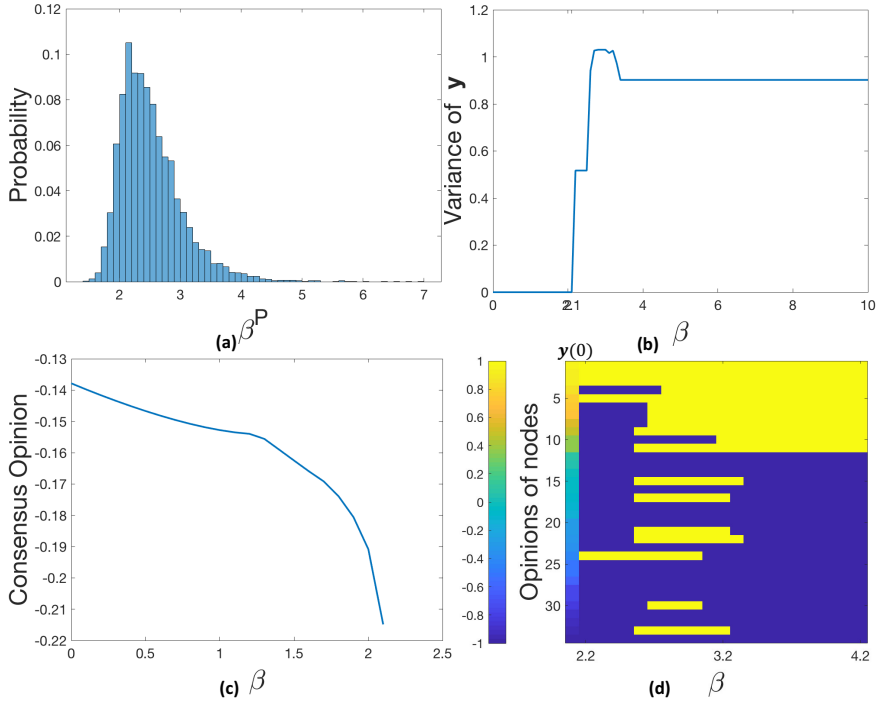


Figure 2.5: **For the Karate network.** *(a) the distribution of β^P for $10,000$ random opinion vectors (uniform on $[-1,1]$); For one of the opinion vectors, (b) the variance of all converged $\mathbf{y}$ as β increases from 0 to 10; (c) consensus opinion values for $\beta \in [0, 2.1]$; (d) final converged opinions for each of the nodes.*

2.5.2 The Influence of the Initial Opinions $\mathbf{y}(0)$

In this experiment, we investigate the influence of $\mathbf{y}(0)$ on the consensus opinion value and the mean polarized opinion. We observed that the consensus value as well as the mean polarized opinion are strongly correlated with the mean of $\mathbf{y}(0)$, as shown in Fig 2.6. Meanwhile, in the case of polarization (Fig 2.6(b)), opinion vectors with similar initial means may result in quite different polarized states because the placement of the opinions on the graph nodes differs. Also, $\mathbf{y}(0)$ with different means could result in similar polarized states with the same mean polarized opinion.

We also investigate the influence of the initial opinions on real-world dataset. Tw:Club with real opinions on whether Barcelona was getting the first place in La-liga 2016, and Tw: Sport with opinions on whether Juventus or Real Madrid is winning the Champions League final in 2015, have the same network but different opinion vectors [70], thus suitable for this evaluation. We found that the β^P is 11.7 for Tw:Club and 3.3 for Tw:Sport. The results indicate that the support behavior

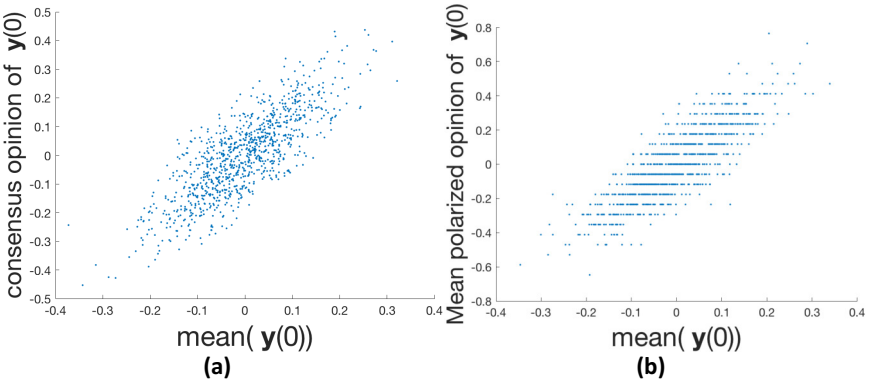


Figure 2.6: **For $1,000$ random $\mathbf{y}(0)$ on Karate network.** *(a) consensus opinion when $\beta = 1$; (b) mean polarized opinion when $\beta = 10$.*

for different football clubs in Tw: Sport gets polarized more easily than a single YES/No question of Tw:Club. With BEBA, we are able to quantify how easily people's opinions on an event may get polarized.

2.5.3 The Influence of the Network G

In this experiment, we study how the network topology affects the β^P value and the stationary opinions of our model. To this end, we generate random networks of the three models with the same number of nodes and similar number of edges, and intialize the same (set of) opinion vectors $\mathbf{y}(0)$ for them.

We observe that different network properties result in different dynamics of polarization. Shown in Fig 2.7(a) are the distributions of the β^P values on the three models for the same set of $\mathbf{y}(0)$. The BA model has a larger standard deviation of the β^P values, which appears to be due to 'hub' nodes whose opinions strongly affect the value of β^P. The ER model has similar mean of β^P to the BA model, which is larger than that of the WS model. As the WS model with the rewiring probability 1 essentially approaches the ER model, our WS network with less randomness (i.e., a rewiring probability of 0.2) in Fig 2.7(a) shows a tendency to get polarized more easily than the ER model. It indicates that, for the same set of opinion vectors on different issues, the more randomness the network has, the more robust the network is against polarization. To further verify this, we do similar experiments with the same set of opinion vectors on the WS models with more rewiring probabilities of 0.1, 0.3, and 0.8, see Fig 2.8(a). It shows that as the rewiring probability of the WS model increases, the mean of β^P becomes larger, which confirms our observation that the randomness in networks correlates with the networks' resilience against polarization.

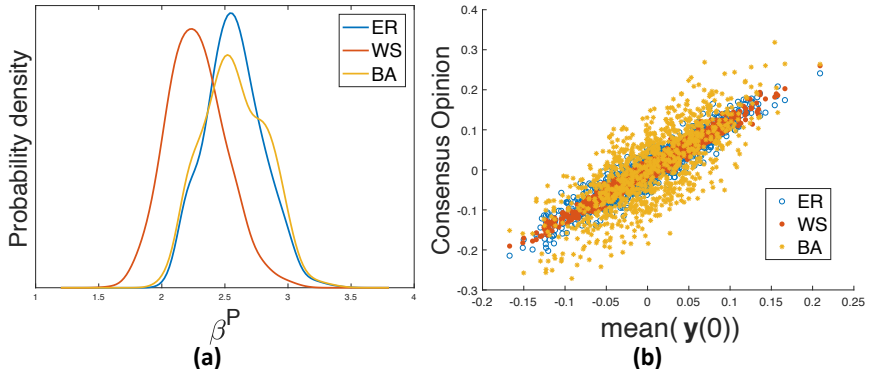


Figure 2.7: **Based on one ER model** ($n = 100, \rho = 0.0606$)**, one WS model** ($n = 100, K = 6, \sigma = 0.2$)**, and one BA model** ($n = 100, M_0 = 4, M = 3$)**.** *(a) distribution of* β^P *for* $1,000$ *random opinion vectors; (b) for* $1,000$ *opinion vectors, the relation between the consensus value and the mean* $\mathbf{y}(0)$ *when* $\beta = 1$.

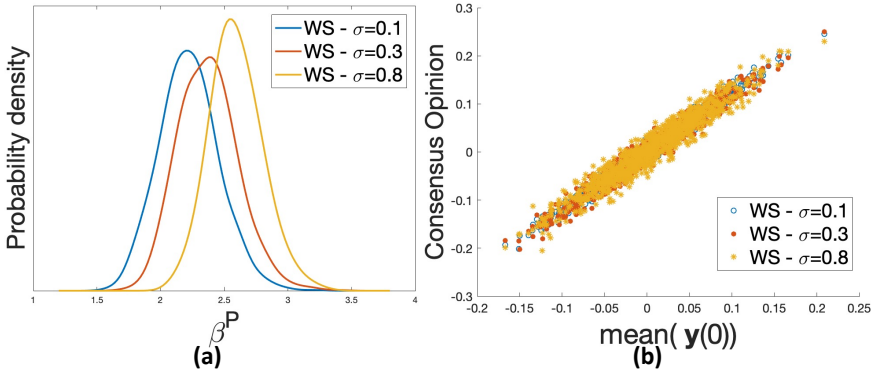


Figure 2.8: **Based on three WS models with different rewiring probabilities** ($n = 100, K = 6, \sigma = 0.1, 0.3, 0.8$)**.** *(a) distribution of* β^P *for* $1,000$ *random opinion vectors; (b) for* $1,000$ *opinion vectors, the relation between the consensus value and the mean* $\mathbf{y}(0)$ *when* $\beta = 1$.

The consensus values reached by the same set of opinion vectors on the three types of networks are plotted in Fig 2.7(b). The shapes of scatter plots become increasingly compact from the BA model, to the ER model, and then to the WS model, indicating the largest and the smallest variance on the consensus opinions for the BA and the WS network, respectively. The large variance for the BA mode is caused by the 'hub' nodes. Comparing to the ER mode, the WS mode here does not have much randomness, thus its consensus opinion varies the least. Fig 2.8(b) also confirms that the WS model with a smaller rewiring probability (i.e., less randomness) has a more compact shape. Similar to the results shown in Fig 2.6,

we also compare the influence of $\mathbf{y}(0)$ on three different types of random networks. The finding is consistent with that of Fig 2.7(b), see Fig 2.14 in Appendix 2.C.

The placement of the edges and the parameters in each model also affect the opinion dynamics. We take the ER model as the example and investigate the influence of G with a fixed and a changing ρ for one random opinion vector. On $1,000$ ER networks with $\rho = 0.4$, the β^P as well as the consensus opinion for that opinion vector vary, see Fig 2.9. If we increase ρ from a small value, which still guarantees a connected network, to 1, we observe quite different β^P for that opinion vector even with similar values of ρ. When ρ gets closer to 1, meaning that the network gets more connected, β^P becomes more stable, see Fig 2.10. The results are similar for the consensus value, and the polarized opinion.

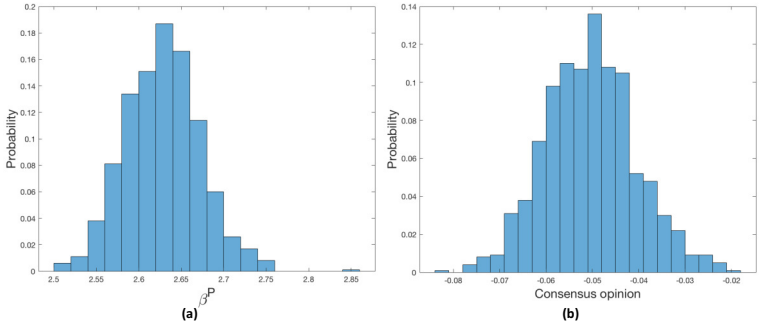


Figure 2.9: **For a random opinion vector $\mathbf{y}(0)$ with mean -0.0395, on $1,000$ ER models with $n = 100$ and $\rho = 0.4$.** *(a) the value of β^P for that $\mathbf{y}(0)$; (b) the consensus opinion reach by $\mathbf{y}(0)$ when $\beta = 1$.*

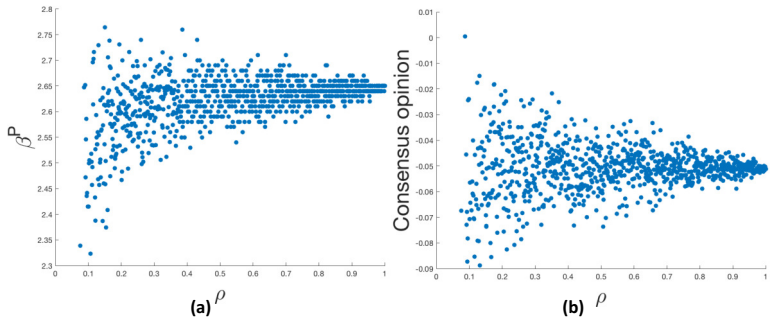


Figure 2.10: **For a random opinion vector $\mathbf{y}(0)$, on ER models with $n = 100$ and $\rho \in (0, 1]$.** *(a) the value of β^P for that $\mathbf{y}(0)$; (b) the consensus opinion reach by $\mathbf{y}(0)$ when $\beta = 1$.*

2.5.4 Real-world Dataset Analysis

Using the six real-world twitter datasets [70, 71], we investigate how easily each event gets polarized opinions, namely the value of β^P. It is shown in Table 2.2 that political events concerning elections in the first row (Tw:UK for the British election 2015, Tw:Delhi for the Delhi Assembly election 2013, and Tw:US for the US Presidential election 2016) are less likely to polarize since they require a relatively high β^P. However, the 2016 US presidential election shows a tendency to get polarized more easily than the other two elections with a lower β^P. On the other hand, the TV (Tw:GoT for the promotion of the TV show 'Games of Thrones' in 2015) and sport (Tw:Sport) events are more likely to get polarized, except when people have to bet on a result (Tw:Club) instead of supporting.

Table 2.2: β^P **for real-world twitter datasets.**

Network	β^P	Network	β^P	Network	β^P
Tw:UK	7.5	Tw:Delhi	7.7	Tw:US	4.9
Tw:GoT	2.9	Tw:Sport	3.3	Tw:Club	11.7

2.5.5 Opinion Manipulation under BEBA.

We also investigate the following question as a potential application of BEBA on opinion manipulation: how will the opinion dynamics be influenced by edge addition or deletion in networks? We use the Karate network to study it experimentally.

We observe that in order to maximally decrease the consensus opinion by editing one edge, adding the edge between the most opinionated disconnected negative nodes is the best choice if allowed a single edge addition; while deleting the edge between the most opinionated connected positive nodes is the best choice if allowed a single edge deletion. Similarly, the maximal decrease of the consensus value can be achieved by adding the edge between the most positively opinionated nodes or deleting the edge between the most negatively opinionated nodes. See Figs 2.11 and 2.12.

Another interesting finding is that the connections between nodes with opposing equivalent opinions (i.e., in terms of absolute value) have almost no influence on the consensus value, see Fig 2.13. In contrast, when the network gets polarized, the neighbors of the neutral nodes have more significant influence on the mean polarized opinions.

2.6 Conclusion and Future work

Modeling how opinions evolve when individuals interact in social networks is an important computational social science challenge that has received renewed atten-

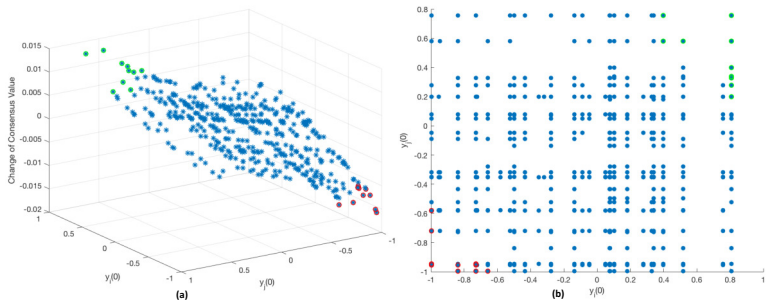


Figure 2.11: **Add one edge on Karate network to change the consensus opinion - $\beta = 1$.** *Top 10 best choices are highlighted: green for increase and red for decrease.*

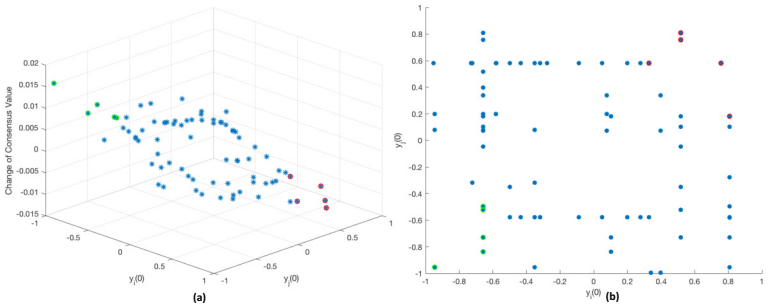


Figure 2.12: **Delete one edge on Karate network to change the consensus opinion - $\beta = 1$.** *Top 5 best choices are highlighted: green for increase and red for decrease.*

tion recently. The availability of realistic models of this type may have substantial real-life impact on a variety of applications, from political campaign design, to conflict prevention and mitigation. A large number of models have been proposed in the literature towards this end. To the best of our knowledge, however, none of them model the so-called Backfire Effect: the fact that individuals, when exposed to a strongly opposing view, will not be moderated, but rather become more entrenched in their opinion.

Here we proposed the BEBA model, which models both Biased Assimilation and Backfire Effect. It is governed by one parameter (which can vary over the individuals), called the entrenchment parameter, determining the strength of both. The BEBA model naturally generates different behaviors: from convergence to a consensus, to polarization. Theoretical and empirical analyses demonstrate that the resulting model is not only practical, its behavior also provides an interesting view on the interplay between network structure, the entrenchment parameter, and the opinions.

These properties make BEBA a useful tool for simulating the effect of inter-

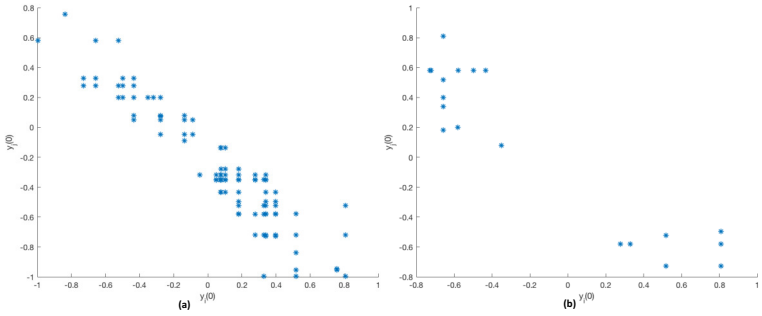


Figure 2.13: **Influence of edge edition on consensus.** *(a) Additions and (b) Deletions that cause minor change (i.e., $< 10^{-3}$) in consensus values on Karate network.*

ventions, such as editing the network (e.g. by facilitating communication between particular pairs of individuals), altering the initial opinions (e.g. through targeted information campaigns), or affecting the entrenchment of particular individuals (e.g. through education). It has the potential to help with correcting the misinformation in the real world.

However, BEBA has its limitations. For example, it would be interesting to investigate a variant of the model where the updated opinions naturally fall into the range $[-1, 1]$ without the clipping we applied in Eq (2.5). Also, it would be interesting to explore the different parameters for the Backfire Effect and Biased Assimilation. We plan to explore these directions in the future.

Acknowledgment

We thank the authors of [70, 71] for sharing the real-world Twitter datasets.

Appendices

2.A Proof of Theorem 1

Only one node in the environment

Recall that there is one node with a fixed opinion $p \in [-1, 1]$ in the environment. The opinion of the agent is updated as mentioned in Eq. (2.6),

Lemma 3. *If $w + \beta p y(t) + 1 \leq 0$, the opinion of the agent stays at $\operatorname{sgn}(y(t))$ for all $t' > t$.*

Proof. As shown in the updating rule that when $w + \beta p y(t) + 1 \leq 0$, $y(t+1) = \operatorname{sgn}(y(t))$. $w + \beta p y(t) + 1 \leq 0$ is equivalent to $\beta p y(t) \leq -w - 1 < 0$. Knowing

that $|y(t+1)| = 1 \geq |y(t)|$,

$$\beta p y(t+1) \leq -w - 1.$$

Therefore, $y(t') = \mathrm{sgn}(y(t+1)) = \mathrm{sgn}(y(t))$ for all $t' > t$. □

Lemma 4. *If $w+\beta p y(t)+1 > 0$, there exist two fixed points where $y(t+1) = y(t)$: p and $-\frac{1}{\beta p}$. p is attracting while $-\frac{1}{\beta p}$ is repelling.*

Proof. The converged opinion y of the agent should satisfy

$$f(y) = \frac{wy + \beta p^2 y + p}{w + \beta p y + 1},$$

$$f(y) - y = \frac{-\beta p y^2 + (\beta p^2 - 1)y + p}{w + \beta p y + 1} = \frac{u(y)}{v(y)} = 0, \tag{2.7}$$

where

$$u(y) = -\beta p y^2 + (\beta p^2 - 1)y + p,$$
$$v(y) = \beta p y + w + 1.$$

By solving $u(y) = 0$, which is equivalent to $f(y) - y = 0$ since $u(y) > 0$, the two fixed points of $f(y)$ are: p and $-\frac{1}{\beta p}$.

Next, we prove that p is attracting and $-\frac{1}{\beta p}$ is repelling.

$$f'(y) = \frac{w(w + \beta p^2 + 1)}{(w + \beta p y + 1)^2} \geq 0,$$

$|f'(y)| = f'(y)$, then $f'(p) = \frac{w}{w+\beta p^2+1} < 1$, thus attracting; while $f'(-\frac{1}{\beta p}) = \frac{w+\beta p^2+1}{w} > 1$, thus repelling. □

Lemma 5. *If $w + \beta p y(t) + 1 > 0$ and $p y(t) \geq 0$, $y = p$.*

Proof. If $p = 0$, $y(t+1) = \frac{w}{w+1} y(t)$, as the iteration goes, $\lim_{t\to\infty} y(t) = 0$;

If $p y(t) > 0$, e.g., they are both positive

- when $0 < y(t) < p$, $y(t+1) - y(t) = \frac{u(y(t))}{v(y(t))} > 0$, thus $y(t+1) > y(t)$, the agent's opinion increases until it reaches p;
- when $p < y(t) < 1$, $y(t+1) - y(t) < 0$, the agent's opinion decreases to p.

□

Lemma 6. *If $w + \beta p y(t) + 1 > 0$ and $p y(t) < 0$,*

1. *If* $\left|\frac{1}{\beta p}\right| > 1$, $\lim_{t\to\infty} y(t) = p$.
2. *If* $\left|\frac{1}{\beta p}\right| \leq 1$,
   (a) *If* $|y(t)| < \left|\frac{1}{\beta p}\right|$, $y = p$;
   (b) *If* $y(t) = -\frac{1}{\beta p}$, $y(t') = -\frac{1}{\beta p}$ *for all* $t' \geq t$;
   (c) *If* $\left|\frac{1}{\beta p}\right| < |y(t)| \leq 1$, $y = \text{sgn}(y(t))$.

Proof. Assume $y(t) \in (0, 1]$ and $p \in (-1, 0)$,

- if $\left|\frac{1}{\beta p}\right| > 1$, all $y(t) \in (0, 1] < -\frac{1}{\beta p}$, $y(t)$ is attracted to p as the updating goes;
- if $\left|\frac{1}{\beta p}\right| = 1$, $y(t)$ is repelled by the extreme point and goes to the attracting one unless it starts with $-\frac{1}{\beta p}$ at time t;
- if $\left|\frac{1}{\beta p}\right| < 1$, when $0 < y(t) < -\frac{1}{\beta p}$, $y(t+1) - y(t) = \frac{u(y(t))}{v(y(t))} < 0$, $y(t+1) < y(t)$, the agent's opinion decreases to p; when $y(t) = -\frac{1}{\beta p}$, $y(t)$ stays there; when $y(t) > -\frac{1}{\beta p}$, $y(t+1) > y(t)$, the agent's opinion increases to the extreme value on its side.

□

A group of nodes in the environment

Assume there is a set of m neighbour with different fixed opinions, $\mathbf{p} = (p_1, p_2, ..., p_m)$, $m > 1$. We denote

- $q = \sum_j p_j^2$ the sum of the squares of the fixed opinions.
- $s = \sum_j p_j$ the sum of the fixed opinions.
- $m = \sum_j 1$ the number of nodes in the environment.

Lemma 7. $mq - s^2 \geq 0$, *which is* $m\sum_j p_j^2 \geq (\sum_j p_j)^2$.

Proof.

$$m\sum_j p_j^2 - (\sum_j p_j)^2 = \frac{1}{2}\sum_i \sum_j (p_i - p_j)^2 \geq 0.$$

□

The agent's opinion is updated by

$$y(t+1) = \begin{cases} \text{sgn}(y(t)) & \text{if } w + \beta s y(t) + m \leq 0, \\ \frac{wy(t) + \beta q y(t) + s}{w + \beta s y(t) + m} & \text{otherwise.} \end{cases} \tag{2.8}$$

Lemma 8. *If $w + \beta s y(t) + m > 0$, there exist two fixed points where $y(t+1) = y(t)$:*

$$y^a = \frac{\beta q - m + \sqrt{\Delta}}{2\beta s}, \quad y^r = \frac{\beta q - m - \sqrt{\Delta}}{2\beta s},$$

where $\Delta = (\beta q - m)^2 + 4\beta s^2$. y^a is attracting while y^r is repelling.

Proof. The function is $f(y) = \frac{wy + \beta q y + s}{w + \beta s y + m}$. The two fixed points satisfy $f(y) = y$. $|f'(y)| = f'(y)$ since

$$\begin{aligned} f'(y) =& \frac{(w + \beta q)(w + m) - \beta s^2}{(\beta s y + w + m)^2} \\ =& \frac{w(w+m) + \beta q w + \beta(qm - s^2)}{(\beta s y + w + m)^2} > 0. \end{aligned}$$

For $y^a = \frac{\beta q - m + \sqrt{\Delta}}{2\beta s}$, $f'(y^a) < 1$ because

$$\begin{aligned} f'(y^a) - 1 =& -\frac{1}{2}\frac{(m - \beta q)^2 + 4\beta s^2 + (2w + m + \beta q)\sqrt{\Delta}}{(\beta s y^a + w + m)^2} \\ <& 0. \end{aligned}$$

For $y^r = \frac{\beta q - m - \sqrt{\Delta}}{2\beta s}$, $f'(y^r) > 1$ because

$$\begin{aligned} f'(y^r) - 1 =& -\frac{1}{2}\frac{(m - \beta q)^2 + 4\beta s^2 - (2w + m + \beta q)\sqrt{\Delta}}{(\beta s y^r + w + m)^2} \\ =& -\frac{1}{2}\frac{A}{B}. \end{aligned}$$

$\frac{A}{B} < 0$ since $B > 0$ and it can be proved as below that $A < 0$.

$$\begin{aligned} & \left[(m - \beta q)^2 + 4\beta s^2\right]^2 - \left[(2w + m + \beta q)\sqrt{\Delta}\right]^2 \\ =& 4\left[(m - \beta q)^2 + 4\beta s^2\right]\left[\beta(s^2 - qm) - w(m + w + \beta q)\right] \\ <& 0. \end{aligned}$$

Therefore, y^a is attracting and y^r is repelling. □

2.B Proof of Theorem 2

Recall that $y_i(t) \in (-1, 0) \cup (0, 1)$. Given any opinion vector $\mathbf{y}(0)$ of a given connected network $G = (V, E)$, the opinions can be divided into two groups V_1 and V_2 at any time t: a) $\forall i \in V_1$, $y_i(t) > 0$; b)$\forall i \in V_2$, $y_i(t) < 0$, and $V =$

$V_1 \cup V_2$. Denote $n_i^s(t)$ the number of node i's neighbors node that are in the **s**ame group with i at time t, and $n_i^d(t)$ the number of neighbors in the **d**ifferent group. Specifically, they are denoted as

$$n_i^s(t) = |N(i)^s|, N(i)^s = \{j | j \in N(i), \text{ and } y_i(t)y_j(t) > 0\},$$
$$n_i^d(t) = |N(i)^d|, N(i)^d = \{k | k \in N(i), \text{ and } y_i(t)y_k(t) < 0\}.$$

Lemma 9. *For node $i \in V$ fix $\beta_i = \beta > 0$, if $\beta > \frac{1}{[\min(|\mathbf{y}(0)|)]^2}$, $\lim_{t\to\infty} |y_i(t)| = 1$.*

Proof. For node $i \in V$, the opinion is updated with BEBA. If $\gamma = 1 + \sum_{j\in N(i)} w_{ij} \leq 0$, $y_i(t+1)$ reaches the extreme value in one iteration due to strong backfire effect.

While when $\gamma > 0$, for any $t > 0$, $y_i(t+1)$ is updated as

$$y_i(t)\frac{1 + \sum_{j\in N(i)^s} w_{ij}\frac{y_j(t)}{y_i(t)} + \sum_{k\in N(i)^d} w_{ik}\frac{y_k(t)}{y_i(t)}}{1 + \sum_{j\in N(i)^s} w_{ij} + \sum_{k\in N(i)^d} w_{ik}} = y_i(t)\frac{C}{D}. \tag{2.9}$$

When $\beta > \frac{1}{[\min(|\mathbf{y}(t)|)]^2}$, for all $k \in N(i)^d$, $w_{ik} = \beta y_i(t)y_k(t) + 1 < 0$. The sums in Eq. (2.9) satisfy: $\sum_{j\in N(i)^s} w_{ij}\frac{y_j(t)}{y_i(t)}$, $\sum_{j\in N(i)^s} w_{ij}$, $\sum_{k\in N(i)^d} w_{ik}\frac{y_k(t)}{y_i(t)} > 0$, and $\sum_{k\in N(i)^d} w_{ik} < 0$.

Now we focus on the node that has the most moderate opinion, namely the node with absolute value of opinion $\min|\mathbf{y}(t)|$ at each time step, starting from time 0. Knowing $C, D > 0$,

$$C - D = \sum_{j\in N(i)^s} w_{ij}(\frac{y_j(t)}{y_i(t)} - 1) + \sum_{k\in N(i)^d} w_{ik}(\frac{y_k(t)}{y_i(t)} - 1). \tag{2.10}$$

Since $y_i(t)$ has the smallest absolute opinion value, for any $j \in N(i)^s$, $\frac{y_j(t)}{y_i(t)} \geq 1$, thus $C > D$, $\frac{C}{D} > 1$, and $|y_i(t+1)| > |y_i(t)|$.

After every iteration from time t to $t+1$, the opinion of the most moderate node becomes more extreme, until it reaches the absolute value of 1, thus for any $i \in V$, $\lim_{t\to\infty} |y_i(t)| = 1$. □

Lemma 10. *For node $i \in V$, if $\beta < \frac{1}{[\max(|\mathbf{y}(0)|)]^2}$, there exists a unique $y^* \in [-\max(|\mathbf{y}(0)|), \max(|\mathbf{y}(0)|)]$ such that $\lim_{t\to\infty} y_i(t) = y^*$ for all $i \in V$.*

Proof. When $\beta < \frac{1}{[\max(|\mathbf{y}(0)|)]^2}$, $\gamma = 1 + \sum_{j\in N(i)} w_{ij} > 0$ because for any $j \in N(i)$, $w_{ij} = \beta y_i(t)y_j(t) + 1 > 0$.

For any $t > 1$, $y_i(t+1)$ is updated as in Eq. (2.9), however, the sums have different values: $\sum_{j\in N(i)^s} w_{ij}\frac{y_j(t)}{y_i(t)}$, $\sum_{j\in N(i)^s} w_{ij}$, $\sum_{k\in N(i)^d} w_{ik} > 0$, and $\sum_{k\in N(i)^d} w_{ik}\frac{y_k(t)}{y_i(t)} < 0$.

Then we focus on the most opinionated node, which means the node has the largest absolution value of its opinion $\max |\mathbf{y}(t)|$, starting from time 0. Knowing $D > 0$,

- when $C > 0$, $C - D$ is shown in Eq. (2.10). With i being the most opinionated node, $\frac{y_j(t)}{y_i(t)} \leq 1$ for all $j \in N(i)^s$; $\frac{y_k(t)}{y_i(t)} < 0$ for all $k \in N(i)^d$. Therefore, $C < D$, $0 < \frac{C}{D} < 1$ and $|y_i(t+1)| < |y_i(t)|$.
- when $C = 0$, $y_i(t+1) = 0$.
- when $C < 0$, $-C - D$ is shown in Eq. (2.11). As $-1 \leq \frac{y_k(t)}{y_i(t)} \leq 0$ for $k \in N(i)^d$, $-C - D < 0$, $0 < \left|\frac{C}{D}\right| < 1$, thus $|y_i(t+1)| < |y_i(t)|$.

$$-2 - \sum_{j \in N(i)^s} w_{ij}(\frac{y_j(t)}{y_i(t)} + 1) - \sum_{k \in N(i)^d} w_{ik}(\frac{y_k(t)}{y_i(t)} + 1). \tag{2.11}$$

At every time step, the most opinionated node get moderated until they reach consensus - there is no such node and the updating process stops because consensus is reached. □

Lemma 11. *For node $i \in V_1$, $y_i(0) = y_0$, where $0 < y_0 < 1$; $\forall i \in V_2$, $y_i(0) = -y_0$. If $\beta = \frac{1}{y_0^2}$, $y_i(t) = y_i(0)$ for all $t \geq 0$.*

Proof. When $\beta = \frac{1}{y_0^2}$, $w_{ij} = \frac{1}{y_0^2} y_i(t) y_j(t)$. At time 1,

$$y_i(1) = \frac{y_i(0) + 2n_i^s(0) y_i(0)}{1 + 2n_i^s(0)} = y_i(0).$$

For any $t \geq 1$,

$$y_i(t+1) = \frac{y_i(t) + 2n_i^s(t) y_i(t)}{1 + 2n_i^s(t)} = y_i(t) = y_i(0).$$

□

2.C Supporting Figure

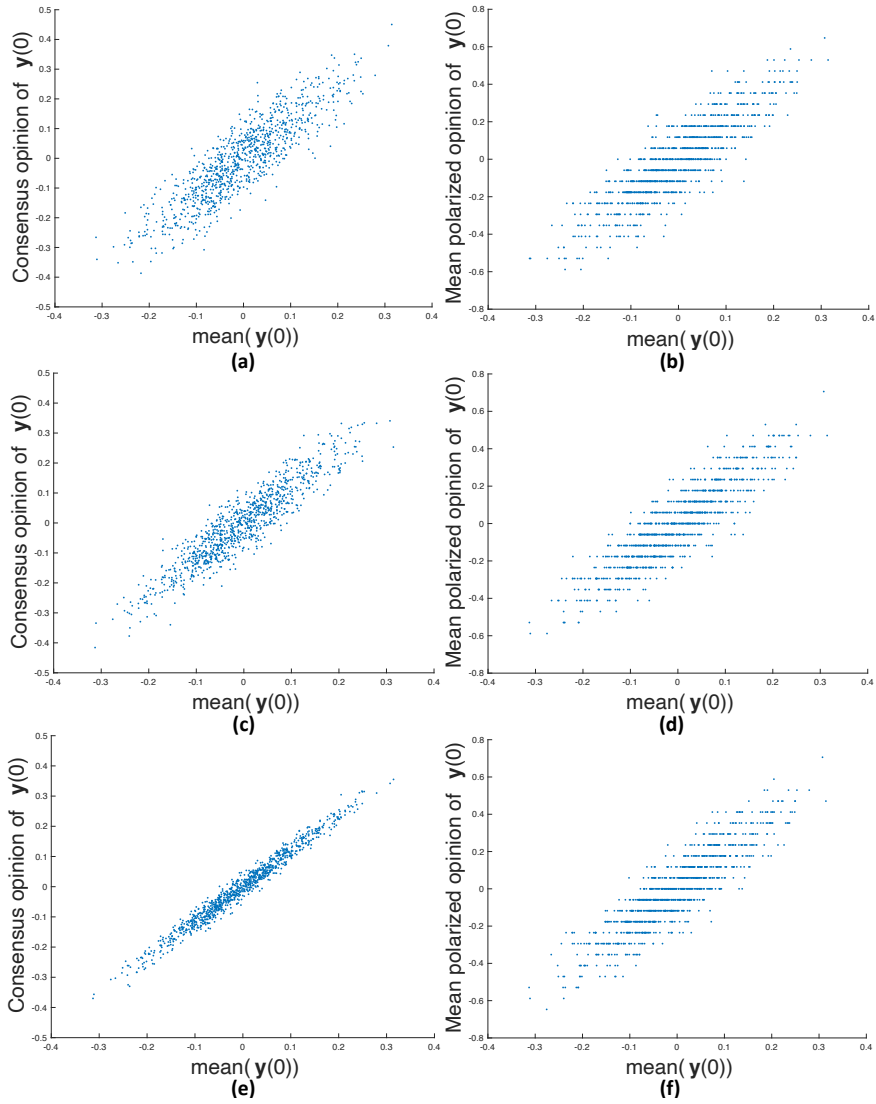


Figure 2.14: **For** $1,000$ **random** $\mathbf{y}(0)$**.** *(a) and (b) on a BA model ($n = 34, M_0 = 3, M = 2$); (c) and (d) on an ER model ($n = 34, \rho = 0.139$); (e) and (f) on a WS model ($n = 34, K = 4, \sigma = 0.2$). The left column of (a), (c), (e) - the relation between the consensus opinion and the mean $\mathbf{y}(0)$ when $\beta = 1$; the right column of (b), (d), (f) - the relation between the mean polarized opinion and the mean $\mathbf{y}(0)$ when $\beta = 10$.*

3

The Normalized Friedkin-Johnsen Model

Abstract The formation of opinions in a social context has long been studied by sociologists. A well-known model is due to Friedkin and Johnsen (further referenced as the *FJ model*), which assumes that individuals hold an immutable *internal opinion* while they *express* an opinion that may differ from it but is more in agreement with the expressed opinions of their friends. Formally, the *expressed opinion* is modeled as the weighted average of the individual's internal opinion and the expressed opinions of their neighbors. This model has been used in recent research originating from the computer science community, studying the origination and reduction of conflict on social networks, how echo chambers arise and can be burst, and more.

Yet, we argue that the FJ model in its elementary form is not suitable for some of these purposes. Indeed, the FJ model entails that the more friends one has, the less one's internal opinion matters in the formation of one's expressed opinion. Arguing that this may not be realistic, we propose a modification of the FJ model that normalizes the influence of one's friends and keeps the influence of one's internal opinion constant. This normalization was in fact suggested by Friedkin and Johnsen, but it has been ignored in much of the recent computer science literature.

In this chapter, we present the details of the normalized model, and investigate the consequences of this normalization, both theoretically and empirically.

3.1 Introduction and Motivation

How people form their opinions has long been the subject of research in the field of social sciences [1, 2]. More recently, such models for opinion formation and dynamics (e.g., [24]) have been used by computer scientists and computational social scientists to study how to quantify and control notions of controversy, disagreement, polarization and conflict on social networks [6,7], e.g. by manipulating the opinions of a small set of particular individuals, or by locally changing the network structure [8–10]. Opinion formation models serve as the fundamental part of these studies.

Background. Many opinion formation models have been proposed and studied based on the influence through social interactions [13–18]. The Friedkin-Johnsen (FJ) Model [2] is a very popular extension of the DeGroot Model [1] that is used often [9, 10, 23]. In the model, individuals are assumed to have two types of opinions: the *internal opinion* and the *expressed opinion*. The internal opinions are assumed to be immutable, and represent individuals' innate opinion about matters. In the absence of any influence by others, this is the opinion an individual would express. However, the actual expressed opinion *will* be affected by one's friends/neighbors (e.g. due to a desire for social acceptance), and is modeled as the weighted average of the individual's own internal opinion and their neighbors' expressed opinions. The opinions are formed through continuous averaging in the model. Later on, the expressed opinion vector in FJ Model was interpreted as the Nash equilibrium in the social game of opinion formation, in which people get social costs as payoffs [24].

Motivation. A feature of the FJ model is that an individual's internal opinion matters less the more friends that individual has (or the stronger those friendships are). This may not be realistic, and for this reason Friedkin and Johnsen themselves suggested that the influence of a friend's expressed opinion on one's own expressed opinion should be normalized [2]. This would ensure that the relevance of one's internal opinion is independent on the number of friends and strength of these friendships.

Yet, this normalization, which is important in particular in studies that investigate how to engineer the connectivity of the network so as to achieve a certain goal (e.g. reducing some measure of conflict, maximizing some measure of influence, etc.), is often ignored in recent work.

In this chapter, we study the relevance of the normalization. First, we make the normalization explicit by proposing a minor variant of the FJ model: the Normalized Friedkin-Johnsen (NFJ) model. Then, we investigate theoretically how NFJ Model differs qualitatively from the FJ model. In particular, we focus on a recently discovered conservation law of conflict [23], which stated that for opinions that follow the FJ model, the sum of measures for internal conflict, external

conflict, and controversy sums to a constant. We show that this conservation law no longer holds under the NFJ model, which provides an opportunity for eliminating conflict. Finally, we investigate empirically how the NFJ and FJ models yield different quantifications for important measures of conflict.

3.2 The Normalized Friedkin-Johnsen Model, and a Theoretical Analysis

This section contains the details of the proposed model, but first we need to introduce some notation.

Notation. Let $G = (V, E, w)$ be a network, where $V = \{1, \ldots, n\}$ is the set of nodes, $E \in V \times V$ is the set of $m = |E|$ edges, and w is a weight function mapping an edge $e \in E$ onto its weight $w(e) \geq 0$. We denote with $\mathbf{W}$ the weighted adjacency matrix (with zero diagonal), defined by $w_{ij} = w(i, j)$ iff $\{i, j\} \in E$ and $w_{ij} = 0$ otherwise. With $N(i)$ we denote the set of neighboring nodes of node i: $N(i) \triangleq \{j \in V | (j, i) \in E\}$ (i.e., node j is a friend who has influence on node i in social networks). Let $\boldsymbol{e}$ denote the vector of ones of appropriate size. Furthermore, let $\mathbf{d} \triangleq \mathbf{W}^T\mathbf{e}$ denote the vector containing the weighted (in-)degrees of all nodes, and $\mathbf{D} \triangleq \text{diag}(\mathbf{d})$ the diagonal degree matrix. Then the Laplacian matrix is defined as $\mathbf{L} \triangleq \mathbf{D} - \mathbf{W}$. Note here the notations are related to in-degrees of nodes in directed networks, and they correspond to degrees (either in-degree or out-degree) for undirected networks.

3.2.1 The Normalized Friedkin-Johnsen model

Before discussing the NFJ model, we first discuss two logical predecessors: a model due to DeGroot, and the vanilla FJ model.

The DeGroot model [1] formalizes opinion formation as a repeated averaging process of one's opinion with one's neighbors. In the model, every person $i \in V$ updates his/her opinion $s_i(t + 1)$ at time $t + 1$ as the weighted sum of their own opinion (with weight w_{ii}) and those of the neighbours (with weight w_{ij} for neighbor j) at time t. Note that w_{ii} is independent from any w_{ij}, and represents the node's believe in its own opinion. Given an undirected weighted graph $G = (V, E, w)$, the updating rule is defined as:

$$s_i(t + 1) = \frac{w_{ii}s_i(t) + \sum_{j \in N(i)} w_{ij}s_j(t)}{w_{ii} + \sum_{j \in N(i)} w_{ij}}. \tag{3.1}$$

In 1990, Friedkin and Johnsen extended the DeGroot model to have two different kinds of opinions [2]: a fixed internal opinion s_i, which is private to each individual, and a public expressed opinion z_i. The expressed opinions are the

weighted sum of the node's own internal opinion and the expressed opinions of the neighbors:

$$z_i = \frac{w_{ii}s_i + \sum_{j\in N(i)} w_{ij}z_j}{w_{ii} + \sum_{j\in N(i)} w_{ij}}. \tag{3.2}$$

Expressed in matrix-vector notation, and with $w_{ii} = 1$ (a common assumption in the literature), this equation is solved by (3.3) below at equilibrium [24]:

$$\mathbf{z} = (\mathbf{L} + \mathbf{I})^{-1}\,\mathbf{s}. \tag{3.3}$$

When $w_{ii} = 1$, $z_i + \sum_{j\in N(i)} w_{ij}(z_i - z_j) = s_i$, which is $\mathbf{z} + \mathbf{L}\mathbf{z} = (\mathbf{I} + \mathbf{L})\mathbf{z} = \mathbf{s}$ in matrix-vector form. Therefore, given fixed $\mathbf{s}$, $\mathbf{z}$ is solved as in Equation (3.3).

In the proposed NFJ Model, we consider that the influence from neighbors should be normalized by the number of neighboring nodes or the total strength of the incident edges, because people's internal opinions will not be less important if they have more friends. We discuss it for directed graphs – undirected graphs can be regarded as a special case (note that most of the existing literature focuses on undirected networks only). In directed networks, only the incoming edges contribute to the opinion formation process. We consider edge $(i, j) \in E$ as the edge from node i to node j, so the element $d_i \triangleq \sum_{j\neq i} w_{ji}$ of $\mathbf{d}$ is the (weighted) in-degree of node i.

In the proposed NFJ Model, the expressed opinion is updated as follows:

$$z_i = \begin{cases} s_i, & \text{if } d_i = 0 \\ \frac{as_i + \frac{\sum_{j\in N(i)} w_{ji}z_j}{d_i}}{a+1}, & \text{otherwise.} \end{cases} \tag{3.4}$$

Thus, in the NFJ model, it is assumed that each node puts the same weight $w_{ii} = a$ (instead of $w_{ii} = 1$) on its internal opinion, independently of the network weights, i.e., independently of the number and weights of incoming edges. Note that when $d_i = 1$, the node follows exactly the updating rule in the vanilla FJ Model. It is worth mentioning that a similar updating rule is also used by Abebe et al. for studying opinion dynamics with varying susceptibility to persuasion, which can be interpreted as different values of a for each node [47, 75]. However, our assumptions and focuses are different.

Assuming that $d_i \neq 0$ for all i, the set of linear Equations (3.4) is solved by Equation (3.5) in a similar way Equation (3.2) is solved, where $\mathbf{K} = \frac{1}{a}\mathbf{D}^{-1}\boldsymbol{L}^T$ is a normalized Laplacian:

$$\mathbf{z} = (\mathbf{K} + \mathbf{I})^{-1}\,\mathbf{s}. \tag{3.5}$$

3.2.2 Implications of the Normalization on the Quantification of Conflict in Networks

Based on FJ Model, several conflict measures have been proposed in the recent computer science literature. Four measures in particular were highlighted in [23]:

- Internal Conflict ic ($= \sum_i (s_i - z_i)^2$) quantifies the extend to which individuals' internal and expressed opinions differ.
- External Conflict ec ($= \sum_{(i,j)\in E} w_{ij} (z_i - z_j)^2$) quantifies the extend to which the expressed opinions of neighbors are in disagreement with each other.
- Controversy c ($= \sum_i z_i^2$) does not depend on the network structure, and simply quantifies how much the opinion varies across the individuals in the network.
- Resistance r ($= \sum_i s_i z_i$) is the inner product between expressed and internal opinion vectors, and also the sum of external conflict and controversy.

Matrix expressions for these quantities in terms of $\mathbf{s}$ and $\mathbf{z}$ are shown in Table 3.1. These measures were proposed for undirected networks. See more details of the measures in Section 4.3.1 of the following chapter.

Table 3.1: Conflict Measures based on FJ Model

Name	$\mathbf{z}$	$\mathbf{s}$
ic	$\mathbf{z}^T\mathbf{L}^2\mathbf{z}$	$\mathbf{s}^T(\mathbf{L}+\mathbf{I})^{-1}\mathbf{L}^2(\mathbf{L}+\mathbf{I})^{-1}\mathbf{s}$
ec	$\mathbf{z}^T\mathbf{L}\mathbf{z}$	$\mathbf{s}^T(\mathbf{L}+\mathbf{I})^{-1}\mathbf{L}(\mathbf{L}+\mathbf{I})^{-1}\mathbf{s}$
c	$\mathbf{z}^T\mathbf{z}$	$\mathbf{s}^T(\mathbf{L}+\mathbf{I})^{-2}\mathbf{s}$
r	$\mathbf{z}^T\mathbf{s}$	$\mathbf{s}^T(\mathbf{L}+\mathbf{I})^{-1}\mathbf{s}$

It was shown by Chen et al. that the first three together give rise to a conservation law of conflict [23], indicating that reducing one kind of conflict implies that another must be increased. Formally:

$$ic + 2ec + c = \mathbf{s}^T\mathbf{s}. \tag{3.6}$$

Note that the expressions in the right column of Table 3.1 are all quadratic forms $\mathbf{s}^T\mathbf{M}_*\mathbf{s}$ for some middle matrix $\mathbf{M}_*$ that depends on the conflict measure of interest $* \in \{ic, ec, c, r\}$ (e.g., $(\mathbf{L}+\mathbf{I})^{-1}\mathbf{L}(\mathbf{L}+\mathbf{I})^{-1}$ for ec). The middle matrices share the same eigenvectors with $\mathbf{L}$, and their eigenvalues can be expressed as a scalar function of the eigenvalues of $\mathbf{L}$ [23]. Figure 3.1 illustrates this relation, with λ representing an eigenvalue of $\mathbf{L}$ while λ_* is the eigenvalue of one of the

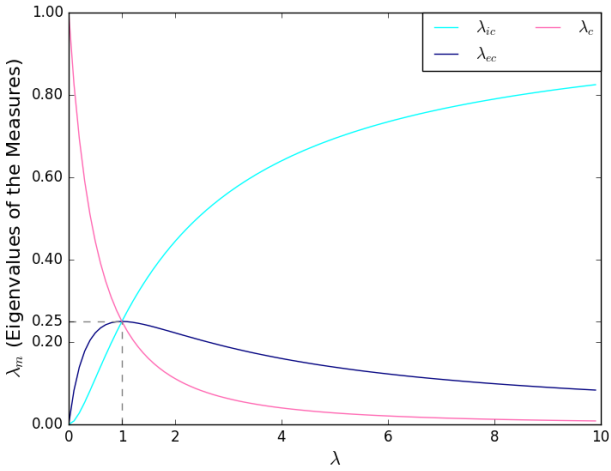


Figure 3.1: Conservation Law of Conflict [23]

middle matrices. The conservation law is reflected in a similar relation amongst the eigenvalues of the three middle matrices, and this for any eigenvalue λ of the Laplacian.

This figure also illustrates that for a given $\mathbf{s}$, a conflict measure will be larger if $\mathbf{s}$ aligns better with an eigenvectors of $\mathbf{L}$ for which λ_* is larger. Arguably the most interesting measure is ec, which increases at first and then decreases when $\mathbf{s}$ becomes less smooth (i.e., with larger eigenvalues)[1]. More intuitions concerning this can be found in Section 4.3.2 of the next chapter. We will discuss this in greater detail with experimental results in the next section.

As a first theoretical analysis of the NFJ model as compared with the FJ model, it is interesting to investigate whether this conservation law still holds in the NFJ model. We start from the conflict measures, and then investigate this only for directed networks. Referring to their definitions based on the FJ model, we define the three conflict measures in the conservation law as follows:

$$ic = \sum_i (s_i - z_i)^2, \quad ec = \frac{1}{a} \sum_{i,j} \frac{w_{ij}}{d_j} (z_i - z_j)^2, \quad c = \sum_i z_i^2$$

In the NFJ model, the conflict measures are very similar to the ones in [23] (i.e., ic and c stay the same). However, ec is different because the importance of the opinion differences over existing edges is also normalized by the in-degrees of the incident nodes. Based on Equation (3.5), the three measures in the new model are expressed in matrix-vector form as in Table 3.2, where $\mathbf{N}_{ec}$ is

[1] Here smooth/low-frequency represents that the close-by nodes hold similar opinions - corresponding to smaller eigenvalues, and high-frequency means nodes differ more with nodes around in their opinions, which corresponds with larger eigenvalues.

Table 3.2: Conflict Measures based on NFJ Model

Name	$\mathbf{z}$	$\mathbf{s}$
ic	$\mathbf{z}^T\mathbf{K}^T\boldsymbol{K}\mathbf{z}$	$\mathbf{s}^T\left(\mathbf{K}^T+\mathbf{I}\right)^{-1}\mathbf{K}^T\boldsymbol{K}\left(\mathbf{K}+\mathbf{I}\right)^{-1}\mathbf{s}$
ec	$\mathbf{z}^T\mathbf{N}_{ec}\mathbf{z}$	$\mathbf{s}^T\left(\mathbf{K}^T+\mathbf{I}\right)^{-1}\mathbf{N}_{ec}\left(\mathbf{K}+\mathbf{I}\right)^{-1}\mathbf{s}$
c	$\mathbf{z}^T\mathbf{z}$	$\mathbf{s}^T\left(\mathbf{K}^T+\mathbf{I}\right)^{-1}\left(\mathbf{K}+\mathbf{I}\right)^{-1}\mathbf{s}$

$$\mathbf{N}_{ec} = \frac{1}{a}\mathrm{diag}\left(\mathbf{D}^{-1}\mathbf{W}^T\mathbf{e}+\mathbf{W}\mathbf{D}^{-1}\mathbf{e}\right)-\frac{1}{a}\left(\mathbf{D}^{-1}\mathbf{W}^T+\mathbf{W}\mathbf{D}^{-1}\right). \tag{3.7}$$

The definition of ec in the new model is inspired by the conservation law of conflict. After finding that the conservation law no longer holds in the NFJ model, we introduce an additional term, denoted as x shown in Equation (3.8) below, such that the law can be restored. It is equivalent to finding two matrices $\mathbf{N}_{ec}$ and $\mathbf{N}_x$, which sum to $\mathbf{K}^T+\mathbf{K}$.

$$ic+ec+c+x=\boldsymbol{s}^T\boldsymbol{s} \tag{3.8}$$

So we have $\mathbf{N}_{ec}$ as in Equation (3.7), and $\mathbf{N}_x$ below

$$\mathbf{N}_x = \frac{1}{a}\mathrm{diag}\left(\mathbf{D}^{-1}\mathbf{W}^T\mathbf{e}-\mathbf{W}\mathbf{D}^{-1}\mathbf{e}\right), \tag{3.9}$$

$$x = \mathbf{z}^T\mathbf{N}_x\mathbf{z} = \frac{1}{a}\sum_i z_i^2\sum_{j\neq i}\left(\frac{w_{ji}}{d_i}-\frac{w_{ij}}{d_j}\right). \tag{3.10}$$

If x cannot be interpreted as a relevant measure of conflict, it can be seen as an opportunity for eliminating conflict: it is then conceivable that the network can be edited (e.g. by adding or removing edges, or by changing weights) so as to reduce all of ic, ec, and c while increasing x. I.e., the sum of the three conflict measures can be minimized by maximizing x. According to Equation (3.8), x can be expressed as in Equation (3.10). It shows that the network edits for conflict optimization (i.e., maximizing x) should consider both how opinionated nodes are (i.e., the values of z_i^2) and the importance of the node's influence on all its neighbors (i.e., the value of $\sum_{j\neq i}\frac{w_{ij}}{d_j}$ since $\sum_{j\neq i}\frac{w_{ji}}{d_i}=1$). Meanwhile, when it comes to comparing the amount of conflict between networks of similar sizes, x indicates that the more opinionated nodes are of minor importance in influencing their neighbors (i.e., small $\sum_{j\neq i}\frac{w_{ij}}{d_j}$), the less total conflict (i.e., $ic+ec+c$) there will be. The interpretation of x, and on how it can be maximized, are subject of our current research.

3.3 Discussion and Experiments

This section discusses the difference of the NFJ model to the original model, using synthetic as well as real-world networks, which are of varying sizes.

3.3.1 Opinion Formation

We start from a very simple network as shown in Figure 3.2, and assign each node with internal opinions where green means $s_i = 1$ and red represents $s_i = -1$. According to Table 3.3, node 1 and node 7, which are the centers of the two star-subgraphs, have expressed opinions opposite to the internal ones in the old model, while they remain on their "original side" in the new model. It is clear that the normalization can have a big impact in this opinion formation model, and it corresponds to the suggested assumption in the original FJ Model [2] as $\sum_j w_{ij} = 1$ and $w_{ij} \in [0, 1]$. Surprisingly, it is usually neglected in works based on this model.

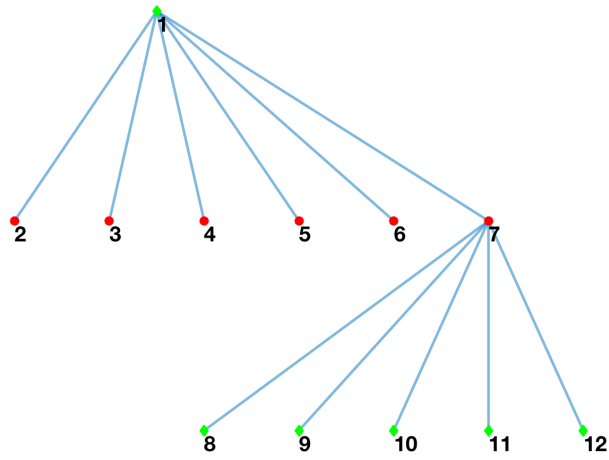


Figure 3.2: Network Example 1

Table 3.3: Expressed Opinions at Equilibrium ($a = 1$)

Node	1	2	3	4	5	6
s	1	-1	-1	-1	-1	-1
z_{FJ}	-0.27	-0.64	-0.64	-0.64	-0.64	-0.64
z_{NFJ}	0.33	-0.33	-0.33	-0.33	-0.33	-0.33
Node	7	8	9	10	11	12
s	-1	1	1	1	1	1
z_{FJ}	0.27	0.64	0.64	0.64	0.64	0.64
z_{NFJ}	-0.33	0.33	0.33	0.33	0.33	0.33

The NFJ model ensures that people value their own internal opinions with an importance independent of the environment, such that the number of friends or the strength of these friendships will not affect people's adherence to their own opinions.

3.3.2 Quantifying Conflict

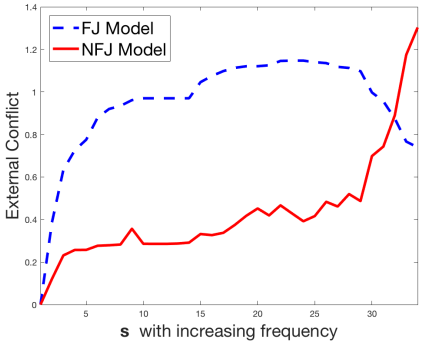
In addition to the conflict eliminator in the conservation law, we will give evidence that this normalized model is different from the original one in terms of conflict measures (i.e., here we focus on external conflict ec as it was arguably the most interesting measure [23]). This model seems to be better as it preserves the controversial discussion within social networks, instead of "diminishing" it with too much opinion averaging. We consider different sizes of synthetic random networks and real-world social networks: 1) the Karate network of friendships between 34 members [69]; 2) a Watts-Strogatz random network with the small world property of 500 nodes; 3) and a real-world Facebook social network containing friend circles [76] of 4039 nodes.

In the original FJ Model, external conflict increases first and then it decreases slowly as the eigenvalues of the Laplacian matrix $\mathbf{L}$ increases, shown in Figure 3.1. In other words, when the vector of internal opinions $\mathbf{s}$ aligns with the eigenvectors of increasing frequency, ec reaches the maxima at a certain point (i.e., $\lambda = 1$). However, the higher the frequency on the graph for $\mathbf{s}$, the more the conflicts there should be in the network because this is how controversy arose. It shows the real conflict between people holding "opposite" (potentially differing) opinions, because high-frequency $\mathbf{s}$ means more differences over existing edges.

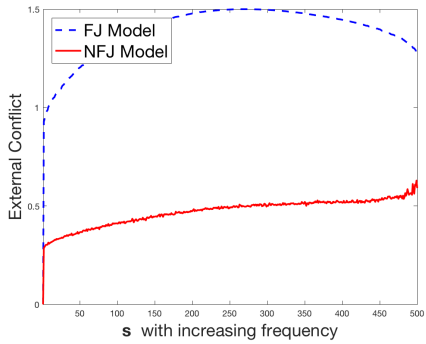
In the experiments, we follow [23] to use the signs of the eigenvectors of the network Laplacian matrix $\mathbf{L}$ as the internal opinion vector $\mathbf{s}$, which correspond to different frequencies (i.e., eigenvalues). In order to make a clearer comparison between both models, we scale the magnitude of the edge weights. We can see that the old model has decreased amount of conflict for high-frequency $\mathbf{s}$ since every node is influenced by more neighbors holding opposite opinions. This is due to too much opinion averaging.

On the contrary, from Figure 3.3, we can see that the high-frequency internal opinions correspond to larger external conflict if we use the new model. This is because the NFJ model limits the overall amount of external influence by the normalization, thus the opinions are not over-averaged and the conflict measure reflects the "real" (i.e., internal) opinion divergence to some extend. Note that external conflict is what exists between people holding opposite opinions internally. It means even if they express themselves differently, one of them should realize the other is on the same side with him/her internally. Therefore, the more people differ from their neighbors on the graph in terms of internal opinions (i.e., $\mathbf{s}$ shows higher frequency), the more conflict there should be. It is consistent with the results of our proposed NFJ Model.

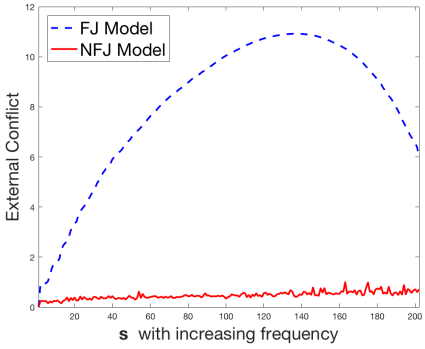
This chapter only presents a first look at the normalized Friedkin-Johnsen (NFJ) Model, and there are a lot of interesting tasks to be done in the near future. For example, the evolution of opinion dynamics, network conflict risk problems under the new model, the discussion on the parameter a (i.e., the self-appraisal [7]),



(a) Karate ($n = 34$, $m = 78$)



(b) small world random network ($n = 500$, $m = 1500$)



(c) Facebook Network ($n = 4039$, $m = 88234$)

Figure 3.3: Conflict Comparison in Networks

networks with different type of nodes (e.g., introducing stubborn nodes who only express their own internal opinions), and so on.

Also, instead of doing nomalization, which discounts the neighboring influences, we can switch the sign of the moderation. In other words, one instance could be that two very opinionated people who hate each other will never moderate the opinions of the other person, on the contrary, their opinions will be reinforced through the connection. Therefore, it leads to a non-linear model of opinion formation. Another study direction is considering higher dimensions of opinions because different issues do not necessarily correspond to different social networks. People within a social network communicate about various issues, and their attitudes on one issue may have influence on other issues, which means minimizing conflict on one issue might actually increase conflict on another. A higher dimensional opinion vector seems to be closer to people's daily life and is more interesting for future study.

4

Quantifying and Minimizing Risk of Conflict in Social Networks

Abstract Controversy, disagreement, conflict, polarization and opinion divergence in social networks have been the subject of much recent research. In particular, researchers have addressed the question of how such concepts can be quantified given people's prior opinions, and how they can be optimized by influencing the opinion of a small number of people or by editing the network's connectivity.

Here, rather than optimizing such concepts given a specific set of prior opinions, we study whether they can be optimized in the average case and in the worst case over all sets of prior opinions. In particular, we derive the worst-case and average-case conflict risk of networks, and we propose algorithms for optimizing these.

For some measures of conflict, these are non-convex optimization problems with many local minima. We provide a theoretical and empirical analysis of the nature of some of these local minima, and show how they are related to existing organizational structures.

Empirical results show how a small number of edits quickly decreases its conflict risk, both average-case and worst-case. Furthermore, it shows that minimizing average-case conflict risk often does not reduce worst-case conflict risk. Minimizing worst-case conflict risk on the other hand, while computationally more challenging, is generally effective at minimizing both worst-case as well as average-case conflict risk.

4.1 Introduction and Motivation

The study of how opinions form through social interactions with others with potentially differing opinions has long been studied in the social sciences (see e.g. [1, 2]). Today, online social networks offer unprecedented access to both social interactions and publicly expressed opinions on controversial matters. This now allows one to quantitatively study differences of opinions on a large scale, as well as to moderate them through targeted interventions. This newfound ability offers new opportunities for conflict prevention and mitigation, as well as for more effective marketing campaigns.[1]

Background. Much prior research has focused on opinions on political matters [11, 12, 26]. However, recent work has often studied the problem in a more generic manner (independent of the topic of controversy) [10, 77, 78]. The identification of controversial issues has been studied using tools from sentiment analysis [27, 28], as well as by relying on the structure of the social network and the distribution of opinions across it [11, 12, 30, 79]. Besides identifying or quantifying controversy or conflict, the question of how it can be influenced has received increasing amounts of attention [8, 10, 31]. Strategies that have been considered include editing the graph (or even designing it from scratch), and attempting to alter the opinions of a small number of individuals [8–10, 24, 32].

Most of these results are based on the opinion formation model by Friedkin and Johnsen [2], which extended the DeGroot model of opinion averaging [1]. In Friedkin and Johnsen's model, individuals are assumed to hold an ('a priori') *internal opinion*, while they may *express* an opinion that may differ from it but that is more socially acceptable (i.e. more similar to their friends' opinions). To model this, it is assumed that individuals are connected to each other in a social network, and that individuals' expressed opinion is a weighted average of their own internal opinion and their neighbors' expressed opinions, with weights representing the strength of the connections in the network.

Shortcomings in the state-of-the-art. An important problem with Friedkin and Johnsen's model is that, while external opinions are hard to measure, access to internal opinions is near-impossible in practice. Another shortcoming of the dominant line of research attempting to reduce conflict by editing the social network is that it tends to focus on a single or a given set of controversial topics. Yet, different issues do not generally correspond to different social networks, such that editing a social network to minimize conflict on one issue may actually increase conflict on another.

Contributions in this paper. In this paper, we depart from the existing literature

[1] It also creates risks: it could allow oppressive governments to design more effective propaganda, or hostile actors to incite conflict rather than prevent it. These risks are an additional reason for these matters to be studied by the scientific community.

in focusing on *risk of conflict*, rather than on conflict around one particular issue. In this way, we overcome both shortcomings of prior work discussed above. We still rely on Friedkin and Johnsen's model of opinion formation to quantify the risk of networks to conflict (which we discuss in detail in Sec. 4.2). However, the proposed quantifications are independent of any particular set of internal (or external) opinions, depending purely on the topology of the network. In this way, we bypass the problem that quantifying internal opinions is beyond reach in practice. Moreover, attempting to reduce the *risk* of conflict, leads to more robust network editing strategies than reducing conflict for one particular assignment of internal opinions.

More specifically, we propose two measures of conflict risk: the *worst-case conflict risk (WCR)* and the *average-case conflict risk (ACR)*, respectively quantifying the amount of conflict *in the worst-case*, and *on average*, over all possible internal opinions. Subsequently, we demonstrate how both WCR and ACR can be minimized by locally editing the network. We do this for a number of pre-existing measures of conflict and disagreement discussed in Sec. 4.3, most notably the *internal conflict* (the extent to which individuals are torn by expressing an opinion that differs from their internal opinion), *external conflict* (the extent to which neighboring individuals express different opinions), and *controversy* (the overall variation in expressed opinion). A side-result in this paper is an equality relating these different conflict measures, leading to what we refer as a *conservation law of conflict*: the sum of the internal conflict, twice the external conflict, and controversy is a constant.

In Sec. 4.4 we propose two types of algorithms (one coordinate descent, and one conditional gradient descent) to locally edit the social network to reduce the WCR and ACR for a number of these measures of conflict. Empirical results are provided in Sec. 4.5, evaluating the effectiveness of the proposed algorithms at reducing risk of conflict, providing additional insight into the local minima of the measures, and discussing conflict risk in random network models.

We end with related work in Sec. 4.6 and conclusions in Sec. 4.7.

Notation. Let $G = (V, E, w)$ be an undirected positive-weighted network with $V = \{1, \ldots, n\}$ the set of nodes, $E \in V \times V$ the set of $m = |E|$ edges (with $(i, j) \in E$ iff $(j, i) \in E$), and w a weight function mapping an edge $e \in E$ onto its weight $w(e) > 0$. We denote with $\mathbf{A}$ the (symmetric) adjacency matrix (with zero diagonal), defined by $a_{ij} = w(i, j)$ iff $(i, j) \in E$ and $a_{ij} = 0$ otherwise. With $N(i)$ we denote the set of neighboring nodes of node i: $N(i) \triangleq \{j \in V | (i, j) \in E\}$. Let $\mathbf{1}$ denote the vector of ones of appropriate size. Furthermore, let $\mathbf{d} \triangleq \mathbf{A1}$ denote the vector containing the weighted degrees of all nodes, and $\mathbf{D} \triangleq \text{diag}(\mathbf{d})$ the diagonal degree matrix. Then the Laplacian matrix is defined as $\mathbf{L} \triangleq \mathbf{D} - \mathbf{A}$.

4.2 Opinion Formation Models

Here we briefly discuss the models of opinion formation on social networks, as formalized above, related to the present paper.

The dynamic model. According to the DeGroot model [1], people's opinions are updated gradually through repeated communication. In the model, every person $i \in V$ has an opinion $s_i(t)$ at time t, and it is influenced by its direct neighbors so as to evolve into a different opinion $s_i(t+1)$ in the next time step. More precisely, their opinion is updated as the weighted sum of their own opinion (with weight w_{ii}) and those of the neighbors (with weight w_{ij} for neighbor j). Given a weighted graph $G = (V, E, w)$, and the opinions $s_i(t)$ of the nodes at time t, the updating rule is defined as:

$$s_i\left(t+1\right) = \frac{w_{ii}s_i\left(t\right) + \sum_{j\in N(i)} w_{ij}s_j\left(t\right)}{w_{ii} + \sum_{j\in N(i)} w_{ij}} \tag{4.1}$$

This model formalizes opinion formation as a repeated averaging process of one's opinion with one's neighbors.

The static model. In 1990, Friedkin and Johnsen extended the model by DeGroot to have two different kinds of opinions [2]: an internal opinion s_i and an expressed opinion z_i. The internal opinions of every person are assumed fixed, while the expressed opinions are influenced by the node's own internal opinion as well the expressed opinions of the neighbors, as follows:

$$z_i = \frac{w_{ii}s_i + \sum_{j\in N(i)} w_{ij}z_j}{w_{ii} + \sum_{j\in N(i)} w_{ij}}. \tag{4.2}$$

Expressed in matrix-vector notation, and with $w_{ii} = 1$ (a common assumption in the literature that we also make in this paper), this equation is solved by (4.3) below at equilibrium [24], i.e., $z_i + \sum_{j\in N(i)} w_{ij}(z_i - z_j) = s_i$ thus $(\mathbf{I}+\mathbf{L})\mathbf{z} = \mathbf{s}$:

$$\mathbf{z} = \left(\mathbf{L} + \mathbf{I}\right)^{-1}\mathbf{s}. \tag{4.3}$$

In this model, the internal opinion s_i of node i is considered a constant, and private to each individual, while the expressed opinion z_i is public, and a compromise between the internal opinion of node i and the expressed opinion of node i's neighbors.

Remark 1. *In this paper, we will generally assume that the internal opinions are mean-centered. Note that in that case, also* $\mathbf{z}$ *will be mean-centered. As opinions are arguably relative, this assumption should not incur any loss of generality. Rather on the contrary: some measures of opinions are affected by the mean of* $\mathbf{s}$ *(as we will point out later), which is arguably undesirable, such that assuming* $\mathbf{s}$ *has zero mean enhances the usability of the proposed measures.*

Table 4.1: Measures for conflict in undirected networks

Name	$\mathbf{z}$	$\mathbf{s}$
internal conflict: ic	$\mathbf{z}^T\mathbf{L}^2\mathbf{z}$	$\mathbf{s}^T(\mathbf{L}+\mathbf{I})^{-1}\mathbf{L}^2(\mathbf{L}+\mathbf{I})^{-1}\mathbf{s}$
external conflict: ec	$\mathbf{z}^T\mathbf{L}\mathbf{z}$	$\mathbf{s}^T(\mathbf{L}+\mathbf{I})^{-1}\mathbf{L}(\mathbf{L}+\mathbf{I})^{-1}\mathbf{s}$
controversy: c	$\mathbf{z}^T\mathbf{z}$	$\mathbf{s}^T(\mathbf{L}+\mathbf{I})^{-2}\mathbf{s}$
resistance: r	$\mathbf{z}^T\mathbf{s}$	$\mathbf{s}^T(\mathbf{L}+\mathbf{I})^{-1}\mathbf{s}$

4.3 Conflict and Conflict Risk

In this paper, we rely on Friedkin and Johnsen's model of opinion formation and discuss a number of (previously known) measures of conflict in terms of the internal opinions $\mathbf{s}$ and expressed opinions $\mathbf{z} = (\mathbf{L}+\mathbf{I})^{-1}\mathbf{s}$. Note that we will often use the term *conflict* in a more generic manner in this paper, to signify conflict, controversy, or disagreement more generally.

In Sec. 4.3.1 we survey the measures of conflict and discuss how they can be computed using matrix-vector operations. Section 4.3.2 introduces an intriguing though intuitive connection between some of these measures. Finally, in Sec. 4.3.3 we discuss how the risk of conflict, as quantified by the proposed measures, can be formulated, both in the worst case (WCR) and in the average-case (ACR).

4.3.1 Conflict Measures

Table 4.1 provides an overview of the proposed measures, which we will discuss in greater detail below.

Internal Conflict ic**.** The internal conflict measure is designed to quantify the extent to which individuals' internal and expressed opinions differ.

Definition 4.3.1. *The internal conflict* ic *is the sum of squares of the differences between individual internal and expressed opinions:*

$$ic = \sum_i (z_i - s_i)^2 .$$

The following proposition provides a convenient matrix-vector expression for it. The proof is elementary:

$$ic = (\mathbf{z}-\mathbf{s})^T(\mathbf{z}-\mathbf{s}),\ \ \mathbf{z}-\mathbf{s} = \left[(\mathbf{L}+\mathbf{I})^{-1}-\mathbf{I}\right]\mathbf{s} = -\mathbf{L}(\mathbf{L}+\mathbf{I})^{-1}\mathbf{s} = -\mathbf{L}\mathbf{z}.$$

Proposition 4.3.1. $ic = \mathbf{z}^T\mathbf{L}^2\mathbf{z} = \mathbf{s}^T(\mathbf{L}+\mathbf{I})^{-1}\mathbf{L}^2(\mathbf{L}+\mathbf{I})^{-1}\mathbf{s}$.

External Conflict ec**.** Arguably the most relevant measure in practice, the external conflict measure quantifies the extent to which the expressed opinions of neighbors are in disagreement with each other. Formally:

Definition 4.3.2. *The external conflict ec is the weighted sum of squares of the pairwise differences between the expressed opinions of neighbors in the network:*

$$ec = \sum_{(i,j)\in E} w_{ij} \left(z_i - z_j\right)^2.$$

Again, it can be expressed conveniently in matrix-vector form:

Proposition 4.3.2. $ec = \mathbf{z}^T \mathbf{L} \mathbf{z} = \mathbf{s}^T \left(\mathbf{L} + \mathbf{I}\right)^{-1} \mathbf{L} \left(\mathbf{L} + \mathbf{I}\right)^{-1} \mathbf{s}$.

The proposed measure of external conflict is closely related to the so-called Network Disagreement Index (NDI) in [6], except that there are two different opinions in our work: it is equal to the NDI evaluated on the external opinions.

Controversy c. Given the expressed opinions, the controversy does not depend on the network structure, and simply quantifies how much the opinion varies across the individuals in the network:

Definition 4.3.3. *The controversy c is the sum of the squares of the expressed opinions:*

$$c = \sum_i z_i^2.$$

Again, this can be trivially expressed in matrix-vector form:

Proposition 4.3.3. $c = \mathbf{z}^T \mathbf{z} = \mathbf{s}^T \left(\mathbf{L} + \mathbf{I}\right)^{-2} \mathbf{s}$.

The controversy c is equivalent with the polarization index proposed by Matakos et al. [10], although they normalized the measure by n, the number of nodes in the network. For zero mean $\mathbf{s}$ (and hence zero mean $\mathbf{z}$), as we assume in this paper, the controversy is also equivalent to the Global Disagreement Index (GDI) [6], defined as:

$$\gamma\left(\mathbf{x}\right) := \sum_{i<j} \left(x_i - x_j\right)^2 \tag{4.4}$$

More specifically, the GDI is a constant factor n times larger than the controversy.

Resistance r. The final measure we wish to discuss is the *resistance*.[2]

Definition 4.3.4. *The resistance r is the inner product between expressed and internal opinion vectors:*

$$r = \sum_i s_i z_i.$$

[2]Its suggested name stems from its mathematical form, which is closely related to the effective resistance in graphs [80]: $R_{ij} = \left(\mathbf{e}_i - \mathbf{e}_j\right)^T \mathbf{L}^+ \left(\mathbf{e}_i - \mathbf{e}_j\right)$, thus it is called resistance. In a graph, the effective resistance between two nodes i and j is: $\left(\mathbf{e}_i - \mathbf{e}_j\right)^T \mathbf{L}^+ \left(\mathbf{e}_i - \mathbf{e}_j\right)$. $\mathbf{e}_i$ has one at position i and zeros elsewhere. If $\mathbf{s} = \mathbf{e}_i - \mathbf{e}_j$ where only the opinions of the two nodes count,

$$r = \mathbf{s}^T \left(\mathbf{L} + \mathbf{I}\right)^{-1} \mathbf{s} = \left(\mathbf{e}_i - \mathbf{e}_j\right)^T \left(\mathbf{L} + \mathbf{I}\right)^{-1} \left(\mathbf{e}_i - \mathbf{e}_j\right).$$

It can again be expressed in matrix-vector notation:

Proposition 4.3.4. $r = \mathbf{s}^T\mathbf{z} = \mathbf{s}^T(\mathbf{L}+\mathbf{I})^{-1}\mathbf{s}$.

The resistance was in fact introduced earlier by Musco et al. [8] (where it was given no name). It was introduced there as the sum of the controversy and external conflict:

Proposition 4.3.5. *Resistance is the sum of external conflict and controversy:* $r = ec + c$.

Their work included an algorithm for optimizing the network to reduce conflict given a specified internal opinion vector $\mathbf{s}$, and took advantage of the fact that resistance is matrix-convex in $\mathbf{L}$.

Summary. Thus, each of the measures can be written in the form

$$* = \mathbf{s}^T\mathbf{M}_*\mathbf{s},$$

where $*$ is one of ic, ec, c, or r, and $\mathbf{M}_{ic} = (\mathbf{L}+\mathbf{I})^{-1}\mathbf{L}^2(\mathbf{L}+\mathbf{I})^{-1}$, $\mathbf{M}_{ec} = (\mathbf{L}+\mathbf{I})^{-1}\mathbf{L}(\mathbf{L}+\mathbf{I})^{-1}$, $\mathbf{M}_c = (\mathbf{L}+\mathbf{I})^{-2}$, and $\mathbf{M}_r = (\mathbf{L}+\mathbf{I})^{-1}$.

We note in passing that the matrices $\mathbf{L}$ and $(\mathbf{L}+\mathbf{I})$ obviously have the same eigenspaces, such that they commute – i.e. the factors in the expressions for $\mathbf{M}_*$ can be freely rearranged.

4.3.2 A Conservation Law of Conflict

In this section, we state an identity that implies that the different measures of conflict act like communicating vessels: reducing one implies that another one must be increased.

Theorem 12 (Conservation law of conflict)**.** *Given a network and an internal opinion vector* $\mathbf{s}$*, then the sum of* ic*,* $2ec$*, and* c *is a constant equal to* $\mathbf{s}^T\mathbf{s}$*:*

$$ic + 2ec + c = \mathbf{s}^T\mathbf{s}.$$

Proof. $ic+2ec+c = \mathbf{s}^T\left(\mathbf{L}+\mathbf{I}\right)^{-1}\left(\mathbf{L}^2+2\mathbf{L}+\mathbf{I}\right)\left(\mathbf{L}+\mathbf{I}\right)^{-1}\mathbf{s} = \mathbf{s}^T(\mathbf{L}+\mathbf{I})^{-1}(\mathbf{L}+\mathbf{I})^2(\mathbf{L}+\mathbf{I})^{-1}\mathbf{s} = \mathbf{s}^T\mathbf{s}$. □

Note that the constant $\mathbf{s}^T\mathbf{s}$ could be regarded as *the internal controversy*: the inherent controversy on a particular topic. The conservation law essentially states that in a social network, this inherent controversy is divided over external conflict, internal conflict, and a remaining amount of controversy. The relative proportions of each of these measures of conflict depend on the structure of the network in relation to the internal opinion vector $\mathbf{s}$.

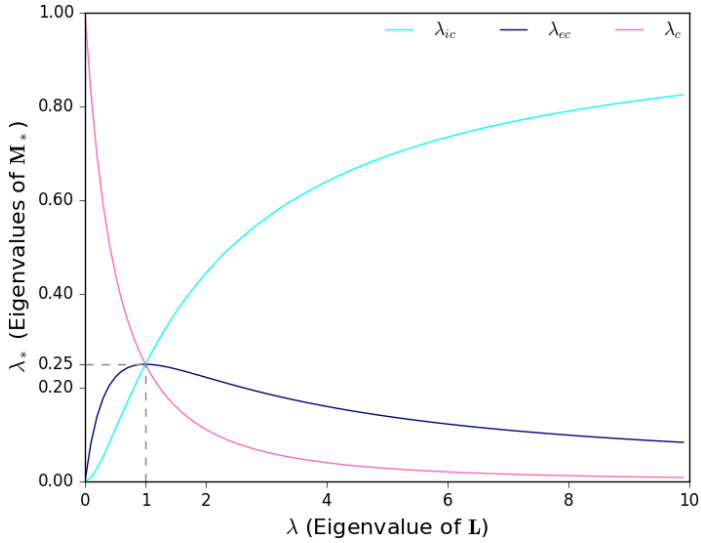


Figure 4.1: Eigenvalues in the Conservation Law.

To understand this better, let $\mathbf{L} = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^T$ be the eigenvalue decomposition of $\mathbf{L}$. It is known from graph signal processing [81] that eigenvectors corresponding to small eigenvalues are slowly varying over the graph (i.e. the i'th and j'th entries of such an eigenvalue tend to be similar if i and j are nearby in the graph), while the large eigenvalues correspond to eigenvectors that fluctuate rapidly over the graph. The eigenvalue decompositions of the diagonal matrices with eigenvalues $\mathbf{M}_*$ are then given by:

$$\mathbf{M}_{ic} = \mathbf{U}\mathbf{\Lambda}^2(\mathbf{\Lambda}+\mathbf{I})^{-2}\mathbf{U}^T,\ \mathbf{M}_{ec} = \mathbf{U}\mathbf{\Lambda}(\mathbf{\Lambda}+\mathbf{I})^{-2}\mathbf{U}^T,\ \mathbf{M}_c = \mathbf{U}(\mathbf{\Lambda}+\mathbf{I})^{-2}\mathbf{U}^T.$$

In other words, any eigenvalue λ of the Laplacian $\mathbf{L}$ yields a corresponding eigenvalue of the $\mathbf{M}_*$ matrices as follows:

$$\lambda_{ic} = \frac{\lambda^2}{(\lambda+1)^2}, \qquad \lambda_{ec} = \frac{\lambda}{(\lambda+1)^2}, \qquad \lambda_c = \frac{1}{(\lambda+1)^2}.$$

These eigenvalues are plotted as a function of the eigenvalue λ of the Laplacian in Fig. 4.1.[3]

Note that λ_{ic} increases with λ, λ_c decreases with λ, and λ_{ec} first increases to reach a maximum value of 0.25 at $\lambda = 1$ after which it decreases again.

For a fixed 2-norm of the internal opinion vector $\mathbf{s}$, the measure of conflict with $\mathbf{M}_*$ is larger the more it is aligned with the eigenvectors corresponding to

[3] Note that the conservation law is reflected in this figure in the following equality, as can be visually verified from Fig. 4.1: $\lambda_{ic} + 2\lambda_{ec} + \lambda_c = 1$.

the largest eigenvalues of $\mathbf{M}_*$. Figure 4.1 shows that this differs for the different measures.

For $\mathbf{s}$ aligning with the smoothest eigenvectors of the Laplacian (i.e. those corresponding to small eigenvalues λ of the Laplacian), the internal and external conflicts are small, but the controversy is large as internal opinions remain unmoderated by dissenting neighbors. This situation would arise when the graph contains different (nearly) disconnected communities, and within each community the internal opinion is constant, while between the communities the internal opinions differ. As $\mathbf{s}$ becomes more aligned with less smooth eigenvectors (i.e. with larger eigenvalues), the external conflict starts to increase because conflicts between neighbors are starting to arise. For the same reason, the internal conflict starts to increase, and the controversy starts to decrease. The external conflict reaches its maximum when $\mathbf{s}$ is aligned with eigenvectors of $\mathbf{L}$ with eigenvalue $\lambda \approx 1$. As λ keeps increasing, meaning $\mathbf{s}$ aligns with more high-frequency eigenvectors, the moderating effect of neighbors starts to become more important, resulting in a decrease of external conflict as well as the controversy. Essentially, the conflict is increasingly internalized in a network where neighbors often have different internal opinions.

4.3.3 Conflict Risk of a Network

The measures from Sec. 4.3.1 quantify the various types of conflict given an internal opinion vector $\mathbf{s}$. Prior work (see Sec. 4.6) has focused on tweaking the network or the opinions of a selection of individuals to reduce such measures. Often, however, the internal opinions are not accessible. More fundamentally, one might wish to minimize conflict on more than one, including yet unknown controversial issues. We therefore argue that it is more sensible to engineer a network so as to reduce the *risk* of conflict, rather than the conflict for one specific internal opinion vector $\mathbf{s}$. We propose two ways of quantifying risk of conflict, discussed in turn below.

Average-case Conflict Risk (ACR). The ACR is defined as the expected conflict, where the expectation is taken w.r.t. the internal opinions. To evaluate this, a probabilistic model for the internal opinions is needed, and we propose to use the uniform distribution over all vectors from $\{-1, 1\}^n$, such that $E\left[\mathbf{s}\mathbf{s}^T\right] = \mathbf{I}$. Thus:

$$\begin{aligned}\text{ACR}_* &= E\left[\mathbf{s}^T\mathbf{M}_*\mathbf{s}\right] = E\left[\text{Tr}\left(\mathbf{s}\mathbf{s}^T\mathbf{M}_*\right)\right] = \text{Tr}\left(E\left[\mathbf{s}\mathbf{s}^T\right]\mathbf{M}_*\right), \\ &= \text{Tr}(\mathbf{M}_*).\end{aligned}$$

Worst-case Conflict Risk (WCR). This is an alternative (and more robust) measure, defined as the maximum conflict over all possible internal opinion vectors $\mathbf{s} \in \{-1, 1\}^n$:

$$\text{WCR}_* = \max_{\mathbf{s}\in\{-1,1\}^n} \mathbf{s}^T\mathbf{M}_*\mathbf{s}.$$

Note that $\mathbf{M}_* \succeq \mathbf{0}$ for all measures, such that this is an instance of Boolean Quadratic Maximization (BQM) problem [82,83]. While this problem is NP-hard, it can be approximated by solving the following semidefinite programming (SDP) relaxation of the problem (here, $\mathbf{\Sigma}$ is a symmetric real-valued matrix):

$$\begin{aligned} u_* = \max_{\mathbf{\Sigma}} \; & \text{Tr}(\mathbf{\Sigma}\mathbf{M}_*), \\ \text{s.t. } & \mathbf{\Sigma} \succeq \mathbf{0}, \\ & \text{diag}(\mathbf{\Sigma}) = \mathbf{1}. \end{aligned}$$

Nesterov [83] proved that this strategy achieves a $\frac{2}{\pi}$ approximation:

$$\frac{2}{\pi} u_* \leq \text{WCR}_* \leq u_*.$$

To derive an estimate for the worst-case $\mathbf{s} \in \{-1, 1\}^n$ from $\mathbf{\Sigma}$, Goemans and Williamson's randomized rounding strategy [84] can be used: Let $\mathbf{\Sigma} = \mathbf{C}\mathbf{C}^T$ be a Cholesky decomposition of $\mathbf{\Sigma}$, and let $\mathbf{x} \in \mathbb{R}^n$ be a randomly sampled vector from some rotation-invariant distribution. Then, for $\mathbf{s} = \text{sign}(\mathbf{C}\mathbf{x})$, it holds that (where the expectation is over the random vector $\mathbf{x}$):

$$\frac{2}{\pi} \text{WCR}_* \leq E[\mathbf{s}^T \mathbf{M}_* \mathbf{s}] \leq \text{WCR}_*.$$

I.e., the estimated worst-case opinion vector achieves a conflict that is not smaller than $\frac{2}{\pi}$ the actual worst-case conflict.

SDPs can be solved in polynomial time: $O(n^{4.5})$. While this is still a high complexity, in practice such SDPs can be solved without further optimizations for thousands of nodes on commodity machines, and results for the Maximum Cut problem suggest that scaling is possible much beyond that (to millions of nodes) by exploiting tight approximations, further relaxations, or dedicated optimization approaches [85,86].

4.4 Minimizing the Conflict Risk

4.4.1 Algorithms

Here we discuss how the ACR and WCR can be optimized by adding or deleting edges in the network. Note that only the resistance is known to be convex, such that we should not hope for convergence to a global optimum. Yet, we argue that the question of convexity is purely academic here: in practice, graph edits can typically be made only in small amounts, either because of budget constraints, or because of practical considerations. For example, a company may wish to increase its productivity by organizing a team-building event or reorganizing office space so

as to create new conflict-risk reducing connections, but such operations are costly and cannot in practice redesign the complete network structure. Thus, what we should be interested in is a fast decrease of the ACR or WCR given the number of edges added or deleted, rather than eventual convergence to a possible local minimum – let alone a global one.

The edits we consider are edge additions or deletions, or more precisely the increase or decrease of edge weights as long as they remain in the range $[0, 1]$. We keep them within this range because it makes no sense to talk about a negative edge strength in social networks, and there is a bound on the strength of connections. Our algorithms can easily be adapted to handle different bounds.

Below, we discuss two algorithmic approaches to this end: one is a conditional gradient method, and suggests a number of edge additions or deletions simultaneously. The other is a coordinate descent method, and suggests adding or deleting just a single edge.

The optimization problems. Let $\mathbf{A}_0$ be the initial adjacency matrix, and $\mathbf{A}$ the optimized adjacency matrix with corresponding matrix $\mathbf{M}_*$. With $\|\cdot\|_1$ the entry-wise one-norm, the optimization problems for ACR and WCR are thus:

$$\begin{aligned}
\text{ACR:} \quad & \min_{\mathbf{A}} \operatorname{Tr}(\mathbf{M}_*), \\
& \text{s.t. } 0 \leq \mathbf{A} \leq 1, \quad \text{and} \quad \|\mathbf{A} - \mathbf{A}_0\|_1 \leq 2k. \\
\text{WCR:} \quad & \min_{\mathbf{A}} \max_{\mathbf{s} \in \{-1,1\}^n} \mathbf{s}^T \mathbf{M}_* \mathbf{s}, \\
& \text{s.t. } 0 \leq \mathbf{A} \leq 1, \quad \text{and} \quad \|\mathbf{A} - \mathbf{A}_0\|_1 \leq 2k,
\end{aligned}$$

where k is a bound on the sum of absolute values of weight changes (the factor 2 stems from the fact that $\mathbf{A}$ is symmetric). The entry-wise one-norm on $\mathbf{A} - \mathbf{A}_0$ ensures this difference tends to be sparse, such that only few edge weights tend to be updated at the minimum.

For the WCR, this problem is complicated by the inner maximization. We handle this optimization problem by alternating optimization: before each conditional gradient or coordinate descent step, we solve the inner maximization as detailed in the previous section, and then assume $\mathbf{s}$ to be fixed. We found however, that robustness of this strategy can be increased by using not a single $\mathbf{s}$, but a small set of ℓ vectors $\mathbf{s}$ all obtained by randomized rounding. More specifically, written in terms of $\mathbf{S} \in \{-1, 1\}^{n \times \ell}$ containing these different $\mathbf{s}$ vectors as its columns, we solve:

$$\begin{aligned}
\text{Robust WCR:} \quad & \min_{\mathbf{A}} \operatorname{Tr}(\mathbf{S}^T \mathbf{M}_* \mathbf{S}), \\
& \text{s.t. } 0 \leq \mathbf{A} \leq 1, \quad \text{and} \quad \|\mathbf{A} - \mathbf{A}_0\|_1 \leq 2k.
\end{aligned}$$

Thus, rather than minimizing the risk of conflict for one given worst-case opinion vector, the average over a set of approximately worst-case opinion vectors is min-

imized. The added robustness of this strategy stems from the fact that different approximately worst-case opinion vectors can be similarly bad, such that editing the graph to reduce risk for one can increase risk for another. In this case, the alternating minimization would fail. Minimizing the risk averaged over a set approximately worst-case opinion vectors thus increases robustness. Note that for $\mathbf{S} = \mathbf{I}$, the WCR reduces to the ACR. Thus, it suffices to discuss the optimization of the WCR in what follows. Both conditional gradient and coordinate descent first compute the gradient of the ACR and WCR. The gradients for the different measures are summarized in Table 4.2.

Conditional gradient descent [87,88]. The conditional gradient method seeks a step $\mathbf{\Delta}$ most aligned with the gradient, while respecting the constraints after taking a finite step along that direction. More specifically, this step direction is found by solving:

$$\min_{\mathbf{\Delta}} \text{Tr}\left(\frac{\partial \text{Tr}\left(\mathbf{S}^T\mathbf{M}_*\mathbf{S}\right)}{\partial \mathbf{L}} \cdot (\text{diag}(\mathbf{\Delta}\mathbf{1}) - \mathbf{\Delta})\right),$$
$$\text{s.t. } 0 \leq \mathbf{A} + \mathbf{\Delta} \leq 1, \quad \text{and} \quad \|\mathbf{\Delta}\|_1 \leq 2k',$$

where $k' << k$ limits the step size. Here, the objective computes the inner product between the gradient with respect to $\mathbf{L}$ and $\text{diag}(\mathbf{\Delta}\mathbf{1}) - \mathbf{\Delta}$, as changing $\mathbf{A}$ by adding $\mathbf{\Delta}$ amounts to a step of $\text{diag}(\mathbf{\Delta}\mathbf{1}) - \mathbf{\Delta}$ on the Laplacian. Note again that these constraints induce sparsity in the solution vector. The experiments indeed confirmed that often $\mathbf{\Delta}$ contains exactly $2k'$ 1's or -1's.

Coordinate descent. The coordinate descent method first computes the gradient with respect to the (symmetric) adjacency matrix from the gradient with respect to the Laplacian (as listed in Table 4.2):

$$\frac{\partial \text{Tr}\left(\mathbf{S}^T\mathbf{M}_*\mathbf{S}\right)}{\partial a_{ij}} = \frac{\partial \text{Tr}\left(\mathbf{S}^T\mathbf{M}_*\mathbf{S}\right)}{\partial l_{ii}} + \frac{\partial \text{Tr}\left(\mathbf{S}^T\mathbf{M}_*\mathbf{S}\right)}{\partial l_{jj}} - 2\frac{\partial \text{Tr}\left(\mathbf{S}^T\mathbf{M}_*\mathbf{S}\right)}{\partial l_{ij}}.$$

Positive $\frac{\partial \text{Tr}\left(\mathbf{S}^T\mathbf{M}_*\mathbf{S}\right)}{\partial a_{ij}}$ means that reducing $a_{ij} > 0$ will reduce the objective. Conversely, negative $\frac{\partial \text{Tr}\left(\mathbf{S}^T\mathbf{M}_*\mathbf{S}\right)}{\partial a_{ij}}$ means that increasing $a_{ij} < 1$ will reduce the objective. Thus, the algorithm takes the $\frac{\partial \text{Tr}\left(\mathbf{S}^T\mathbf{M}_*\mathbf{S}\right)}{\partial a_{ij}}$ with largest absolute value for which either $a_{ij} > 0$ and $\frac{\partial \text{Tr}\left(\mathbf{S}^T\mathbf{M}_*\mathbf{S}\right)}{\partial a_{ij}} > 0$, or for which $a_{ij} < 1$ and $\frac{\partial \text{Tr}\left(\mathbf{S}^T\mathbf{M}_*\mathbf{S}\right)}{\partial a_{ij}} < 0$. In the former case, the algorithm sets $a_{ij} = a_{ji} = 0$, and in the latter it sets $a_{ij} = a_{ji} = 1$.

Conditional gradient versus coordinate descent. The coordinate descent method is computationally obviously easier, but convergence may be slower than with the conditional gradient method. They are compared with each other in the empirical results section.

Table 4.2: Middle Matrices and Gradients

$*$	$\mathbf{M}_*$	ACR: $\frac{\partial \mathrm{Tr}(\mathbf{M}_*)}{\partial \mathbf{L}}$	WCR: $\frac{\partial \mathrm{Tr}\left(\mathbf{S}^T\mathbf{M}_*\mathbf{S}\right)}{\partial \mathbf{L}}$
ic	$(\mathbf{L}+\mathbf{I})^{-2}\mathbf{L}^2$	$2(\mathbf{L}+\mathbf{I})^{-2} - 2(\mathbf{L}+\mathbf{I})^{-3}$	$\mathbf{L}(\mathbf{L}+\mathbf{I})^{-2}\mathbf{S}\mathbf{S}^T(\mathbf{L}+\mathbf{I})^{-1}$ $+(\mathbf{L}+\mathbf{I})^{-1}\mathbf{S}\mathbf{S}^T(\mathbf{L}+\mathbf{I})^{-2}\mathbf{L}$
ec	$(\mathbf{L}+\mathbf{I})^{-2}\mathbf{L}$	$-(\mathbf{L}+\mathbf{I})^{-2} + 2(\mathbf{L}+\mathbf{I})^{-3}$	$(\mathbf{L}+\mathbf{I})^{-2}\mathbf{S}\mathbf{S}^T(\mathbf{L}+\mathbf{I})^{-2}$ $-\mathbf{L}(\mathbf{L}+\mathbf{I})^{-2}\mathbf{S}\mathbf{S}^T(\mathbf{L}+\mathbf{I})^{-2}\mathbf{L}$
c	$(\mathbf{L}+\mathbf{I})^{-2}$	$-2(\mathbf{L}+\mathbf{I})^{-3}$	$-(\mathbf{L}+\mathbf{I})^{-1}\mathbf{S}\mathbf{S}^T(\mathbf{L}+\mathbf{I})^{-2}$ $-(\mathbf{L}+\mathbf{I})^{-2}\mathbf{S}\mathbf{S}^T(\mathbf{L}+\mathbf{I})^{-1}$
r	$(\mathbf{L}+\mathbf{I})^{-1}$	$-(\mathbf{L}+\mathbf{I})^{-2}$	$-(\mathbf{L}+\mathbf{I})^{-1}\mathbf{S}\mathbf{S}^T(\mathbf{L}+\mathbf{I})^{-1}$

Table 4.3: Gradient matrix elements for size n complete graph

Matrix	Diagonal	Off-diagonal
$-(\mathbf{L}+\mathbf{I})^{-2}$	$-\frac{n+3}{(n+1)^2}$	$-\frac{n+2}{(n+1)^2}$
$-2(\mathbf{L}+\mathbf{I})^{-3}$	$-2\frac{n^2+3n+4}{(n+1)^3}$	$-2\frac{n^2+3n+3}{(n+1)^3}$
$-(\mathbf{L}+\mathbf{I})^{-2} + 2(\mathbf{L}+\mathbf{I})^{-3}$	$\frac{n^2+2n+5}{(n+1)^3}$	$\frac{n^2+3n+4}{(n+1)^3}$

4.4.2 Local Optima of the ACR for Different Risk Measures

As pointed out, only the resistance is known to be convex, such that the ACR and WCR are prone to local minima. Relying on the gradients in Table 4.2, we can prove the following proposition.

Proposition 4.4.1. *The complete graph forms local minimum for the ACR of conflict measures ec, c, and r.*

Proof. The adjacency matrix of a size n complete graph consists of 0 on the diagonal and 1 elsewhere, thus the corresponding Laplacian matrix has $n-1$ on the diagonal and -1 elsewhere. In Table 4.3, the elements in the corresponding ACR gradients with respect to the Laplacian are shown. We will show from these that no feasible step can be found that improves the objectives for a complete graph.

Indeed, for a complete graph (with all weights equal to 1), edge weights can only be decreased. However, decreasing the weight of the edges increases the objective: for a step of $-\delta$ on w_{ij}, the external conflict is increased by $2\frac{n-1}{(n+1)^3}\delta$, the controversy by $\frac{4}{(n+1)^3}\delta$ and the resistance by $\frac{2}{(n+1)^2}\delta$. For $n > 1$ these changes are strictly positive, such that the ACR would be increased after decreasing any w_{ij} by $1 \geq \delta > 0$. □

Derivative results. A number of results immediately follow from this proposition. Recall that resistance is convex on $\mathbf{L}$ [8], so this local minimum is a global one. Furthermore, note that from the conservation law, it follows directly that the

Table 4.4: Risks for complete graph of size n

$*$	ic	ec	c	r
$\mathrm{Tr}\left(\mathbf{M}_*\right)$	$\frac{n^2(n-1)}{(n+1)^2}$	$\frac{n(n-1)}{(n+1)^2}$	$\frac{n(n+3)}{(n+1)^2}$	$\frac{2n}{n+1}$

Table 4.5: Dataset summary statistics.

Network	Karate	Facebook	ER	BA	WS
Nodes	34	4039	n	n	n
Edges	78	88234	m	m_1	$\frac{nK}{2}$
Avg degree	4.5882	43.6910	$\frac{2m}{n}$	$\frac{2m_1}{n}$	K

gradient of $ic + 2ec + c$ is equal to 0. Thus, it is trivial to show that for ic, a complete graph is a local maximum of the ACR. Finally, for a complete graph of size n (i.e., the number of node is n, $n > 1$), the values of ACR for different conflict measures are given in Table 4.4. Using this table, it can be shown that larger complete graph has smaller conflict risks than two smaller complete graphs with the same total number of nodes. For complete graphs of size n_1, n_2, and $n_1 + n_2$ ($n_1, n_2 \geq 3, n_1, n_2 \in \mathbb{Z}$),

$$\mathrm{Tr}\left[\mathbf{M}_{ec}\left(n_1\right)\right] + \mathrm{Tr}\left[\mathbf{M}_{ec}\left(n_2\right)\right] > \mathrm{Tr}\left[\mathbf{M}_{ec}\left(n_1 + n_2\right)\right].$$

(As long as $n_1 n_2 - n_1 - n_2 - 3 \geq 0$, the above inequality holds, which can be proved using $\mathrm{Tr}\left(\mathbf{M}_{ec}\right)$ in Table 4.4.)

We also showed empirically that for the ec a set of disconnected components are optimal where each component is either a clique, a sufficiently long chain, or a tree where each leaf node is separated by at least two edges from a bifurcation node (see Sec. 4.5 for details).

4.5 Empirical Evaluation

4.5.1 Datasets

We use real social networks as well as synthetic data shown in Table 4.5. The real-world datasets we use are the Karate network with 34 nodes and a Facebook network consisting of 4039 users. The Karate network is a social network of friendships between 34 members of a Karate club [69]. The Facebook network contains friend circles and was collected through the Facebook app surveys [76].

The synthetic data includes three random network models: Erdős-Rényi (ER) random networks with binomial degree distribution; Barabási-Albert (BA) random networks with power-law degree distribution; and Watts-Strogatz (WS) small world random networks.

Table 4.6: ACR for random networks of size $n = 1000, m \approx 5000$.

ACR	ic	ec	c	r
ER	796.6	94.1	15.3	109.3
BA	759.3	109.7	21.2	131.0
WS	804.2	91.2	13.3	104.5

4.5.2 Experimental Findings

We investigate the following questions: (1) What types of networks have the highest risks for what types of conflict measures; (2) What are the local minima of the ACR for the various measures; (3) For the external conflict: how do the actual conflict, ACR, and WCR evolve as the ACR or WCR is being minimized; (4) How do the coordinate and conditional gradient descent methods compare for the external conflict. Note that some results are summarized, and our implementation is available for reproducibility. [4]

4.5.2.1 Conflict risk for different measures in random networks

We investigated how the ACR for different conflict measures compare to each other across ER, BA, and WS models. We generated random networks of very similar sizes and densities according to these models, and we compared their ACR for different conflict measures.

Across a wide range of graph densities, the WS network is consistently the most high-risk for ic, while the BA network is consistently the most high-risk for ec. For c and r the most high-risk network depends on the density, although usually the BA or ER networks carry the highest risk. Table 4.6 gives an example.

These findings can be interpreted in terms of the properties of the random network models. In the WS network, the ic is probably high due to the short path lengths and high clustering coefficient, which causes opinions to be strongly moderated. In the BA network, the existence of high-degree *hubs* along with a fat tail of small-degree nodes may cause considerable ec between these hubs (which are strongly moderated) and their surrounding nodes (which are moderated only by very few nodes).

4.5.2.2 Empirical study of the local optima of ACR with different conflict measures

We used the coordinate and conditional gradient descent methods to optimize the ACR (i.e., $\mathrm{Tr}\,(\mathbf{M}_*)$) until convergence, to investigate the structure of the network

[4] All code is available at https://github.com/aida-ugent/conflictrisk-public

at the local minima. The following findings complement and corroborate the theoretical analysis of the local minima from the previous section.

Internal Conflict In our experiments, after convergence the network always contains no edges. As in that case internal and expressed opinions coincide, the ic is then equal to zero, this is obviously the global minimum.

External Conflict In our experiments, the local minima always contained sets of disconnected subgraphs that are cliques, trees, and chains, and sometimes cliques with a chain attached to one of its nodes. Yet, the particular local minimum found differs for different initial graphs, and also slightly for the different algorithms and choices of k'.

Controversy The local minimum found is always the completely connected graph. While this problem is not known to be convex, we conjecture that it has only one local minimum.

Resistance We know from theory that this ACR minimization for resistance is a convex problem. Thus, the minimum found is always the global minimum, namely the complete graph.

Clearly the ec, which is arguably the most relevant among the conflict measures in practice, also exhibits the most complex behavior. One example of how the network changes when minimizing the ec is shown in Fig. 4.2, where the bottom network is the local minimum for the network on the top. Typical adjustments during both the coordinate descent and the conditional gradient algorithm are: a chain of three nodes always forms a triangle (see node 25, 48, 50); two nodes at the same end of a chain/tree will always be connected (see node 14, 27); connections that are not strong enough will break (see node 36 between node 12 and 15).

Remark 2. *Interestingly, the structures at the local optima of the ACR for* ec *seem to correspond with common management structures in companies: a flat organization corresponds to a clique, while a hierarchical organization corresponds to a tree. Management practice may well have evolved this way in part because it minimizes conflict.*

In the sequel, for conciseness we focus on the ec alone, as this is arguably the most useful and most interesting measure.

4.5.2.3 Effectiveness of minimizing ACR versus WCR for ec

Here we investigate the effectiveness of both ACR and WCR. In particular, we investigated on one ER network and the Karate network how the ACR, WCR, and the conflict for three different internal opinion vectors, evolved over consecutive iterations. The three fixed opinion vectors include a random vector $\mathbf{s}_1$, and two

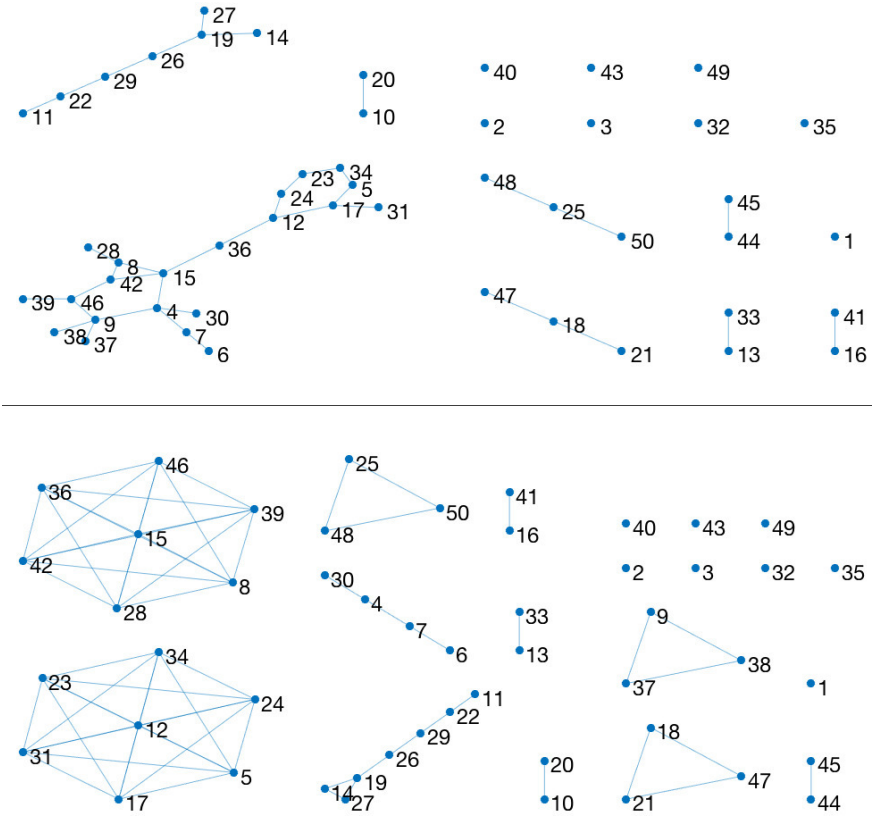


Figure 4.2: Optimization of the ACR of ec on an ER network ($n = 50, p = 0.03$) with gradient descent ($k' = 2$).

vectors found as $\text{sign}(v)$ where v is an eigenvector of the Laplacian: the 10th smallest (i.e. low-frequency on the graph, $\mathbf{s}_2$) as well as the $n - 10$th (i.e. high-frequency on the graph, $\mathbf{s}_3$).

Figure 4.3 shows that the optimization for ACR will not necessarily improve the WCR, and also does not improve the ec for the low-frequency vector $\mathbf{s}_2$, while the optimization over the WCR always decreases also the ACR and the risks for all three given opinion vectors. The fact that the WCR is an upper bound for the ACR as well as for the conflict for any given internal opinion vector probably explains this. Yet, it is remarkable that minimizing the more robust measure WCR does not seem to reduce much the rate at which also the ACR reduces.

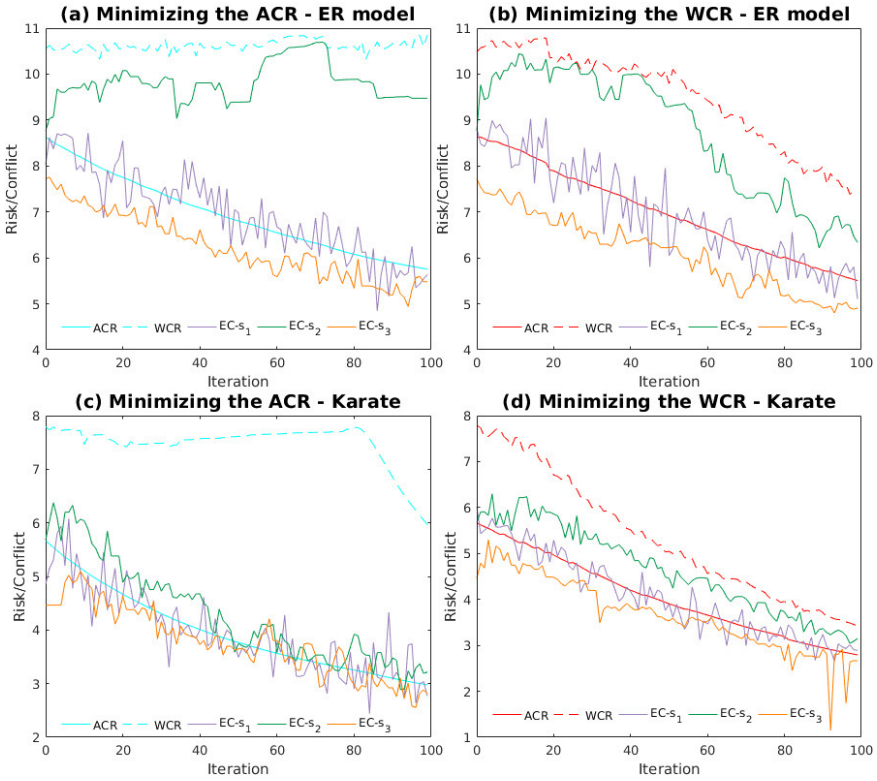


Figure 4.3: The ACR, WCR, and conflict for the three described internal opinion vectors over consecutive iterations. (a), (b) are based on an ER model ($n = 50, m = 60$) with gradient descent $k' = 1$; (c), (d) on Karate with coordinate descent.

4.5.2.4 How does the performance of conditional gradient descent compare to that of coordinate descent?

The following experiment illustrates our observation that conditional gradient descent typically converges to a better local minimum than coordinate descent. This may be because conditional gradient descent can make larger steps at each iteration, thus allowing it to escape bad local minima more easily. Figure 4.4 shows an example of their different performances, which is consistent with our theoretical conclusion in Sec. 4.4 about local optima structures, i.e., larger complete graphs contains less external conflict ACR than smaller ones adding to the same size.

4.5.2.5 Real-world networks

We now summarize the main findings here. The ACR for the Karate network is minimized by forming a complete network for ec, c and r, and the network without

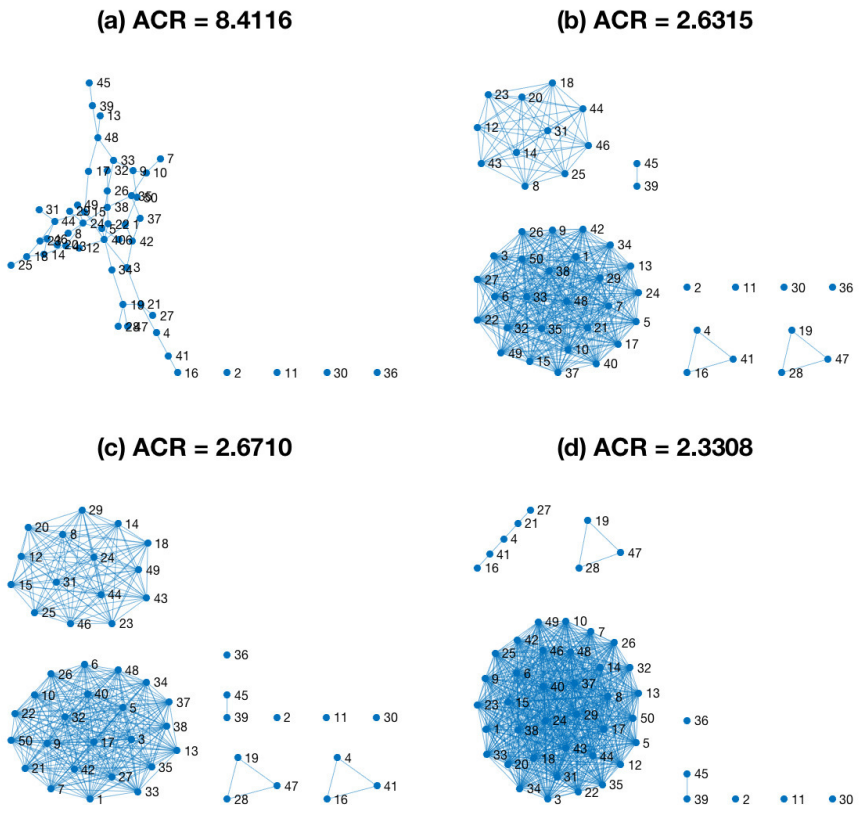


Figure 4.4: Optimal results using the two algorithms. (a) is the original graph; (b) is the result of coordinate descent; (c) is the result of gradient descent with $k' = 5$ at each iteration; (d) is the result of gradient descent with $k' = 25$.

edge for ic. Connections within the 'friend circles' in the Facebook network are found to be strengthened when minimizing the ec ACR, while those between circles are gradually deleted: the originally connected network is ultimately divided into several connected components as the optimization continues. It takes 3 to 5 seconds for one coordinate step on Facebook dataset at the beginning and the time increases as edges are added, which is acceptable in practice.

4.6 Related Work

Social network analysis research almost invariably relies on data from online social media and microblogging sites. In particular Twitter [11, 25, 26, 77] is often the scene of controversial debates. Notable studies are Conover et al., who performed

research on the retweet and mention networks from Twitter, and differentiated between the two mechanisms [11], and Garimella et al., who used conversation graphs obtained from twitter to quantify controversy for any topic [25]. While online social media expose the users to various kinds of opinions, the effects of 'filter bubbles' and 'echo chamber' have attracted increasing amounts of attention in recent years [89, 90]: when people only get information that corroborates their own opinions and communicate only with like-minded people, there is a risk that society will be increasingly fragmented and polarized, although there is an ongoing debate about this issue [89].

Research about polarization and controversy has so far mostly focused on political issues. Morales et al. studied the emergence of political polarization and quantified its effects by a polarization index [26]. Akoglu quantified the political polarity of individuals and political issues by doing classification and ranking tasks [12]. It defines a node classification task on edge-signed (+/-) bipartite opinion network, then predicts latent political classes of people and opinion subjects and ranks people and issues.

Opinion formation models are not always used; some prior work focuses on the underlying structure of the social network, or assumes there are only two groups for 'pro' and 'contra'. Coletto et al. used only local patterns of user interactions (motifs) [79]. Guerra et al. focused on the nodes in the community boundaries [77]. Random Walk Controversy (RWC) scores are used to quantify controversy in [25] as the difference between the properties of a random walk ending in different opinion partitions. Amin et al. studied the problem of identifying and separating polarization using a matrix factorization based gradient descent algorithm [30].

Different measures have been proposed for quantifying polarization or controversy. Modularity is regarded as a traditional measure for polarization [78], but Guerra et al. argue that it is not a good measure since non-polarized networks may also be divided into modular communities in [77]. Then they proposed their novel polarization metric P based on boundary nodes and found that polarized networks tend to have low concentration on high-degree nodes in the boundary between two communities. The Social Network Distance (SND) is a distance measure that quantifies the likelihood of evolution of one snapshot of a social network into another snapshot under a chosen opinion dynamic model in [91]. To quantify controversy in social networks in any topic domain, a three-step pipeline is proposed in [25]. It was found that the RWC outperformed many other controversy measures, including the betweenness, embedding, boundary connectivity, and dipole moment.

A major and increasingly important focus of research is whether polarization and controversy can be *engineered*, e.g. by editing the graph or affecting opinions of a selected set of individuals. In [31], the edge-recommendation problem is

studied based on the endorsement graph, with the goal to reduce the controversy score (namely the RWC), and the acceptable probability of the recommended edge is taken into account. The addition of edges is discussed in [24] in order to reduce the social cost, namely the lack of agreement in the network, and it is argued as intuitive because the exposure to opposite opinions can reduce disagreement. The expressed opinion vector $\mathbf{z}$ above is obtained at the Nash Equilibrium in the social game of opinion formation [24]. Moreover, they firstly studied the problem of moderating people's opinions to reduce the polarization. Based on the same opinion formation dynamics, the promotion problem called the **CAMPAIGN** was studied in [9]. It aimed to promote a product by setting the expressed opinions of k nodes to 1 such that the overall opinions $g(\mathbf{z})$ over the network can be maximized. The expressed opinion z_i represents the affection of node i for the product, and it lies in the range from 0 to 1. This work provides a good example of shifting from the problem of measuring opinion differences to the area of influence maximization.

4.7 Conclusions and Further work

Research into the formation of conflict, disagreement, and related concepts was until recently the subject of the social sciences only. Today however, the fact that opinion formation takes place increasingly on online social platforms creates new possibilities to address related issues from a computer science perspective, building on models of opinion formation from the social sciences. Specifically, it creates the potential to quantify, mitigate, and reduce conflict and disagreement. Prior research on this topic has focused on a single issue of controversy, and the reduction of conflict on this issue, in particular by manipulating the structure of the network.

In this paper we included a small survey of existing measures, and identified an insightful identity between them that amounts to a conservation law of conflict. However, we also argued that reducing one of these measures of conflict for a single issue is problematic, reducing conflict on a single issue may increase it for another. Indeed, in practice a network is not tied to a single issue, and even when it is, the individual opinions may be hard to gauge. To resolve this, we take a novel perspective on this problem, focusing on identifying a limited number of edges to add or remove in the network so as to reduce the *risk* of conflict, both on average and in the worst-case over all possible opinions. We have demonstrated the usefulness of these characterizations of conflict risk, studied their behavior in a range of networks, developed effective algorithms for optimizing them, and confirmed that their minimization minimizes actual risk on some random opinion assignments.

In further work, we plan to investigate further the theoretical properties of these

measures, in particular of the worst-case risk. Additionally, we plan to improve our implementations and investigate other algorithmic improvements for enhanced scalability.

5

ALPINE: Active Link Prediction using Network Embedding

Abstract Many real-world problems can be formalized as predicting links in a partially observed network. Examples include Facebook friendship suggestions, the prediction of protein–protein interactions, and the identification of hidden relationships in a crime network. Several link prediction algorithms, notably those recently introduced using network embedding, are capable of doing this by just relying on the observed part of the network. Often, whether two nodes are linked can be *queried*, albeit at a substantial cost (e.g., by questionnaires, wet lab experiments, or undercover work). Such additional information can improve the link prediction accuracy, but owing to the cost, the queries must be made with due consideration. Thus, we argue that an *active learning* approach is of great potential interest and developed ALPINE (Active Link Prediction usIng Network Embedding), a framework that identifies the most useful link status by estimating the improvement in link prediction accuracy to be gained by querying it. We proposed several query strategies for use in combination with ALPINE, inspired by the optimal experimental design and active learning literature. Experimental results on real data not only showed that ALPINE was scalable and boosted link prediction accuracy with far fewer queries, but also shed light on the relative merits of the strategies, providing actionable guidance for practitioners.

5.1 Introduction

Network embedding and link prediction: Network embedding methods [36], also known as graph representation learning methods, map nodes in a graph onto low-dimensional real vectors, which can then be used for downstream tasks such as graph visualization, link prediction, node classification, and more. Our focus in this paper was on the important downstream task of link prediction.

The purpose of link prediction in networks is to predict future interactions in a temporal network (e.g., social links among members in a social network) or to infer missing links in static networks [33]. Applications of link prediction in networks range from predicting social network friendships, consumer-product recommendations, citations in citation networks, to protein–protein interactions. While classical approaches for link prediction [34] remain competitive for now, link prediction methods based on the state-of-the-art network embedding methods already match and regularly exceed them in performance [35].

Active learning for link prediction: An often-ignored problem affecting all methods for link prediction, and those based on network embedding in particular, is the fact that obtaining information on the connectivity of a network can be challenging, slow, or expensive. As a result, in practice, networks are often only partially observed [48], while for many node pairs, the link status remains unknown. For example, an online consumer-product network is far from complete as the consumption offline or on other websites is hard to track; some crucial relationships in crime networks can be hidden intentionally; in biological networks (e.g., protein interaction networks), wet lab experiments may have established or ruled out the existence of links between certain pairs of biological entities (e.g., interactions between proteins), while due to limited resources, for most pairs of entities, the link status remains unknown. Moreover, in many real-world networks, new nodes continuously stream in with very limited information on their connectivity to the rest of the network.

In many of these cases, a budget is available to query an "oracle" (e.g., human or expert system) for a limited number of as-yet unobserved link statuses. For instance, wet lab experiments can reveal missing protein–protein interactions, and questionnaires can ask consumers to indicate whether they have seen particular movies before or have a friendship relation with a particular person. Of course, the link statuses of some node pairs are more informative than those of the others. Given the typically high cost of such queries, it is thus of interest to identify those node pairs for which the link status is unobserved, but for which knowing it would add the most value. Obviously, this must be performed before the query is made and thus before the link status is known.

This kind of machine learning setting, where the algorithm can decide for which data points (here: node pairs) it wishes to obtain a training label (here:

link status), is known as *active learning*. While active learning for the particular problem of link prediction is not new [92–95], it has received far less attention than active learning for standard classification or regression problems, and the use of active learning for link prediction based on network embedding methods has to the best of our knowledge remained entirely unexplored. Studying this is the main aim of this paper: *Can we design active learning strategies that identify the node pairs with unobserved link status, of which knowing the link status would be maximally informative for the network embedding-based link prediction?* To determine the utility of a candidate node pair, we focused on the link prediction task: querying a node pair's link status is deemed more useful if the embedding found with this newly obtained link status information is better *for the important purpose of link prediction*.

Partially observed networks: To solve this problem, a distinction should be made between node pairs that are known to be unlinked and node pairs for which the link status is not known. In other words, the network should be represented as a partially observed network, with three node pair statuses: linked, unlinked, and unknown. The node pairs with unknown status are then the candidates for querying, and if the result of a query indicates that they are not linked, actual information is added. This contrasts with much of the state-of-the-art in network embedding research, where unlinked and unknown statuses are not distinguished.

Thus, the active learning strategies proposed will need to build on a network embedding method that naturally handles partially observed networks. Given such a method, we then need an active learning query strategy for identifying the unknown candidate link statuses with the highest utility. After querying the oracle for the label of the selected link status, we can use it as additional information to retrain the network embedding model. In this way, more and more informative links and non-links become available for training the model, maximally improving the model's link prediction ability with a limited number of queries.

The ALPINE framework: We proposed the ALPINE (Active Link Prediction usIng Network Embedding) framework, the first method using active learning for link prediction based on network embedding, and developed different query strategies for ALPINE to quantify the utilities of the candidates. Our proposed query strategies included simple heuristics, as well as principled approaches inspired by the active learning and experimental design literature. ALPINE was based on a network embedding model called Conditional Network Embedding (CNE) [37], whose objective function is expressed analytically. There are two reasons why we chose to build our work on CNE. The first is that, as we will show, CNE can be easily adapted to work for partially observed networks, as opposed to other popular network embedding methods (including those based on random walks). The second reason is that CNE is an analytical approach (not relying on random walks or sampling strategies), and thus allowed us to derive mathematically principled

active learning query strategies. Yet, we note that ALPINE can be applied also to other existing or future network embedding methods with these properties.

Illustrative example: To illustrate the idea of ALPINE, we give an example on the Harry Potter network [96]. The network originally had 65 nodes and 223 *ally* links, but we only took its largest connected component of 59 characters and 218 connections. Note that enemy relation was not considered. We assumed that the network was partially observed: the links and non-links for all characters except "Harry Potter" (the cyan star) were assumed to be fully known, and for Harry Potter, only his relationship with "Rubeus Hagrid" (the green plus) was observed as linked and with "Lord Voldemort" (the red x) as unlinked. The goal was to predict the status of the unobserved node pairs, i.e., whether the other nodes (the circles) were allies of "Harry Potter". Suppose we have a budget to query five unobserved relationships. We thus want to select the five most informative ones.

ALPINE can quantify the informativeness of the unknown link statuses, using different query strategies. Shown in Table 5.1 are the top five queries selected by strategies **max-deg.**, **max-prob.**, and **max-ent.**, which are defined in Section 5.4. Nodes mentioned in the table are highlighted with character names in Figure 5.1. Strategy **max-deg.** suggests to query the relationships among Harry and the high-degree nodes—those who are known to have many allies. Strategy **max-prob.** selects nodes that are highly likely to be Harry's friends based on the observed part of the network. Finally, **max-ent.** proposes to query the most uncertain relationships. A more detailed discussion of these results and a thorough formal evaluation of ALPINE are left for Section 5.5, but the reader may agree that the proposed queries are indeed intuitively informative for understanding Harry's social connections.

Table 5.1: Top-5 Query Selections for the Three Strategies of ALPINE.

Strategy	**max-deg.**	**max-prob.**	**max-ent.**
1	Ron Weasley	Ron Weasley	Albus Dumbledore
2	Albus Dumbledore	Hermione Granger	Grawp
3	Hermione Granger	Albus Dumbledore	Minerva McGonagall
4	Ginny Weasley	Grawp	Severus Snape
5	Sirius Black	Minerva McGonagall	Aragog

Contributions: The **main contributions** of this paper are:

- We proposed the ALPINE framework for actively learning to embed partially observed networks by identifying the node pairs with an as-yet unobserved link status of which the link status is estimated to be maximally informative for the embedding algorithm (Section 5.3).
- To identify the most informative candidate link statuses, we developed several active learning query strategies for ALPINE, including simple heuris-

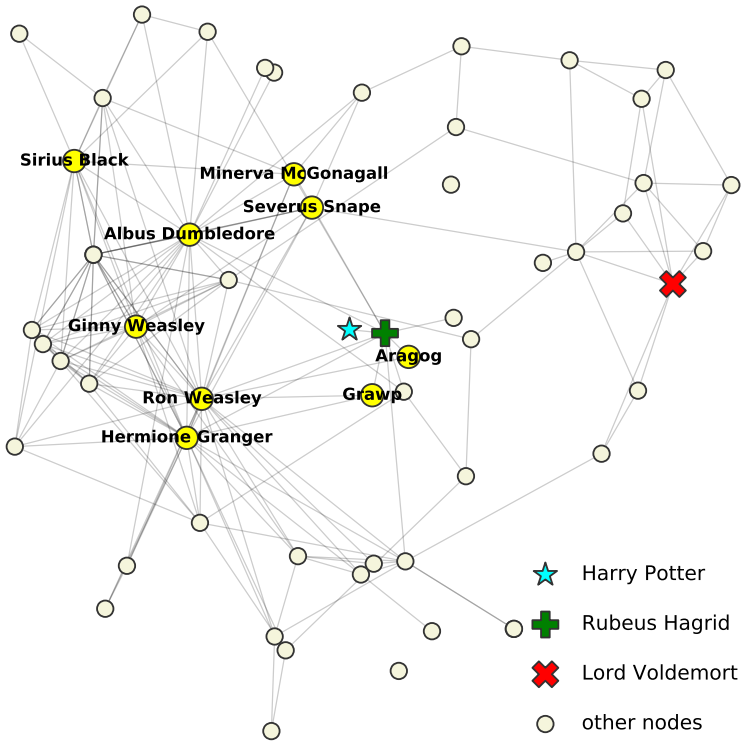


Figure 5.1: Harry Potter network with suggestions from Table 5.1 highlighted.

tics, uncertainty sampling, and principled variance reduction methods based on D-optimality and V-optimality from experimental design (Section 5.4).

- Through extensive experiments [1], (1) we showed that CNE adapted for partially observed networks was more accurate for link prediction and more time efficient than when considering unobserved link statuses as unlinked (as most state-of-the-art embedding methods do), and (2) we studied the behaviors of different query strategies under the ALPINE framework both qualitatively and quantitatively (Section 5.5).

5.2 Background

Before introducing the problem we studied and the framework we proposed, in this section, we first survey the relevant background and related work on active learning and network embedding.

[1] The source code of this work is available at https://github.com/aida-ugent/alpine_public, accessed on 17 October, 2020)

5.2.1 Active Learning and Experimental Design

Active learning is a subfield of machine learning, which aims to exploit the situation where learning algorithms are allowed to actively choose (part of) the training data from which they learn, in order to perform better. It is particularly valuable in domains where training labels are scarce and expensive to acquire [97–99], and thus where a careful selection of the data points for which a label should be acquired is important. The success of an active learning approach depends on how much more effective its choice of training data is, when compared to random acquisition, also known as *passive learning*. Mapped onto the context of our work, the unlabeled "data points" are node pairs with an unknown link status, and an active learning strategy would aim to query the link statuses of those that are most informative for the task performed by the network embedding model. Of particular interest to the current paper is *pool-based* active learning, where a pool of unlabeled data points is provided, and a subset from this pool may be selected by the active learning algorithm for labeling by a so-called oracle (this could be, e.g., a human expert or a biological experiment). In the present context, this would mean that the link status of only some of the node pairs can be queried.

Active learning is closely related to *optimal experimental design* in statistics [100], which aims to design optimal "experiments" (i.e., the acquisition of training labels) with respect to a statistical criterion and within a certain cost budget. The objective of experimental design is usually to minimize a quantity related to the (co)variance matrix of the estimated model parameters or of the predictions this model makes on the test data points. In models estimated by the maximum likelihood principle, a crucial quantity in experimental design is the *Fisher information*: it is the reciprocal of the estimator variance, thus allowing one to quantify the amount of information a data point carries about the parameters to be estimated.

While studied for a long time in statistics, the idea of estimator variance minimization first showed up in the machine learning literature for regression [101], and later, the Fisher information was used to judge the asymptotic values of the unlabeled data for classification [102]. Yet, despite this related work in active learning and the rich and mature statistical literature on experimental design for classification and regression problems, to the best of our knowledge, the concept of variance reduction has not yet been studied for link prediction in networks or network embedding.

5.2.2 Network Embedding and Link Prediction

The confluence of neural network research and network data science has led to numerous network embedding methods being proposed in the past few years. Given a (undirected) network $\mathcal{G} = (V, E)$ with nodes V and edges $E \subseteq \binom{V}{2}$, the goal of a network embedding model is to find a mapping $f : V \to \mathbb{R}^d$ from

nodes to d-dimensional real vectors. The embedding of a network is denoted as $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n]^T \in \mathbb{R}^{n \times d}$. In short, network embedding methods aim to find embeddings such that first-order [37] or sometimes higher order [103, 104] proximity information between nodes is well approximated by some distance measure between the embeddings of the nodes. In this way, they aim to facilitate a variety of downstream network tasks, including graph visualization, node classification, and link prediction.

For the link prediction task, a network embedding model uses a function $g : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ evaluated on $\mathbf{x}_i$ and $\mathbf{x}_j$ to compute the probability of nodes i and j being linked. In practice, g can be found by training a classifier (e.g., logistic regression) on a set of linked and unlinked node pairs, while it can also follow directly from the network embedding model. The method CNE, on which most of the contributions in the present paper were based, belongs to the latter type. With $\mathbf{A} \in \{0, 1\}^{n \times n}$ the adjacency matrix of a network $\mathcal{G} = (V, E)$ (i.e., $a_{ij} = 1$ if $\{i, j\} \in E$ and zero otherwise), the goal of CNE is to find an embedding, denoted as $\mathbf{X}^*$, that maximizes the probability of the network given that embedding [37]:

$$P(\mathcal{G}|\mathbf{X}) = \prod_{\{i,j\} \in E} P(a_{ij} = 1|\mathbf{X}) \cdot \prod_{\{k,l\} \notin E} P(a_{kl} = 0|\mathbf{X}), \tag{5.1}$$

where $P(a_{ij} = 1|\mathbf{X}) = g(\mathbf{x}_i, \mathbf{x}_j)$ for a suitably defined g (see [37] for details).

A problem with all existing network embedding methods we are aware of is that they treat *unobserved* link statuses in the same way they treat *unlinked* statuses. For example, in methods based on a skip gram with negative sampling, such as DeepWalk [103] and node2vec [104], the random walks for determining node similarities traverse via known links, avoiding the unobserved node pairs and thus treating them in the same way as the unlinked ones. Similarly, the more recently introduced Graph Convolutional Networks (GCNs) [105] allow nodes to recursively aggregate information from their neighbors, again without distinguishing unobserved from unlinked node pair statuses. We argue that this makes existing methods suboptimal for link prediction in the practically common situation when networks are only partially observed. This has gone largely unnoticed in the literature, probably because partially observed networks do not tend to be used in the empirical evaluation setups in the papers where these methods were introduced.

The failure to recognize the crucial distinction between unobserved and unlinked node pair statuses has also precluded research on active learning in this context. Indeed, the pool will be a subset of the set of unobserved node pairs, and an unlinked result of a query will add value to the embedding algorithm only if the unobserved status was not already treated as unlinked. Thus, in order to do active network embedding for link prediction, it is essential to distinguish the two links' status.

While CNE was not originally introduced for embedding partially observed networks, it is easily adapted for this purpose (this is not the case, e.g., for skip

gram-based methods based on random walks). We will show how this is performed in Section 5.3.1. This is an important factor contributing to our choice for using this model in this paper.

5.2.3 Related Work

Our work sits at the intersection of three topics: active learning, network embedding, and link prediction. There exists work on the combinations of any two, but not all three of them. Most prominently, link prediction is a commonly considered downstream task of network embedding [36], but active learning has not been studied in this context. Research on active network embedding has focused on node classification [98, 106, 107], but link prediction was not considered as a downstream task in this literature. Finally, active learning for link prediction is not new either [92, 93, 95], but so far, it has been studied in combination with more classical techniques than network embedding. Thus, to the best of our knowledge, the present paper is the first one studying the use of active learning for network embedding with link prediction as a downstream task. Given the promise network embedding methods have shown for link prediction, we believe this is an important gap to be filled.

Work on active learning for graph-based problems has focused on node and graph classification, as well as on various tasks at the link level [99, 108]. The graph classification task considers data samples as graph objects, useful, e.g., for drug discovery and subgraph mining [109], while the node classification aims to label nodes in graphs [110–113]. Active learning has also been used for predicting the sign (positive or negative) of edges in signed networks, where some of the edge labels are queried [92]. Similarly, given a graph, Jia et al. [94] studied how edge flows can be predicted by actively querying a subset of the edge flow information, to help with sensor placement in water supply networks. All these methods are only laterally relevant to the present paper, in that their focus is not on link prediction.

The methods most strongly related to our work are the ActiveLink framework [93] and the approach called HALLP [95].

ActiveLink is designed for link prediction in knowledge graphs and has as its aim to speed up the training of the deep neural link predictors by actively selecting the training data [93]. More specifically, the method selects samples from the *multitriples* constructed according to the training triples. As link prediction in a knowledge graph with only one relation (one predicate) is equivalent with link prediction in a standard graph as considered in this paper, ActiveLink appears to be a meaningful baseline for our paper. However, when applied to a standard graph in this way, ActiveLink would work by querying *all* node pairs involving a set of selected nodes (or, in terms of the adjacency matrix, it would query entire rows or

columns). Thus, this strategy would learn the entire neighborhood of the selected nodes, but very little about the rest of the network. While this strategy may be sensible in the case of a knowledge graph where information about one relation may also be informative about other relations, it is clearly not useful for standard graphs. This excludes ActiveLink as a reasonable baseline.

HALLP (Hybrid Active Learning approach for Link Prediction), on the other hand, proposes an active learning method for link prediction in *standard* networks [95], demonstrating a benefit over passive learning. However, both HALLP's link prediction method (based on a Support Vector Machine (SVM) [114]) and its active learning query strategy utilize a pre-determined set of features rather than a learned model such as a network embedding. The query strategy attempts to select node pairs for which the link prediction is most uncertain. Specifically, HALLP considers two link prediction models for calculating these uncertainties for the candidate node pairs, defining the utility of a candidate node pair as follows:

$$u_{\mathbf{A}}(i,j) = c_1 u_{local}(i,j) + c_2 u_{global}(i,j). \tag{5.2}$$

Here, $u_{local}(i,j)$ is the so-called local utility that measures the uncertainty of a linear SVM link prediction model, $u_{global}(i,j)$ is the so-called global utility based on the hierarchical random graph link predictor [115], and c_1 and c_2 are two coefficients. Interestingly, HALLP uses all unlinked node pairs as candidate node pairs. Thus, HALLP also does not distinguish between the unlabeled and unlinked statuses, and the discovery of non-links is of no use because they will continue to be treated as unlinked. Despite these qualitative shortcomings, HALLP can be used as a baseline for our work, and we included it as such in our quantitative evaluation in Section 5.5.3.

5.3 The ALPINE Framework

In this section, we first show how CNE can be modified to use only the observed information, after formally defining the concept of a partially observed network. Then, we introduce the problem of active link prediction using network embedding and propose ALPINE, which tackles the problem.

5.3.1 Network Embedding for Partially Observed Networks

Network embedding for partially observed networks differs from general network embedding in the way it treats the unknown link statuses. It uses only the observed links and non-links to train the model, where the unobserved part does not participate. We defined an (undirected) Partially Observed Network (PON) as follows:

Definition 1. *A Partially Observed Network (PON) is a tuple* $\mathcal{G} = (V, E, D)$ *where* V *is a set of* $n = |V|$ *nodes and* $E \subseteq \binom{V}{2}$ *and* $D \subseteq \binom{V}{2}$ *the sets of*

node pairs with observed linked and observed unlinked status, respectively, where $E \cap D = \emptyset$*. Thus,* $K \triangleq E \cup D$ *represents the observed (known) part, and* $U \triangleq \binom{V}{2} \setminus K$ *is the set of node pairs for which the link status is unobserved.*

For convenience, we may also represent a PON by means of its adjacency matrix $\mathbf{A}$, with $\mathbf{A} \in \{0, 1, \text{null}\}^{n \times n}$ and a_{ij} at row i and column j equal to null if $\{i, j\} \in U$, to one if $\{i, j\} \in E$, and to zero if $\{i, j\} \in D$.

Most network embedding methods (and methods for link prediction more generally) do not treat the known unlinked status differently from the unknown status, such that the networks are embedded with possibly wrong link labels. This appears almost inevitable in methods based on random walks (indeed, it is unclear how one could, in a principled manner, distinguish unlinked from unknown statuses in random walks), but also many other methods, such as those based on matrix decompositions, suffer from this shortcoming. We now proceed to show how CNE, on the other hand, can be quite straightforwardly modified to elegantly distinguish unlinked from unknown status, by maximizing the probability only for the observed part K of the node pairs:

$$P(\mathcal{G}|\mathbf{X}) = \prod_{\{i,j\} \in E} P(a_{ij} = 1|\mathbf{X}) \cdot \prod_{\{k,l\} \in D} P(a_{kl} = 0|\mathbf{X}). \tag{5.3}$$

In this way, we do not have to assume that the unobserved links are absent, as state-of-the-art methods do. Note that we only consider the known information in E and D (rather than $\notin E$), and the unknowns represented by U are not used for training, as defined by Equation (5.3). Furthermore, the link probability in CNE is formed analytically because the embedding is found by solving a Maximum Likelihood Estimation (MLE) problem: $\mathbf{X}^* = \text{argmax}_{\mathbf{X}} P(\mathcal{G}|\mathbf{X})$. Based on this, later in Section 5.4, we will show how it also allows us to quantify the utility of an unknown link status for active learning.

5.3.2 Active Link Prediction using Network Embedding—The Problem

After embedding the PON, we can use the model to predict the unknown link statuses. Often, however, an "oracle" can be queried to obtain the unknown link status of node pairs from U at a certain cost (e.g., through an expensive wet lab experiment). If this is the case, the query result can be added to the known part of the network, after which the link predictions can be improved with this new information taken into account. By carefully querying the most informative nodes, active learning aims to maximally benefit from such a possibility at a minimum cost. More formally, this problem can be formalized as follows.

Problem 1 (ALPINE)**.** *Given a partially observed network* $\mathcal{G} = (V, E, D)$*, a network embedding model, a budget* k*, a query-pool* $P \subseteq U$*, and a target set*

$T \subseteq U$ containing all node pairs for which the link statuses are of primary interest, how can we select k node pairs from the pool P such that, after querying the link status of these node pairs, adding them to the respective set E or D depending on the status, and retraining the model, the link predictions made by the network embedding model for the target set T are as accurate as possible?

The pool P defines the candidate link statuses, which are unobserved but accessible (i.e., unknown but can be queried with a cost), while the target set T is the set of link statuses that are directly relevant to the problem at hand. Of course, in solving this problem, both the link prediction task and the active learning query strategy should be based only on the observed information $K = E \cup D$.

The problem is formalized in its general form and can become specific depending on the data accessibility (represented by P) and the link prediction task (represented by T). The pool P may contain all the unobserved information or only a small subset of it. Sets T and P may coincide, overlap non-trivially, or be disjoint, depending on the application. We experiment with various options in our quantitative evaluation in Section 5.5.3.

5.3.3 The ALPINE Framework

To tackle the problem of active network embedding for link prediction, we proposed ALPINE, a pool-based [99] active link prediction approach using network embedding and, to the best of our knowledge, the first method for this task. Our implementation and evaluation of ALPINE was based on CNE, but we stress that our arguments can be applied in principle to any other network embedding method of which the objective function can be expressed analytically. The framework works by finding an optimal network embedding for a given PON $\mathcal{G} = (V, E, D)$, selecting one or a few candidate node pairs from the pool $P \subseteq U$ with $U = \binom{V}{2} \setminus (E \cup D)$ to query the connectivity according to a query strategy, updating the PON with the new knowledge provided by querying the oracle, and re-embedding the updated PON. The process iterates until a stopping criterion is met, e.g., the budget is exhausted or the predictions are sufficiently accurate.

The PON can be embedded by the modified CNE, and the active learning query strategy, which evaluates the informativeness of the unlabeled node pairs, is the key to our pool-based active link prediction with network embedding. Defined by a *utility function* $u_{\mathbf{A},\mathbf{X}} : V \times V \to \mathbb{R}$, the query strategy ranks the unobserved link statuses and selects the top ones for querying. The utility quantifies how useful knowing that link status is estimated to be for the purpose of increasing the link prediction accuracy for node pairs in T. Specifically, each query strategy will select the next query for an appropriate $u_{\mathbf{A},\mathbf{X}}$ as:

$$\underset{\{i,j\} \in P}{\operatorname{argmax}} \quad u_{\mathbf{A},\mathbf{X}}(i,j).$$

In practice, not just the single best node pair (i.e., argmax) is selected at each iteration, but the s best ones, with s referred to as the step size.

In summary, given a PON $\mathcal{G} = (V, E, D)$, a network embedding model, a query strategy defined by its utility function $u_{\mathbf{A},\mathbf{X}}$, a pool $P \subseteq U$, a target set $T \subseteq U$, a step size s, and a budget k (number of link statuses in P that can be queried), ALPINE iteratively queries an oracle for the link status of s node pairs, selected as follows:

- At iteration $i = 0$, initialize the pool as $P_0 = P$, and the set of node pairs with known link status as $K_0 = E \cup D$, and initialize $\mathcal{G}_0 = \mathcal{G}$ and $\mathbf{A}_0 = \mathbf{A}$;
- Then, repeat:
  1. Compute the optimal embedding $\mathbf{X}_i^*$ for $\mathcal{G}_i$;
  2. Find the set of queries $Q_i \subseteq P_i$ of size $|Q_i| = \min(s, k)$ with the largest utilities according to $u_{\mathbf{A}_i,\mathbf{X}_i^*}$ (and T);
  3. Query the oracle for the link statuses of node pairs in Q_i, set $P_{i+1} \leftarrow P_i \setminus Q_i$, and set $\mathcal{G}_{i+1}$ equal to $\mathcal{G}_i$ with node pairs $\{i, j\} \in Q_i$ added to the set of known linked or unlinked node pairs (depending on the query result), then set $\mathbf{A}_{i+1}$ accordingly;
  4. Set $k \leftarrow k - |Q_i|$, and break if k is zero.

In this formulation, ALPINE stops when the budget is used up. An optional criterion is surpassing a pre-defined accuracy threshold on T.

5.4 Query Strategies for ALPINE

Now, we introduce a set of active learning query strategies for ALPINE, each of which is defined by its utility function $u_{\mathbf{A},\mathbf{X}}$. For reference, an overview of all strategies is provided in Table 5.2.

The first four (**page-rank.** **max-deg.**, **max-prob.**, and **min-dis.**) are heuristic approaches that use the node degree and link probability information. The fifth (**max-ent.**) is uncertainty sampling, and the last two (**d-opt.** and **v-opt.**) are based on variance reduction. These last three strategies are directly inspired by the active learning and experimental design literature for classical prediction problems (regression and classification). From the utility functions in the last column of the table, we see that the first two strategies do not depend on the embedding model, while the other five are all embedding based. Only for the last strategy (**v-opt.**) is the utility function a function of the target set T.

Table 5.2: Summary of the Query Strategies for ALPINE.

Strategy	Definition	Utility Function
page-rank.	PageRank score sum	$u_{\mathbf{A}}(i,j) = \text{PR}_i + \text{PR}_j$
max-deg.	Degree sum	$u_{\mathbf{A}}(i,j) = \sum_{k:(i,k)\in E} a_{ik} + \sum_{l:(j,l)\in E} a_{jl}$
max-prob.	Link probability	$u_{\mathbf{A},\mathbf{X}^*}(i,j) = P(a_{ij}=1\|\mathbf{X}^*)$
min-dis.	Node pair distance	$u_{\mathbf{A},\mathbf{X}^*}(i,j) = -\|\|\mathbf{x}_i^* - \mathbf{x}_j^*\|\|_2$
max-ent.	Link entropy	$u_{\mathbf{A},\mathbf{X}^*}(i,j) = -\sum_{a_{ij}=0,1} P(a_{ij}\|\mathbf{X}^*)\log P(a_{ij}\|\mathbf{X}^*)$
d-opt.	Parameter variance reduction	$u_{\mathbf{A},\mathbf{X}^*}(i,j) = u_{\mathbf{x}_i^*}(i,j) + u_{\mathbf{x}_j^*}(i,j)$
v-opt.	Prediction variance reduction	$u_{\mathbf{A},\mathbf{X}^*}(i,j) = \sum_{k:(i,k)\in T} u^{ik}(i,j) + \sum_{l:(j,l)\in T} u^{jl}(i,j)$

5.4.1 Heuristics

The heuristic strategies includes the degree- and probability-based approaches for evaluating the utility of the candidate node pairs. Intuitively, one might expect the connections between high-degree nodes to be important in shaping the network structure; thus, we proposed two degree-related strategies: **page-rank.** and **max-deg.** Meanwhile, as networks are often sparse, queries that result in the discovery of new links—rather than the discovery of non-links—are considered more informative, and this idea leads to the **max-prob.** and **min-dis.** strategies.

With strategy **page-rank.**, each candidate node pair is evaluated as the sum of both nodes' PageRank scores [116]: $u_{\mathbf{A}}(i,j) = \text{PR}_i + \text{PR}_j$, while for **max-deg.**, the utility is defined as the sum of the degrees: $u_{\mathbf{A}}(i,j) = \sum_{k:\{i,k\}\in E} a_{ik} + \sum_{l:\{j,l\}\in E} a_{jl}$. The probability-based strategies both tend to query node pairs that are highly likely to be linked. This is true by definition for **max-prob.**: $u_{\mathbf{A},\mathbf{X}^*}(i,j) = P(a_{ij}=1|\mathbf{X}^*)$ and approximately true for **min-dis.**: $u_{\mathbf{A},\mathbf{X}^*}(i,j) = -||\mathbf{x}_i^* - \mathbf{x}_j^*||_2$, as nearby nodes in the embedding space are typically linked with a higher probability.

5.4.2 Uncertainty Sampling

Uncertainty sampling is perhaps the most commonly used query strategy in active learning [99]. It selects the least certain candidate to label, and entropy is widely used as the uncertainty measure. In active network embedding for link prediction, a node pair with an unknown link status can be labeled as unlinked (zero) or linked (one). According to the link probabilities obtained from the learned network embedding model, the entropy of a node pair's link status is:

$$u_{\mathbf{A},\mathbf{X}^*}(i,j) = -P_{ij}^* \log(P_{ij}^*) - (1-P_{ij}^*)\log(1-P_{ij}^*),$$

where $P_{ij}^* = P(a_{ij}=1|\mathbf{X}^*)$. It defines the **max-ent.** strategy that selects the most uncertain candidate link status to be labeled by the oracle. Intuitively, knowing the most uncertain link status maximally reduces the total amount of uncertainty in the unobserved part, although indirect benefits of the queried link status on the model's capability to predict the status of other node pairs are not accounted for.

5.4.3 Variance Reduction

In the literature on *experimental design*, a branch of statistics that is closely related to active learning, the optimality criteria are concerned with two types of variance: the variance of the parameter estimates (D-optimality) and the variance of the predictions using those parameter estimates (V-optimality) [100]. We propose to quantify the utility of the candidate link statuses, based on how much they contribute to the variance reduction that was of our interest. D-optimality [117] aims to minimize the parameter variance estimated through the inverse determinant of the Fisher information. V-optimality [118, 119] minimizes the average prediction variance over a specified set of data points, which corresponds to the target set T for our problem. Since both optimality criteria largely depend on the Fisher information matrix, we give details on the Fisher information for CNE first. Then, the two variance reduction query strategies is formally introduced.

5.4.3.1 The Fisher Information for the Modified CNE

In ALPINE, the modified CNE finds a locally optimal embedding $\mathbf{X}^*$ as the Maximum Likelihood Estimator (MLE) for the given PON $\mathcal{G} = (V, E, D)$, i.e., $\mathbf{X}^*$ maximizes $P(\mathcal{G}|\mathbf{X})$ in Equation (5.3) w.r.t. $\mathbf{X}$. The variance of an MLE can be quantified using the Fisher information [120]. More precisely, the Cramer–Rao bound [121] provides a lower bound on the variance of an MLE by the inverse of the Fisher information: $\text{Var}(\mathbf{X}^*) \succeq \mathcal{I}(\mathbf{X}^*)^{-1}$. Although the Fisher information can often not be computed exactly (as it requires knowledge of the data distribution), it can be effectively approximated by the *observed* information [122]. For the modified CNE, this observed information *for the MLE* $\mathbf{x}_i^*$, the embedding of node i, is given by (proof in Appendix 5.A):

$$\mathcal{I}(\mathbf{x}_i^*) = \gamma^2 \sum_{\{i,j\} \notin U} P_{ij}^*(1 - P_{ij}^*)(\mathbf{x}_i^* - \mathbf{x}_j^*)(\mathbf{x}_i^* - \mathbf{x}_j^*)^T, \tag{5.4}$$

where γ is a CNE parameter. Thus, the variance of node i's MLE embedding $\mathbf{x}_i^*$ is bounded: $\text{Var}(\mathbf{x}_i^*) \succeq \mathcal{I}(\mathbf{x}_i^*)^{-1}$.

5.4.3.2 Parameter Variance Reduction with D-Optimality

D-optimality stands for the determinant optimality, with which we want to minimize the estimator variance, or equivalently maximize the determinant of the Fisher information, through querying the labels of the candidate node pairs [100, 117]. The estimated parameter in CNE is the embedding $\mathbf{X}$. The utility of each candidate node pair $\{i, j\}$ is determined by the estimated variance *reduction* it causes on the estimator—in particular the embeddings of both nodes i and j. As those estimated variances are lower bounded by the inverse of their Fisher informa-

tion, the **d-opt.** strategy seeks to minimize the bounds by maximizing the Fisher information.

Intuitively, the determinant of the Fisher information measures the curvature of the likelihood with respect to the estimator. A large curvature means a small variance as $\text{Var}(\mathbf{x}_i) \succeq \mathcal{I}(\mathbf{x}_i)^{-1}$, corresponding to a large value of D-optimality. The smaller the bound of the parameter variance, or equivalently the more information, the more stable the embedding, and thus, the more accurate the link predictions. This motivates the **d-opt.** strategy.

The estimated information increase of knowing a candidate link status, which we also called the informativeness of that link status, can thus be quantified by the changes in the determinants of the Fisher information matrices of the embeddings of both nodes. Querying the link status of node pair $\{i, j\} \in P$ will reduce the variance matrix bounds $\mathcal{I}(\mathbf{x}_i^*)^{-1}$ and $\mathcal{I}(\mathbf{x}_j^*)^{-1}$, as it creates additional information on their optimal values. For $\mathbf{x}_i^*$, its new Fisher information assuming $\{i, j\} \in P$ has a known status, is denoted as $\mathcal{I}^j(\mathbf{x}_i^*)$ in Equation (5.5), and similarly for $\mathbf{x}_j^*$ leading to $\mathcal{I}^i(\mathbf{x}_j^*)$. Using Equation (5.4), it is easy to see that $\mathcal{I}^j(\mathbf{x}_i^*)$ can be calculated as an additive update to $\mathcal{I}(\mathbf{x}_i^*)$:

$$\mathcal{I}^j(\mathbf{x}_i^*) = \mathcal{I}(\mathbf{x}_i^*) + \gamma^2 P_{ij}^*(1 - P_{ij}^*)(\mathbf{x}_i^* - \mathbf{x}_j^*)(\mathbf{x}_i^* - \mathbf{x}_j^*)^T. \tag{5.5}$$

That estimated information increase is caused by the difference of determinants between the old and the new Fisher information of $\mathbf{x}_i^*$ (and similarly for $\mathbf{x}_j^*$), shown in Equation (5.6). As it is a rank one update in Equation (5.5), we can apply the matrix determinant lemma [123] and write this amount of information change as in Equation (5.7):

$$\begin{aligned} u_{\mathbf{x}_i^*}(i, j) &= \det\left[\mathcal{I}^j(\mathbf{x}_i^*)\right] - \det\left[\mathcal{I}(\mathbf{x}_i^*)\right], & (5.6)\\ &= \gamma^2 P_{ij}^*(1 - P_{ij}^*)(\mathbf{x}_i^* - \mathbf{x}_j^*)^T \mathcal{I}(\mathbf{x}_i^*)^{-1}(\mathbf{x}_i^* - \mathbf{x}_j^*)\det\left[\mathcal{I}(\mathbf{x}_i^*)\right]. & (5.7) \end{aligned}$$

Combining the information change from both nodes, the utility function of **d-opt.** for a node pair $\{i, j\} \in P$ is formally defined as:

$$u_{\mathbf{A},\mathbf{X}^*}(i, j) \quad = u_{\mathbf{x}_i^*}(i, j) + u_{\mathbf{x}_j^*}(i, j). \tag{5.8}$$

Finally, using Equation (5.7), the estimated information increase caused by knowing the status of $\{i, j\} \in P$ proves to be always positive and equal to:

$$u_{\mathbf{A},\mathbf{X}^*}(i, j) =$$
$$\gamma^2 P_{ij}^*(1 - P_{ij}^*)(\mathbf{x}_i^* - \mathbf{x}_j^*)^T \left[\det\left[\mathcal{I}(\mathbf{x}_i^*)\right]\mathcal{I}(\mathbf{x}_i^*)^{-1} + \det\left[\mathcal{I}(\mathbf{x}_j^*)\right]\mathcal{I}(\mathbf{x}_j^*)^{-1}\right](\mathbf{x}_i^* - \mathbf{x}_j^*).$$

5.4.3.3 Prediction Variance Reduction with V-Optimality

V-optimality aims to select training data so as to minimize the variance of a set of predictions obtained from the learned model [100, 118, 119]. The definition

naturally fits the active network embedding for link prediction problem definition, where we only care about the predictions of the target node pairs in T. Therefore, the goal of the **v-opt.** strategy is to minimize the link prediction variance for the target set T.

With CNE, the link prediction function g follows naturally from the model $P^*_{ij} \triangleq g(\mathbf{x}^*_i, \mathbf{x}^*_j) = P(a_{ij} = 1|\mathbf{X}^*)$. What the V-optimality utility function quantifies is then the estimated reduction that a candidate link status in the pool can have on all the target prediction variance—Var(P^*_{ij}) for $(i, j) \in T$. The challenge to be addressed is thus the computation of the reduction in the variance Var(P^*_{ij}). We outline how this can be performed in two steps:

1. First, generate the expression of the prediction variance;
2. Then, define the query strategy as the utility function that quantifies the variance reduction.

The prediction variance Var(P^*_{ij}) can be computed using the first-order analysis (details in Appendix 5.B) and decomposed into contributions from both end nodes, as in Equation (5.9):

$$\text{Var}(P^*_{ij}) \quad = \text{Var}_{\mathbf{x}^*_i}(P^*_{ij}) + \text{Var}_{\mathbf{x}^*_j}(P^*_{ij}) + 2\text{Cov}_{\mathbf{x}^*_i,\mathbf{x}^*_j}(P^*_{ij}). \tag{5.9}$$

Then, the bounds on the parameter variance can be used to bound the variance on the estimated probabilities—Var$_{\mathbf{x}^*_i}(P^*_{ij})$—as in Equation (5.10):

$$\text{Var}_{\mathbf{x}^*_i}(P^*_{ij}) \geq \left[\gamma P^*_{ij}(1 - P^*_{ij})\right]^2 (\mathbf{x}^*_i - \mathbf{x}^*_j)^T \mathcal{I}(\mathbf{x}^*_i)^{-1}(\mathbf{x}^*_i - \mathbf{x}^*_j), \tag{5.10}$$

and similarly for Var$_{\mathbf{x}^*_j}(P^*_{ij})$.

Now, we can quantify the reduction caused by knowing the link status of a candidate node pair. As discussed before, knowing the link status of a node pair $\{i, j\} \in P$, represented by Equation (5.5), leads to a reduction of the bounds $\mathcal{I}(\mathbf{x}^*_i)^{-1}$ and $\mathcal{I}(\mathbf{x}^*_j)^{-1}$, thus on Var$_{\mathbf{x}^*_i}(P^*_{ij})$ and Var$_{\mathbf{x}^*_j}(P^*_{ij})$, and on Var$(P^*_{ij})$ due to Equation (5.9). The last term Cov$_{\mathbf{x}^*_i,\mathbf{x}^*_j}(P^*_{ij})$ in Equation (5.9) does not result in any variance change, so it can be ignored. Putting things together allows defining the V-optimality utility function for **v-opt.** and proves a theorem for computing it.

Definition 2. *The V-optimality utility function $u_{\mathbf{A},\mathbf{X}^*}$ evaluated at $\{i, j\} \in P$ quantifies the reduction in the bound on the sum of the variances Var(P^*_{kl}) (see Equation (5.9) and (5.10)) of all P^*_{kl} for $\{k, l\} \in T$, achieved by querying $\{i, j\}$.*

Theorem 1. *The V-optimality utility function is given by:*

$$u_{\mathbf{A},\mathbf{X}^*}(i, j) \quad = \textstyle\sum_{k:\{i,k\}\in T} u^{ik}(i, j) + \sum_{l:\{j,l\}\in T} u^{jl}(i, j),$$

where:

$$u^{ik}(i, j) = (\gamma P^*_{ik}(1 - P^*_{ik}))^2(\mathbf{x}^*_i - \mathbf{x}^*_k)^T(\mathcal{I}(\mathbf{x}^*_i)^{-1} - \mathcal{I}^j(\mathbf{x}^*_i)^{-1})(\mathbf{x}^*_i - \mathbf{x}^*_k),$$
$$u^{jl}(i, j) = (\gamma P^*_{jl}(1 - P^*_{jl}))^2(\mathbf{x}^*_j - \mathbf{x}^*_l)^T(\mathcal{I}(\mathbf{x}^*_j)^{-1} - \mathcal{I}^i(\mathbf{x}^*_j)^{-1})(\mathbf{x}^*_j - \mathbf{x}^*_l).$$

Due to the fact that the Fisher information update in Equation (5.5) is rank one, we can apply the Sherman–Morrison formula [124] to $\mathcal{I}(\mathbf{x}_i^*)^{-1} - \mathcal{I}^j(\mathbf{x}_i^*)^{-1}$ and rewrite $u^{ik}(i,j)$ (and similarly for $u^{jl}(i,j)$) as:

$$u^{ik}(i,j) = \frac{\gamma^4 P_{ij}^*(1-P_{ij}^*)}{1+\gamma^2 P_{ij}^*(1-P_{ij}^*)d_{jj}(\mathbf{x}_i^*)} \left[P_{ik}^*(1-P_{ik}^*)\right]^2 d_{kj}(\mathbf{x}_i^*)^2,$$

where $d_{jj}(\mathbf{x}_i^*) = (\mathbf{x}_i^* - \mathbf{x}_j^*)^T \mathcal{I}(\mathbf{x}_i^*)^{-1}(\mathbf{x}_i^* - \mathbf{x}_j^*)$, $d_{kj}(\mathbf{x}_i^*) = (\mathbf{x}_i^* - \mathbf{x}_k^*)^T \mathcal{I}(\mathbf{x}_i^*)^{-1}$ $(\mathbf{x}_i^* - \mathbf{x}_j^*)$. Unsurprisingly, the variance reduction is always positive.

5.5 Experiments and Discussion

To evaluate our work, we first studied empirically how partial network embedding with the modified CNE benefited the link prediction task. Then, we investigated the performance of ALPINE with the different query strategies qualitatively and quantitatively. Specifically, we focused on the following research questions in this section:

Q1 What is the impact of distinguishing an "observed unlinked" from an "unobserved" status of a node pair for partial network embedding?

Q2 Do the proposed active learning query strategies for ALPINE make sense qualitatively?

Q3 How do the different active learning query strategies for ALPINE perform quantitatively?

Q4 How can the query strategies be applied best according to the results?

Data: We used eight real-world networks of varying sizes in the experiments:

1. The **Harry Potter** network (used also in Section 5.1) is from the corresponding novel. We used only the ally relationships as edges and its largest connected component, which yielded a network with 59 nodes for the most important characters and 218 ally links among them [96];

2. **Polbooks** is a network of 105 books about U.S. politics among which 441 connections indicate the co-purchasing relations [125];

3. **C.elegans** is a neural network of C.elegans with 297 neurons and 2148 synapses as the links [73];

4. **USAir** is a network of 332 airports connected through 2126 airlines [126];

5. **MP_cc** is a Twitter network we gathered in April 2019 for the Members of Parliament (MP) in the U.K., which originally contained 650 nodes. We only used its largest connected component of 567 nodes and 49,631 friendship (i.e., mutual follow) connections;

6. **Polblogs_cc** is the largest connected component of the U.S. Political Blogs Network [125], containing 1222 nodes and 16,714 undirected hyperlink edges;

7. PPI is a protein–protein interaction network with 3890 proteins and 76,584 interactions [127], and we used its largest connected component **PPI_cc** of 3852 nodes and 37,841 edges after deleting the self-loops;

8. **Blog** is a friendship network of 10,312 bloggers from BlogCatalog, containing 333,983 links [128].

5.5.1 The Benefit of Partial Network Embedding

An important hypothesis underlying this work is that distinguishing an "observed unlinked" status of a node pair from an "unknown/unobserved" status, as opposed to treating both as absent, which is commonly performed, will enhance the performance of network embedding. We now empirically investigated this hypothesis by comparing CNE with its variant that performs partial network embedding: (1) the original CNE defined by its objective function in Equation (5.1), which does not make this distinction, and (2) the modified version that optimizes Equation (5.3), which we called CNE_K (i.e., CNE for the Knowns), which does make the distinction. Specifically, we compared the model fitting time and the link prediction accuracy for both:

1. CNE: fit the entire network where the unobserved link status is treated as unlinked;

2. CNE_K: fit the model only for the observed linked and unlinked node pairs.

Setup: To construct a PON, we first initialized the observed information by randomly sampling a node pair set $K_0 = E \cup D$ that contained a proportion r_0 of the complete information. The complete information means the total number of links in the complete graph for a given number of nodes. For example, $r_0 = 10\%$ means that 10% of the network link statuses are observed as either linked or unlinked: if the network has n nodes, $|K_0| = 10\% \times \frac{n(n-1)}{2}$. The differentiation between the observed unlinked (D) and the unobserved (U) for CNE_K is made by not using node pairs outside K_0 for training. The observed K_0 is guaranteed connected as this is a common assumption for network embedding methods. Then,

we embedded the same PON using both CNE and CNE_K on a machine with an Intel Core i7 CPU 4.20 GHz and 32 GB RAM.

Results: From the results shown in Figure 5.2, we see that CNE_K was not only more time efficient, but also provided more accurate link predictions. The ratio r_0 of observed information varied for datasets because the larger the network, the more time consuming the computations are. The time differences for a small observed part were enough to highlight the time efficiency of CNE_K. The two measures examined were: AUC_U—the prediction AUC score for all unobserved node pairs $(i, j) \in U$ containing $1 - r_0$ network information; and t(s)—the model fitting time in seconds. Both values are averaged—for each r_0 averaged over 10 different PONs and each PON with 10 different embeddings (i.e., CNE has local optima) for the first four datasets—while it is 5×5 for the fifth and sixth and 3×3 for the last dataset.

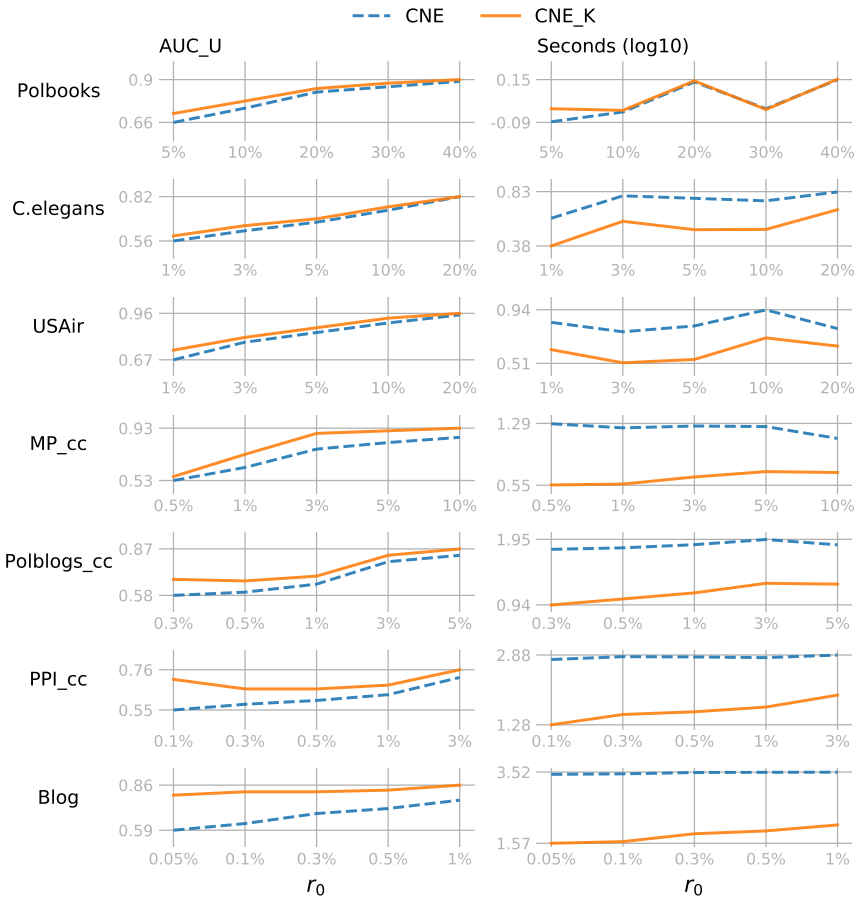


Figure 5.2: Comparison of CNE_K and CNE.

It is not surprising that the fitting time of CNE_K was almost always shorter than the original CNE as CNE_K only fits the observed information. One exception was the Polbooks network, on which both methods used similar amounts of time, because the network size was not large. However, as the network size increased, CNE_K showed increasing time efficiency. Especially for the Blog network: with $r_0 = 0.05\%$, CNE_K was 76 times faster than CNE. CNE_K thus enabled network embedding to scale more easily to large networks.

In addition to time efficiency, CNE_K always achieved a higher AUC than CNE, since CNE will try to model the absence of an edge even where there might actually be an (unobserved) edge. In other words, CNE was trained on data with a substantial amount of label noise: 0 labels (absent edge) that actually must be a 1 (present edge), while CNE_K only used those labels that were known to be correct. Partial network embedding for the knowns is especially useful in settings where only a small part of a large network is observed and the goal is to predict the unobserved links.

5.5.2 Qualitative Evaluation of ALPINE

In Section 5.1, we used the Harry Potter network to illustrate the idea of ALPINE with three of our query strategies, which focused on predicting the unknown links for a target node—"Harry Potter"—who has very limited observed information to the rest of the network. Now, we complete this qualitative evaluation with the same setting for other strategies: **page-rank.**, **d-opt.**, and **v-opt.**; **min-dis.** was omitted as it approximates **max-prob.**.

Table 5.3 shows the top five suggestions, and the relevant characters are highlighted in Figure 5.3 with their names. Since CNE achieved different local optima, here, we used a different two-dimensional visualization to better display the names. All the suggestions were reasonable and could be explained from different perspectives, proving that ALPINE with those query strategies made sense qualitatively. Similar to **max-deg.** and **max-prob.**, **page-rank.** and **d-opt.** had the same top three suggestions: Hermione, Ron, and Albus, which are essential allies of Harry. Knowing whether Harry is linked to them will give a clear big picture of his social relations. The results can further be analyzed according to the strategy definitions.

Table 5.3: Top-5 Query Selections for the other Three Strategies of ALPINE.

Strategy	page-rank.	d-opt.	v-opt.
1	Ron Weasley	Hermione Granger	Arthur Weasley
2	Albus Dumbledore	Ron Weasley	Fluffy
3	Hermione Granger	Albus Dumbledore	Charlie Weasley
4	Vincent Crabbe Sr.	Severus Snape	Albus Dumbledore
5	Neville Longbottom	Ginny Weasley	Ron Weasley

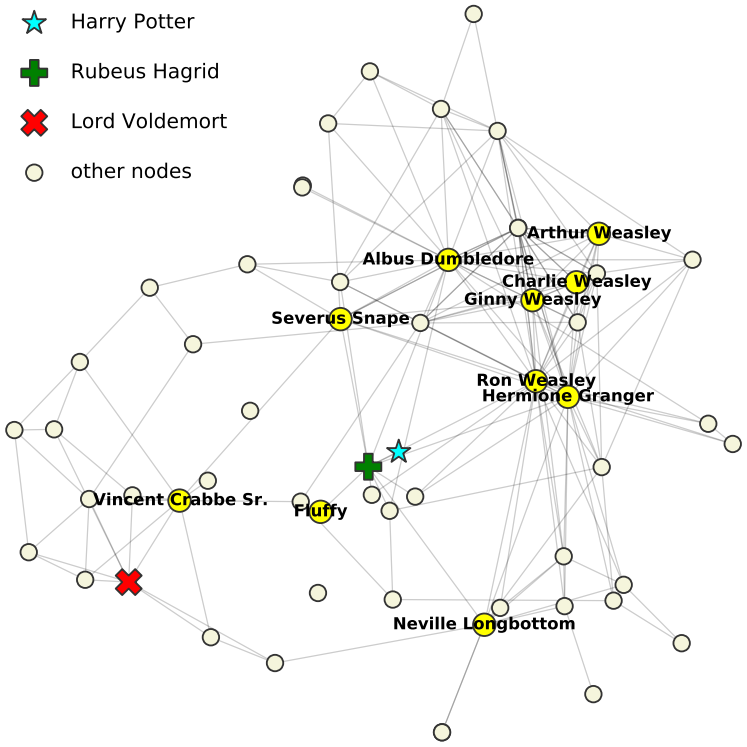


Figure 5.3: Harry Potter network with suggestions from Table 5.3 highlighted.

Strategy **page-rank.**, as **max-deg.**, aims to find out Harry's relationships with the *influencers*—nodes that are observed to have many links. With this type of strategy, we learned which *influencers* Harry is close to, as well as his potential allies connecting to them; and conversely for his unlinked *influencers*.

The **d-opt.** strategy selects queries based on the parameter variance reduction. It implies that by knowing whether Harry is linked to the suggested nodes, the node embeddings will have a smaller variance, such that the entire embedding space is more stable, and thus, the link predictions are more reliable. For example, Severus, who ranks the fourth here (also the fourth with **max-ent.**), was not an obvious ally of Harry, but he helps secretly and is essential in shaping the network structure. The suggestions were considered uncertain and contributed to the reduction of the parameter variance.

The **v-opt.** strategy quantifies the informativeness of the unobserved link statuses by the amount of estimated prediction variance reduction they cause. It suggests that Harry's relationships to the Weasley family are informative for minimizing the prediction variance for him. It makes sense as this family is well connected with Rubeus, who is Harry's known ally, and also connects well with other nodes.

As for Fluffy, it was observed to be connected only to Rubeus and unlinked to all other nodes except Harry. Knowing whether Fluffy and Harry are linked greatly reduced the variance on the prediction for the unobserved, because there was no other information for it.

We concluded that, intuitively, the query strategies resulted in expected behavior, although we caution against overinterpretation of this subjective qualitative evaluation. The next quantitative evaluation provided an objective assessment of the merits of the query strategies, relative to passive learning, to each other, and to the single pre-existing method of which we are aware.

5.5.3 Quantitative Evaluation of ALPINE

In the quantitative evaluation, we mainly wanted to compare the performance of different query strategies from Section 5.4 with passive learning, as well as with the state-of-the-art baseline method HALLP [95]. Passive learning is represented by the random strategy that uniformly selects node pairs from the pool. As for HALLP, we implemented its query strategy shown in Equation (5.2) (since the source code is not publicly available), and set c_1 and c_2 both to one. Note that, as we wanted to compare query strategies in the fairest possible way, the link prediction was performed using CNE_K also for HALLP.

Setup: As before, we constructed a PON by randomly initializing the observed node pair set K_0 with a given ratio r_0, while making sure K_0 was connected. Then, we applied ALPINE with different query strategies for a budget k and a step size s. More specifically, we investigated three representative different cases depending on the pool P and the target set T:

1. $P = U$ and $T = U$: all the unobserved information was accessible, and we were interested in knowing all link statuses in U;

2. $P \subset U$ and $T = U$: only part of U was accessible, and we still wanted to predict the entire U as accurately as possible;

3. $P \subset U$, $T \subset U$, and $P \cap T = \emptyset$: only part of U was accessible, and we were interested in predicting a different set of unobserved link status that was inaccessible.

For all datasets, we investigated four values of r_0: $[3\%, 10\%, 30\%, 80\%]$, to see how the percentage of the observed information affected the strategy performance. All quantitative experiments used a step size depending on the network size: 1% of the network information. For a network with n nodes, it means that $s = 1\% * \frac{n(n-1)}{2}$ unobserved candidate link statuses will be selected for querying in each iteration. Then, the budget k, pool size $|P|$, and target set size $|T|$ were multiples of s for different cases. The random strategy was a baseline for all three cases,

while the HALLP strategy was only used in the first case because it was designed only for this setup.

Below, we first discuss our findings for each of the three cases on the five smallest datasets. After that, we discuss some results on the two larger networks for the most scalable query strategies only.

5.5.3.1 Case 1: $P = U$ and $T = U$

In the first case, we had the pool of all unobserved link statuses and wanted to predict all the unknowns. Shown in Figure 5.4 are the results, in which each row represents a dataset with its step size and each column corresponds to one r_0 value. For every individual subplot, the AUC_U is the AUC score for all the initially unobserved link statuses—those not included in K_0. The budget k was set to 10 steps, i.e., $k = 10s$, resulting in 10 iterations. Iteration 0 was the initial performance before the active learning, so there were always $\frac{k}{s} + 1$ scores. In other words, given the budget $k = 10s$, even for $r_0 = 80\%$, we did not query the entire pool and reached only 90% of the network information. The AUC scores were averaged over several different random PONs, and each PON defined by a K_0 was initialized with different random embeddings (10×10 for the first four networks and 5×5 for the last and largest one). Each random strategy score was further averaged over three runs.

In general, the active learning strategies outperformed the **rand.** strategy. We saw that when the observed part was relatively small—3% or 10%— the degree-related strategies that did not depend on the embedding usually performed very well, and the random strategy was not always the worst. As more information was observed when r_0 increased (see the plots from the left to the right in each row), we did not only see that the active learning strategies, such as **v-opt.** and **max-ent.**, began to dominate and passive learning became the worst, but also the increase of the starting AUC_U. Zooming in to individual subplots, we saw that ALPINE boosted link prediction accuracy with far fewer queries for the active learning strategies, compared to passive learning. Overall, when the observed information was very limited, the embedding-independent strategies **page-rank.** and **max-deg.** outperformed the others; while for sufficiently enough information, **v-opt.** and **max-ent.** were the better choices. We speculated that this was the case as the embedding must be of sufficient quality for the embedding-dependent strategies to work, which requires a certain minimum amount of data. Worth noticing is that **d-opt.** showed similar performance across different values of r_0, which will be discussed further in Section 5.5.4.

As for the HALLP strategy, which aimed to query the most uncertain node-pairs and thus was similar in spirit to **max-ent.**, the performance was very variable. In some cases, it performed quite well, as shown in the top right subplot, beating **v-opt.** in the first few iterations, while on the MP_cc network, it was one of the

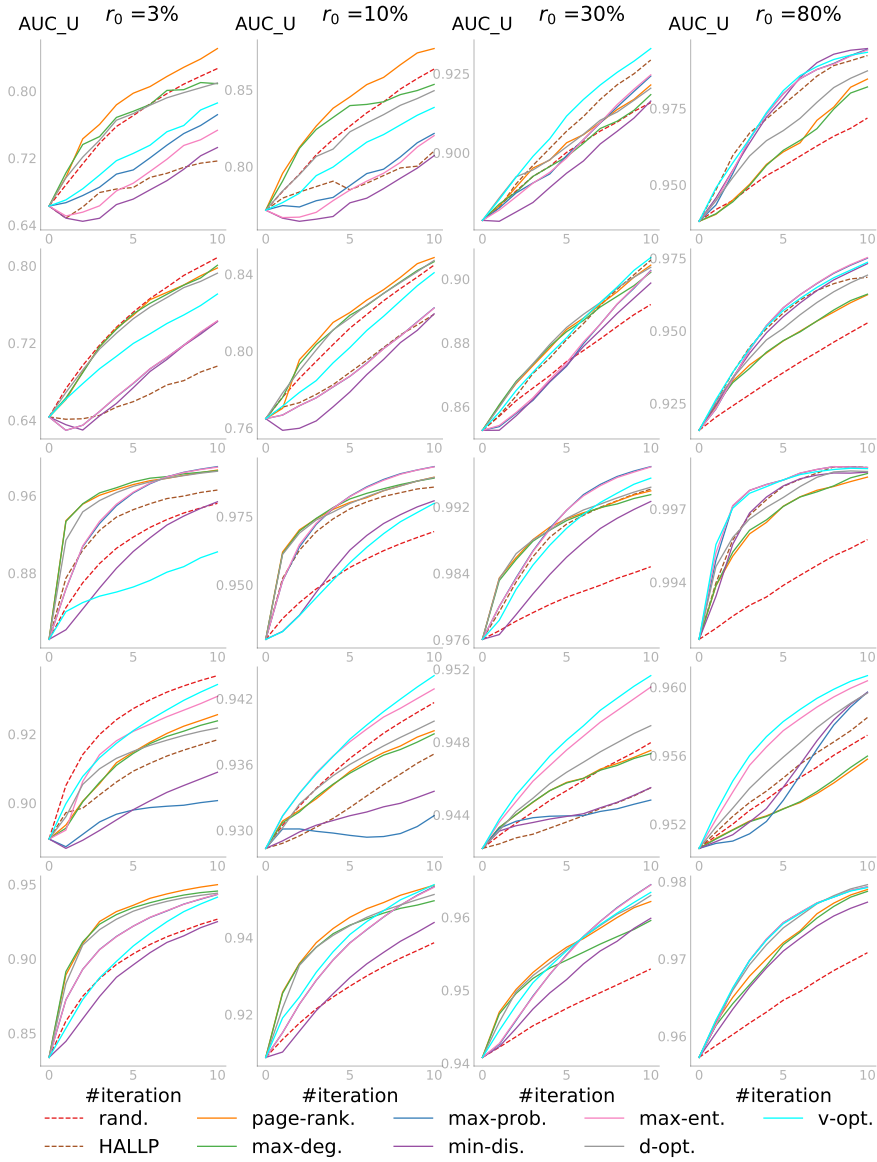


Figure 5.4: ALPINE: $P = U$ and $T = U$. Row 1: Polbooks ($s = 54$); Row 2: C. elegans ($s = 439$); Row 3: USAir ($s = 549$); Row 4: MP_cc ($s = 1604$); Row 5: Polblogs_cc ($s = 7460$).

worst strategies. In addition to that, the runtime of HALLP was much longer than that of the other strategies; thus, some of the subplots do not have the HALLP result. The runtime analysis for one iteration of the query process on a server with

an Intel Xeon Gold CPU 3.00 GHz and 256 GB RAM is shown in Table 5.4 below. The results were averaged in the same way as in Figure 5.4 for the four values of r_0 and then further averaged over the r_0 values. Across different datasets, HALLP was by far the most computationally expensive strategy as it had to run two link predictors.

Table 5.4: Runtime for one query in seconds - Case-1.

Data	rand.	page-rank.	max-deg.	max-prob.	min-dis.	max-ent.	d-opt.	v-opt.	HALLP
Polbooks	0.001	0.031	0.008	0.004	0.027	0.004	0.093	0.482	12.33
C.elegans	0.005	0.108	0.042	0.028	0.18	0.029	0.675	5.469	148.4
USAir	0.006	0.134	0.052	0.035	0.231	0.036	0.881	7.309	232.6
MP_cc	0.016	0.707	0.16	0.117	0.693	0.125	2.746	28.90	1074
Polblogs_cc	0.092	1.153	0.707	0.675	3.264	0.709	12.31	226.0	12022

5.5.3.2 Case 2: $P \subset U$ and $T = U$

In the second case, we applied ALPINE with a smaller pool, while we were still interested in predicting all the unknown link statuses. The experiment setting was similar to the previous case, but the pool size $|P|$ was set to 10 times the step size—$10s$—and the budget $k = 5s$, i.e., only five iterations were performed. The candidates in the pool were randomly sampled from the unobserved part for each PON in our experiments.

Figure 5.5 shows the results for this case. Compared to the first case, the AUC_U was lower for each individual subplot as the accessible information in the pool was more limited. The results confirmed again that all active learning strategies were better than passive learning, Shown more clearly in the last row in Figure 5.5 on the Polblogs_cc network is that the three strategies **page-rank.**, **max-deg.**, and **d-opt.** were the winning group for the first two r_0 values. However, in the third and fourth subplot in the same row, **v-opt.**, **max-ent.**, and **d-opt.** performed best. The **d-opt.** strategy stayed as one of the top strategies across different percentages of the observed information.

5.5.3.3 Case 3: $P \subset U$, $T \subset U$, and $P \cap T = \emptyset$

We imposed further constraints in the third case: not only the pool P of node pairs that could be queried was limited, but also the set T of target node-pairs for which we wanted to predict the status was limited. Moreover, both sets were not intersecting. As in the second case, the budget was set to $k = 5s$ and $|P| = 10s$. The target set size was now taken to be $|T| = 5s$. Both P and T were sampled randomly from U before the querying started.

The results in Figure 5.6 confirmed again that active learning outperformed passive learning. One might expect **v-opt.** to perform the best in this case because it was the only strategy that explicitly considered T. Although it was shown to

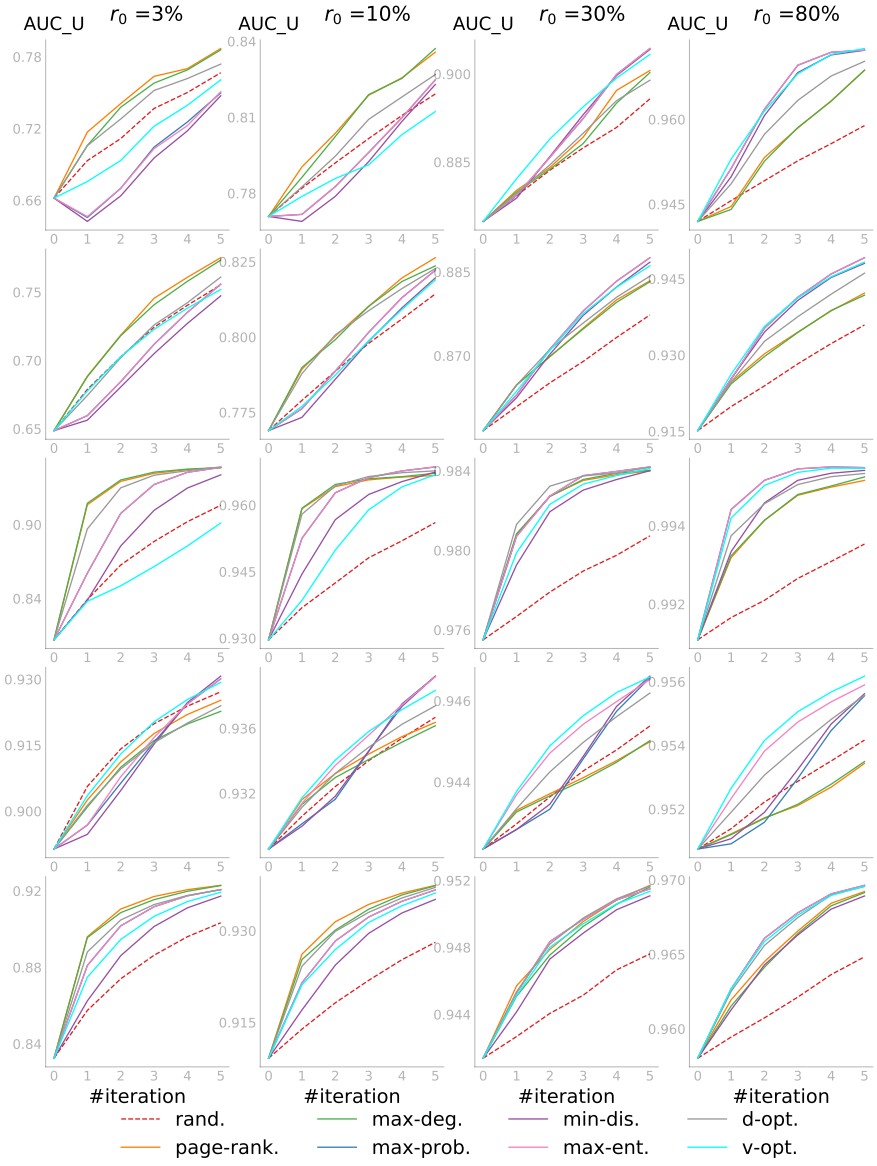


Figure 5.5: ALPINE—$P \subset U$ and $T = U$. Row 1: Polbooks ($s = 54$); Row 2: C. elegans ($s = 439$); Row 3: USAir ($s = 549$); Row 4: MP_cc ($s = 1604$); Row 5: Polblogs_cc ($s = 7460$).

perform quite well in some subplots especially for the first iteration, the quality of the embedding affected its performance. Indeed, as we observed before, the reliability of all embedding-based strategies depended largely on how well the

network was embedded, which became much better as r_0 increased.

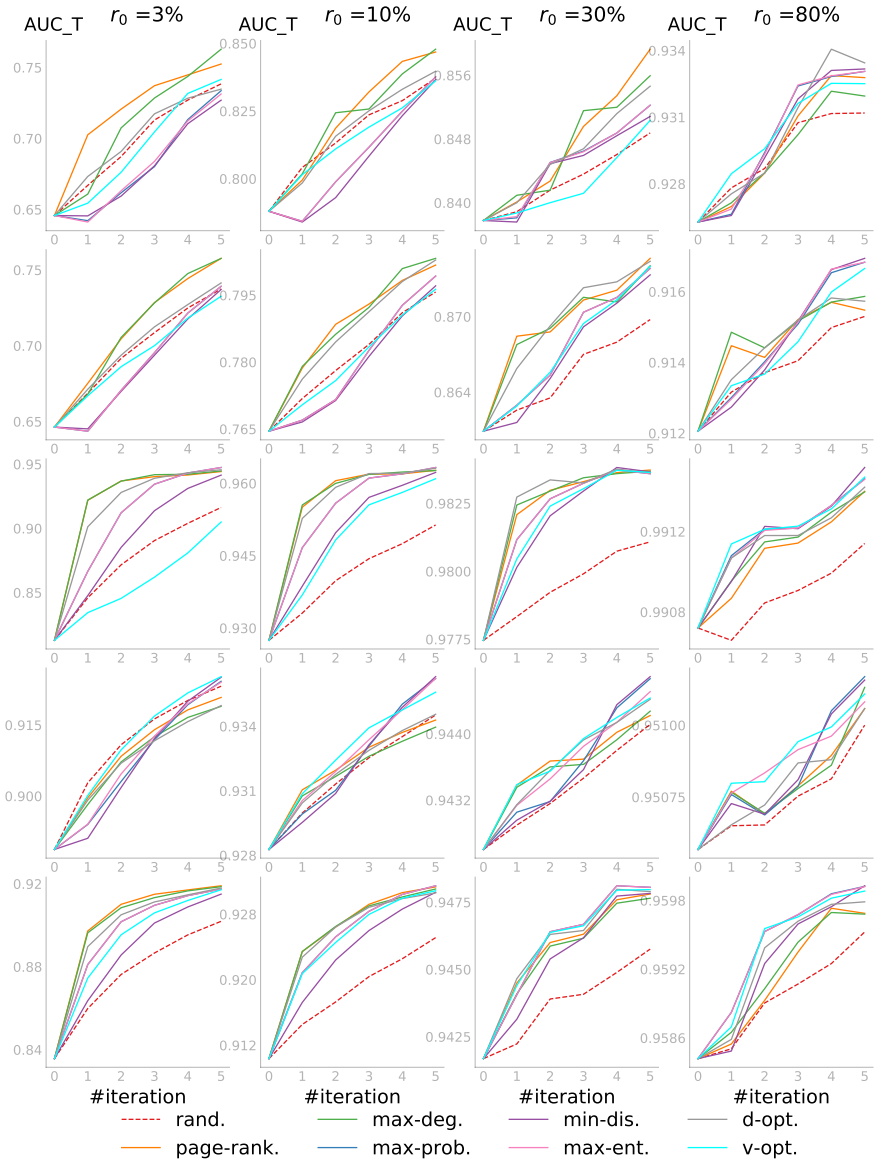


Figure 5.6: ALPINE: $P \subset U$, $T \subset U$, and $P \cap T = \emptyset$. Row 1: Polbooks ($s = 54$); Row 2: C. elegans ($s = 439$); Row 3: USAir ($s = 549$); Row 4: MP_cc ($s = 1604$); Row 5: Polblogs_cc ($s = 7460$).

Compared to the previous two cases, the results here were not as smooth even after averaging. The reason was that the score AUC_T depended not only on P,

but also largely on T, which were both randomly sampled. Whether P contained candidate node pairs that were informative for T affected the score. Overall, the embedding-independent strategies—**page-rank.** and **max-deg.**—had the top performance when r_0 was small; and the embedding-based strategies became increasingly competitive if more information was observed.

5.5.3.4 Evaluations on Two Larger Networks

Finally, we conducted a quantitative evaluation on two larger networks: PPI_cc and Blog. The results are shown in Figures 5.7 and 5.8 and confirmed the observations we made on the five smaller networks. Figure 5.7 shows the PPI_cc results for the three cases with seven query strategies, excluding **v-opt.** and HALLP, as they were computationally too expensive. The AUC scores were averaged over five sets of random initial K_0, P, and T, and each set with five initial embeddings. The last column looks bumpy since the score was already very high and small randomness in the embedding could cause a slight difference. Figure 5.8 shows the results for the second and third case with three values of r_0. Case 1 was omitted because embedding the Blog network with a large observed part was already quite expensive, and evaluating all the unobserved candidates when r_0 was small made it computationally too demanding.

5.5.4 Discussion

Our experiments showed that ALPINE in its general form can be adapted for various problem settings, and active learning performed consistently and substantially better than passive learning regardless of which of the investigated query strategies was applied. Now, we discuss how the strategies can be optimally applied based on our observations, with advice and insights that may help a practitioner select the best query strategy given the properties of the data and available computational resources.

Among the seven active learning query strategies we developed, **page-rank.** and **max-deg.** did not depend on the network embedding while the other five were embedding based. Thus, as with limited observed information, the network embedding might be of poor quality, in such cases, **page-rank.** and **max-deg.** were seen to outperform the others.

The embedding-based strategies began to dominate when more information was observed and the embedding quality improved. The **max-ent.** and **v-opt.** had the top performance, but **d-opt.** had a more stable high performance across different values of r_0. Based on those observations, we recommend a mixed strategy that starts from the degree-related and then switches to other embedding-based strategies.

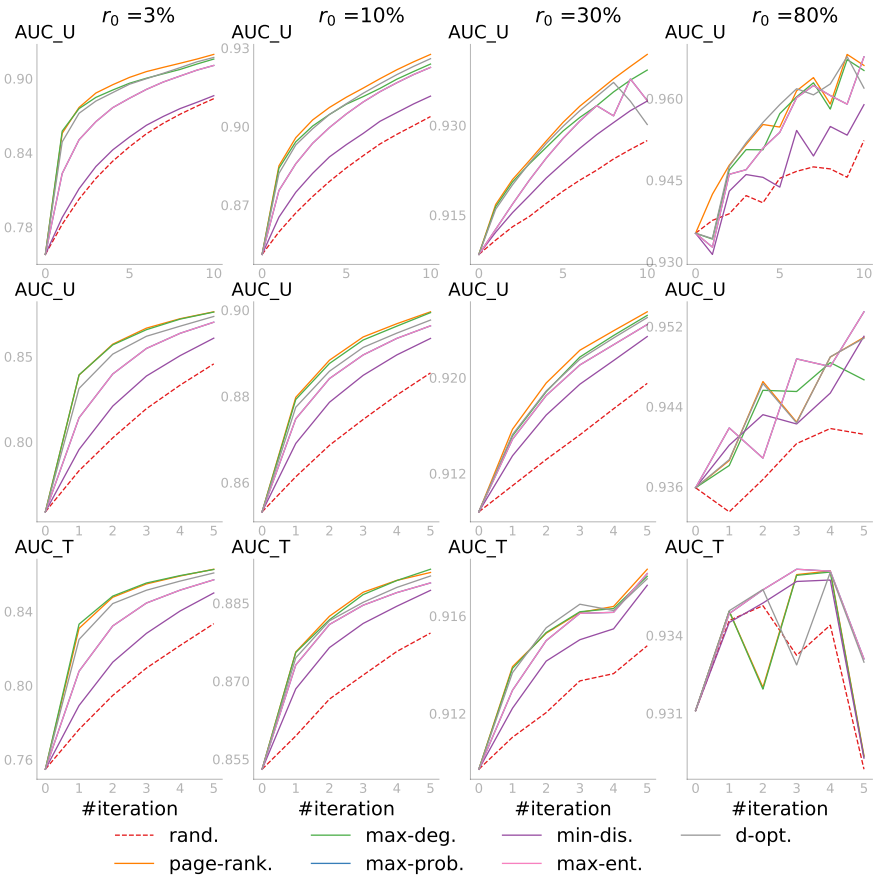


Figure 5.7: ALPINE on PPI_cc with $s = 74170$. Row 1: Case 1; Row 2: Case 2; Row 3: Case 3.

The complexity of the utility computation depended on the sizes of P and T, as well as the network. Normally, the larger the pool, the more expensive the computations were, as we had to consider more candidate node pairs. All query strategies, including **rand.**, required a similar computing time when given the same size of P. A notable exception is **v-opt.**, which was computationally more expensive. Yet, if we had a sufficiently accurate network embedding model, e.g., see the last columns in Figures 5.4–5.6, **v-opt.** was almost always the most accurate, especially for the first few iterations. Thus, when the cost of querying was high as compared to the cost of computations, **v-opt.** was preferable as soon as enough data were available such that the embedding was sufficiently accurate. If computational cost was a bottleneck though, **max-ent.** and **d-opt.** were computationally less expensive substitutes for **v-opt.**, with comparable accuracies.

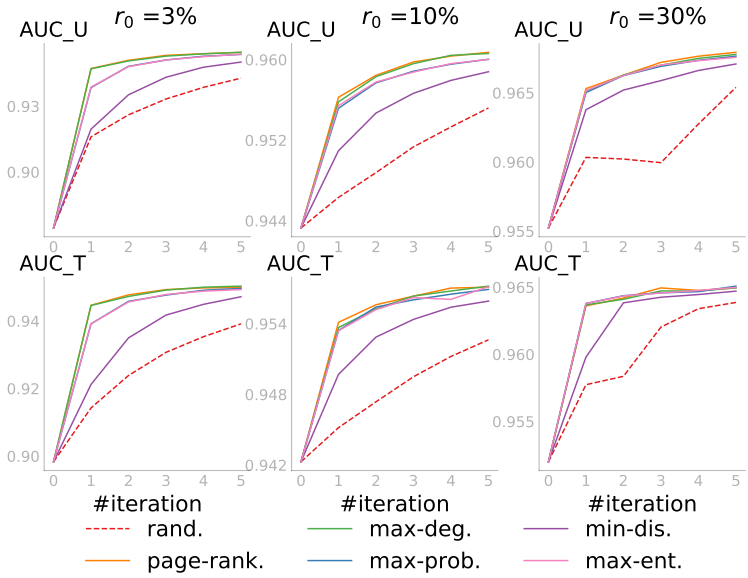


Figure 5.8: ALPINE on Blog with s = 531,635. Row 1: Case 2; Row 2: Case 3.

The experiments aimed to show how active learning, compared to passive learning, benefited the network embedding based link prediction, namely CNE. Therefore, following the line of research in active learning, we restricted our baselines to only the random and the state-of-the-art active learning strategies for link prediction [93, 95, 99, 110]. However, it would also be interesting to compare ALPINE with CNE against other types of link prediction methods to gain more insights. For example, a comparison of our work with a state-of-the-art link prediction approach (e.g., SEAL [129] according to [130, 131]) could be used to show whether the differentiation between the unknown and the unlinked status together with active learning would improve the link prediction accuracy in general. Note that this type of comparison can be biased as we had three types of link statuses, while other link prediction methods usually have only two. There are also other network embedding methods that can be used in combination with the ALPINE framework; thus the comparison among CNE and other base models can be considered. That leaves many possible opportunities for research to be built on this work.

5.6 Conclusions

Link prediction is an important task in network analysis, tackled increasingly using network embeddings. It is particularly important in partially observed networks,

where finding out whether a node pair is linked is time consuming or costly, such that for a large number of node pairs, it is not known if they are linked. We proposed to make use of active learning in this setting and studied the problem of active learning for link prediction using network embedding in this paper.

More specifically, we proposed the ALPINE framework, a method that actively learns to embed partially observed networks to achieve better link predictions, by querying the labels of the most informative unobserved link statuses. We developed several utility functions for ALPINE to quantify the utility of a node pair: some heuristically motivated and some derived as variance reduction methods based on D-optimality and V-optimality from optimal experimental design.

We implemented ALPINE in combination with Conditional Network Embedding (CNE). To accomplish this, we first adapted CNE to work for partially observed networks. Through experimental investigation, we found that this modified version of CNE was not only more time efficient, but also more accurate for link prediction—an important side-result of the present paper.

We then empirically evaluated the performance of the utility functions we developed for ALPINE, both qualitatively and quantitatively, providing insights into the merits of ALPINE and advice for practitioners on how to optimally apply this method to different problem settings.

More broadly, the application of active learning to the link prediction problem in general, which is usually for partially observed networks, could help us to build more realistic and practical methods. Taking this work as a starting point, we see interesting future directions, including the investigation of a mixed strategy, batch mode active learning for ALPINE, and the application of ALPINE to the cold-start problem in recommender systems. Meanwhile, a thorough comparison of ALPINE with CNE against general link prediction methods, as well as the choice of the base network embedding model to be used with the ALPINE framework remain to be further investigated.

Acknowledgment

We thank Ahmad Mel for helping collect the MP network data.

Appendices

5.A The Observed Fisher Information Matrix

CNE is defined as in Equation (5.1), aiming to find an embedding that maximizes the graph probability. To compute the Fisher information of CNE, we first need to compute the score, which is the partial derivative of the log likelihood function

$\log P(\mathcal{G}|\mathbf{X})$ with respect to the parameter. The parameters in CNE is the embedding matrix $\mathbf{X} \in \mathbb{R}^{n\times d}$, thus the score for one node embeddings $\mathbf{x}_i$ for $i \in V$ is [37]:

$$s(\mathbf{x}_i) = \frac{\partial \log P(\mathcal{G}|\mathbf{X})}{\partial \mathbf{x}_i} = \gamma \sum_{j\neq i}(P_{ij} - a_{ij})(\mathbf{x}_i - \mathbf{x}_j), \tag{5.11}$$

where $\gamma = \frac{1}{\sigma_1^2} - \frac{1}{\sigma_2^2}$ is a parameter in CNE, and P_{ij} represents $P(a_{ij} = 1|\mathbf{X})$.

Then the Fisher Information, defined as the variance of the score is $\mathcal{I}(\mathbf{x}_i) = \mathbb{E}\left[s(\mathbf{x}_i)s(\mathbf{x}_i)^T\right]$:

$$\mathcal{I}(\mathbf{x}_i) = \gamma^2 \sum_{j\neq i} P_{ij}(1 - P_{ij})(\mathbf{x}_i - \mathbf{x}_j)(\mathbf{x}_i - \mathbf{x}_j)^T. \tag{5.12}$$

The observed Fisher information that take into account only the observed part is thus,

$$\mathcal{I}(\mathbf{x}_i) = \gamma^2 \sum_{\{i,j\}\notin U} P_{ij}(1 - P_{ij})(\mathbf{x}_i - \mathbf{x}_j)(\mathbf{x}_i - \mathbf{x}_j)^T. \tag{5.13}$$

Full Hessian. When considering the entire embedding matrix $\mathbf{X}$, its Fisher Information is its full Hessian $\mathbf{H} \in \mathbb{R}^{nd\times nd}$ consisting of $n\times n$ blocks of size $d\times d$. The diagonal blocks $\mathcal{I}_{ii}(\mathbf{X}) = \mathcal{I}(\mathbf{x}_i)$, and the off-diagonal blocks $\mathcal{I}_{ij}(\mathbf{X})$ are defined as

$$\mathcal{I}_{ij}(\mathbf{X}) = \mathbb{E}\left[\frac{\partial \log P(\mathcal{G}|\mathbf{X})}{\partial \mathbf{x}_i}\frac{\partial \log P(\mathcal{G}|\mathbf{X})}{\partial \mathbf{x}_j}^T\right] = \gamma^2 P_{ij}(1-P_{ij})(\mathbf{x}_i-\mathbf{x}_j)(\mathbf{x}_j-\mathbf{x}_i)^T. \tag{5.14}$$

5.B The Prediction Variance

As mentioned, the prediction variance is computed via a first-order analysis of the prediction, and we provide the details here. The prediction $P_{ij} = P(a_{ij} = 1|\mathbf{X})$ is a function of $\mathbf{x}_i$ and $\mathbf{x}_j$, denoted $f(\mathbf{x}_i, \mathbf{x}_j)$, and it can be approximated by its first-order Taylor expansion at the MLE $\mathbf{X}^*$:

$$f(\mathbf{x}_i, \mathbf{x}_j) = f(\mathbf{x}_i^*, \mathbf{x}_j^*) + \frac{\partial f(\mathbf{x}_i^*, \mathbf{x}_j^*)}{\partial \mathbf{x}_i}(\mathbf{x}_i - \mathbf{x}_i^*) + \frac{\partial f(\mathbf{x}_i^*, \mathbf{x}_j^*)}{\mathbf{x}_j}(\mathbf{x}_j - \mathbf{x}_j^*). \tag{5.15}$$

Therefore, the prediction variance $\text{Var}(P_{ij})$ is

$$\begin{aligned}\text{Var}(P_{ij}) =& \frac{\partial f(\mathbf{x}_i^*, \mathbf{x}_j^*)}{\partial \mathbf{x}_i}^T \text{Var}(\mathbf{x}_i)\frac{\partial f(\mathbf{x}_i^*, \mathbf{x}_j^*)}{\partial \mathbf{x}_i} + \frac{\partial f(\mathbf{x}_i^*, \mathbf{x}_j^*)}{\partial \mathbf{x}_j}^T \text{Var}(\mathbf{x}_j)\frac{\partial f(\mathbf{x}_i^*, \mathbf{x}_j^*)}{\partial \mathbf{x}_j} \\ &+ 2\frac{\partial f(\mathbf{x}_i^*, \mathbf{x}_j^*)}{\partial \mathbf{x}_i}^T \text{Cov}(\mathbf{x}_i, \mathbf{x}_j)\frac{\partial f(\mathbf{x}_i^*, \mathbf{x}_j^*)}{\partial \mathbf{x}_j}.\end{aligned} \tag{5.16}$$

According to $\text{Var}(P_{ij})$, $\text{Var}(P^*_{ij})$ at the MLE then is

$$\text{Var}(P^*_{ij}) = \frac{\partial P^*_{ij}}{\partial \mathbf{x}_i}^T \text{Var}(\mathbf{x}^*_i) \frac{\partial P^*_{ij}}{\partial \mathbf{x}_i} + \frac{\partial P^*_{ij}}{\partial \mathbf{x}_j}^T \text{Var}(\mathbf{x}^*_j) \frac{\partial P^*_{ij}}{\partial \mathbf{x}_j} + 2 \frac{\partial P^*_{ij}}{\partial \mathbf{x}_i}^T \text{Cov}(\mathbf{x}^*_i, \mathbf{x}^*_j) \frac{\partial P^*_{ij}}{\partial \mathbf{x}_j}. \tag{5.17}$$

If we write the three terms in the equation above in a simpler form for brevity, i.e., as $\text{Var}_{\mathbf{x}^*_i}(P^*_{ij})$, $\text{Var}_{\mathbf{x}^*_j}(P^*_{ij})$, and $\text{Cov}_{\mathbf{x}^*_i,\mathbf{x}^*_j}(P^*_{ij})$, we get the expression of $\text{Var}(P^*_{ij})$ in Equation (5.9). Now we look at the bounds and take the first term for example, for which we first need to compute $\frac{\partial P_{ij}}{\partial \mathbf{x}_i}$ [37].

$$\frac{\partial P_{ij}}{\partial \mathbf{x}_i} = P_{ij} \frac{\partial \log P_{ij}}{\partial \mathbf{x}_i} = \gamma P_{ij}(1 - P_{ij})(\mathbf{x}_i - \mathbf{x}_j). \tag{5.18}$$

Then we have the first term $\text{Var}_{\mathbf{x}^*_i}(P^*_{ij})$ as follows, and it comes to the bound in Equation (5.10).

$$\begin{aligned} \text{Var}_{\mathbf{x}^*_i}(P^*_{ij}) =& \gamma^2 \left[P^*_{ij}(1 - P^*_{ij})\right]^2 (\mathbf{x}^*_i - \mathbf{x}^*_j)^T \text{Var}(\mathbf{x}^*_i)(\mathbf{x}^*_i - \mathbf{x}^*_j), & (5.19) \\ \geq& \gamma^2 \left[P^*_{ij}(1 - P^*_{ij})\right]^2 (\mathbf{x}^*_i - \mathbf{x}^*_j)^T \mathcal{I}(\mathbf{x}^*_i)^{-1} (\mathbf{x}^*_i - \mathbf{x}^*_j). & (5.20) \end{aligned}$$

6

Adversarial Robustness of Probabilistic Network Embedding for Link Prediction

Abstract In today's networked society, many real-world problems can be formalized as predicting links in networks, such as Facebook friendship suggestions, e-commerce recommendations, and the prediction of scientific collaborations in citation networks. Increasingly often, link prediction problem is tackled by means of network embedding methods, owing to their state-of-the-art performance. However, these methods lack transparency when compared to simpler baselines, and as a result their robustness against adversarial attacks is a possible point of concern: could one or a few small adversarial modifications to the network have a large impact on the link prediction performance when using a network embedding model? Prior research has already investigated adversarial robustness for network embedding models, focused on classification at the node and graph level. Robustness with respect to the link prediction downstream task, on the other hand, has been explored much less.

This paper contributes to filling this gap, by studying adversarial robustness of Conditional Network Embedding (CNE), a state-of-the-art probabilistic network embedding model, for link prediction. More specifically, given CNE and a network, we measure the sensitivity of the link predictions of the model to small adversarial perturbations of the network, namely changes of the link status of a node pair. Thus, our approach allows one to identify the links and non-links in the

network that are most vulnerable to such perturbations, for further investigation by an analyst. We analyze the characteristics of the most and least sensitive perturbations, and empirically confirm that our approach not only succeeds in identifying the most vulnerable links and non-links, but also that it does so in a time-efficient manner thanks to an effective approximation.

6.1 Introduction

Networks are used to model entities and the relations among them, so they are capable of describing a wide range of data in real world, such as social networks, citation networks, and networks of neurons. The recently proposed Network Embedding (NE) methods can be used to learn representations of the non-iid network data such that networks are transformed into the tabular form. The tabular data can then be fed to solve several network tasks, such as visualization, node classification, recommendation, and link prediction. We focus on link prediction that aims to predict future or currently missing links [33] as it has been widely applied in our lives. Examples include Facebook friendship suggestions, Netflix recommendations, predictions of protein-protein interactions, etc.

Many traditional link prediction approaches have been proposed [34], but the task is tackled increasingly often by the NE methods due to their state-of-the-art performance [35]. However, the NE methods lack transparency, e.g., Graph Neural Networks (GNNs) [50], when compared to simpler baselines. Thus, similar to many other machine learning algorithms [49], they could be vulnerable to adversarial attacks. It has been shown that simple imperceptible changes of the node attribute or the network topology can result in wrongly predicted node labels, especially for GNNs [51,52]. Meanwhile, adversarial attacks are easy to be found in our daily *online* lives, such as in recommender systems [132–134].

Robustness of NE methods for link prediction is important. Attacking link prediction methods can be used to hide sensitive links, while defending can help identify the interactions hidden intentionally, e.g., important connections in crime networks. Moreover, as links in online social networks represent the information sources and exposures, from the dynamic perspective, manipulations of network topology can be used to affect the formation of public opinions on certain topics, e.g., via exposing a targeted group of individuals to certain information sources, which is risky. The problem we want to investigate is: *Could one or a few small adversarial modifications to the network topology have a large impact on the link prediction performance when using a network embedding model?*

Existing adversarial robustness studies for NE methods mainly consider classification at the node and graph level, which investigates whether the labels will be wrongly predicted due to adversarial perturbations. It includes semi-supervised node classification [51,53,54,135–142], and graph classification [52,55,56]. Only

a few works consider the link-level task [131, 143–145], leaving robustness of NE methods for link prediction insufficiently explored.

To fill the gap, we study the adversarial robustness of Conditional Network Embedding (CNE) [37] for the link prediction task. CNE is a state-of-the-art probabilistic NE model that preserves the first-order proximity, of which the objective function is expressed analytically. Therefore, it provides mathematically principled explainability [146]. Moreover, compared to other NE models, such as those based on random walks [103, 104], CNE is more friendly to link prediction because the link probabilities follow directly from the model so there is no need to further train a classifier for links with the node embeddings. However, there has been no study on the adversarial robustness of CNE for link prediction.

In our work, we consider only the network topology as input, meaning that there is no node attribute. More specifically, given CNE and a network, we measure the sensitivity of the link predictions of the model to small adversarial perturbations of the network, i.e., the changes of the link status of a node pair. The sensitivity is measured as the impact of the perturbation on the link predictions. Intuitively, we quantify the impact as the KL-divergence between the two link probability distributions learned by the model from the clean and the corrupted network through re-training. While the re-training can be expensive, we develop effective and efficient approximations based on the gradient information, which is similar to the computation of the regularizer in Virtual Adversarial Training (VAT) [147]. Our main contributions are:

- We propose to study the adversarial robustness of a probabilistic network embedding model CNE for link prediction;
- Our approach allows us to identify the links and non-links in the network that are most vulnerable to adversarial perturbations for further investigation;
- With two case studies, we explain the robustness of CNE for link prediction through (a) illustrating how structural perturbations affect the link predictions; (b) analyzing the characteristics of the most and least sensitive perturbations, providing insights for adversarial learning for link prediction.
- We show empirically that our gradient-based approximation for measuring the sensitivity of CNE for link prediction to small structural perturbations is not only time-efficient but also significantly effective.

6.2 Related Work

Robustness in machine learning means that a method can function correctly with erroneous inputs [148]. The input data may contain random noise embedded, or adversarial noise injected intentionally. The topic became a point of concern when

the addition of noise to an image, which is imperceptible to human eyes, resulted in a totally irrelevant prediction label [49]. Robustness of models against noisy input has been investigated in many works [149–151], while adversarial robustness usually deals with the worst-case perturbations on the input data.

Network tasks at the node, link, and graph level are increasingly done by network embedding methods, which include shallow models and GNNs [152]. Shallow models either preserve the proximities between nodes (e.g., DeepWalk [103], LINE [153], and node2vec [104]) or factorize matrices containing graph information [154, 155] to effectively represent the nodes as vectors. GNNs use deep structure to extract node features by iteratively aggregating their neighborhood information, e.g., Graph Convolutional Networks (GCNs) [105] and GraphSAGE [156].

Adversarial learning for networks includes three types of studies: attack, defense, and certifiable robustness [157–159]. Adversarial attacks aim to maximally degrade the model performance through perturbing the input data, which include the modifications of node attributes or changes of the network topology. Examples of attacking strategies for GNNs include the non-gradient based NETTACK [51], Mettack using meta learning [136], SL-S2V with reinforcement learning [52], and attacks by rewiring for graph classification [55]. The defense strategies are designed to protect the models from being attacked in many different ways, e.g., by detecting and recovering the perturbations [138], applying adversarial training [49] to resist the worst-case perturbation [139], or transferring the ability to discriminate adversarial edges from exploring clean graphs [142]. Certifiable robustness is similar in essence to adversarial defense, but it focuses on guaranteeing the reliability of the predictions under certain amounts of attacks. The first provable robustness for GNNs was proposed to certify if a node label will be changed under a bounded attack on node attributes [141], and later a similar certificate for structural attack was proposed [54]. There are also robustness certifications for graph classification [56, 160] and community detection [161]. The most popular combination is GNNs for node or graph classification, while the link-level tasks have been explored much less.

Early studies on robustness for link-level tasks usually target traditional link prediction approaches. That includes link prediction attacks that aim to solve specific problems in the social context, e.g., to hide relationships [162, 163] or to disguise communities [164], and works that restrict the perturbation type to only adding or only deleting edges [165–167], which could result in less efficient attacks or defenses. The robustness for NE based link prediction is much less investigated than classification, and is considered more often as a way to evaluate the robustness of the NE method, such as in [144, 168, 169]. To the best of our knowledge, there are only two works on adversarial attacks for link prediction based on NE: one targeting the GNN-based SEAL [129] with structural perturbations and one targeting GCN with iterative gradient attack [143].

6.3 Preliminaries

In this section, we provide the preliminaries of our work, including the notations, the probabilistic network embedding model CNE that we use for link prediction, and the virtual adversarial training method to which the our idea is similar.

6.3.1 Link Prediction with Probabilistic Network Embedding

Network embedding methods map nodes in a network onto a lower dimensional space as real vectors or distributions, and we work with the former type. Given a network $G = (V, E)$, where V and E are the node and edge set, respectively, a network embedding model finds a mapping $f : V \rightarrow \mathbb{R}^d$ for all nodes as $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, ...\mathbf{x}_n]^T \in \mathbb{R}^{n \times d}$. Those embeddings $\mathbf{X}$ can be used to visualize the network in the d-dimensional space; classify nodes based on the similarity between vector pairs; and predict link probabilities between any node pair.

To do link prediction, a network embedding model requires a function g of vectors $\mathbf{x}_i$ and $\mathbf{x}_j$ to calculate the probability of nodes i and j being linked. This can be done by training a classifier with the links and non-links, or the function follows naturally from the model. Conditional Network Embedding (CNE) is the probabilistic model on which our work is based, and of which the function g directly follows [37]. Suppose there is an undirected network $G = (V, E)$ with its adjacency matrix $\mathbf{A}$, where $a_{ij} = 1$ if $(i, j) \in E$ and 0 otherwise, CNE finds an optimal embedding $\mathbf{X}^*$ that maximizes the probability of the graph conditioned on that embedding. It maximizes its objective function:

$$P(G|\mathbf{X}) = \prod_{(i,j)\in E} P(a_{ij} = 1|\mathbf{X}) \prod_{(k,l)\notin E} P(a_{kl} = 0|\mathbf{X}). \tag{6.1}$$

To guarantee that the connected nodes are embedded closer and otherwise farther, the method uses two half normal distributions for the distance d_{ij} between nodes i and j conditioned on their connectivity. By optimizing the objective in Eq. (6.1), CNE finds the most informative embedding $\mathbf{X}^*$ and the probability distribution $P(G|\mathbf{X}^*)$ that defines the link predictor $g(\mathbf{x}_i, \mathbf{x}_j) = P(a_{ij} = 1|\mathbf{X}^*)$.

Many network embedding methods purely map nodes into vectors of lower dimensions and focus on node classification, such as the random-walk based ones [103, 104, 153] and GCNs [105]. Those methods require an extra step to measure the similarities between the pairs of node embeddings for link prediction. Comparing to them, CNE is a better option for link prediction. Moreover, CNE provides good explainability for link predictions as g can be expressed analytically [146].

6.3.2 Virtual Adversarial Attack

Adversarial training achieved great performance for the supervised classification problem [49], and Virtual Adversarial Training (VAT) is better for the semi-supervised

setting [147]. By identifying the most sensitive 'virtual' direction for the classifier, VAT uses regularization to smooth the output distribution. The regularization term is based on the virtual adversarial loss of possible local perturbations on the input data point. Let $x \in \mathbb{R}^d$ and $y \in Q$ denote the input data vector of dimension d and the output label in the space of Q, respectively. The labeled data is defined as $\mathcal{D}_l = \left\{x_l^{(n)}, y_l^{(n)} | n = 1, ..., N_l\right\}$, the unlabeled data as $\mathcal{D}_{ul} = \left\{x_{ul}^{(m)} | m = 1, ..., N_{ul}\right\}$, and the output distribution as $p(y|x, \theta)$ parametrized by θ. To quantify the influence of any local perturbation on x_* (either x_l or x_{ul}), VAT has the Local Distribution Smoothness (LDS),

$$\text{LDS}(x_*, \theta) := D\left[p(y|x_*, \hat{\theta}), p(y|x_* + r_{vadv}, \theta)\right] \tag{6.2}$$

$$r_{vadv} := \text{argmax}_{r;||r||_2 \leq \epsilon} D\left[p(y|x_*, \hat{\theta}), p(y|x_* + r, \theta)\right], \tag{6.3}$$

where D can be any non-negative function that measures the divergence between two distributions, and $p(y|x, \hat{\theta})$ is the current estimate of the true output distribution $q(y|x)$. The regularization term is the average LDS for all data points.

Although VAT was designed for classification with tabular data, the idea of it is essentially similar to our work, i.e., we both quantify the influence of local virtual adversarial perturbations. For us, that is the link status of a node pair. As we have not yet included the training with a regularization term in this work, we now focus on finding the r_{vadv} in Eq. (6.3). That is to identify the most sensitive perturbations that will change the link probabilities the most.

6.4 Quantifying the Sensitivity to Small Perturbations

With the preliminaries, we now formally introduce the specific problem we study in this paper. That is, to investigate if there is any small perturbations to the network that have large impact on the link prediction performance. The small perturbations we look into are the edge flips, which represent either the deletion of an existing edge or the addition of a non-edge. It means that we do not restrict the structural perturbations to merely addition or merely deletion of edges.

Intuitively, that impact of any small virtual adversarial perturbation can be measured by re-training the model. But re-training, namely re-embedding the network using CNE, can be computationally expensive. Therefore, we also investigate on approximating the impact both practically with incremental partial re-embedding, and theoretically with the gradient information.

6.4.1 Problem Statement and Re-Embedding (RE)

The study of the adversarial robustness for link prediction involves identifying the worst-case perturbations on the network topology, namely the changes of the network topology that influence the link prediction results the most. For imperceptibility, we focus on the small structural perturbation of individual edge flip in this work. Thus, our specific problem is defined as

Problem 1 (Impact of a structural perturbation)**.** *Given a network $G = (V, E)$, a network embedding model, how can we measure the impact of each edge flip in the input network on the link prediction results of the model?*

Intuitively, the impact can be measured by assuming the edge flip as a virtual attack, flip the edge and retrain the model with the virtually corrupted network, after which we know how serious the attack is. That means we train CNE with the clean graph $G = (V, E)$ to obtain the link probability distribution $P^* = P(G|\mathbf{X}^*(\mathbf{A}))$. After flipping one edge, we get the corrupted graph $G' = (V, E')$, retrain the model, and obtain a different link probability $Q^* = Q(G'|\mathbf{X}^*(\mathbf{A}'))$. Then we measure the impact of the edge flip as the KL-divergence between P^* and Q^*. In this way, we also know how the small perturbation changes the node embeddings, which helps explain the influence of the virtual attack.

If the virtual edge flip is on node pair (i, j), $a'_{ij} = 1 - a_{ij}$ where a_{ij} is the corresponding entry in the adjacency matrix of the clean graph $\mathbf{A}$ and a'_{ij} of the corrupted graph $\mathbf{A}'$. Re-embedding G' with CNE results in probability $Q^*(i, j)$, then the impact of flipping (i, j), which we consider as the sensitivity of the model to the perturbation on that node pair, denoted as $s(i, j)$, is:

$$s(i, j) = KL\left[P^*||Q^*(i, j)\right]. \tag{6.4}$$

Measured practically, this KL-divergence is the actual impact for each possible edge flip on the predictions. The optimal embeddings $\mathbf{X}^*(\mathbf{A})$ and $\mathbf{X}^*(\mathbf{A}')$ not only explain the influenced link predictions but also exhibit the result of the flip.

Ranking the node pairs in the network by the sensitivity measure for all node pairs allows us to identify the most and least sensitive links and non-links for further investigation. However, re-embedding the entire network can be computationally expensive, especially for large networks. The sensitivity measure can be approximated both empirically and theoretically, and we will show how this can be done in the rest of this section.

6.4.2 Incremental Partial Re-Embedding (IPRE)

Empirically, one way to decrease the computational cost is to incrementally re-embed only the two corresponding nodes of the flipped edge. In this case, our assumption is that the embeddings of all nodes except the two connecting the flipped

edge (i.e., node i and j) will stay unchanged since the perturbation is small and local. We call it Incremental Partial Re-Embedding (IPRE), which allows only the changes of $\mathbf{x}_i$ and $\mathbf{x}_j$ if (i, j) is flipped. It means that the impact of the small perturbation on the link probabilities is restricted within the one-hop neighborhood of the two nodes, resulting in the changed link predictions between node i and j with the rest of the nodes. The definition of the impact in Eq. (6.4) still holds and only the ith and jth columns and rows in the link probability matrix have non-zero values. Comparing to RE, IPRE turns out to be a faster and effective approximation, which we will show with experiments.

6.4.3 Theoretical Approximation of the KL-Divergence

Incrementally re-embedding only the two nodes of the flipped edge is faster but it is still re-training of the model. Although our input is non-iid, in contrast to the tabular data used in VAT [147], we can form our problem as in Eq. (6.5), of which the solution is the most sensitive structural perturbation for link prediction.

$$\Delta\mathbf{A} := \text{argmax}_{\Delta\mathbf{A};||\Delta\mathbf{A}||=2} KL\left[P(G|\mathbf{X}^*(\hat{\mathbf{A}})), P(G|\mathbf{X}^*(\hat{\mathbf{A}} + \Delta\mathbf{A}))\right]. \quad (6.5)$$

CNE has its link probability distribution expressed analytically, so the impact of changing the link status of node pair (i, j), represented by the KL-divergence in Eq. (6.4) can be approximated theoretically. Given the clean graph G, CNE learns the optimal link probability distribution $P^* = P(G|\mathbf{X}^*(\mathbf{A}))$ whose entry is $P^*_{kl} = P(a_{kl} = 1|\mathbf{X}^*)$. Let $Q^*(i, j)$ be the optimal link probability distribution of the corrupted graph G' with only (i, j) flipped from the clean graph. The impact of the flip $s(i, j)$ can be decomposed as,

$$s(i, j) = KL\left[P^*||Q^*(i, j)\right] = \sum\left[p\log\frac{p}{q} + (1-p)\log\frac{1-p}{1-q}\right], \quad (6.6)$$

where p and q are entries of P^* and $Q^*(i, j)$ respectively. We can approximate $s(i, j)$ at G, or equivalently, at P^*, as G is close to G' thus P^* is close to $Q^*(i, j)$.

The first-order approximation of $s(i, j)$ is a constant because at G its gradient $\frac{\partial KL[P^*||Q^*(i,j)]}{\partial a_{ij}} = 0$, so we turn to the second-order approximation in Eq. (6.7), which, evaluated at G, is $\tilde{s}(i, j)$ in Eq. (6.8). That requires the gradient of each link probability w.r.t the edge flip, i.e., $\frac{\partial p}{\partial a_{ij}} = \frac{\partial P^*_{kl}}{\partial a_{ij}}$. Now we will show how to compute it with CNE.

$$s(i, j) \approx \frac{\partial KL\left[P^*||Q^*(i, j)\right]}{\partial a_{ij}}\Delta\mathbf{A} + \frac{1}{2}\frac{\partial^2 KL\left[P^*||Q^*(i, j)\right])}{\partial a_{ij}^2}\Delta\mathbf{A}^2, \quad (6.7)$$

$$\tilde{s}(i, j) = \frac{1}{2}\sum\frac{1}{p\,(1-p)}\left[\frac{\partial p}{\partial a_{ij}}\right]^2. \quad (6.8)$$

The gradient. At the graph level, the gradient of a link probability P^*_{kl} for node pair (k, l) w.r.t the input graph $\mathbf{A}$ is $\frac{\partial P^*_{kl}}{\partial \mathbf{A}} = \frac{\partial P^*_{kl}}{\partial \mathbf{X}^*(\mathbf{A})} \frac{\partial \mathbf{X}^*(\mathbf{A})}{\partial \mathbf{A}}$. While at the node pair level, the gradient of P^*_{kl} w.r.t. a_{ij} is

$$\frac{\partial P^*_{kl}}{\partial a_{ij}} = \frac{\partial P^*_{kl}}{\partial \mathbf{x}^*(\mathbf{A})} \frac{\partial \mathbf{x}^*(\mathbf{A})}{\partial a_{ij}} \tag{6.9}$$

$$= \mathbf{x}^{*T}(\mathbf{A}) \mathbf{E}_{kl} \mathbf{E}_{kl}^T \left[\frac{-\mathbf{H}}{\gamma^2 P^*_{kl}(1 - P^*_{kl})} \right]^{-1} \mathbf{E}_{ij} \mathbf{E}_{ij}^T \mathbf{x}^*(\mathbf{A}), \tag{6.10}$$

where for clearer presentation we flatten the matrix $\mathbf{X}$ to a vector $\mathbf{x}$ that is $nd \times 1$, $\mathbf{E}_{kl}$ is a column block matrix consisting of n blocks of size $d \times d$ where the k-th and l-th block are positive and negative identity matrix $\mathbf{I}$ and $-\mathbf{I}$ of the right size respectively and 0s elsewhere, and $\mathbf{H}$ is the full Hessian below

$$\mathbf{H} = \gamma \sum_{u \neq v} \left[(P^*_{uv} - a_{uv}) \mathbf{E}_{uv} \mathbf{E}_{uv}^T - \gamma P^*_{uv}(1 - P^*_{uv}) \mathbf{E}_{uv} \mathbf{E}_{uv}^T \mathbf{x}^*(\mathbf{A}) \mathbf{x}^{*T}(\mathbf{A}) \mathbf{E}_{uv} \mathbf{E}_{uv}^T \right].$$

The gradient reflects the fact that the change of a link status in the network influences the embeddings $\mathbf{x}^*$, and then the impact is transferred through $\mathbf{x}^*$ to the link probabilities of the entire graph. In other words, if an important relation (in a relatively small network) is perturbed, it could cause large changes in many P^*_{kl}s, deviating them from their predicted values with the clean graph.

The gradient in Eq. (6.10) is exact and measures the impact all over the network. However, the computation of the inverse of the full Hessian can be expensive when the network size is large. But fortunately, $\mathbf{H}$ can be well approximated with its diagonal blocks [146], which are of size $d \times d$ each block. So we can approximate the impact of individual edge flip with $\tilde{s}(i, j)$ at a very low cost using

$$\frac{\partial P^*_{kl}}{\partial a_{ki}} = (\mathbf{x}^*_k - \mathbf{x}^*_l)^T \left[\frac{-\mathbf{H}_k}{\gamma^2 P^*_{kl}(1 - P^*_{kl})} \right]^{-1} (\mathbf{x}^*_k - \mathbf{x}^*_i), \tag{6.11}$$

where $\mathbf{H}_k = \gamma \sum_{l:l \neq k} \left[(P^*_{kl} - a_{kl}) \mathbf{I} - \gamma P^*_{kl}(1 - P^*_{kl})(\mathbf{x}^*_k - \mathbf{x}^*_l)(\mathbf{x}^*_k - \mathbf{x}^*_l)^T \right]$ is the kth diagonal block of $\mathbf{H}$. Here P^*_{kl} is assumed to be influenced only by $\mathbf{x}_k$ and $\mathbf{x}_l$, thus only the edge flips involving node k or l will result in non-zero gradient for P^*_{kl}. It essentially corresponds to IPRE, where only the attacked nodes are allowed to move in the embedding space. In fact, as the network size grows, local perturbations are not likely to spread the influence broadly. We will show empirically this theoretical approximation is both efficient and effective.

6.5 Experiments

For the purpose of evaluating our work, we first focus on illustrating the robustness of CNE for link prediction with two case studies, using two networks of relatively small sizes. Then we evaluate the approximated sensitivity for node pairs on larger networks. The research questions we want to investigate are:

- How to understand the sensitivity of CNE to an edge flip for link prediction?
- What are the characteristics of the most and least sensitive perturbations for link prediction using CNE?
- What are the quality and the runtime performance of the approximations?

Data. The data we use includes six real world networks of varying sizes. **Karate** is a social network of 34 members in a university karate club, which has 78 friendship connections [69]. **Polbooks** network describes 441 Amazon co-purchasing relations among 105 books about US politics [125]. **C.elegans** is a neural network of the nematode C.elegans with 297 neurons linked by 2148 synapses [73]. **USAir** is a transportation network of 332 airports as nodes and 2126 airlines connecting them as links [126]. **MP** is the largest connected part of a Twitter friendship network for the Members of Parliament (MP) in the UK during April 2019, having 567 nodes and 49631 edges [40]. **Polblogs** is a network with 1222 political blogs as nodes and 16714 hyperlinks as undirected edges, which is the largest connected part of the US political blogs network from [125].

Setup. We do not have train-test split, because we want to measure the sensitivity of *all* link probabilities of CNE to *all* small perturbations of the network. The CNE parameters are $\sigma_2 = 2$, $d = 2$ for the case studies, $d = 8$ for evaluating the approximation quality, learning rate is 0.2, $\max\_\text{iter} = 2k$, and $\text{ftol} = 1e-7$.

6.5.1 Case Studies

The first two research questions will be answered with the case studies on Karate and Polbooks, which are relatively small thus can be visualized clearly. Both networks also have ground-truth communities, which contributes to our analysis. With Karate, we show how the small perturbations influence link probabilities via node embeddings. On Polbooks, we analyze the characteristics of the most and least sensitive perturbations. Note that we use the dimension 2 for both the visualization of CNE embeddings and the calculation of the sensitively.

Karate. To show the process of attacking CNE link prediction on Karate, we illustrate and analyze how the most sensitive edge deletion and addition affect the model in predicting links. With the RE approach, we measure the model sensitivity to single edge flip and find the top 5 sensitive perturbations in Table 6.1. The most sensitive deletion of link $(1, 12)$ disconnects the network, and we do not consider this type of perturbation in our work because it is obvious and easy to be detected. We see the other top sensitive perturbations are all cross-community, and we pick node pairs $(1, 32)$ and $(6, 30)$ for further study.

Fig. 6.1 shows the CNE embeddings of the clean Karate and the perturbed graphs, where the communities are differentiated with green and red color. CNE

Table 6.1: The Top 5 Sensitive Perturbations

Rank	Node Pair	s(i, j)	A[i, j]	Community?
1	(1, 12)	12.30	1	within
2	(1, 32)	2.52	1	cross
3	(20, 34)	1.96	1	cross
4	(6, 30)	1.75	0	cross
5	(7, 30)	1.75	0	cross

embeddings might have nodes overlap when $d = 2$, such as node 6 and 7, because they have the same neighbors, but this will not be a problem if d is higher.

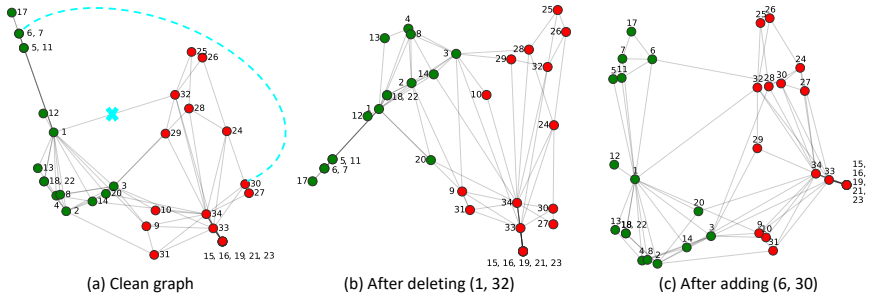


Figure 6.1: Case study on Karate with the most sensitive perturbations.

The deletion of edge $(1, 32)$ is marked with a cross in Fig. 6.1 (a), after which the changed node embeddings are shown in Fig. 6.1 (b). Although being rotated, the relative locations of the nodes change a lot, especially node 1, 32, and those in the boundary between the communities, e.g., node 3 and 10. Node 1 is pushed away from the red nodes, and as the center of the green nodes, it plays an essential role in affecting many other link probabilities. Comparing to other cross-community edges, $(1, 32)$ is the most sensitive because both nodes have each other as the only cross-community link. So the deletion largely decreases the probability of their neighbors connecting to the other community. Moreover, node 1 has a high degree. Therefore, it makes sense that this is the most sensitive edge deletion.

The addition of edge $(6, 30)$ is marked as a dashed arc in Fig. 6.1 (a), and the case is similar for $(7, 30)$. Adding the edge changes the node locations as shown in Fig. 6.1 (c). The distant tail in green that ends with node 17 moves closer to the red community. Note that both node 6 and 30 had only the within-community links before the perturbation. Even though their degrees are not very high, the added edge changes the probabilities of many cross-community links from almost zero to some degree of existence, pulling nodes to the other community.

Polbooks. Polbooks has three types of political books, which are liberal (L), neu-

Table 6.2: The Top Sensitive and Non-Sensitive Perturbations

Edge Deletion - S				Edge Deletion - Non-S			
Rank	Node Pair	s(i,j)	Community	Rank	Node Pair	s(i,j)	Community
1	(46, 102)	16.91	N-L	5460	(72, 75)	0.033	L-L
15	(7, 58)	14.64	N-C	5459	(8, 12)	0.034	C-C
Edge Addition - S				**Edge Addition - Non-S**			
Rank	Node Pair	s(i,j)	Community	Rank	Node Pair	s(i,j)	Community
2	(3, 98)	15.53	C-L	5458	(37, 39)	0.035	C-C
3	(3, 87)	15.42	C-L	5454	(8, 47)	0.036	C-C
4	(28, 33)	14.98	N-C	5451	(33, 35)	0.038	C-C
5	(25, 98)	14.96	C-L	5449	(30, 71)	0.039	L-L
6	(25, 91)	14.92	C-L	5438	(66, 75)	0.042	L-L

tral (N), and conservative (C), marked with colors red, purple, and blue, respectively. Shown in Table 6.2 are the most and least sensitive perturbations, where the left column are the Top 2 deletions and the middle and right columns are the top 5 additions. We do so as real networks are usually sparse. The rank is based on the sensitivity measure, thus the non-sensitive perturbations are ranked bottom (i.e., 5460). Then we will mark the those perturbations in the CNE embeddings, for edge deletions and additions separately.

The edge deletions are marked in Fig. 6.2, and we see the most sensitive ones are cross-community while the least sensitive ones are within-community. Similar to the Karate case, node pair $(46, 102)$ has each other as the only cross-community link, after deleting which the node embeddings will be affected significantly. Edge $(7, 58)$ is in the boundary between liberal and conservative nodes, and it has a neutral book. As the predictions in the boundary are already uncertain, one edge deletion would fluctuate many predictions, resulting in high sensitivity. The least-sensitive edge deletions are not only within-community, but are also between high-degree nodes, i.e., $d_{72} = 22$, $d_{75} = 16$, $d_8 = d_{12} = 25$. These nodes have already been well connected to nodes of the same type, thus they have stable embeddings and the deletions have little influence on relevant predictions.

We mark the edge additions separately for the sensitive and non-sensitive perturbations in Fig. 6.3, to contrast their difference. The left Fig. 6.3 (a) shows the top 5 sensitive edge additions are all cross-community, and all include at least one node at the distant place from the opposing community, i.e., nodes 33, 91, 87, 98. Being distant means those nodes have only the within-community connections, while adding a cross-community link would confuse the link predictor on the predictions for many relevant node pairs. Meanwhile, as the sensitive perturbations involve low-degree nodes, they are usually unnoticeable while weighted highly by

those nodes. The non-sensitive edge additions are similar to the non-sensitive deletions in the sense that both have the pair of nodes embedded closely. As long as the two nodes are mapped closely in the embedding space, it makes little difference if they are connected and the node degree does not matter much.

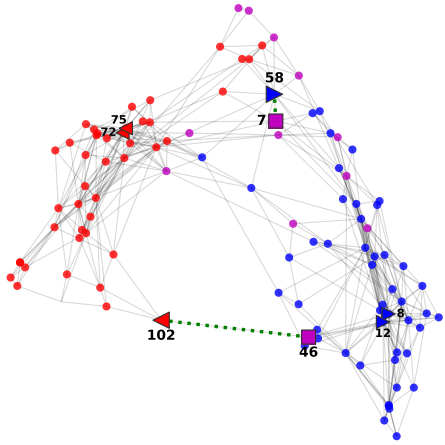


Figure 6.2: Case study on Polbooks with the most and least sensitive edge deletion.

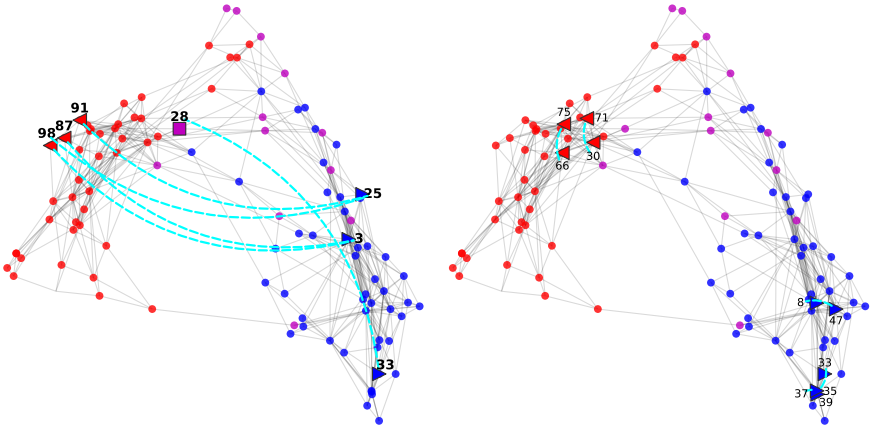


Figure 6.3: Case study on Polbooks with the most and least sensitive edge addition.

Interestingly, our observations in the case studies agree only partially with a heuristic community detection attack strategy called DICE [164], which has been used as a baseline for attacking link prediction in [143]. Inspired by modularity, DICE randomly disconnect internally and connect externally [164], of which the goal is to hide a group of nodes from being detected as a community. Our analy-

sis agrees with connecting externally, while for link prediction the disconnection should also be *external*, meaning that disconnecting internally might not work for link prediction. If the internal disconnection are sampled to node pairs that are closely positioned, the attack will have the little influence. Therefore, it might not be suitable to use DICE for link prediction attacks.

6.5.2 Quality and Runtime of the Approximations

We use the sensitivity measured by re-embedding (RE) as the ground truth impact of the small perturbations. The quality of an approximation is determined by how close it is to the ground truth. As the sensitivity is a ranked measure, we use the normalized discounted cumulative gain (NDCG) to evaluate the quality of the empirical approximation IPRE and the theoretical approximation with the diagonal Hessian blocks Approx. The closer the NDCG value is to 1, the better. We do not include the theoretical approximation with the exact Hessian because it can be more computationally expensive than RE for large networks. To show the significance, the p-value of each NDCG is found with randomization test of 1,000 samples. The runtime for computing the sensitivity of one edge flip is recorded on a server with Intel Xeon Gold CPU 3.00GHz and 1024GB RAM.

Shown in Table 6.3 are the quality of the approximations on five real-world networks. The first two columns show how well IPRE and Approx approximate RE, and the third column shows how well Approx approximates IPRE. We see the NDCG values in the table are all significantly high. Comparing to Approx, IPRE better approximates RE, and as the network size gets relatively large, the NDCG is alway larger than 0.99, indicating that the larger the network, the more local the impact of a small perturbation. For Approx, the NDCG for approximating RE are high across datasets, but it is even higher for IPRE. The reason is that both Approx and IPRE essentially make the same assumption that the influence of the perturbation will be spread only to the one-hop neighborhood.

Table 6.3: Quality of the Approximations - NDCG

ground truth	RE				IPRE	
approximation	IPRE		Approx		Approx	
	NDCG	p-value	NDCG	p-value	NDCG	p-value
Polbooks ($n = 105$)	0.9691	0.0	0.9700	0.0	0.9873	0.0
C.elegans ($n = 297$)	0.9977	0.0	0.9880	0.0	0.9905	0.0
USAir ($n = 332$)	0.9902	0.0	0.9697	0.0	0.9771	0.0
MP ($n = 567$)	0.9985	0.0	0.9961	0.0	0.9960	0.0
Polblogs ($n = 1222$)	0.9962	0.0	0.9897	0.0	0.9899	0.0

The approximations are not only effective, but also time-efficient. We see in

Table 6.4 that RE is the slowest, IPRE is faster, and Approx is significantly much faster than the previous two empirical approaches, especially for larger networks. On the Polblogs network, Approx is 36k times faster than RE and 22k times faster than IPRE. It shows that our method also scales to large networks.

Table 6.4: Runtime in seconds

	RE	IPRE	Approx
Polbooks	0.889	0.117	0.00012
C.elegans	2.819	0.568	0.00045
USAir	6.206	0.781	0.00043
MP	8.539	2.289	0.00116
Polblogs	45.456	27.648	0.00124

6.6 Conclusion

In this work we study the adversarial robustness of a probabilistic network embedding model CNE for the link prediction task by measuring the sensitivity of the link predictions of the model to small adversarial perturbations of the network. Our approach allows us to identify the most vulnerable links and non-links that if perturbed will have large impact on the model's link prediction performance, which can be used for further investigation, such as defending attacks by protecting those. With two case studies, we analyze the characteristics of the most and least sensitive perturbations for link prediction with CNE. Then we empirically confirm that our theoretical approximation of the sensitivity measure is both effective and efficient, meaning that the worst-case perturbations for link prediction using CNE can be identified successfully in a time-efficient manner with our method. For future work, we plan to explore the potential of our theoretical approximation to construct a regularizer for adversarially robust network embedding and to develop certifiable robustness for link prediction.

7 Conclusion

In this last chapter, we conclude the main research outputs of the thesis, summarize the findings of each problem investigated, and interpret how they can be used to contribute to our lives. Then, we discuss potential research directions for the future that can be built on the presented work.

7.1 Conclusion

This thesis consists of five research works on the two main topics: opinion formation and dynamics; and the vital network task of link prediction. Starting from the opinion formation process happening *on* social networks, we turn to the task of predicting links *of* networks in general because of the challenges in dealing with opinions on different issues. To better model the opinion formation process, we proposed the BEBA model to account for the backfire effect and biased assimilation simultaneously in Chapter 2 and the NFJ model that can normalize the influence one receives from friends in Chapter 3. These models can be used as the fundamentals for research works that measure opinion differences to quantify controversy, polarization, and conflict. Then we presented a new way of optimizing conflict in social networks that requires no opinions on specific issues in Chapter 4. That is to consider how resilient specific network topologies are to opinions over any set of topics. Our method can be applied to minimize the risk of conflict in organizations, companies, or any social media platform, preventing the community or even society and nation from being more divided. Lastly, we delve into the links

of networks and focus on building more data-efficient and robust link prediction methods using the network embedding approach in Chapters 5 and 6. The detailed findings and how they can be applied will be concluded in the remainder of this chapter.

The BEBA model proposed in Chapter 2 is the first DeGroot-type model that considers both the backfire effect and biased assimilation. The model uses one parameter, which we call the entrenchment parameter, as well as the opinion difference over the existing edges, to control the extent to which an individual opinion is influenced by a neighboring opinion. Theoretical results showed that the model naturally leads to opinion convergence from consensus to polarization. The empirical results demonstrated that the model not only makes sense for real-world data but also provides insights on how the opinion formation process is influenced by three factors: the initial opinion distribution over the network, the network topology, and the entrenchment parameter. Thus, our work is of potential help for designing effective intervention strategies on public opinions when there is the need to correct misinformation or fake news, Moreover, it can be used to defend malicious public opinion manipulation through changing the opinions, the network connections, or the individual entrenchment degree. However, the model in its current form has its limitations. Like many existing opinion formation models, BEBA considers opinions for each individual in the network on only one topic, while people usually communicate with each other on various issues. Those issues are not independent of each other. It is possible to incorporate opinion vectors of higher dimensions. However, the correlation among the issues remains a point of concern, which is one of the challenges we address in Chapter 4.

With the NFJ model studied in Chapter 3, we found a conflict eliminator via the theoretical analysis. The empirical investigation also showed that the NFJ model preserves controversy, which we could observe in reality, because it avoids too much opinion averaging. One interpretation of the normalization we introduced could be that people have limited energy and attention, so the portion allocated to the environmental information does not necessarily increase with the number of friends. Instead, as one makes more friends, each of them would get less opportunity to communicate with the person. Similarly, it would be interesting to investigate how the inner source of influence in the form of self-appraisal [7], represented as w_{ii} in the FJ model definition, will change and differ for people with different personalities because it also influences the formation of opinions. Many other exciting studies are also possible. One of the most interesting ideas, in our opinion, was to discount the environmental influence by the similarity of the opinions instead of the number of neighbors (i.e., the normalization). Essentially, both methods avoid over-averaging opinions, but the similarity-based approach might be more realistic. Similar people are more likely to be friends. Likewise, similar opinions on the same topics can trigger more significant influence. In contrast,

opposing opinions that are too different might backfire, which can be observed as arguing or fighting, meaning that the moderation of relatively extreme opinions rarely happens. This type of model is non-linear, while NFJ stays linear. The non-linear opinion formation model is what we investigated in Chapter 2 as chronologically BEBA was proposed later than the study of NFJ .

In Chapter 4, we tackled the problem of quantifying and minimizing conflict, namely the opinion divergence, in social networks without knowing any opinions. The work was motivated by two shortcomings of the state-of-the-art: the difficulty in obtaining opinions in practice and the fact that minimizing the conflict on one issue could lead to the rise of conflict on another in the same social network. Along the way of solving the problem, we discovered an interesting conservation law of conflict after summarizing existing literature. It indicates that conflict on one single issue remains a constant in a social network. Observing this, we departed from the literature to focus on the resilience of the network topology against conflict on all possible issues, which corresponds to a novel notion we introduced as the risk of conflict. Then we developed two algorithms for optimizing the risk of conflict in the average and the worst case over all opinion distributions of the network. Our theoretical and empirical results demonstrated the characteristics of the network structure with minimal conflict (for different kinds of measures). The experiments also showed that optimizing the worst-case conflict risk by editing the network structure is more effective for both cases, albeit more computationally expensive. Therefore, it could be meaningful to delve deeper into the risk of conflict in the worst case, both theoretically for more properties and empirically for higher efficiency. In that chapter, our research focus changed from the opinion formation process on the network to the topology of the network. To effectively reduce the risk of conflict, it is essential to know all the link information in the social network of interest. However, networks are usually partially observed, which turns out to be a vital network problem of link prediction, That problem is our focus for the subsequent two research contributions presented in Chapters 5 and 6.

Based on a state-of-the-art link prediction method using network embedding, namely CNE, we developed an active learning framework for link prediction on partially observed networks in Chapter 5 — the ALPINE framework, for improving data efficiency. In ALPINE, we first adapted the CNE model to embed the network with only the observed links and non-links, meaning that the observed non-links and unknown link status are treated differently. Experiments showed that the modified CNE using only the known information is both more time-efficient and effective for the link prediction task. With a set of query strategies we developed for use in combination with ALPINE, the most informative unknown link status can be identified, queried, and added into the training data to improve the link prediction performance. Qualitative evaluation of ALPINE confirmed that our method indeed identified the most informative unknown link status. Results of our

quantitative experiments further illustrated the merits of ALPINE and shed light on how to apply the proposed query strategies to real-world problems suitably. In practice, we regularly deal with partial information for problem-solving, so we hope that ALPINE can contribute to real-world problems that have already been formalized as link prediction, calling for more effective and efficient solutions.

Besides the data efficiency of link prediction methods, it is also important to ensure reliable link prediction results. Therefore, we investigated the adversarial robustness of the same probabilistic network embedding model CNE in Chapter 6. More specifically, the last paper chapter focused on measuring the model's sensitivity to small adversarial perturbations of the network connections, i.e., the change of the link status. With the proposed method, the most vulnerable links and non-links can be identified; thus, if they are verified or protected, the link probabilities provided by the model can be trusted. Because if they are ensured to have the correct signs (i.e., 1 for links and 0 for non-links), other link statuses are not significantly impacting the model performance. We illustrated with two case studies that our sensitivity measure is reasonable and empirically confirmed that the theoretical approximation we developed for it could also identify the most sensitive perturbations successfully, which saves much more time and is of significantly good quality. In its current form, the paper is ready for an extension for developing an adversarially robust version of CNE. If we can construct a regularizer using the theoretical approximation as in Virtual Adversarial Training (VAT) [147], the most sensitive perturbations would be taken into consideration during the model training. In this way, the robustness is embedded in the model and independent of the network. It is also possible to develop a robustness certification for link prediction based on our theoretical approximation. While certified robustness for node classification has been investigated, there is no mention of a similar notion for link prediction. More details concerning this direction will be discussed later in future work.

Chapters 5 and 6 use the same network embedding method CNE [37], and they have similar mathematics as both considered the gradient information of the model. However, the ideas are not the same, and the gradients are used for different purposes. In Section 5.4.3 of Chapters 5, the two variance reduction strategies used the fisher information, which can be derived as the Hessian of the *log*-likelihood w.r.t. the embeddings ($\mathbf{X}$). So we computed the partial derivative of the log-likelihood function (i.e., $\log P$) w.r.t. the parameter as in Appendix 5.A. While in Section 6.4.3 of Chapter 6, we calculate the gradient of the link probability (i.e., not $\log P$) with respect to the edge flip a_{ij}, during which we need a similar derivation as in Eq. (6.9): first the gradient of a link probability to $\mathbf{x}$, then the gradient of $\mathbf{x}$ to a_{ij}.

Limitations. That being said, our methods are far from perfect. Now we list the limitations for each contribution, which await future work.

- The BEBA model. The BEBA model suffers from a common issue for all opinion formation models; that is, there lacks the opinion data for validating if it accurately models the opinion formation process. In fact, the backfire effect is not supported by solid evidence due to non-robust experimental results. Without solid validation, we are still skeptical: are we modeling a real social phenomenon or just our feelings? Thankfully, our real-world data analysis showed promising results on the validity of the BEBA model. We hope to get more opinion data with the help of studies in natural language processing for opinion formation model validation. Another issue we want to address is the clipping in the current BEBA. It would be great to develop a variant of the model where the updated opinions fall naturally into the range of $[-1, 1]$. Moreover, we have only used a uniform entrenchment parameter for all nodes in a social network, so it would be more realistic to incorporate different values of β for the nodes.

- The NFJ model. Although the normalization has led to more sensible behavior than the FJ model, it is still not clear if the weights people put on their own internal opinions, i.e., w_{ii}, are constant. It is not realistic for w_{ii} to decreas as $|N(i)|$ increase. However, w_{ii} might change due to other reasons, such as reading, of which the measuring could be even more challenging than getting real-valued opinions. Meanwhile, that change of w_{ii}, as well as the value of w_{ii} itself, could differ among people. So the NFJ model still has a long way to go, and the primary focus could be the investigation on w_{ii}.

- Risk of conflict in social networks. One issue of the work is the scalability of the implemented algorithms. The largest network we experimented with is the Facebook network of 4039 nodes. Suppose we want to apply it to minimize the risk of conflict for a big company or a university whose network sizes are undoubtedly larger; our method might not work efficiently. In particular, if we want to optimize for the worst case, which performs better but is more computationally expensive. Therefore, developing more efficient solutions for the two optimization problems is necessary.

- ALPINE. Our work of ALPINE also suffers from the scalability issue, especially for the two query strategies that stem from D-optimality and V-optimality. The computations for evaluating the utilities of the candidate link statuses are indeed quite demanding in large-scare networks. It would be beneficial to improve the efficiency on this. Meanwhile, we used a common assumption for network embedding methods to initialize the PONs as connected. However, the realistic problem setting might not guarantee the initial connectivity. So it is worth investigating a mixed strategy that can cope with disconnected PONs, which could be similar to the cold-start problem. It would also be interesting to see the performance of ALPINE with

other network embedding models besides CNE. In principle, our framework works for any network embedding that can be expressed analytically, but its compatibility with different NE models waits to be studied. Lastly, taking into account that the datasets we experimented with are not necessarily ground truth, we might consider applying ALPINE to the signed networks, which are naturally suitable here. There are three types of connections in signed networks, i.e., the positive, negative, and unlinked, so we can embed node pairs with positive links closer, negative links farther, and query from the unlinked pairs. Many other possible cases can also be explored, such as four types of link statuses of positive, negative, unlinked, and unknown.

- Robustness for link prediction. For the simplicity of studying robustness for link prediction, we assumed the imperceptibility of the perturbations as a single edge flip on the networks. However, this might not be precise. As we can see in the case studies, those sensitive links and non-links could be pretty evident, i.e., they cross communities. It is tricky that although they are conspicuous in relatively small networks, they can be imperceptible in large networks (i.e., of millions of nodes). Thus, we need to define the notion of imperceptibility in our context with due consideration.

7.2 Directions for Future Work

With the five contributions from Chapters 2 to 6, we present our research on opinion dynamics and link prediction in this thesis, which are two important problems in computational social science and network science. Of course, there are many other exciting research directions for future work, including the search for more realistic and up-to-date opinion formation models; the study of emerging phenomena in our digital world (e.g., echo chamber or filter bubbles) using computational tools; the detection of malicious intervention on public opinion formation; the investigation of the interplay between opinion dynamics and link formation in networks (i.e., people that are more like-minded are more likely to be friends, and it is also true that friends tend to have similar opinions); and more. Among all, the ideas below are of most interest to us.

Opinion Embedding. Graph representation learning methods are capable of learning the representations of the nodes in networks [170], such as the CNE model we have been using for link prediction [37]. In addition to individuals in a social network who hold opinions, their opinions on different topics can also be represented as nodes, resulting in a heterogeneous network with two types of nodes (i.e., one for people and one for opinion) and links (i.e., for friendship and opinion). Usually, an opinion on a particular topic is within the range $[-1, 1]$, i.e., -1 for 'against', 1 for 'for', and the absolute value stands for the support level. It means that, for

each issue, we can introduce two nodes for the extreme opinions on both sides, e.g., node topic1_against and topic1_for, and the connections between people and the opinion nodes represent their opinions with the edge weights controlling the support level. Illustrated in Figure 7.1 is the basic idea of opinion embedding for a single topic.

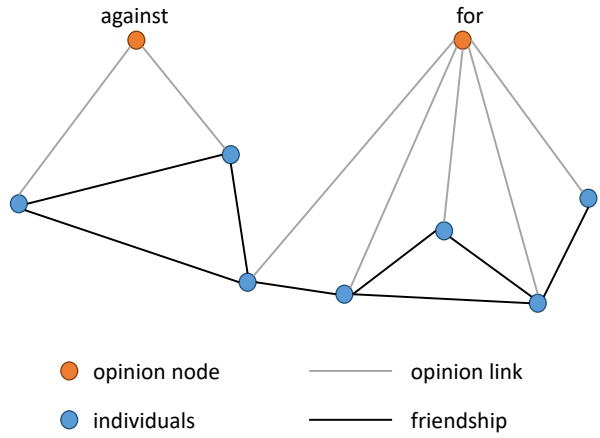


Figure 7.1: Illustration of Opinion Embedding for a single topic.

There are at least two advantages of opinion embedding. The first one is that the visualization of the network not only shows the supporting groups for both sides of each topic, but also reveals the correlations among different topics (i.e., suggested by the distances between the opinion nodes in the embedding space). It helps us understand and keep track of the social changes happening at a faster pace today due to the internet. The second benefit is that, by tracking the trajectory of the opinion nodes dynamically, the method is of potential help for defending malicious opinion manipulation, especially during political elections. If any abnormal or significant change of the node embeddings is detected, people will be warned of the risk that there might be malicious intervention. However, there is a challenge we face here. Unlike the nodes representing people, the two opinion nodes for each issue have ordinal differences (i.e., from -1 to 1). So we plan to find ways to encode the difference into the network structure or the network embedding procedure in the future.

Opinion Formation with Graph Neural Networks. Another way to address the challenge of high dimensional opinion vectors over a social network could be to use Graph Neural Networks (GNNs) [50], which is a class of network representation learning methods. Designed mainly for the network task of semi-supervised node classification, GNNs iteratively aggregate the neighborhood information (e.g., the node features) to represent the nodes as a set of real vectors. The

feature propagation in GNNs is essentially an averaging process over the (local) network connections. The process is similar to opinion formation on social networks if we consider the node opinions over different issues as the node features. Recent research has shown that while most GNNs are good at the semi-supervised node classification on networks with strong homophily (i.e., where similar nodes are linked), they achieve much less satisfying performance on graphs with heterophily (i.e., where most linked node pairs belong to different classes) [171, 172]. The assumption of GNNs on the network homophily may be problematic when they are applied to heterophilous networks. However, this is not a problem for social networks with homophily [173]. Therefore, it would be interesting to use learning methods like GNNs to investigate opinion formation, in contrast to the traditional ways of using mathematically defined updating rules. Furthermore, current research on how opinions are formed and information is diffused on social networks might help explain the patterns of feature propagation in GNNs.

Robustness Certification for Link Prediction. Last but not least, following our study on the robustness for link prediction in Chapter 6, we discuss how the results can be used to develop robustness certification for link prediction. Relevant research has been done for the classification of nodes or graphs, in which a target node or graph is certified to have always the true label under bounded perturbations [54, 56, 141, 160], while for community detection, a set of nodes are certified to be grouped into the desired communities under limited attacks [161]. However, similar concepts have not been considered for link prediction yet.

The robustness certification on classifications studies the worst-case margin between the true label and the predicted label under bounded attacks, e.g., on the network structure [54]. Given a network $G = (\mathbf{A}, \mathbf{X})$ where $\mathbf{A}$ and $\mathbf{X}$ are its adjacency and node feature matrices, respectively, the robustness certification of a model (e.g., GCN [105]), represented by its parameter θ, on a target node t can be given by first calculating the worst-case margin as:

$$m^t(y^*, y) := \text{minimize}_{\tilde{\mathbf{A}}} f_\theta^t(\mathbf{X}, \mathcal{T}(\tilde{\boldsymbol{A}}))_{y^*} - f_\theta^t(\mathbf{X}, \mathcal{T}(\tilde{\boldsymbol{A}}))_y, \tag{7.1}$$

$$\text{subject to } \tilde{\mathbf{A}} \in \mathcal{A}(\mathbf{A}), \tag{7.2}$$

where $f_\theta^t(\mathbf{X}, \mathcal{T}(\tilde{\boldsymbol{A}}))_y$ is the probability that the target node t is predicted to be in class y under the admissible perturbation $\tilde{\mathbf{A}}$ within the considered perturbation space defined as $\mathcal{A}(\mathbf{A})$. Then if the margin $m^t(y^*, y) > 0$ for all classes $y \neq y^*$, the target node t is certified to be classified correctly by the model under $\mathcal{A}$ when the ground truth network structure $\mathbf{A}^*$ can be reached via $\mathcal{A}$. It means that any possible perturbations within $\mathcal{A}$ will *not* lead to the change in the predicted class for node t, i.e., the probability of the node belonging to y^* is always the largest.

Similarly, the link prediction performance can be certifiable robust to bounded perturbations on the network structure. It turns out to be a much simpler problem than that for node classification because link prediction is basically a binary clas-

sification as a link can only exist or not. Therefore, the target to certify can be extended from a single link to a set of link statuses of our interest. To be specific, given a node pair set T, the certification is to ensure that under admissible perturbation within space $\mathcal{A}$, the link prediction results from our model on T can still be trusted. To do that, we need to define a similar worst-case margin as:

$$m^T := max(AUC_T(\tilde{\mathbf{A}})) - min(AUC_T(\tilde{\mathbf{A}})), \tag{7.3}$$

$$\text{subject to } \tilde{\mathbf{A}} \in \mathcal{A}(\mathbf{A}). \tag{7.4}$$

If $m^T \leq \epsilon$, the predictions of link statuses in T are certifiable robust with a margin of error up-bounded by ϵ. Therefore, this method can provide a quantification of reliability for link prediction on a set of node pairs, which is transferred from the uncertainty in data inputs as the perturbation space $\mathcal{A}$. We plan to figure out the mathematics for this idea in the future so the theoretical and empirical analysis can both be done.

Concerning data efficiency, there are many trendy machine learning approaches using transfer learning [174] or self-supervise learning [175], together with pre-training [176] that have achieved excellent performance in recent years [177–180]. For the link prediction task, they could be of help. For example, we might use the knowledge extracted from graphs of similar types and domains for predicting links on a new network such that the method can be more general. Based on it, further general models across domains might also be worth investigating. Alternatively, we could explore more on applying self-supervised learning for link prediction when no ground truth data is available because most relevant research focuses on the classification task of nodes or graphs.

We hope that this thesis can contribute to the research community and help us understand how our digitized societies affect our opinions and social relationships. Meanwhile, as recent decades have witnessed the rapid growth of Artificial Intelligence (AI), one could wonder: *Can machine form opinions?* If machine opinion formation for more complex AI is too far away or even impossible, can we use machines for investigating human opinion formation via controlling the information exposures? Let us wait together to see what surprise the future will bring us.

References

[1] M. H. DeGroot. *Reaching a Consensus*. Journal of the American Statistical Association, 69(345):118–121, 1974.

[2] N. E. Friedkin and E. C. Johnsen. *Social influence and opinions*. Journal of Mathematical Sociology, 15(3-4):193–206, 1990.

[3] M. O. Jackson. *Social and Economic Networks*. Princeton University Press, 2008.

[4] C. Castellano, S. Fortunato, and V. Loreto. *Statistical physics of social dynamics*. Reviews of Modern Physics, 81:591–646, 2009.

[5] D. M. Lazer, A. Pentland, D. J. Watts, S. Aral, S. Athey, N. Contractor, D. Freelon, S. Gonzalez-Bailon, G. King, H. Margetts, et al. *Computational social science: Obstacles and opportunities*. Science, 369(6507):1060–1062, 2020.

[6] P. Dandekar, A. Goel, and D. T. Lee. *Biased assimilation, homophily, and the dynamics of polarization*. Proceedings of the National Academy of Sciences, 110(15):5791–5796, 2013.

[7] P. Jia, A. MirTabatabaei, N. E. Friedkin, and F. Bullo. *Opinion Dynamics and the Evolution of Social Power in Influence Networks*. SIAM Review, 57(3):367–397, 2015.

[8] C. Musco, C. Musco, and C. E. Tsourakakis. *Minimizing Polarization and Disagreement in Social Networks*. In Proceedings of the 27th World Wide Web Conference, pages 369–378, 2018.

[9] A. Gionis, E. Terzi, and P. Tsaparas. *Opinion Maximization in Social Networks*. In Proceedings of the 13th SIAM International Conference on Data Mining, pages 387–395, 2013.

[10] A. Matakos, E. Terzi, and P. Tsaparas. *Measuring and moderating opinion polarization in social networks*. Data Mining and Knowledge Discovery, 31(5):1480–1505, 2017.

[11] M. Conover, J. Ratkiewicz, M. Francisco, B. Goncalves, F. Menczer, and A. Flammini. *Political Polarization on Twitter*. Proceedings of the International AAAI Conference on Web and Social Media, 5(1):89–96, 2011.

[12] L. Akoglu. *Quantifying Political Polarity Based on Bipartite Opinion Networks*. Proceedings of the International AAAI Conference on Web and Social Media, 8(1):2–11, 2014.

[13] R. L. Berger. *A Necessary and Sufficient Condition for Reaching a Consensus using DeGroot's Method*. Journal of the American Statistical Association, 76(374):415–418, 1981.

[14] R. A. Holley and T. M. Liggett. *Ergodic Theorems for Weakly Interacting Infinite Systems and the Voter Model*. The Annals of Probability, 3(4):643–663, 1975.

[15] J. T. Cox. *Coalescing Random Walks and Voter Model Consensus Times on the Torus in Zd*. The Annals of Probability, pages 1333–1366, 1989.

[16] K. Sznajd-Weron and J. Sznajd. *Opinion evolution in closed community*. International Journal of Modern Physics C, 11(06):1157–1165, 2000.

[17] S. Galam. *Minority opinion spreading in random geometry*. The European Physical Journal B - Condensed Matter and Complex Systems, 25(4):403–406, 2002.

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

[19] A. Bizyaeva, A. Franci, and N. E. Leonard. *A General Model of Opinion Dynamics with Tunable Sensitivity*. arXiv preprint arXiv:2009.04332, 2020.

[20] G. Deffuant, D. Neau, F. Amblard, and G. Weisbuch. *Mixing beliefs among interacting agents*. Advances in Complex Systems, 3(01n04):87–98, 2000.

[21] G. Deffuant, F. Amblard, G. Weisbuch, and T. Faure. *How can extremism prevail? A study based on the relative agreement interaction model*. Journal of Artificial Societies and Social Simulation, 5(4), 2002.

[22] P. Clifford and A. Sudbury. *A Model for Spatial Conflict*. Biometrika, 60(3):581–588, 1973.

[23] X. Chen, J. Lijffijt, and T. De Bie. *Quantifying and Minimizing Risk of Conflict in Social Networks*. In Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pages 1197–1205, 2018.

[24] D. Bindel, J. Kleinberg, and S. Oren. *How Bad is Forming Your Own Opinion?* Games and Economic Behavior, 92:248–265, 2015.

[25] K. Garimella, G. D. F. Morales, A. Gionis, and M. Mathioudakis. *Quantifying Controversy on Social Media*. ACM Transactions on Social Computing, 1(1):1–27, 2018.

[26] A. J. Morales, J. Borondo, J. C. Losada, and R. M. Benito. *Measuring political polarization: Twitter shows the two sides of Venezuela*. Chaos: An Interdisciplinary Journal of Nonlinear Science, 25(3):033114, 2015.

[27] Y. Choi, Y. Jung, and S.-H. Myaeng. *Identifying Controversial Issues and Their Sub-topics in News Articles*. In Proceedings of the 3rd Pacific-Asia Workshop on Intelligence and Security Informatics, pages 140–153, 2010.

[28] Y. Mejova, A. X. Zhang, N. Diakopoulos, and C. Castillo. *Controversy and Sentiment in Online News*. arXiv preprint arXiv:1409.8152, 2014.

[29] M. Thelwall. *The Heart and Soul of the Web? Sentiment Strength Detection in the Social Web with SentiStrength*, pages 119–134. Springer International Publishing, 2017.

[30] M. T. Al Amin, C. Aggarwal, S. Yao, T. Abdelzaher, and L. Kaplan. *Unveiling polarization in social networks: A matrix factorization approach*. In Proceedings of the 36th IEEE Conference on Computer Communications, pages 1–9, 2017.

[31] K. Garimella, G. D. F. Morales, A. Gionis, and M. Mathioudakis. *Balancing Opposing Views to Reduce Controversy*. arXiv preprint arXiv:1611.00172, 2016.

[32] K. Garimella, G. De Francisci Morales, A. Gionis, and M. Mathioudakis. *Reducing Controversy by Connecting Opposing Views*. In Proceedings of the 10th ACM International Conference on Web Search and Data Mining, pages 81–90, 2017.

[33] D. Liben-Nowell and J. Kleinberg. *The Link-Prediction Problem for Social Networks*. Journal of the American Society for Information Science and Technology, 58(7):1019–1031, 2007.

[34] V. Martínez, F. Berzal, and J.-C. Cubero. *A Survey of Link Prediction in Complex Networks*. ACM Computing Surveys, 49(4):1–33, 2016.

[35] A. C. Mara, J. Lijffijt, and T. De Bie. *Benchmarking Network Embedding Models for Link Prediction: Are We Making Progress?* In Proceedings of the 7th IEEE International Conference on Data Science and Advanced Analytics, pages 138–147, 2020.

[36] P. Cui, X. Wang, J. Pei, and W. Zhu. *A Survey on Network Embedding*. IEEE Transactions on Knowledge and Data Engineering, 31(5):833–852, 2019.

[37] B. Kang, J. Lijffijt, and T. De Bie. *Conditional Network Embeddings*. In Proceedings of the 7th International Conference on Learning Representations, 2019.

[38] X. Chen, P. Tsaparas, J. Lijffijt, and T. De Bie. *Opinion dynamics with backfire effect and biased assimilation*. PLOS ONE, 16(9):1–17, 2021.

[39] X. Chen, J. Lijffijt, and T. De Bie. *The Normalized Friedkin-Johnsen Model (A Work-in-progress Report)*. In the ECML PKDD 2018-PhD Forum, 2018.

[40] X. Chen, B. Kang, J. Lijffijt, and T. De Bie. *ALPINE: Active Link Prediction Using Network Embedding*. Applied Sciences, 11(11):5043, 2021.

[41] X. Chen, B. Kang, J. Lijffijt, and T. De Bie. *Adversarial Robustness of Probabilistic Network Embedding for Link Prediction*. arXiv preprint arXiv:2107.01936, 2021.

[42] E. Gilbert, T. Bergstrom, and K. Karahalios. *Blogs Are Echo Chambers: Blogs Are Echo Chambers*. In Proceedings of 42nd Hawaii International Conference on System Sciences, pages 1–10, 2009.

[43] C. G. Lord, L. Ross, and M. R. Lepper. *Biased Assimilation and Attitude Polarization: The Effects of Prior Theories on Subsequently Considered Evidence.* Journal of Personality and Social Psychology, 37(11):2098–2109, 1979.

[44] U. Krause et al. *A discrete nonlinear and non-autonomous model of consensus formation*. Communications in Difference Equations, 2000:227–236, 2000.

[45] B. Nyhan and J. Reifler. *When Corrections Fail: The Persistence of Political Misperceptions*. Political Behavior, 32(2):303–330, 2010.

[46] A. E. Allahverdyan and A. Galstyan. *Opinion Dynamics with Confirmation Bias*. PLOS ONE, 9(7):1–14, 2014.

[47] R. Abebe, J. Kleinberg, D. Parkes, and C. E. Tsourakakis. *Opinion Dynamics with Varying Susceptibility to Persuasion*. In Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pages 1089–1098, 2018.

[48] Y. Zhao, Y.-J. Wu, E. Levina, and J. Zhu. *Link Prediction for Partially Observed Networks*. Journal of Computational and Graphical Statistics, 26(3):725–733, 2017.

[49] I. J. Goodfellow, J. Shlens, and C. Szegedy. *Explaining and Harnessing Adversarial Examples*. In Proceedings of the 3rd International Conference on Learning Representations, 2015.

[50] M. Gori, G. Monfardini, and F. Scarselli. *A New Model for Learning in Graph Domains*. In Proceedings of 2005 IEEE International Joint Conference on Neural Networks, volume 2, pages 729–734, 2005.

[51] D. Zügner, A. Akbarnejad, and S. Günnemann. *Adversarial Attacks on Neural Networks for Graph Data*. In Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pages 2847–2856, 2018.

[52] H. Dai, H. Li, T. Tian, X. Huang, L. Wang, J. Zhu, and L. Song. *Adversarial Attack on Graph Structured Data*. In Proceedings of the 35th International Conference on Machine Learning, pages 1115–1124, 2018.

[53] K. Xu, H. Chen, S. Liu, P.-Y. Chen, T.-W. Weng, M. Hong, and X. Lin. *Topology Attack and Defense for Graph Neural Networks: An Optimization Perspective*. In Proceedings of the 28th International Joint Conference on Artificial Intelligence, pages 3961–3967, 2019.

[54] D. Zügner and S. Günnemann. *Certifiable Robustness of Graph Convolutional Networks under Structure Perturbations*. In Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pages 1656–1665, 2020.

[55] Y. Ma, S. Wang, T. Derr, L. Wu, and J. Tang. *Graph Adversarial Attack via Rewiring*. In Proceedings of the 27th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, page 1161?1169, 2021.

[56] H. Jin, Z. Shi, V. J. S. A. Peruri, and X. Zhang. *Certified Robustness of Graph Convolution Networks for Graph Classification under Topological Attacks*. In Proceedings of the 34th Annual Conference on Neural Information Processing Systems, volume 33, pages 8463–8474, 2020.

[57] X. Chen, P. Tsaparas, J. Lijffijt, and T. De Bie. *Opinion Dynamics with Backfire Effect and Biased Assimilation*. In the 15th International Workshop on Mining and Learning with Graphs (MLG), 2019.

[58] R. S. Baron, S. I. Hoppe, C. F. Kao, B. Brunsman, B. Linneweh, and D. Rogers. *Social Corroboration and Opinion Extremity*. Journal of Experimental Social Psychology, 32(6):537–560, 1996.

[59] A. Corner, L. Whitmarsh, and D. Xenias. *Uncertainty, scepticism and attitudes towards climate change: biased assimilation and attitude polarisation*. Climatic Change, 114(3-4):463–478, 2012.

[60] C. G. Lord and C. A. Taylor. *Biased Assimilation: Effects of Assumptions and Expectations on the Interpretation of New Evidence*. Social and Personality Psychology Compass, 3(5):827–841, 2009.

[61] D. Chong and J. N. Druckman. *Framing Public Opinion in Competitive Democracies*. The American Political Science Review, 101(4):637–655, 2007.

[62] P. M. Herr. *Consequences of Priming: Judgment and Behavior*. Journal of Personality and Social Psychology, 51(6):1106–1115, 1986.

[63] T. Wood and E. Porter. *The Elusive Backfire Effect: Mass Attitudes' Steadfast Factual Adherence*. Political Behavior, 41:135–163, 2019.

[64] B. Swire-Thompson, J. DeGutis, and D. Lazer. *Searching for the Backfire Effect: Measurement and Design Considerations*. Journal of Applied Research in Memory and Cognition, 9(3):286–299, 2020.

[65] C. Monti, G. De Francisci Morales, and F. Bonchi. *Learning Opinion Dynamics From Social Traces*. In Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pages 764–773, 2020.

[66] M. Del Vicario, A. Scala, G. Caldarelli, H. E. Stanley, and W. Quattrociocchi. *Modeling confirmation bias and polarization*. Scientific Reports, 7(1):1–9, 2017.

[67] D. Kempe, J. Kleinberg, S. Oren, and A. Slivkins. *Selection and influence in cultural dynamics*. Network Science, 4(1):1–27, 2016.

[68] A. Das, S. Gollapudi, and K. Munagala. *Modeling Opinion Dynamics in Social Networks*. In Proceedings of the 7th ACM International Conference on Web Search and Data Mining, pages 403–412, 2014.

[69] W. W. Zachary. *An Information Flow Model for Conflict and Fission in Small Groups*. Journal of Anthropological Research, 33(4):452–473, 1977.

[70] A. Zarezade, A. De, H. Rabiee, and M. G. Rodriguez. *Cheshire: An Online Algorithm for Activity Maximization in Social Networks*. arXiv preprint arXiv:1703.02059, 2017.

[71] A. De, I. Valera, N. Ganguly, S. Bhattacharya, and M. Gomez-Rodriguez. *Learning and Forecasting Opinion Dynamics in Social Networks*. In Proceedings of the 30th Annual Conference on Neural Information Processing Systems, volume 29, 2016.

[72] B. Bollobás. *Random Graphs*. Number 73 in Cambridge Studies in Advanced Mathematics. Cambridge University Press, 2 edition, 2001.

[73] D. J. Watts and S. H. Strogatz. *Collective dynamics of 'small-world' networks*. Nature, 393(6684):440–442, 1998.

[74] R. Albert and A.-L. Barabási. *Statistical mechanics of complex networks*. Reviews of Modern Physics, 74:47–97, 2002.

[75] R. Abebe, T.-H. H. Chan, J. Kleinberg, Z. Liang, D. Parkes, M. Sozio, and C. E. Tsourakakis. *Opinion Dynamics Optimization by Varying Susceptibility to Persuasion via Non-Convex Local Search*. ACM Transactions on Knowledge Discovery from Data, 16(2):1–34, 2021.

[76] J. Leskovec and J. J. Mcauley. *Learning to Discover Social Circles in Ego Networks*. In Proceedings of the 26th Annual Conference on Neural Information Processing Systems, volume 25, 2012.

[77] P. Guerra, W. Meira Jr., C. Cardie, and R. Kleinberg. *A Measure of Polarization on Social Media Networks Based on Community Boundaries*. Proceedings of the International AAAI Conference on Web and Social Media, 7(1):215–224, 2013.

[78] M. E. Newman. *Modularity and community structure in networks*. Proceedings of the National Academy of Sciences, 103(23):8577–8582, 2006.

[79] M. Coletto, K. Garimella, A. Gionis, and C. Lucchese. *A Motif-Based Approach for Identifying Controversy*. Proceedings of the International AAAI Conference on Web and Social Media, 11(1):496–499, 2017.

[80] W. Ellens, F. Spieksma, P. Van Mieghem, A. Jamakovic, and R. Kooij. *Effective graph resistance*. Linear Algebra and its Applications, 435(10):2491–2506, 2011.

[81] D. I. Shuman, N. S. K, P. Frossard, A. Ortega, and P. Vandergeynst. *The Emerging Field of Signal Processing on Graphs: Extending High-Dimensional Data Analysis to Networks and Other Irregular Domains*. IEEE Signal Processing Magazine, 30(3):83–98, 2013.

[82] Z.-Q. Luo, W.-K. Ma, A. M.-C. So, Y. Ye, and S. Zhang. *Semidefinite Relaxation of Quadratic Optimization Problems*. IEEE Signal Processing Magazine, 27(3):20–34, 2010.

[83] Y. Nesterov. *Semidefinite relaxation and nonconvex quadratic optimization*. Optimization Methods and Software, 9(1-3):141–160, 1998.

[84] M. X. Goemans and D. P. Williamson. *Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming*. Journal of the ACM, 42(6):1115–1145, 1995.

[85] T. D. Bie and N. Cristianini. *Fast SDP Relaxations of Graph Cut Clustering, Transduction, and Other Combinatorial Problems*. Journal of Machine Learning Research, 7:1409–1436, 2006.

[86] P.-W. Wang and J. Z. Kolter. *The Mixing method for Maxcut-SDP problem*. In Proceedings of the NIPS Workshop on Learning in High Dimensions with Structure, 2016.

[87] M. Frank and P. Wolfe. *An algorithm for quadratic programming*. Naval Research Logistics Quarterly, 3(1-2):95–110, 1956.

[88] E. S. Levitin and B. T. Polyak. *Constrained minimization methods*. USSR Computational Mathematics and Mathematical Physics, 6(5):1–50, 1966.

[89] R. K. Garrett. *Echo chambers online?: Politically motivated selective exposure among Internet news users*. Journal of Computer-Mediated Communication, 14(2):265–285, 2009.

[90] E. Bakshy, S. Messing, and L. A. Adamic. *Exposure to ideologically diverse news and opinion on Facebook*. Science, 348(6239):1130–1132, 2015.

[91] V. Amelkin, P. Bogdanov, and A. K. Singh. *A Distance Measure for the Analysis of Polar Opinion Dynamics in Social Networks*. In Proceedings pf the 33rd IEEE International Conference on Data Engineering, pages 159–162, 2017.

[92] N. Cesa-Bianchi, C. Gentile, F. Vitale, and G. Zappella. *A Linear Time Active Learning Algorithm for Link Classification*. In Proceedings of the 26th Annual Conference on Neural Information Processing Systems, volume 25, 2012.

[93] N. Ostapuk, J. Yang, and P. Cudré-Mauroux. *ActiveLink: Deep Active Learning for Link Prediction in Knowledge Graphs*. In Proceedings of the 28th World Wide Web Conference, pages 1398–1408, 2019.

[94] J. Jia, M. T. Schaub, S. Segarra, and A. R. Benson. *Graph-based Semi-Supervised & Active Learning for Edge Flows*. In Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pages 761–771, 2019.

[95] K.-J. Chen, J. Han, and Y. Li. *HALLP: A Hybrid Active Learning Approach to Link Prediction Task*. Journal of Computers, 9(3):551–556, 2014.

[96] C. Evans, J. Friedman, E. Karakus, and J. Pandey. *PotterVerse*. https://github.com/efekarakus/potter-network, 2014.

[97] K. Brinker. *Incorporating Diversity in Active Learning with Support Vector Machines*. In Proceedings of the 20th International Conference on Machine Learning, pages 59–66, 2003.

[98] H. Cai, V. W. Zheng, and K. C.-C. Chang. *Active Learning for Graph Embedding*. Arxiv Prepr. Arxiv:1705.05085, 2017.

[99] B. Settles. *Active Learning Literature Survey*. Technical report, University of Wisconsin–Madison, 2009.

[100] A. Atkinson, A. Donev, R. Tobias, et al. *Optimum Experimental Designs, with SAS*. Oxford University Press, 2007.

[101] D. A. Cohn, Z. Ghahramani, and M. I. Jordan. *Active Learning with Statistical Models*. Journal of Artificial Intelligence Research, 4:129–145, 1996.

[102] T. Zhang and F. J. Oles. *A Probability Analysis on the Value of Unlabeled Data for Classification Problems*. In Proceedings of the 17th International Conference on Machine Learning, pages 1191–1198, 2000.

[103] B. Perozzi, R. Al-Rfou, and S. Skiena. *DeepWalk: Online Learning of Social Representations*. In Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pages 701–710, 2014.

[104] A. Grover and J. Leskovec. *node2vec: Scalable Feature Learning for Networks*. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pages 855–864, 2016.

[105] T. N. Kipf and M. Welling. *Semi-Supervised Classification with Graph Convolutional Networks*. In Proceedings of the 5th International Conference on Learning Representations, 2017.

[106] X. Chen, G. Yu, J. Wang, C. Domeniconi, Z. Li, and X. Zhang. *ActiveHNE: Active Heterogeneous Network Embedding*. In Proceedings of the 28th International Joint Conference on Artificial Intelligence, pages 2123–2129, 2019.

[107] Z. Yang, W. Cohen, and R. Salakhudinov. *Revisiting Semi-Supervised Learning with Graph Embeddings*. In Proceedings of the 33rd International Conference on Machine Learning, pages 40–48, 2016.

[108] C. C. Aggarwal, X. Kong, Q. Gu, J. Han, and S. Y. Philip. *Active Learning: A Survey*. In Data Classification: Algorithms and Applications, pages 571–605. CRC Press, 2014.

[109] X. Kong, W. Fan, and P. S. Yu. *Dual Active Feature and Sample Selection for Graph Classification*. In Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pages 654–662, 2011.

[110] M. Bilgic, L. Mihalkova, and L. Getoor. *Active Learning for Networked Data*. In Proceedings of the 27th International Conference on Machine Learning, pages 79–86, 2010.

[111] N. Cesa-Bianchi, C. Gentile, F. Vitale, and G. Zappella. *Active Learning on Trees and Graphs*. Arxiv Prepr. Arxiv:1301.5112, 2013.

[112] A. Guillory and J. A. Bilmes. *Label Selection on Graphs*. In Proceedings of the 23rd Annual Conference on Neural Information Processing Systems, volume 22, 2009.

[113] Z. Yang, J. Tang, and Y. Zhang. *Active Learning for Streaming Networked Data*. In Proceedings of the 23rd ACM International Conference on Information and Knowledge Management, pages 1129–1138, 2014.

[114] C. Cortes and V. Vapnik. *Support-Vector Networks*. Machine Learning, 20(3):273–297, 1995.

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

[116] L. Page, S. Brin, R. Motwani, and T. Winograd. *The PageRank Citation Ranking: Bringing Order to the Web*. Technical report, Stanford InfoLab, 1999.

[117] K. Smith. *On the Standard Deviations of Adjusted and Interpolated Values of an Observed Polynomial Function and its Constants and the Guidance they give Towards a Proper Choice of the Distribution of Observations*. Biometrika, 12(1/2):1–85, 1918.

[118] W. J. Welch. *Computer-Aided Design of Experiments for Response Estimation*. Technometrics, 26(3):217–224, 1984.

[119] S. Liu and H. Neudecker. *A V-optimal design for Scheffé's polynomial model*. Statistics & Probability Letters, 23(3):253–258, 1995.

[120] E. L. Lehmann and G. Casella. *Theory of Point Estimation*. Springer Science & Business Media, 2006.

[121] C. R. Rao. *Information and the Accuracy Attainable in the Estimation of Statistical Parameters*. In Breakthroughs in Statistics, pages 235–247. Springer, 1992.

[122] B. Efron and D. V. Hinkley. *Assessing the Accuracy of the Maximum Likelihood Estimator: Observed Versus Expected Fisher Information*. Biometrika, 65(3):457–483, 1978.

[123] D. A. Harville. *Matrix Algebra From a Statistician's Perspective*, 1998.

[124] J. Sherman and W. J. Morrison. *Adjustment of an Inverse Matrix Corresponding to a Change in One Element of a Given Matrix*. The Annals of Mathematical Statistics, 21(1):124–127, 1950.

[125] L. A. Adamic and N. Glance. *The Political Blogosphere and the 2004 U.S. Election: Divided They Blog*. In Proceedings of the 3rd International Workshop on Link Discovery, pages 36–43, 2005.

[126] M. S. Handcock, D. R. Hunter, C. T. Butts, S. M. Goodreau, and M. Morris. *statnet: An R package for the Statistical Modeling of Social Networks*. Web page http://www.csde.washington.edu/statnet, 2003.

[127] B.-J. Breitkreutz, C. Stark, T. Reguly, L. Boucher, A. Breitkreutz, M. Livstone, R. Oughtred, D. H. Lackner, J. Bähler, V. Wood, et al. *The BioGRID Interaction Database: 2008 update*. Nucleic Acids Research, 36(suppl_-1):D637–D640, 2007.

[128] R. Zafarani and H. Liu. *Social Computing Data Repository at ASU*. Arizona State University, School of Computing, Informatics and Decision Systems Engineering, 2009. Available from: http://socialcomputing.asu.edu.

[129] M. Zhang and Y. Chen. *Link Prediction Based on Graph Neural Networks*. In Proceedings of the 32nd Annual Conference on Neural Information Processing Systems, volume 31, 2018.

[130] P. Li, Y. Wang, H. Wang, and J. Leskovec. *Distance Encoding: Design Provably More Powerful Neural Networks for Graph Representation Learning*. In Proceedings of the 34th Annual Conference on Neural Information Processing Systems, volume 33, pages 4465–4478, 2020.

[131] W. Lin, S. Ji, and B. Li. *Adversarial Attacks on Link Prediction Algorithms Based on Graph Neural Networks*. In Proceedings of the 15th ACM Asia Conference on Computer and Communications Security, pages 370–380, 2020.

[132] H. Zhang, Y. Li, B. Ding, and J. Gao. *Practical Data Poisoning Attack against Next-Item Recommendation*. In Proceedings of the 29th World Wide Web Conference, pages 2458–2464, 2020.

[133] Z. Liu and M. Larson. *Adversarial Item Promotion: Vulnerabilities at the Core of Top-N Recommenders That Use Images to Address Cold Start*. In Proceedings of the 30th World Wide Web Conference, pages 3590–3602, 2021.

[134] G. Yang, N. Z. Gong, and Y. Cai. *Fake Co-visitation Injection Attacks to Recommender Systems*. In Proceedings of the 24th Annual Network and Distributed System Security Symposium, 2017.

[135] D. Zügner, O. Borchert, A. Akbarnejad, and S. Guennemann. *Adversarial Attacks on Graph Neural Networks: Perturbations and their Patterns*. ACM Transactions on Knowledge Discovery from Data, 14(5):1–31, 2020.

[136] D. Zügner and S. Günnemann. *Adversarial Attacks on Graph Neural Networks via Meta Learning*. In Proceedings of the 7th International Conference on Learning Representations, 2019.

[137] D. Zhu, Z. Zhang, P. Cui, and W. Zhu. *Robust Graph Convolutional Networks Against Adversarial Attacks*. In Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pages 1399–1407, 2019.

[138] H. Wu, C. Wang, Y. Tyshetskiy, A. Docherty, K. Lu, and L. Zhu. *Adversarial Examples for Graph Data: Deep Insights into Attack and Defense*. In Proceedings of the 28th International Joint Conference on Artificial Intelligence, pages 4816–4823, 2019.

[139] Feng, Fuli and He, Xiangnan and Tang, Jie and Chua, Tat-Seng. *Graph Adversarial Training: Dynamically Regularizing Based on Graph Structure*. IEEE Transactions on Knowledge and Data Engineering, 33(6):2493–2504, 2021.

[140] A. Bojchevski and S. Günnemann. *Certifiable Robustness to Graph Perturbations*. In Proceedings of the 33rd Annual Conference on Neural Information Processing Systems, volume 32, 2019.

[141] D. Zügner and S. Günnemann. *Certifiable Robustness and Robust Training for Graph Convolutional Networks*. In Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pages 246–256, 2019.

[142] X. Tang, Y. Li, Y. Sun, H. Yao, P. Mitra, and S. Wang. *Transferring Robustness for Graph Neural Network against Poisoning Attacks*. In Proceedings of the 13th ACM International Conference on Web Search and Data Mining, pages 600–608, 2020.

[143] J. Chen, X. Lin, Z. Shi, and Y. Liu. *Link Prediction Adversarial Attack via Iterative Gradient Attack*. IEEE Transactions on Computational Social Systems, 7(4):1081–1094, 2020.

[144] A. Bojchevski and S. Günnemann. *Adversarial Attacks on Node Embeddings via Graph Poisoning*. In Proceedings of the 36th International Conference on Machine Learning, pages 695–704, 2019.

[145] Q. Dai, X. Shen, L. Zhang, Q. Li, and D. Wang. *Adversarial Training Methods for Network Embedding*. In Proceedings of the 28th World Wide Web Conference, pages 329–339, 2019.

[146] B. Kang, J. Lijffijt, and T. De Bie. *ExplaiNE: An Approach for Explaining Network Embedding-based Link Predictions*. arXiv preprint arXiv:1904.12694, 2019.

[147] T. Miyato, S.-i. Maeda, M. Koyama, and S. Ishii. *Virtual Adversarial Training: A Regularization Method for Supervised and Semi-Supervised Learning*. IEEE Transactions on Pattern Analysis and Machine Intelligence, 41(8):1979–1993, 2018.

[148] IEEE. *IEEE Standard Glossary of Software Engineering Terminology*. IEEE Std 610.12-1990, pages 1–84, 1990. doi:10.1109/IEEESTD.1990.101064.

[149] B. Mirzasoleiman, K. Cao, and J. Leskovec. *Coresets for Robust Training of Deep Neural Networks against Noisy Labels*. In Proceedings of the 34th Annual Conference on Neural Information Processing Systems, volume 33, pages 11465–11477, 2020.

[150] C. Zheng, B. Zong, W. Cheng, D. Song, J. Ni, W. Yu, H. Chen, and W. Wang. *Robust Graph Representation Learning via Neural Sparsification*. In Proceedings of the 37th International Conference on Machine Learning, pages 11458–11468, 2020.

[151] Q. Dai, Q. Li, J. Tang, and D. Wang. *Adversarial Network Embedding*. In Proceedings of the 32nd AAAI Conference on Artificial Intelligence, volume 32, 2018.

[152] X. Liu and J. Tang. *Network Representation Learning: A Macro and Micro View*. AI Open, 2:43–64, 2021.

[153] J. Tang, M. Qu, M. Wang, M. Zhang, J. Yan, and Q. Mei. *LINE: Large-scale Information Network Embedding*. In Proceedings of the 24th World Wide Web Conference, pages 1067–1077, 2015.

[154] X. Wang, P. Cui, J. Wang, J. Pei, W. Zhu, and S. Yang. *Community Preserving Network Embedding*. In Proceedings of the 31st AAAI Conference on Artificial Intelligence, volume 31, page 203?209, 2017.

[155] J. Qiu, Y. Dong, H. Ma, J. Li, K. Wang, and J. Tang. *Network Embedding as Matrix Factorization: Unifying DeepWalk, LINE, PTE, and node2vec*. In Proceedings of the 11th ACM International Conference on Web Search and Data Mining, pages 459–467, 2018.

[156] W. Hamilton, Z. Ying, and J. Leskovec. *Inductive Representation Learning on Large Graphs*. In Proceedings of the 31st Annual Conference on Neural Information Processing Systems, volume 30, 2017.

[157] L. Sun, Y. Dou, C. Yang, J. Wang, P. S. Yu, L. He, and B. Li. *Adversarial Attack and Defense on Graph Data: A Survey*. arXiv preprint arXiv:1812.10528, 2018.

[158] W. Jin, Y. Li, H. Xu, Y. Wang, S. Ji, C. Aggarwal, and J. Tang. *Adversarial Attacks and Defenses on Graphs: A Review, A Tool and Empirical Studies*. ACM SIGKDD Explorations Newsletter, 22(2):19–34, 2021.

[159] L. Chen, J. Li, J. Peng, T. Xie, Z. Cao, K. Xu, X. He, and Z. Zheng. *A Survey of Adversarial Learning on Graphs*. arXiv preprint arXiv:2003.05730, 2020.

[160] Gao, Zhidong and Hu, Rui and Gong, Yanmin. *Certified Robustness of Graph Classification against Topology Attack with Randomized Smoothing*. In Proceedings of the 21st IEEE Global Communications Conference, pages 1–6, 2020.

[161] J. Jia, B. Wang, X. Cao, and N. Z. Gong. *Certified Robustness of Community Detection against Adversarial Structural Perturbation via Randomized Smoothing*. In Proceedings of the 29th World Wide Web Conference, pages 2718–2724, 2020.

[162] A. M. Fard and K. Wang. *Neighborhood randomization for link privacy in social network analysis*. World Wide Web, 18(1):9–32, 2015.

[163] M. Waniek, K. Zhou, Y. Vorobeychik, E. Moro, T. P. Michalak, and T. Rahwan. *How to Hide One's Relationships from Link Prediction Algorithms*. Scientific Reports, 9(1):1–10, 2019.

[164] M. Waniek, T. P. Michalak, M. J. Wooldridge, and T. Rahwan. *Hiding individuals and communities in a social network*. Nature Human Behaviour, 2(2):139–147, 2018.

[165] K. Zhou, T. P. Michalak, M. Waniek, T. Rahwan, and Y. Vorobeychik. *Attacking Similarity-Based Link Prediction in Social Networks*. In Proceedings of the 18th International Conference on Autonomous Agents and Multiagent Systems, pages 305–313, 2019.

[166] K. Zhou, T. P. Michalak, and Y. Vorobeychik. *Adversarial Robustness of Similarity-based Link Prediction*. In Proceedings of the 19th IEEE International Conference on Data Mining, pages 926–935, 2019.

[167] S. Yu, M. Zhao, C. Fu, J. Zheng, H. Huang, X. Shu, Q. Xuan, and G. Chen. *Target Defense Against Link-Prediction-Based Attacks via Evolutionary Perturbations*. IEEE Transactions on Knowledge and Data Engineering, 33(2):754–767, 2021.

[168] S. Pan, R. Hu, G. Long, J. Jiang, L. Yao, and C. Zhang. *Adversarially Regularized Graph Autoencoder for Graph Embedding*. In Proceedings of the 27th International Joint Conference on Artificial Intelligence, pages 2609–2615, 2018.

[169] M. Sun, J. Tang, H. Li, B. Li, C. Xiao, Y. Chen, and D. Song. *Data Poisoning Attack against Unsupervised Node Embedding Methods*. arXiv preprint arXiv:1810.12881, 2018.

[170] W. L. Hamilton, R. Ying, and J. Leskovec. *Representation learning on graphs: Methods and applications*. arXiv preprint arXiv:1709.05584, 2017.

[171] J. Zhu, Y. Yan, L. Zhao, M. Heimann, L. Akoglu, and D. Koutra. *Beyond Homophily in Graph Neural Networks: Current Limitations and Effective Designs*. Proceedings of the 34th Annual Conference on Neural Information Processing Systems, 33, 2020.

[172] J. Zhu, R. A. Rossi, A. Rao, T. Mai, N. Lipka, N. K. Ahmed, and D. Koutra. *Graph Neural Networks with Heterophily*. In Proceedings of the 35th AAAI Conference on Artificial Intelligence, volume 35, pages 11168–11176, 2021.

[173] M. McPherson, L. Smith-Lovin, and J. M. Cook. *Birds of a Feather: Homophily in Social Networks*. Annual Review of Sociology, 27(1):415–444, 2001.

[174] S. J. Pan and Q. Yang. *A Survey on Transfer Learning*. IEEE Transactions on Knowledge and Data Engineering, 22(10):1345–1359, 2010.

[175] X. Liu, F. Zhang, Z. Hou, L. Mian, Z. Wang, J. Zhang, and J. Tang. *Self-supervised Learning: Generative or Contrastive*. IEEE Transactions on Knowledge and Data Engineering, pages 1–1, 2021. doi:10.1109/TKDE.2021.3090866.

[176] X. Han, Z. Zhang, N. Ding, Y. Gu, X. Liu, Y. Huo, J. Qiu, L. Zhang, W. Han, M. Huang, Q. Jin, Y. Lan, Y. Liu, Z. Liu, Z. Lu, X. Qiu, R. Song, J. Tang, J.-R. Wen, J. Yuan, W. X. Zhao, and J. Zhu. *Pre-Trained Models: Past, Present and Future*. AI Open, 2021. doi:https://doi.org/10.1016/j.aiopen.2021.08.002.

[177] S. Yuan, H. Zhao, Z. Du, M. Ding, X. Liu, Y. Cen, X. Zou, Z. Yang, and J. Tang. *WuDaoCorpora: A Super Large-scale Chinese Corpora for Pre-training Language Models*. AI Open, 2:65–68, 2021.

[178] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding*. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Volume 1, pages 4171–4186, 2019.

[179] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, and D. Amodei. *Language Models are Few-Shot Learners*. In Proceedings of the 34th Annual Conference on Neural Information Processing Systems, volume 33, pages 1877–1901, 2020.

[180] J. Qiu, Q. Chen, Y. Dong, J. Zhang, H. Yang, M. Ding, K. Wang, and J. Tang. *GCC: Graph Contrastive Coding for Graph Neural Network Pre-*

Training. In Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pages 1150–1160, 2020.

Social media play an increasingly important role in forming opinions, creating a need for deeper analysis of opinion dynamics and links in networks.