PROTECTING PRIVACY
WHEN MINING AND SHARING USER DATA

A DISSERTATION
SUBMITTED TO THE DEPARTMENT OF COMPUTER SCIENCE
AND THE COMMITTEE ON GRADUATE STUDIES
OF STANFORD UNIVERSITY
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

Aleksandra Korolova

August 2012

This dissertation is online at: http://purl.stanford.edu/db465vb0804

I certify that I have read this dissertation and that, in my opinion, it is fully adequate in scope and quality as a dissertation for the degree of Doctor of Philosophy.

Ashish Goel, Primary Adviser

I certify that I have read this dissertation and that, in my opinion, it is fully adequate in scope and quality as a dissertation for the degree of Doctor of Philosophy.

Timothy Roughgarden

I certify that I have read this dissertation and that, in my opinion, it is fully adequate in scope and quality as a dissertation for the degree of Doctor of Philosophy.

Nina Mishra

Approved for the Stanford University Committee on Graduate Studies.

Patricia J. Gumport, Vice Provost Graduate Education

This signature page was generated electronically upon submission of this dissertation in electronic format. An original signed hard copy of the signature page is on file in University Archives.

Abstract

In today's digital age, online services, such as search engines and social networks, collect large amounts of data about their users and their users' online activities. Large-scale mining and sharing of this data has been a key driver of innovation and improvement in the quality of these services, but has also raised major user privacy concerns.

This thesis aims to help companies find ways to mine and share user data for the purpose of furthering innovation while all the while protecting their users' privacy, and to motivate and help them reason about the privacy-utility trade-offs using a rigorous quantifiable definition of privacy. To achieve this we explore examples of privacy violations, propose privacy-preserving algorithms, and analyze the trade-offs between utility and privacy for several concrete algorithmic problems in search and social network domains.

We propose and execute two novel privacy attacks on an advertising system of a social network that lead to breaches of user privacy. The attacks take advantage of the advertising system's use of users' private profile data, the powerful microtargeting capabilities provided by the system, and the detailed ad campaign performance reports provided to advertisers, in order to infer private information about users. The proposed attacks build a case for a need to reason about data sharing and mining practices using a rigorous definition of privacy, elucidate the privacy and utility trade-offs that may arise in advertising systems that allow fine-grained targeting based on user profile and activity characteristics, and have contributed to changes in the social network's advertising system aimed at increasing the barriers to practical execution of such attacks in the future.

We propose a practical algorithm for sharing a subset of user search data consisting of queries and clicks in a provably privacy-preserving manner. The algorithm protects privacy by limiting the amount of each user's data used and, non-deterministically, throwing away infrequent elements in the data, with the specific parameters of the algorithm being determined by the privacy guarantees desired. The proposed algorithm, and the insights gained from its analysis offer a systematic and

practical approach towards sharing counts of user actions while satisfying a rigorous privacy definition, and can be applied to improve privacy in applications that rely on mining and sharing user search data.

We then present a quantitative analysis of privacy-utility trade-offs in the social recommendations and social data sharing domains using formal models of privacy and utility. For social recommendations, we present a lower bound on the minimum loss in privacy for link-based recommendation algorithms achieving good utility. For social data sharing, we present a theoretical and experimental analysis of the relationship between visibility of connections in the social network and the difficulty for a competing service to obtain knowledge of a large fraction of connections in the network. The methods of analysis introduced and the harsh trade-offs identified can be useful for guiding privacy-conscious development of social products and algorithms, and give a refined understanding of the privacy-utility trade-offs.

Few topics today arouse as much heated discussion as issues of user privacy. This thesis focuses on making practical and constructive strides towards understanding and providing tools for achieving a viable balance between two seemingly opposing needs – user data-driven innovation and privacy.

I dedicate this thesis to

my mom, Aida Korolova

my grandparents, Noemi and Sheftel Mihelovich

my love, Yaron Binur

and to my late advisor, Rajeev Motwani

Acknowledgements

My most favorite part of completing the thesis is having an occasion to thank the people in my life without whose mentorship, generosity with their time, inspiration, friendship, support, and love it would not have happened.

I was incredibly fortunate to start on a path that has led to this thesis under the advisement of Rajeev Motwani. Rajeev always had several problems to suggest, many of them inspired by the practical challenges one of the Internet companies was facing, and whenever I wanted to pursue my own, gently guided me to a problem formulation that could have scientific depth and some real-world impact. In those times when I could not seem to make any progress on the problem at hand, our weekly meeting always gave me a positive boost – Rajeev was never discouraged and always helped move things forward by suggesting a new idea or approach worth trying. Rajeev's mentorship has greatly influenced my taste in problems, and his unwavering support gave me something very valuable – an opportunity and confidence to work on problems I enjoy. Rajeev's ability to make me feel that he had all the time in the world for me, his friendliness, and generosity in sharing his wisdom and experience, made working with him a true joy.

I am deeply grateful to Ashish Goel for taking me on as his student after Rajeev's untimely passing, and becoming the nicest, most supportive and thoughtful advisor imaginable. Ashish's enthusiasm for my ideas made me eager and confident in continuing to pursue them; his always-on-the-spot guidance helped me make them better, and his willingness to share the research ideas he was passionate about and to explore them with me, gave me an invaluable opportunity to develop an interest and gain insights into new areas. I am also grateful to Ashish for being so approachable, patient, and laid-back with me, for the thoughtful advice on all the occasions I asked, and for the gentle encouragement and unwavering support that helped make this thesis happen.

I am grateful to Tim Roughgarden for his brilliant teaching and mentorship early in my Ph.D.

studies, thanks to which I greatly enjoyed learning the fields of algorithmic game theory and mechanism design. I have also greatly appreciated Tim's thoughtfulness in including me in his group's research and social activities, and his friendliness and sincerity when giving advice. But I am most grateful to Tim for his willingness to help whenever I asked. No matter how incoherent my rant about some technical challenge I was facing, Tim was always ready to listen, spend the time understanding the problem, and help me think about it in a better way. I owe a great deal to Tim's profound technical suggestions, astute questions and insightful observations about my work and work-in-progress.

Nina Mishra has been a treasured mentor, collaborator, and sounding board ever since my internship at Microsoft Search Labs in the summer of 2008. Nina has taught me how to become better at every part of research by personal example, by asking the tough but constructive questions, and by generously investing her time to help me iterate on whatever I was doing, be it developing an argument for why an idea is worthy to be pursued, improving the writing in a paper, or practicing and revising a presentation. I am especially grateful to Nina for the times when she sensed that I was struggling, called me up, offered to work together on whatever problem I wished, and then did that tirelessly and enthusiastically.

I am tremendously grateful to Krishnaram Kenthapadi, the collaboration with whom has also started in the summer of 2008 at Search Labs, for being the most reliable, friendly, and patient mentor and collaborator one can wish for. In addition to helping me turn many vague ideas into coherent and readable proofs, and helping turn the tedious experimental evaluations into a fun pair-programming process, through his approach to research Krishnaram has taught me that in order to make progress on any research task at hand, one doesn't have to wait for inspiration to strike, but merely to sit down and start calmly working out the details of it.

Ilya Mironov's amazing ability to take any research paper, and convey its insights in a few crisp intuitive sentences has helped me learn a lot about existing algorithms and privacy research. Ilya has also been a friendly, thought-provoking, and creative collaborator, who has greatly influenced my thinking and helped in research in the last couple of years. The times of brainstorming, debating, and working together with Nina, Krishnaram, and Ilya, have been the most joyful, exciting and rewarding of my Ph.D.

I thank John Mitchell, for serving on my defense and quals committees, for the support and advice throughout the years, and for sharing with me his innovative ideas on teaching and research-based entrepreneurial ventures. I am grateful to Lera Boroditsky for giving me the opportunity to

explore cognitive science together with her and her lab. I thank Cliff Nass for chairing my defense committee.

I am very grateful to my other collaborators, Shubha Nabar, Ying Xu, Alex Ntoulas, Ashwin Machanavajjhala, and Atish Das Sarma, joint work with whom appears in this thesis. I have learned so much from working together with you, and am grateful for the ideas and energy you have given me. I am also grateful to Sep Kamvar, Sreenivas Gollapudi, and Yun-Fang Juan, collaboration with whom may have not resulted in a published paper, but in whom I gained colleagues I enjoy exchanging ideas and working with, and whose opinions and advice on research and life I greatly value and appreciate.

Aneesh Sharma has been a great office mate, and a supportive, thoughtful and caring friend throughout the years. Thank you for listening to my rants, and helping me overcome technical and non-technical challenges with your always calm and positive attitude!

I am grateful to friends in the department for their friendship and cheerful support throughout (thank you, Esteban & Silvia, Zoltan & Kinga, Mukund, Sergei, Pranav, Steven, Dilys, David, Paul, Joni, Adam, Peter, Elie). And I am grateful to my many friends from outside the department, whose love and support kept me sane and happy throughout the Ph.D. years.

Going further back in time, I would not be the person and researcher I am today without the amazing teachers I have had in college and school. Patrick Winston's classes, although nominally about AI, have also influenced my thinking about research and life, Dan Spielman's class and Joe Gallian's REU program gave me the first exciting taste of research, and Philippe Golle and Ayman Farahat mentored me on my first-ever paper related to privacy. The math classes and fakultativ taught by Viktor Gluhov not only gave me a solid foundation in mathematics, but were also the part of school I looked forward to every day. The summer and winter camps organized by Vladimir Litvinskiy were the most educational and the most joyful parts of high school. The Latvian math olympiad movement led by Agnis Andžāns gave me an opportunity, motivation, and support to develop problem-solving skills, to see the world, and to have the skills and credentials to study in the US. I owe my algorithmic thinking and programming skills to Sergey Melnik of Progmeistars, attending whose after-school computer science program was another highlight of high school years. I am also grateful to my three English teachers, Zinaida Jasinskaja, Ljudmila Rumjanceva, and Daina Hitrova, whose lessons gave me a solid foundation for my English knowledge. I thank Denis for showing me the path to studying in the US and for his friendship in our teenage years. Finally, I am grateful to George Soros, whose foundation's programs have shaped the course of my life three times.

Rachel, Yuval, Anat, and Yoav, have been the best "in-law" family one can wish for long before their in-law status was official. Their support, advice, supply of home-made food, ability to comfort and readiness to help on everything from clothes shopping to proof-reading my writings, is something I greatly appreciate and treasure.

My grandparents, Noemi and Sheftel, instilled in me the love of learning and gave me the skills and opportunities to become good at it. Our discussions during walks and family dinners, and their example of attitude to work, family, and life, have profoundly shaped my interests, knowledge, and who I am as a person. I am also very grateful to other members of my family, especially my uncle Semen and my cousin Bella, for the many joyful moments we have shared and for their unwavering support and love.

I am grateful to my mom, Aida, for being my closest friend, my most reliable supporter, the source of comfort, confidence, and happiness. My mom's work and dedication to teaching gave me my love for and knowledge of mathematics, and her dedication to family gave me a tremendously happy childhood. Mamochkin, thank you for always believing in me, always supporting me, and for the sacrifices you have made to give me everything that I have. I love you very much!

Finally, I give my unending gratitude and love to the person who knows this thesis best, and without whose encouragement, creative ideas, support and love throughout the last decade it would not have been possible. Thank you, my dearest Yaron, for the person you have helped me become, for being a reliable, caring, thoughtful, resourceful and inspiring partner, for your love, and for the happiness and joy you bring me. I love you very much!

I gratefully acknowledge the financial support of my studies by Cisco Systems Stanford Graduate Fellowship, NSF Award IIS-0904325, NSF Grant ITR-0331640, TRUST (NSF award number CCF-0424422), and grants from Cisco, Google, KAUST, Lightspeed, and Microsoft.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

In today's digital age, online services, such as social networks, collect large amounts of data about their users and their users' online activities. This data can be beneficial for innovation and improvements of the user experience, but its treatment also presents challenges to individuals' privacy. The thesis analyzes the trade-offs between utility and privacy that arise when online services collect, data-mine and share user data, and develops algorithms that can enable the services to balance those trade-offs. Our study focuses on two types of online services that are at the core of how users explore information and interact online today – search engines and social network services – and analyzes several concrete problems faced by them.

1.1 The Benefits of Mining and Sharing User Data

While sometimes portrayed in a negative light, the data collected by search engines and social networks regarding their users' and their usage of the services, and the algorithms developed for mining and sharing that data, have been a key driver of innovation and improvement in the quality of these services.

Search engines continuously refine the quality of results presented to a particular user based on previous actions of other users who have posed the same or similar queries. The search engines accomplish this by varying the order and type of results presented to users and then recording and analyzing their responses as measured by actions such as time spent on the result page, which of the results are clicked, and whether or not the users attempt to reformulate the query [4, 5, 93, 155]. Thus, each individual using the search engine not only benefits from its service but also contributes

to its further improvement. Beyond improvements in ranking and presentation of results, the search engines have mined and shared the data collected in order to revolutionize spell-checking, make search query suggestions, and reinvent image search, to name only a few of the innovations.

Similarly, the analysis of data and actions of individual users of a social network makes the social network more valuable for all of its users. For example, by studying patterns of connections between existing social network users, the social network service can design algorithms to recommend other users whom a user who is just joining the network may know [82, 167, 170], thereby also increasing the new user's engagement with the network. In addition, by analyzing the items "liked", read or bought by users' friends, the social network service can recommend each user new items to look at, read, or buy [85, 87]. Furthermore, by combining information contained in a user profile with information about social connections and activity on the site, the social network service can present users with more relevant advertising [89, 153], recommend job openings within one's network and industry [11], suggest missing connections [17], and so on.

Besides improving the services search engines and social networks offer directly to their end users, the data collected about users and their activity can be used to improve products these companies offer to members of their ecosystem, including advertisers and API developers. For example, search engines report to advertisers the volume of searches for particular keywords and number of clicks on their links, and social networks provide breakdowns of demographic and interest composition of the audiences clicking on their ads. Such insights greatly benefit the search engines, the social networks, and the advertisers, as they help advertisers improve their campaigns, thus bringing more relevant ads, better user experience and higher revenue to the services [89, 90, 149].

The data and its aggregate releases are also an invaluable resource for the scientific community [76, 123] and for public service, as it can be used for large scale sociological studies. Examples of such studies include: measurement of moods and happiness depending on season, geographic location, and relationship status [66, 72], analysis of the structure and size of human groups and reach of social connections [15, 104], study of disease epidemics [63], forecasts of economic indicators such as travel and unemployment [183], issue analysis for political campaigns [68], and a dynamic census of human opinions [103]. For those who have access to the data, it offers a view into the structure and communication patterns of human society, as well as activities and interests of individuals at an unprecedented scale, granularity, and accuracy.

1.2 Privacy Challenges when Mining and Sharing User Data

As the preceding examples illustrate, the ability to mine and share parts of user data has been instrumental to the improvement of online services for users and other players of the Internet ecosystem. However, the practices needed to achieve these improvements are increasingly raising user privacy concerns [181]. Furthermore, in many contexts, there often exists a fundamental trade-off between the utility of shared data and privacy [43, 46, 50, 61].

As search engines and social networks aim to innovate, offer better products, provide better insights to advertisers, developers, and page managers, and to share more with the scientific and business communities, they incorporate more user data into their algorithms, share it at finer levels of granularity, and mine it for purposes different from the context in which the data was originally collected [84]. Offering more varied and relevant keyword suggestions based on user searches or providing more detailed reports about audiences clicking on the ads or engaging with a page, creates a competitive advantage for search engines and social networks, respectively. Presenting longer query suggestions and exposing more profile and social graph information offers a better user experience. Sharing the results of more targeted analyses provides more scientific and sociological value. But in the quest for improvement, the companies sometimes underestimate the effects these data mining and sharing practices can have on violating the privacy of individuals [53, 109].

The consequences of an inadvertent disclosure of a user profile or activity data as part of the output of an internal data-mining algorithm or as a result of data sharing can be quite severe both for the businesses of the online services, and for the individual users whose private data is disclosed.

For search engines and social network services, the trust of users in their privacy practices is a strategic product and business advantage. Loss of such trust could lead to abandonment or decreased usage of the service [181, 193], which in turn could lead to less data being available to improve the service, and therefore, deterioration in the quality of the service, which would lead to further decrease in usage, thereby creating a vicious cycle harming the business [77]. Respect for user privacy may also be a legally binding commitment, a violation of which can lead to procedural and monetary penalties [56, 57, 88].

From the users' perspective, insufficient privacy protections on the part of a service they use and entrust with their activity, personal or sensitive information could lead to significant emotional, financial, and physical harm. Although not all of these activities are legally permitted, a disclosure of user search activities may potentially lead to embarrassment [22], identity theft, and discrimination when determining insurance premiums [190]. Interest, activity, and behavioral data could be used out

of context by prospective employers [83], in legal proceedings [8,91], to predict sensitive information about users [95,141,157], or to influence people to adhere to stereotypes that society or companies expect of them [10]. It is hard to foresee all the risks that a digital dossier consisting of detailed profile and activity data could pose in the future, but it is not inconceivable that it could wreak havoc on users' lives.

Thus, although users appreciate the continual innovation and improvement in the quality of online services, they are also becoming increasingly concerned about their privacy, and about the ways their online identity and activity data is compiled, mined, and shared [154]. The online services, in turn, strive to strike a balance between mining and sharing user data in order to improve their products and user experience, further their business needs and fulfill obligations to advertisers and partners, and the desire and need to retain user trust and protect user privacy [184].

1.3 Protecting Privacy when Mining and Sharing User Data – Contributions and Structure

This thesis aims to help companies find ways to mine and share user data for the purpose of furthering innovation while all the while protecting their users' privacy, and to motivate and help companies reason about the privacy-utility trade-offs of their practices using a rigorous quantifiable definition of privacy as a foundation. To achieve this we have explored examples of privacy violations, proposed privacy-preserving algorithms, and analyzed trade-offs between utility and privacy for several concrete algorithmic problems in search and social network domains.

1.3.1 Contributions

Specifically, this thesis makes three concrete contributions:

I. Proposal and execution of two novel attacks on an advertising system of a social network that could lead to breaches of user privacy. The attacks took advantage of the advertising system's use of users' private profile data, the powerful microtargeting capabilities provided by the system, and the detailed ad campaign performance reports provided to advertisers, in order to infer private information about users. The proposed attacks and their analysis have presented a new example of a real-world system in which reliance on ad-hoc techniques to preserve privacy fails to achieve that goal, elucidated the privacy and utility trade-offs that may arise in advertising systems that allow fine-grained targeting based on user profile and

activity characteristics, and through the disclosure of findings, have contributed to changes in the social network's advertising system aimed at increasing the barriers to practical execution of such attacks in the future [81, 151].

II. Proposal of a practical algorithm for sharing a subset of user search data consisting of queries and clicks in a provably privacy-preserving manner. The algorithm protects privacy by limiting the amount of each user's data used and, non-deterministically, throwing away infrequent elements in the data, with the specific parameters of the algorithm being determined by the privacy guarantees desired. The proposed algorithm, and the insights gained from its analysis offer a systematic and practical approach towards sharing counts of user actions while satisfying a rigorous privacy definition, and can be applied to improve privacy in search applications that rely on mining and sharing user search data [105, 131].

III. A quantitative analysis of privacy-utility trade-offs in the social recommendations and social data sharing domains using formal models of privacy and utility. For social recommendations, we present a lower bound on the minimum loss in privacy for link-based recommendation algorithms achieving good utility. For social data sharing, we present a theoretical and experimental analysis of the relationship between visibility of connections in the social network and the difficulty for a competing service to obtain knowledge of a large fraction of connections in the network. The methods of analysis introduced and the harsh trade-offs identified can be useful for guiding privacy-conscious development of social network products and algorithms, and a refined understanding of the privacy-utility trade-offs [55, 126].

The presentation of the work is broken up into three main parts according to the contributions.

1.3.2 Part I – A Need for Privacy by Design

In Chapter 2 we detail examples of well-intentioned approaches to privacy-preserving data sharing and data mining that have nonetheless resulted in breaches of user privacy. In Section 2.1 we recap breaches that have occurred as a result of data sharing by two large Internet companies (AOL and Netflix). In Section 2.2 we describe our first contribution, two novel attacks on Facebook's advertising system, that illustrate that privacy breaches can occur also when companies are merely data mining user data for their own purposes, rather than broadly sharing it.

The examples presented in Chapter 2 motivate the study of algorithms for data sharing and mining that can be evaluated using a rigorous privacy definition, and in Chapter 3 we describe a

well-motivated and widely adopted privacy definition, *differential privacy*, due to Dwork et al. [49], which we adopt for the rest of the thesis.

The preliminary version of the work presented in Section 2.2 has appeared in [109] and was a co-winner of the 2011 PET Award for Outstanding Research in Privacy Enhancing Technologies [151][1]. The work has benefited from thoughtful feedback and constructive suggestions of Ashish Goel.

1.3.3 Part II – Algorithms for Sharing and Mining User Data Privately

In Part II we focus on algorithms enabling the search engine to mine and share subsets of user data while preserving user privacy.

1.3.3.1 Releasing Search Queries and Clicks Privately

The question of how to publish an anonymized search log was brought to the forefront by a well-intentioned, but privacy-unaware AOL search log release (Section 2.1.1). Since then a series of ad-hoc techniques have been proposed in the literature (Section 2.1.1.6), though none were known to be provably private. In Chapter 4, we describe a major step towards a solution: we show how queries, clicks and their associated perturbed counts can be published in a manner that rigorously preserves privacy, using the notion of privacy adopted in Chapter 3. Our algorithm is decidedly simple to state and formalizes the intuition of protecting privacy by "throwing away tail queries" into a practical algorithm with rigorous privacy guarantees. Our analysis of the proposed algorithm's privacy guarantees identifies and justifies the deviations from intuition necessary for protecting privacy, and mathematically quantifies exactly how to choose the parameters in order to achieve a desired level of privacy.

Relating back to the trade-off between privacy and utility is the question of whether the data we can safely publish using our proposed algorithm is of any use. Our findings offer a glimmer of hope: we demonstrate that a non-negligible fraction of queries and clicks can indeed be safely published via a collection of experiments on a real search log. In addition, we select two applications, keyword generation and studying human fears, and show that the keyword suggestions and top fear lists generated from the perturbed data resemble those generated from the original data. Thus, although there are other factors besides the privacy of users to consider when publishing search log data, the techniques we develop in Chapter 4 can be of immediate use for improving privacy protections in

[1] http://petsymposium.org/2012/award.
The contents of all hyperlinks referenced in this thesis were archived in June 2012, and are available at http://theory.stanford.edu/~korolova/Thesis/References

applications that rely on mining and sharing user search data, such as search log release or search suggestions.

The work in Chapter 4 is joint with Krishnaram Kenthapadi, Nina Mishra, and Alexandros Ntoulas, and has also benefited from insights gained in discussions with Rakesh Agrawal and Frank McSherry. The preliminary version of the work has appeared in the 18th International World Wide Web Conference [110], where it was one of six nominees for the Best Paper Award[2].

1.3.4 Part III – Quantifying Utility-Privacy Trade-offs for Social Data

In Part III we switch gears towards analyzing the privacy-utility trade-offs in the context of online social networks. In both the problems we analyze, the social connections represent valuable user data that the social network service aims to leverage in order to improve the service it provides, while at the same time protecting privacy. In the first problem, the social network needs to protect its users from adversaries interested in inferring private information about the users' connections. In the second, the social network service needs to protect itself from adversaries interested in gaining possession of the social network graph.

1.3.4.1 Social Recommendations

The ubiquitous adoption of social networks, such as Facebook and LinkedIn, made possible new forms of recommendations – those that rely on one's social connections in order to make personalized recommendations of ads, content, products, and people. Since recommendations may use sensitive information as part of their input, they are associated with privacy risks. In Chapter 5 we study whether personalized graph link-analysis based "social recommendations", or recommendations that are *solely* based on a user's social network, can be made without disclosing previously unknown sensitive links in the social graph to the user receiving recommendations. Our main contributions are intuitive and precise trade-off results between privacy and utility for a formal model of personalized social recommendations and a formal model of privacy, differential privacy. Concretely, we prove lower bounds on the minimum loss in utility for any recommendation algorithm that is differentially private, and strengthen these bounds for two particular utility functions used in social recommendation algorithms.

On the positive side, we show how two known methods can be adapted in order to turn any social

[2] http://www2009.org/best.html

recommendation algorithm into a privacy-preserving one. We then experimentally analyze the quality of recommendations made by privacy-preserving recommendation algorithms on two real-world networks. Both our theoretical analysis and experimental results strongly suggest a harsh trade-off between privacy and utility in this setting, implying that good private social recommendations may be feasible only for a small subset of the users in the social network or for a lenient setting of privacy parameters. Our findings may be applicable for improving privacy protections in applications that rely on mining private or sensitive graph links, and provide a strong motivation for a need to develop non-algorithmic approaches to enabling social recommendations that preserve privacy, such as design of interfaces that empower the users to control which data to exclude from being used as an input in social recommendation algorithms.

The work in Chapter 5 is joint with Ashwin Machanavajjhala and Atish Das Sarma, and has greatly benefited from ideas of and thought-provoking discussions with Arpita Ghosh and Tim Roughgarden. The preliminary version of the work has appeared in [129].

1.3.4.2 Social Graph Visibility

In Chapter 6 we analyze the difficulty of obtaining a large portion of a social network when given access to snapshots of local neighborhoods of some users in it. The question is motivated by the trade-off in the desire of a social network service, such as Facebook or LinkedIn, to enable their users to use the service most effectively to communicate and share information with people in their extended networks and the desire to protect the social graph they have spent time and effort building from falling into the hands of competitors and malicious agents.

We introduce a notion of *lookahead*, which formally captures the extent to which the social network's interface exposes the local neighborhoods in the social network graph to its users. We analyze, both experimentally and theoretically, the fraction of user accounts in the social network that an attacker would need to gain access to in order to learn the structure of a large fraction of the network, as a function of the attacker's strategy for choosing users whose accounts are subverted and as a function of the lookahead chosen by the social network for the user interface. The main contribution is in helping social network services understand and quantify the privacy-utility trade-offs in this setting.

The work in Chapter 6 is joint with Rajeev Motwani, Shubha U. Nabar, and Ying Xu, and the preliminary version of it has appeared in [111].

Part I

A Need for Privacy by Design

Chapter 2

Motivating Examples

In this chapter we detail examples of well-intentioned approaches to privacy-preserving data sharing and data mining that have nonetheless resulted in breaches of user privacy. In Section 2.1 we recap breaches that have occurred as a result of data sharing by two large Internet companies (AOL and Netflix) and have been extensively covered both in the academic literature and public press. In Section 2.2 we describe our first contribution, two novel attacks on Facebook's advertising system, that illustrate that privacy breaches can occur also when companies are merely data mining user data for their own purposes, rather than broadly sharing it. In all cases, the companies in question have made efforts to protect the privacy of their users, and we argue that they have failed because they have relied on intuition, rather than rigorous analysis, for protection. These real-world privacy breach examples build a case for the rest of the thesis work – the failures of heuristic and intuition based approaches to protect privacy motivate the need to start with a rigorous privacy definition, and then design and rigorously analyze algorithms for protecting privacy when mining and sharing user data that satisfy that definition.

2.1 Privacy Breaches when Data Sharing

We present two examples of large Internet companies, AOL and Netflix, sharing an anonymized version of their user data with the public. We describe the motivation for data sharing, the anonymization technique each company used, and the impact of the data sharing on furthering the goals that motivated it. We then describe the privacy breaches that have occurred as a result of the data sharing, their implications for the companies and their users, as well as discuss other natural candidates

for anonymization of the shared data, and why those techniques would also likely fail to protect privacy. The examples illustrate that protecting user privacy when sharing parts or aggregates of user data is a challenging task.

2.1.1 AOL Search Log Release

The AOL search log release [12] announced on August 3rd, 2006 included 21 million web queries posed by more than 500 thousand AOL users over a period of three months.

2.1.1.1 The Data

Search engines keep detailed records of user interactions with the search engine, called *search logs*. They consist of all the information the search engine can conceivably record about the interactions, such as: query-related information (what query was searched for, what results were displayed, what results were clicked, whether the user returned to view other results after the initial clickthrough), user-related information (the IP address and browser version of the device from which the search originated), and performance-related information (duration of time for the search engine to display results), etc. These logs have been successfully used by search engines in order to improve the quality of search results (e.g., to fix spelling errors, suggest related searches, expand acronyms and estimate query popularity over time), to measure their own performance (e.g., the average response time to a query), to identify areas for improvement (e.g., by looking at the query abandonment rates). Search logs are a gold mine for search engine innovation, both for designing better algorithms to find information and for creating new tools to analyze and predict activities in the world, such as real-time flu activity surveillance and early epidemic detection of illnesses based on search volume [63], economic forecasts about levels of activity in purchase categories such as automobile sales [183], and identification of political issues of interest to voters in each state [68].

2.1.1.2 Motivation for Data Release

The motivation for the AOL search log release[1] was to facilitate research on search and collaboration on such research between AOL and interested individuals. Search log data could be an invaluable source of innovation for many. Computer science researchers have been building a case for search log access [19, 194] so that they could study and analyze new information retrieval algorithms via

[1]The original post announcing the release has been removed but a screen cap and a data release announcement email can be found here: http://texturbation.com/blog/aoldata.jpg, http://sifaka.cs.uiuc.edu/xshen/aol/20060803_SIG-IRListEmail.txt

a common benchmark search log, learn about user information needs and query formulation approaches, build new systems tailored for particular types of queries such as questions [76]. Social scientists could investigate the use of language in queries as well as discrepancies between user interests as revealed by their queries versus as revealed by face-to-face surveys [178]. Advertisers could use the logs to understand how users navigate to their pages, gain a better understanding of their competitors, and improve keyword advertising campaigns.

2.1.1.3 Anonymization Technique

On the other hand, a release of unperturbed search logs to the greater public could be catastrophic from a privacy perspective, as users communicate with a search engine in an uninhibited manner, leaving behind an electronic trail of confidential thoughts and painfully identifiable information (e.g., their credit card numbers, decease symptoms, and names of friends or lovers).

In order to protect user privacy while sharing user data AOL had modified the raw search log data prior to its release. They had omitted IP addresses, browser and other user information, and for each user, published only their identifier, query, query time, the rank of the document clicked, and the domain of the destination URL. Furthermore, although AOL searches were tied to AOL usernames in their search logs, the user identifiers were replaced with randomly assigned numeric identifiers prior to publication.

2.1.1.4 Impact (Utility)

Although AOL has taken down the data within days of publication due to public outcry and privacy issues identified, the data has been downloaded, reposted[2], and made searchable by a number of sites. The release was greeted with enthusiasm by the industry and the academic research community [70, 76, 123], with the only reservations being related to the ethics of use due to privacy concerns related to the data [76]. The desire to access such data and potential for its productive utilization outside of search companies is high; workshops [39,136] that granted access to a shared dataset based on MSN Search query log under a strict no-redistribution license have garnered active participation and contributions from the research community.

[2] http://www.gregsadetsky.com/aol-data/

2.1.1.5 Impact (Privacy)

The consequences of the data release were devastating from the privacy perspective. In a matter of days, the identity of user #4417749 had been unmasked by the New York Times [22], and tied to Thelma Arnold, a 62-year old widow from Lilburn, GA, revealing her entire search history and portrait of her most private interests, from landscapers in her town to dating, her dog's habits, and diseases of her friends. Unmasking user #4417749 as Thelma could have been done by anyone with access to the published AOL logs, as the search history contained full names of Thelma's family members, their geographic locations, and other clues helpful in inferring the identity of the user posing the queries. Besides harm to Thelma, and other users whose names and social security numbers were published[3], the AOL search log release may have had other harmful consequences the extent of which is difficult to assess, such as: loss of user trust in AOL as a company, as well as, possibly, in other search engines, increased anxiety regarding the privacy of online activities for users, and hesitation of other companies to share their data to enable broader innovation [76]. As a consequence of an improper release of this private data set the CTO of AOL resigned, two employees were dismissed [196], and a class action lawsuit was filed.

2.1.1.6 Why Simple Tweaks won't Fix It

One can argue that although AOL's anonymization techniques were insufficient for protecting privacy, techniques that are better thought through would be more successful, and following AOL's release, many other ad-hoc anonymization techniques have been proposed. For example, if replacing usernames with numeric ids is not sufficient for protecting privacy, then it is natural to wonder if removing usernames preserves privacy. Such an approach would also likely fail to protect privacy since one could potentially stitch together queries of a search session belonging to the same user via the timestamps. Beyond inferring sessions, the heart of the problem lies in the fact that revealing a single query such as a credit card number breaks privacy.

If the queries themselves are private, then it is natural to wonder if hashing the queries preserves privacy. In fact, that too fails as Kumar et al. [115] nicely argue. They show that tokenizing a query, hashing the tokens and publishing the hashes does not preserve privacy since an adversary who has access to another log can reverse-engineer the tokens by utilizing the frequency with which the query appears in the log.

Jones et al. [97] study an application of simple classifiers to connect a sequence of queries to

[3] http://superjiju.wordpress.com/2009/01/18/aol-seach-query-database/

the location, gender and age of the user issuing the queries, and argue that releasing a query log poses a privacy threat because these three characteristics of the user can be used to create a set of candidate users who might have posed that query. Their more recent work [98] investigates privacy leaks that are possible even when queries from multiple users are grouped together and no user or session identifiers are released.

In short, while many works [1,194] describe seemingly promising ad-hoc techniques for protecting privacy when sharing search log data, the results are, by and large, negative for privacy.

2.1.2 Netflix Prize

The Netflix prize data release [25][4] announced on October 2nd, 2006, included over 100 million ratings given by over 480 thousand users to 17,700 movies.

2.1.2.1 The Data

Netflix is an online movie subscription rental and streaming service that enables its subscribers to watch movies and TV shows online or receive DVDs for watching by mail. One of the main differentiators of Netflix from previous generation movie rental companies such as Blockbuster, is that the service provides recommendations of new movies to watched based on the history of the user's watching and rating activity. Robust movie recommendations is one of the reasons for active subscriber engagement with Netflix's service [25, 117, 125].

2.1.2.2 Motivation for Data Release

The goal of the Netflix prize was to improve their existing movie recommendation algorithm. To achieve this goal, Netflix announced a contest with a million dollar prize for the best prediction algorithm and made an anonymized subset of user ratings data available to all interested participants. The high-profile contest and data release enabled Netflix to engage a broad range of technologists from all over the world in designing a better algorithm, at a tiny fraction of the cost that it would have cost Netflix to hire even some of them [125]. The contest provided researchers interested in developing recommendation algorithms with a training set of an unprecedented size and quality, movie industry professionals – with an insight into viewer ratings, sociologists – with a snapshot of interests of our generation, and so on.

[4] http://www.netflixprize.com

2.1.2.3 Anonymization Technique

On the other hand, it is clear that a release of an unperturbed movie rating database could be catastrophic from a privacy perspective. Users watch and rate a variety of movies reflecting their interests, concerns, and fears in the privacy of their homes, but they would not necessarily be comfortable with the whole world knowing what and when they watch, and how they evaluate it. For example, a teenager exploring their sexuality may not want classmates to become aware of this exploration by identifying that he or she has watched and rated highly a disproportionally large amount of gay/lesbian movies. An adult watching movies about people coping with life-threatening or chronic diseases may not want their employer or insurance company to become aware of it, etc.

In order to protect user privacy Netflix had made modifications to the data before making a subset of it available for download to the contest participants. They removed all user level information (such as name, username, age, geographic location, browser used, etc.), and deliberately perturbed "some of the rating data for some customers [...] in one or more of the following ways: deleting ratings; inserting alternative ratings and dates; and modifying rating dates" [25][5]. The published data consisted of tuples of: a randomly assigned numeric user id, movie, date of rating, and a numeric value of the rating on the scale from 1 to 5.

2.1.2.4 Impact (Utility)

The contest was a tremendous success from a business and publicity perspectives for Netflix, as well as from the perspective of furthering the science of recommendations. During the contest's three year duration 41 thousand teams from 186 different countries have downloaded the data and 5 thousand teams submitted an algorithm. The winning algorithm has shown significant improvement in the quality of recommendations over the algorithm developed by Netflix, and hence, is expected to further increase user engagement and satisfaction with the service [117, 125]. Furthermore, the contest has given high publicity and led to significant progress in the field of recommender systems [23, 107], resulting in establishment of matrix factorization techniques as the dominant methodology for implementing collaborative filtering [108, 177], introduction of two-layer undirected graphical models capable of representing and efficiently learning over large tabular data [160], and progress on modeling temporal dynamics in collaborative filtering [106]. The techniques developed are applicable to other companies recommending items to users, such as Amazon, Pandora, and others.

[5] http://www.netflixprize.com/rules

2.1.2.5 Impact (Privacy)

The consequences of the Netflix prize were dire from the privacy perspective. Narayanan and Shmatikov [147] have shown how to de-anonymize several users in the published dataset by cross-correlating published Netflix ratings with non-anonymous movie ratings on the Internet Movie Database (IMDb) website. While the ratings of movies users made on IMDb did not pose privacy risks as they were made consciously, the re-identification of users using their public ratings enabled the world to also see their private ratings. A surprising statistic about the anonymized Netflix dataset and the de-anonymization algorithm of [147] is how little auxiliary information is needed for reliable cross-correlation – with 8 movie ratings, 99% of records can be uniquely identified; for 68% of records two ratings and dates are sufficient. Moreover, the de-anonymization algorithm is robust to discrepancies in the ratings and dates, and works even though Netflix only published a sample of their user data. The work of [147] has offered a formal mathematical treatment of how a small amount of background information or auxiliary knowledge about an individual can facilitate a fairly reliable de-identification of that individual in a seemingly well-anonymized dataset.

Netflix's failure to fully protect the privacy of the users during the first data release affected their ability to run a follow-up contest with an expanded set of user data, such as gender, age, and location. The announcement of the follow-up contest was greeted with concern by privacy advocates [152] and a privacy law suit [169]. As a result, Netflix has decided to cancel the follow-up contest [88], a missed opportunity both for Netflix, their users, and the scientific community.

Furthermore, as in the AOL case, the failure to protect privacy during data sharing may have had a hard-to-measure negative impact. For example, particularly privacy conscious users may have stopped using Netflix or stopped rating or truthfully rating movies on the service due to lack of trust to the service. The second high-profile privacy breach due to user data release within a year of AOL's has significantly elevated privacy concerns in the minds of users and legislators, and have made other companies even more hesitant to share their data, which is a loss to society impeding innovation.

2.1.2.6 Why Simple Tweaks won't Fix It

It is hard to even conceive of other anonymization techniques for privacy-preserving sharing of Netflix data, given how bare-bones the data set published already was. [147] show that even after removing rating date information from consideration, 84% of subscribers can still be uniquely identified. One can consider excluding movies of sensitive context entirely, but the sensitivities of users may vary

widely, and while some may feel comfortable sharing their rating of a seemingly innocuous movie with the world, others may not be even comfosrtable admitting that they have watched it. One can request users to select which of their ratings they would be comfortable with being included in the contest, but it is unlikely that users would be willing to engage in such a time and thought-consuming effort.

In short, the hopes for coming up with strategies for protecting privacy in this context, and especially, in the context of a follow-up prize competition containing a broader range of user data, were rather glim. Up until the recent work of [133] there was no approach known for sharing Netflix data with guarantees that another clever attack would not be able to thwart the protections, with a tremendous hit to Netflix trustworthiness and to its users.

2.2 Privacy Breaches when Data Mining

In Section 2.1 we reviewed examples of user privacy breaches that were a result of data shared by companies using ad-hoc anonymization techniques to protect privacy. However, it is reasonable to presume that if the companies do not share anonymized versions of their user data and merely use it for their own internal data-mining purposes, then user privacy is protected. In this section we present the first contribution of the thesis, a study of a real-world system designed with an intention to protect privacy but without rigorous privacy guarantees, and experimental evidence that user privacy may be breached not only when user data is shared, but also when it is data-mined while relying on ad-hoc techniques to protect privacy.

The real-world data-mining system we study is the advertising system of the largest online social network at the time of writing, Facebook. We propose, describe, and provide experimental evidence of several novel approaches to exploiting the advertising system, its capabilities for fine-grained targeting and detailed campaign performance reports, in order to obtain private user information. In addition to building the case for developing techniques for mining and sharing user data that satisfy provable privacy guarantees, the disclosure of our findings to Facebook and their response has contributed to making the kinds of attacks identified more difficult to execute in the future [81,151].

Organization. In Section 2.2.1 we give the background on the trade-offs Facebook faces with regards to protecting user privacy while enabling monetization of their data. We describe the data users share with Facebook, the privacy controls available to them, and user privacy expectations in Section 2.2.2.1, and the capabilities provided to advertisers by Facebook in Section 2.2.2.2 We introduce the underlying causes of privacy leaks, our proposed attack blueprints, and present our

experimental evidence of their success in Section 2.2.3. We discuss our results, their implications, and related work in Sections 2.2.4 and 2.2.5. We conclude in Section 2.2.6 with a discussion of Facebook's response to our research disclosure and a discussion of the challenges of designing privacy-preserving microtargeted advertising systems.

2.2.1 Introduction: Facebook, Targeted Advertising, and Privacy

As more people rely on online social networks to communicate and share information with each other, the social networks expand their feature set to offer users a greater range of the type of data they can share. As a result, more types of data about people is collected and stored by these online services, which leads to increased concerns related to its privacy and re-purposing. One of the big concerns users have when they share personal information on social networking sites is the possibility that their personal information may be sold to advertisers [164, 176].

Although leading social networks such as Facebook have refrained from selling the information to advertisers, in order to monetize the data they possess they have created systems that enable advertisers to run highly targeted social advertising campaigns. Not surprisingly, the goals of monetization through enabling highly targeted advertising and protecting the privacy of users' personal information entrusted to the company are at odds. To reconcile these conflicting goals, Facebook has designed an advertising system which provides a separation layer between individual user data and advertisers. Concretely, Facebook collects from advertisers the ad creatives to display and the targeting criteria which the users being shown the ad should satisfy, and delivers the ads to people who fit those criteria [162].

Building an advertising system that serves as an intermediary layer between user data and advertisers is a common approach to user data monetization. As observed by Harper [77], "most websites and ad networks do not "sell" information about their users. In targeted online advertising, the business model is to sell space to advertisers - giving them access to people ("eyeballs") based on their demographics and interests. If an ad network sold personal and contact info, it would undercut its advertising business and its own profitability."

Through experiments, we demonstrate that an advertising system serving as an intermediary layer between users and advertisers is not sufficient to provide the guarantee of "deliver the ad [...] without revealing any personal information to the advertiser" [162, 201], as many of the details of the advertising system's design influence the privacy guarantees the system can provide, and an advertising system without privacy protections built in by design is vulnerable to determined and

sophisticated attackers. We propose and give experimental evidence of feasibility of several new types of attacks for inferring private user information by exploiting the microtargeting capabilities of Facebook's advertising system. The crux of the attacks consists of crafting advertising campaigns targeted to individuals whose privacy one aims to breach and using the ad campaign performance reports to infer new information about them. The first type of attack, **Inference from Impressions**, enables an attacker posing as an advertiser to infer a variety of private information about a user from the fact that he matched the campaign targeting criteria. The second type of attack, **Inference from Clicks**, enables inferences from the fact that a user takes action, such as a click, based on the content of the ad.

We thus make a two-fold contribution, by raising awareness of the many ways that information leakage can happen in microtargeted advertising systems and by providing an example of a real-world system in which internal data mining of users' private data entrusted to the company can lead to privacy breaches.

2.2.2 The Facebook Interface for Users and Advertisers

This section describes the functionality of Facebook from user and advertiser perspectives during Spring and Summer of 2010, the time during which this research was performed.

2.2.2.1 Facebook from the Users' Perspective

In this section we describe the types of information that users can include in their Facebook profiles, the privacy controls available to them, and their privacy concerns.

2.2.2.1.1 User Profile Information When users sign up on Facebook, they are required to provide their real first and last name, email, gender, and date of birth[6]. They are also immediately encouraged to upload a picture and fill out a more detailed set of information about themselves, such as *Basic Information*, consisting of current city, hometown, interested in (women or men), looking for (friendship, dating, a relationship, networking), political and religious views; *Relationships*, consisting of a relationship status (single, in a relationship, engaged, married, it's complicated, in an open relationship, widowed); *Education and Work* information; *Contact Information*, including address, mobile phone, IM screen name(s), and emails; as well as *Likes and Interests*. The *Likes and Interests* profile section can include things such as favorite activities, music, books, movies, TV,

[6]It is against Facebook's Statement of Rights and Responsibilities to provide false personal information http://www.facebook.com/terms.php

as well as *Pages* corresponding to brands, such as Starbucks or Coca Cola, events such as the 2010 Winter Olympics, websites such as TED.com, and diseases, such as AIDS. Any user can *Like* any Page about any topic. Since the launch of Facebook's Open Graph API [179], users are able to *Like* many entities on the web, such as webpages, blog posts, products, and news articles. Users can also post status updates, pictures, and videos, ask questions and share links through Facebook, potentially enabling Facebook to learn further details about their interests through data mining of these pieces of content.

Many Facebook users complete and actively update [71] this variety of information about themselves, thus seamlessly sharing their interests, current activities, thoughts, and pictures with their friends.

2.2.2.1.2 User Privacy Facebook provides the ability to limit who can see the information a user shares on Facebook through a privacy setting specific to each category of information. One can distinguish five significant levels of privacy settings specifying the visibility of a particular type of information: *Everyone, Friends of Friends, Friends Only, Hide from specific people*, and *Only me*. A very natural set of privacy settings, and one for which there is evidence[7] many users would strive for if they had the technical sophistication and patience to navigate Facebook's ever-changing privacy interface, is to restrict the majority of information to be visible to "Friends only", with some basic information such as name, location, a profile picture, and a school (or employer) visible to "Everyone" to enable search and distinguishability from people with the same name. In certain circumstances, one might want to hide particular pieces of one's information from a subset of one's friends (e.g., sexual orientation information from co-workers, relationship status from parents), as well as keep some of the information visible to "Only me" (e.g., date of birth, which is required by Facebook or interest in a certain Page, in order to receive that Page's updates in one's Newsfeed, without revealing one's interest in that Page to anyone).

Facebook users have shown time [112] and again [101] that they expect Facebook to not expose their private information without their control [102]. This vocal view of users, privacy advocates, and legislators on Facebook's privacy changes has recently been acknowledged by Facebook's CEO [201], resulting in a revamping of Facebook's privacy setting interface and a re-introduction of the options to restrict the visibility of all information, including that of *Likes and Interests*. Users are deeply concerned about controlling their privacy according to a Pew Internet and American Life Project

[7] As evidenced by 100,000 people using an open-source privacy scanner *Reclaim Privacy* http://www.reclaimprivacy.org

study [130], which shows that more than 65% of social network users say they have changed the privacy settings for their profile to limit what they share with others.

Facebook users have been especially concerned with the privacy of their data as it relates to the sharing of it with advertisers [164,176]. In particular, the user attitude to Facebook's microtargeted personalized ads is very mixed. A user survey by [172] shows that 54% of users don't mind the Facebook ads, while 40% dislike them, with ads linking to other websites and dating sites gathering the least favorable response. Often, users seem perplexed about the reason behind a particular ad being displayed to them, e.g., a woman seeing an ad for a Plan B contraceptive may wonder what in her Facebook profile led to Facebook matching her with such an ad and feel that the social network calls her sexual behavior into question [171]. When asked about targeted advertisements in the context of their online experience, 72% of respondents feel negatively about targeted advertising based on their web activity and other personal data[8]; 66% of Americans do not want marketers to tailor advertisements to their interests [182], and 52% of survey respondents claim they would turn off behavioral advertising [143].

2.2.2.2 Facebook from the Advertisers' Perspective

2.2.2.2.1 Ad Creation Process and Targeting Options An *ad creative* created using Facebook's self-serve advertising system consists of the destination URL, Title, Body Text, and an optional image.

The unique and valuable proposition [153] that Facebook offers to its advertisers are the **targeting criteria** they are allowed to specify for their ads. As illustrated in Figure 2.1, the advertiser can specify such targeting parameters as Location (including a city), Sex, Age or Age range (including a "Target people on their birthdays" option), Interested In (all, men, or women), Relationship status (e.g., single or married), Languages, Likes & Interests, Education (including specifying a particular college, high school, and graduation years), and Workplaces. The targeting criteria can be flexibly combined, e.g., targeting men who live within 50 miles of San Francisco, are male, 24-30 years old, single, interested in women, Like Skiing, have graduated from Harvard, and work at Apple. If one chooses multiple options for a single criteria, e.g., both "Single" and "In a Relationship" in Relationship status, then the campaign will target people who are "singe **or** in a relationship". Likewise, specifying multiple interests, e.g., "Skiing", "Snowboarding", targets people who like "skiing **or** snowboarding". Otherwise, unrelated targeting criteria such as

[8] http://online.wsj.com/community/groups/question-day-229/topics/how-do-you-feel-about-targeted

age and education are combined using a conjunction, e.g., "exactly between the ages of 24 and 30 inclusive, who graduated from Harvard". During the process of ad creation, Facebook provides a real-time "Estimated Reach" box, that estimates the number of users who fit the currently entered targeting criteria. The diversity of targeting criteria that enable audience microtargeting down to the slightest detail is an advertiser's (and, as we will see, a malicious attacker's) paradise. The advertiser can also specify the time during which to run the ad, daily budget, and max bid for Pay for Impressions (CPM) or Pay for Clicks (CPC) campaigns.

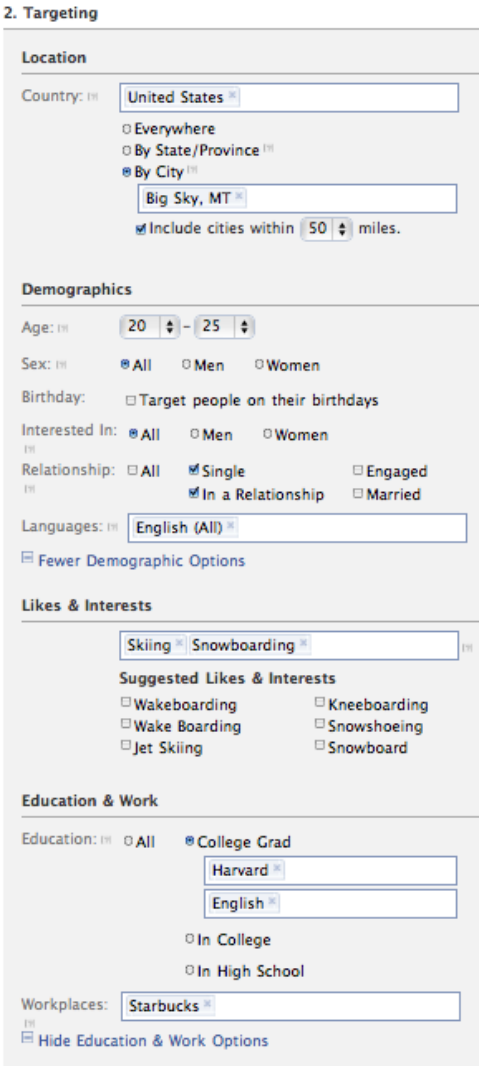


Figure 2.1: Campaign targeting interface

2.2.2.2.2 Matching Ads to People After the ad campaign is created, and every time it is modified, the ad is submitted for approval that aims to verify its adherence to Facebook's advertising guidelines.[9] Based on our experiments it appears that the approval is occasionally performed manually and other times - automatically, and focuses on checking adherence to guidelines of the ad image and text.

For each user browsing Facebook, the advertising system determines all the ads whose targeting criteria the user matches, and chooses the ads to show based on their bids and relevance.

Facebook provides detailed ad campaign performance reports specifying the total number of impressions and clicks the ad has received, the number of unique impressions and clicks, broken up by day, as well as rudimentary responder demographics. The performance report data is reported close to real time.

2.2.2.2.3 Effectiveness of Targeted Ads From the advertisers' perspective, the ability to microtarget users using a diverse set of powerful targeting criteria offers a tremendous new opportunity for audience reach.

[9] http://www.facebook.com/ad_guidelines.php

Specifically on Facebook, in 2010 the biggest advertisers have increased their spending more than 10-fold [192] and the "precise enough" audience targeting is what encourages leading brand marketers to spend their advertising budget on Facebook [153]. Furthermore, Facebook itself recommends targeting ads to "smaller, more specific" groups of users,[10] as such ads are "more likely to perform better".

In a broader context, there is evidence that narrowly targeted ads are much more effective than ordinary ones [143,195] and that very targeted *audience buying* ads, e.g., directed at "women between 18 and 35 who like basketball"[11] are valuable in a search engine ad setting as well.

2.2.3 Proposed Attacks Breaching Privacy

We illustrate that the promise by several Facebook executives [162, 164, 165, 201] that Facebook "[doesn't] share your personal information with services you don't want", and in particular, "[doesn't] give advertisers access to your personal information" [201], "don't provide the advertiser any [...] personal information about the Facebook users who view or even click on the ads" [165] is something that the advertising system has strived to achieve but has not yet fully accomplished. We show that despite Facebook's advertising system serving as an intermediary layer between user data and advertisers, the design of the system, the matching algorithm, and the user data used to determine the fit to the campaign's targeting criteria, combined with the detailed campaign performance reports, has contributed to a system that could have leaked private user information.

We experimentally investigate the workings of Facebook's advertising system and establish that (during the summer of 2010 when this research was done):

- Facebook used private and "Friends Only" user information to determine whether the user matches an advertising campaign's targeting criteria
- The default privacy settings led to many users having a publicly visible uniquely identifying set of features
- The variety of permissible targeting criteria allowed microtargeting an ad to an arbitrary person
- The ad campaign performance reports contained a detailed breakdown of information, including number of unique clicks, respondents' demographic characteristics, and breakdown by time,

[10] http://www.facebook.com/help/?faq=14719
[11] http://blogs.wsj.com/digits/2010/07/15/live-blogging-google-on-its-earnings

which we show leads to an attacker posing as an advertiser being able to design and successfully run advertising campaigns that enable them to:

A. Infer information that people post on Facebook in "Only me", "Friends Only", and "Hide from these people" visibility mode

B. Infer private information not posted on Facebook through ad content and user response

C. Display intrusive and "creepy" ads to individuals

We now describe in detail two novel attacks that exploit the details of the advertising system's design in order to infer private information and our experiments implementing them.[12]

2.2.3.1 Infer Information Posted on Facebook with "Only me", "Friends Only", and "Hide from these people" Privacy Settings through Ad Campaign Match

Attack 1: **Inference from Impressions** is aimed at inferring information that a user has entered on Facebook but has restricted to be visible to "Only me" or "Friends Only." According to the privacy settings chosen by the user, this information should not be available for observation to anyone except the user themself, or to anyone except the user's friends, respectively. The proposed attack will bypass this restriction by running several advertising campaigns targeted at the user and differing only in the targeting criteria corresponding to the unknown private information the attacker is trying to infer. The difference in campaign performance reports of these campaigns will enable the attacker to infer desired private information.

For ease of notation, we represent each advertising campaign as a mixture of conjunctions and disjunctions of boolean predicates, where campaign $A = a_1 \wedge (a_2 \vee a_3)$ targets people who satisfy criteria a_1 (e.g., "went to Harvard") and criteria a_2 (e.g., "Like skiing") or a_3 (e.g., "Like snowboarding").

The necessary and sufficient conditions for the attack's success are: the ability to choose targeting criteria A that identify the user U uniquely[13]; Facebook's user-ad matching algorithm showing the ad only to users who match the ad targeting criteria exactly and using the information of whether U satisfies f_i when determining campaign match; the user U using Facebook sufficiently often so that the ads have a chance to be displayed to U at least once over the observation time period, if U matches the targeting criteria; the advertising system treating campaigns $A_1, \ldots, A_k$ equally.

[12]For ethical reasons, all experiments conducted were either: 1) performed with consent of the people we were attacking or aimed at fake accounts; 2) aimed at Facebook employees involved with the advertising system; 3) aimed at inferring information that we do not plan to store, disclose, or use.

[13]We discuss the feasibility of this in Section 2.2.4.1.

Attack 1 Inference from Impressions

1: **Input:** A user U and a feature F whose value from the possible set of values $\{f_1, \ldots, f_k\}$ we'd like to determine, if it is entered by U on Facebook.
2: Observe the profile information of U visible to the advertiser that can be used for targeting.
3: Construct an ad campaign with targeting criteria A combining background knowledge about U and information visible in U's profile, so that one reasonably believes that only U matches the campaign criteria of A.
4: Run k ad campaigns, $A_1, \ldots, A_k$, such that $A_i = A \wedge f_i$. Use identical and innocuous content in the title and text of all the ads. Specify a very high CPM (or CPC) bid, so as to be reasonably sure the ads would win an auction among other ads for which U is a match.
5: Observe the impressions received by the campaigns over a reasonable time period. If only one of the campaigns, say A_j, receives impressions, from a unique user, conclude that U satisfies f_j. Otherwise, refine campaign targeting criteria, bid, or ad content.

We run several experiments following the blueprint of Attack 1, and experimentally establish that the advertising system satisfies the above conditions. In particular, we establish that Facebook uses "Friends Only" and "Only me" visible user data when determining whether a user matches an advertising campaign, thereby enabling a malicious attacker posing as an advertiser to infer information that was meant by the user to be kept private or "Friends only", violating user privacy expectations and the company's privacy promises [162, 164, 165, 201].

We also remark that a typical user would find Attack 1: **Inference from Impressions** very surprising, as the advertiser is able to gain knowledge about things the user might have listed in their profile even if the user U does not pay attention to or click on the ad.

2.2.3.1.1 Inferring a Friend's Age The first experiment shows that using Facebook's advertising system it is possible to infer the age of a particular person, who has set the information to only be visible by themselves.

We attack a friend of the author, who has entered her birthday on Facebook (because Facebook requires every user to do so) but has specified that she wants it to be private by selecting the "Don't show my birthday in my profile" option in the Information section of her profile and by selecting "Make this visible to Only Me" in the Birthday Privacy Settings. Accordingly, she expects that no one should be able to learn her age, however, our experiments demonstrate that it is not the case.

We know the college she went to and where she works, which happens to be a place small enough that she is the only one at her workplace from that college. Following the blueprint of **Inference from Impressions** we created several identical ad campaigns targeting a female at the friend's place of work who went to the friend's college, with the ads differing only in the age of the person being targeted – 33, 34, 35, 36, or 37. The ads whose age target does not match the friend's age will

not be displayed, and the ad that matches her age will be, as long as the ad creative is reasonably relevant and the friend uses Facebook during the ad campaign period.

From observing the daily stats of the ad campaigns' performance, particularly, the number of impressions each of the ads has received, we correctly inferred the friend's age: 35, as only the ad targeted to a 35-year-old received impressions. The cost of finding out the private information was a few cents. The background knowledge we utilized related to the friend's education and workplace, is also available in her profile and visible to "Friends Only". Based on prior knowledge, we pruned our exploration to the $33 - 37$ age range, but could have similarly succeeded by running more campaigns, or by first narrowing down the age range by running campaigns aimed at "under 30" and "over 30", then "under 40" and "over 40", then "under 34" and "over 34", etc.

2.2.3.1.2 Inferring a Non-friend's Sexual Orientation Similarly, following the same blueprint, we succeeded in correctly inferring sexual orientation of a non-friend who has posted that she is "Interested in women" in a "Friends Only" visibility mode. We achieved Step 3 of the blueprint by targeting the campaign to her gender, age, location, and a fairly obscure interest publicly visible to everyone, and used "Interested in women" and "Interested in men" as the varying values of F.

2.2.3.1.3 Inferring Information other than Age and Sexual Orientation The private information one can infer using techniques that exploit the microtargeting capabilities of Facebook's advertising system, its ad-user matching algorithm, and the ad campaign performance reports, is not limited to user age or sexual orientation. An attacker posing as an advertiser can also infer a user's relationship status, political and religious affiliation, presence or absence of a particular interest, as well as exact birthday using the "Target people on their birthdays" targeting criterion.

Although using private user information obtained through ad campaigns is against Facebook's Terms of Service, a determined malicious attacker would not hesitate to disregard it.

2.2.3.2 Infer Private Information not Posted on Facebook through Microtargeted Ad Creative and User Response to it

The root cause of privacy breaches possible using Attack 1: **Inference from Impressions** is Facebook's use of private data to determine whether the users match targeting criteria specified by the ad campaign. We now present a different kind of attack, Attack 2: **Inference from Clicks**, that takes advantage of the microtargeting capabilities of the system and the ability to observe a user's response to the ad in order to breach privacy. The goal of this attack is to infer information

about users that may not have been posted on Facebook, such as a particular user's interest in a certain topic. The attack proceeds by creating an ad enticing a user U interested in topic T to click on it, microtargeting the ad to U, and using U's reaction to the ad (e.g., a click on it) as an indicator of U's interest in the topic.

Suppose an attacker wants to find out whether a colleague is having marital problems, a celebrity is struggling with drug abuse, or whether an employment candidate enjoys gambling or is trying to get pregnant. Attack 2: **Inference from Clicks** targets the campaign at the individual of interest, designs the ad creative that would engage an individual interested in the issue (e.g., "Having marital difficulties? Our office offers confidential counseling."), and observes whether the individual clicks on the ad to infer the individual's interest in the issue.

Attack 2 Inference from Clicks

1: **Input:** A user U and a topic of interest T.
2: Observe the profile information of U visible to the advertiser that can be used for targeting.
3: Construct targeting criteria A combining background knowledge about U and information visible in U's profile, so that one reasonably believes that only U matches the criteria of A.
4: Run an ad campaign with targeting criteria A and ad content, picture, and text inquiring about T, linking to a landing page controlled by an attacker.
5: Observe whether the ad receives impressions to ensure that it is being shown to U. Make conclusions about U's interest in topic T based on whether the ad receives clicks.

Any user who clicks on an ad devised according to the blueprint of **Inference from Clicks** reveals that the ad's topic is likely of interest to him. However, the user does not suspect that by clicking the ad, he possibly reveals sensitive information about himself in a way tied to his identity, as he is completely unaware what targeting criteria led to this ad being displayed to him, and whether every single user on Facebook or only one or two people are seeing the ad.

For ethical reasons, the experiments we successfully ran to confirm the feasibility of such attacks contained ads of more innocuous content: inquiring whether a particular individual is hiring for his team and asking whether a person would like to attend a certain thematic event.

2.2.3.2.1 Display Intrusive and "Creepy" Ads to Individuals One can also take advantage of microtargeting capabilities in order to display funny, intrusive, or creepy ads. For example, an ad targeting a particular user U could use the user's name in its content, along with phrases ranging from funny, e.g., "Our son is the cutest baby in the world" to disturbing, e.g., "You looked awful at Prom yesterday". For these types of attacks to have the desired effect, one does not need to guarantee the success of Step 3 of Attack 2 – an intrusive ad may be displayed to a wider audience,

but if it uses a particular user's name, it will likely only have the desired effect on that user, since others will likely deem it irrelevant after a brief glance.

2.2.3.3 Other Possible Inferences

The information one can infer by using Facebook's advertising system is not limited to the private profile information and information inferred from the contents of the ads the users click.

Using the microtargeting capability, one can estimate the frequency of a particular person's Facebook usage, determine whether they have logged in to the site on a particular day, or infer the times of day during which a user tends to browse Facebook. To get a sense of how private this information may be or become in the future, consider that according to American Academy of Matrimonial Lawyers, 81% of its members have used or faced evidence from Facebook or other social networks in the last five years [8], with 66% citing Facebook as the primary source, including a case when a father sought custody of kids based on evidence that the mother was on Facebook at the time when she was supposed to attend events with her kids [91].

More broadly, going beyond individual user privacy, one can imagine running ad campaigns in order to infer organization-wide trends, such as the age or gender distribution of employees of particular companies, the amount of time they spend on Facebook, the fraction of them who are interested in job opportunities elsewhere, etc. For example, a data-mining startup Rapleaf has recently used [190] their database of personal data meticulously collected over several years, to compare shopping habits and family status of Microsoft and Google employees. Exploitation of powerful targeting capabilities and detailed campaign performance reports of Facebook's advertising system could potentially facilitate a low-cost efficient alternative to traditional investigative analysis. Insights into interests and behavioral patterns of certain groups could be valuable from the social science perspective, but could also have possibly undesired implications, if exploited, for example, by insurance companies negotiating contracts with small companies, stock brokers trying to gauge future company performance, and others trying to exploit additional information obtained through ad campaigns to their advantage.

2.2.4 Discussion of Attacks and their Replicability

In this section, we discuss the feasibility of selecting campaign features for targeting particular individuals, the additional privacy risks posed by "Connections targeting" capabilities of the Facebook advertising system, the ways in which an attacker can increase confidence in conclusions obtained

through the attacks exploiting the advertising system, and the feasibility of creating fake user accounts.

2.2.4.1 Targeting Individuals

The first natural question that arises with regards to the attack blueprints and experiments presented is whether creating an advertising campaign with targeting criteria that are satisfied only by a particular user is practically feasible. There is strong experimental and theoretical evidence that it is indeed the case.

As pointed out by [174], 87% of all Americans (or 63% in follow-up work by [67]) can be uniquely identified using zip code, birth date, and gender. Moreover, it is easy to establish [52, 145] that 33 bits of entropy are sufficient in order to identify someone uniquely from the entire world's population. Recent work [51] successfully applies this observation to uniquely identify browsers based on characteristics such as user agent and timezone information that browsers make available to websites. Although we did not perform a rigorous study, we conjecture that given the breadth of permissible Facebook ad targeting criteria, it is likely feasible to collect sufficient background knowledge on anyone to identify them uniquely.

The task of selecting targeting criteria matching a person uniquely is in practice further simplified by the default Facebook privacy settings that make profile information such as gender, hometown, interests, and Pages liked visible to everyone. An obscure interest shared by few other people, combined with one's location is likely to yield a unique identification, and although the step of selecting these targeting criteria requires some thinking and experimentation, common sense combined with easily available information on the popularity of each interest or Page on Facebook enables the creation of a desired campaign. For users who have changed their default privacy settings to be more restrictive, one can narrow the targeting criteria by investigating their education and work information through other sources. An attacker, such as a stalker, malicious employer, insurance company, journalist, or lawyer, is likely to have the resources to obtain the additional background knowledge on their person of interest or may have this information provided to them by the person himself through a resume or application. Friends of a user are particularly powerful in their ability to infer private information about the user, as all information the user posts in "Friends Only" privacy mode facilitates their ability to refine targeting and create campaigns aimed at inferring information kept in the "Only me" mode or inferring private information not posted using **Inference from Clicks**.

2.2.4.2 Danger of Friends of Friends, Page and Event Admins

Additional power to successfully design targeting criteria matching particular individuals comes from the following two design choices of Facebook's privacy settings and ad campaign creation interface:

- All profile information except email addresses, IM, phone numbers and exact physical address is by default available to "Friends of Friends".
- The campaign design interface offers options of targeting according to one's *Connections on Facebook*, e.g., targeting users who are/aren't connected to the advertiser's Page, Event, Group, or Application, or targeting users whose friends are connected to a Page, Event, Group, or Application.

While these design choices are aimed at enabling users to share at various levels of granularity and enabling advertisers to take full advantage of social connections and the popularity of their Page(s) and Event(s), they also facilitate the opportunity for a breach of privacy through advertising. For example, an attacker may entice a user to Like a Page or RSVP to an event they organize through prizes and discounts. What a user most likely does not realize is that by Liking a Page or RSVPing to an event he makes himself more vulnerable to the attacks of Section 2.2.3. Furthermore, since the Connections targeting also allows to target friends of users who are connected to a Page, if one's friend Likes a Page, it also makes one vulnerable to attacks from the owner of that Page, leading to a potential privacy breach of one's data without any action on one's part.

2.2.4.3 Mitigating Uncertainty

A critic can argue that there is an inherent uncertainty both on the side of Facebok's system design (in the way that Facebook matches ads to people, chooses which ads to display based on bids, and does campaign performance reporting) and on the side of user usage of Facebook (e.g., which information and how people choose to enter it in their Facebook profile, how often they log in, etc.) that would hinder an attacker's ability to breach user privacy. We offer the following counter-arguments:

Uncertainty in Matching Algorithm. The attacker has the ability to create multiple advertising campaigns as well as to create fake user profiles (see Section 2.2.4.4) matching the targeting criteria of those campaigns in order to reverse-engineer the core aspects of how ads are being matched to users, in what positions they are being displayed, how campaign performance reporting is done, which of the targeting criteria are the most reliable, etc. For example, in the course of our experiments, we identified that targeting by city location did not work as expected, and were able to tweak

the campaigns to rely on state location information. For our experiments and in order to learn the system, we created and ran more than 30 advertising campaigns at the total cost of less than $10, without arousing suspicion.

Uncertainty in User Information. Half of Facebook's users log in to Facebook every day [14], thus enabling a fairly quick feedback loop: if, with a high enough bid, the attacker's campaign is not receiving impressions, this suggests that the targeting criteria require further exploration and tweaking. Hence, although a user might have misspelled or omitted entering information that is known to the attacker through other channels, some amount of experimentation, supplemented with the almost real-time campaign performance reporting, including the number of total and unique impressions and clicks received, is likely to yield a desired campaign.

Uncertainty in Conclusion. Although attacks may not yield conclusions with absolute certainty, they may provide reasonable evidence. A plausible sounding headline saying that a particular person is having marital problems or is addicted to pain killers can cause both embarrassment and harm. The detailed campaign performance reports, including the number of unique clicks and impressions, the ability to run the campaigns over long periods of time, the almost real-time reporting tools, the incredibly low cost of running campaigns, and the lax ad review process, enables a determined attacker to boost his confidence in any of the conclusions.

2.2.4.4 Fake Accounts

As the ability to create fake user accounts on Facebook may be crucial for learning the workings of the advertising system and for more sophisticated attacks, we comment on the ease with which one can create these accounts.

The creation of fake user accounts (although against the Terms of Service) that look real on Facebook is not a difficult task, based on our experiments, anecdotal evidence [13], [14] and others' research [159]. The task can be outsourced to Mechanical Turk, as creation of an account merely requires picking a name, email, and filling out a CAPTCHA. By adding a profile picture, some interests, and some friends to the fake account, it becomes hard to distinguish from a real account. What makes the situation even more favorable for an advertising focused attacker, is that typically fake accounts are created with a purpose of sending spam containing links to other users, an observation Facebook relies upon to mark an account as suspicious [60]; whereas the fake accounts created for the purpose of facilitating attacks of Section 2.2.3 would not exhibit such behavior, and would thus,

[14] http://rickb.wordpress.com/2010/07/22/why-i-dont-believe-facebooks-500m-users/

presumably, be much harder to distinguish from a regular user.

2.2.5 Related Work

The work most closely related to ours is that of Wills and Krishnamurthy [114] and Edelman [53] who have shown that clicking on a Facebook ad, in some cases, revealed to the advertiser the user ID of the person clicking, due to Facebook's failure to properly anonymize the HTTP Referer header. Their work has resulted in much publicity and Facebook has since fixed this vulnerability [96].

The work of [73] observes that ads whose ad creative is neutral to sexual preference may be targeted exclusively to gay men, which could create a situation where a user clicking on the ad would reveal to the advertiser his sexual preference.

Several pranks have used Facebook's self-serve advertising system to show an innocuous or funny ad to one's girlfriend[15] or wife[16]. However, they do not perform a systematic study or suggest that the advertising system can be exploited in order to infer private information.

2.2.6 Why Simple Tweaks won't Fix It

As we have demonstrated, despite Facebook's intentions to protect privacy and the use of user private information only for internal data-mining rather than external data-sharing, the information that has been explicitly marked by users as private or information that they have not posted on the site but is inferable from the content of the ads they click, leaks in a way tied to their identity through the current design of Facebook's advertising system. We describe Facebook's response to our research disclosure, and other seemingly promising ad-hoc solutions towards protecting privacy in microtargeted ad systems, as well as outline the reasons why they would not guarantee privacy, next.

2.2.6.1 Facebook's Response and Other Candidate Solutions

Following the disclosure of our findings to Facebook on July 13, 2010, Facebook promptly implemented changes to their advertising system that make the kinds of attacks we describe much harder to execute.

Their approach was to introduce an additional check in the advertising system, which at the campaign creation stage looks at the "Estimated Reach" of the ad created, and suggests to the

[15] http://www.clickz.com/3640069
[16] http://www.gabrielweinberg.com/blog/2010/05/a-fb-ad-targeted-at-one-person-my-wife.html

advertiser to target a broader audience if the "Estimated Reach" does not exceed a soft threshold of about 20 people. We applaud Facebook's prompt response and efforts in preventing the execution of attacks proposed in this work, but believe that their fix does not fully eliminate the possibility of proposed attacks.

Although we did not perform further experiments, it is conceivable that the additional restriction of sufficiently high "Estimated Reach" can be bypassed in principle for both types of attacks proposed. To bypass the restriction while implementing Attack 1: **Inference from Impressions**, it suffices for the attacker to create more than 20 fake accounts (Section 2.2.4.4) that match the user being targeted in the known attributes. A sufficient number of accounts matching the targeting criteria in the system would permit running the ad, and attacker's control over the fake accounts would enable differentiating between the impressions and clicks of targeted individual and of the fake accounts. To bypass the restriction while implementing Attack 2: **Inference from Clicks**, one can either take a similar approach of creating more than 20 fake accounts, or target the ad to a slightly broader audience than the individual, but further personalize the ad to make it particularly appealing to the individual of interest (e.g., by including the individual's name or location in the ad's text).

Hence, although the minimum campaign reach restriction introduced additional complexity into reliably executing attacks, the restriction does not seem to make the attacks infeasible for determined and resourceful adversaries.

A better solution to protect users from private data inferences using attacks of type 1: **Inference from Impressions** would be to use only profile information designated as visible to "Everyone" by the user when determining whether a user matches a campaign's targeting criteria. If private and "Friends Only" information is not used when making the campaign match decisions, then the fact that a user matches a campaign provides no additional knowledge about this user to an attacker beyond what they could infer by simply looking up their public profile on Facebook.

Although using only fully public information in the advertising system would come closest to delivering on the privacy promises made by Facebook to its users [162, 164, 165, 201], it would also introduce a business challenge for Facebook. As much of the information users share is "Friends Only", using only information shared with "Everyone" would likely degrade the quality of the audience microtargeting that Facebook is able to offer advertisers, and hence create a business incentive to encourage users to share their information more widely in order to monetize better (something that Facebook has been accused of but vehemently denies [164]). Another approach

would be to introduce an additional set of privacy controls to indicate which information the users are comfortable sharing with advertisers; however, this would create significant additional cognitive burden on users navigating an already very complex set of privacy controls [59].

We do not know of a solution that would be fully foolproof against **Inference from Clicks** attacks. The *Power Eye* concept [118, 185], providing consumers with a view of the data used to target the ad upon a mouseover, offers some hope in providing the user with the understanding of the information they might be revealing when clicking on a particular ad. However, the hassle and understanding of privacy issues required to evaluate the breadth of the targeting and the risk that it poses is likely beyond the ability of a typical consumer.

2.2.6.2 Microtargeted Advertising while Preserving Privacy in Other Contexts

The challenges of designing microtargeted advertising systems offering the benefits of fine-grained audience targeting while aiming to preserve user privacy will become applicable to a variety of other companies entrusted with user data and administering their own advertising systems (e.g., Google) as they move to enable better targeting [184]. We have demonstrated that merely using an intermediary layer that handles the matching between users and ads is not sufficient for being able to provide the privacy guarantees users and companies aspire for, and that a variety of seemingly minor design decisions play a crucial role in the ease of breaching user privacy using the proposed novel class of attacks.

The works of [180] and [74] propose systems that perform profiling, ad selection, and targeting on the client's (user's) side and use cryptographic techniques to ensure accurate accounting. These proposals require a shift in the paradigm of online advertising, where the ad brokers relinquish the control of the way profiling and matching is performed and rely on a weaker client-side model of the user, which seems unlikely in the near-term.

2.3 Summary

In Section 2.1 we have described two cases of sharing of seemingly anonymized user data that resulted in privacy breaches. In Section 2.2 we have described a case of seemingly privacy-preserving data mining that resulted in privacy breaches. For each of these data types and sharing and mining goals, we have discussed additional candidate techniques for protecting privacy and highlighted the reasons why each of those techniques would also be unsatisfactory.

We have not considered cases where data was being mindlessly shared; in all of these examples, there was a conscious and good faith effort made to protect the privacy of users whose data was involved. A variety of other efforts for sharing other types of user data illustrate the same point – an utter failure of ad-hoc anonymization techniques to adequately protect privacy [16, 146, 200].

The Internet user data that the companies desire to mine and share has certain very distinct characteristics that are novel to privacy research and have been the source of stumbling in all the privacy protection attempts described in this chapter. The space of possible features mined or shared (such as search queries posed, movies rated, social network profile features specified) is very high-dimensional but each vector representing a particular user is very sparse (as each user poses only a tiny fraction of possible queries, rates a tiny fraction of all movies, and has a limited number of profile features). Moreover, auxiliary information on a particular user is available in abundance, as pieces of information are spread out and shared by users with multiple websites with varying privacy settings, and unlike in previous research in which the features could have been cleanly divided into sensitive and insensitive [187], it is impossible to predict which parts of the data are sensitive and/or available elsewhere to the adversary. The characteristics of the data and the users combined with abundant auxiliary information creates new challenges for protecting privacy, as a few queries, ratings, or profile features may be sufficient to uniquely fingerprint a user [51, 52, 145, 147]. Moreover, the individual queries, ratings, or profile features themselves may be extremely sensitive and a publication of just one of them may violate privacy.

Furthermore, each privacy breach due to improper privacy protection while mining or sharing user data, even if the number of users affected is small, deals a multi-faceted blow: to the reputation of the company that shared this data, to the individual(s) whose privacy was violated, to the willingness of users and other companies to consider sharing their data, as well as to the perpetual availability of the published private data for use in further attacks. Lack of ability to capitalize on the increased availability of online data by sharing and mining it across companies, institutions, and communities due to privacy concerns would be a loss to society and academic scholarship [30], significantly impeding innovation, and the ability of all members of society to gain new insights into humanity's behavior.

We have built a case, supported also by decades of research in cryptography [44], that ad-hoc approaches to protecting privacy inevitably fail in the world of creative and sophisticated adversaries in possession of auxiliary information from multiple sources. However, we and many members of the scientific community [70, 76, 123] believe that the tremendous value of user data for innovation and

scientific discovery warrants a thorough search for approaches to protecting privacy while mining and sharing user data. The way to avoid the failures of ad-hoc techniques to protect privacy is to start with a rigorous, quantifiable privacy definition and design algorithms for mining and sharing data that provably satisfy the definition. We introduce a well-motivated privacy definition and proceed to design algorithms and analyze privacy/utility trade-offs using that definition next.

Chapter 3

A Rigorous Privacy Definition

In this chapter we describe a rigorous definition of privacy, *differential privacy*, introduced by [49] in 2006 that we will rely on in the rest of the thesis, briefly highlight reasons why this definition is particularly well-suited to our problem domain, and present two known approaches for achieving differential privacy. Several recent surveys by Dwork [43, 45, 46] provide a further in-depth overview of the motivation, techniques, and frontiers of differential privacy.

3.1 Differential Privacy

3.1.1 Prior to Differential Privacy

In the last two decades of work focused on releasing statistics about groups of individuals while protecting their privacy, many definitions of what it means to protect privacy have been proposed and utilized. One in particular, *k-anonymity*, has been popular in relation to user data [3, 119] until differential privacy was introduced. In the k-anonymity model, each individual is represented as a tuple of attributes, and those attributes that can be linked with external information in order to uniquely identify an individual are termed quasi-identifiers. The privacy protection of k-anonymity [161, 175] is to guarantee that every set of quasi-identifiers appears in at least k records in the data set, or equivalently, that any particular individual's data is indistinguishable from data of at least $k-1$ other individuals with respect to the quasi-identifiers. k-anonymity is usually achieved through suppression or coarsening of the values of the attributes, for example by omitting names of the individuals and replacing five-digit zipcodes with only their first two digits.

The definition is very restrictive and hard to achieve without significant loss in utility especially in the context of online and social data. To understand why, imagine representing a user's search history as a set of attributes, then trying to specify quasi-identifiers among those attributes, and making each individual's search history indistinguishable from $k-1$ others. Even if we were able to reliably identify the quasi-identifiers among the search queries, the number of which is almost certainly very large, subsequent anonymization of the very sparse user search history representations would likely render them useless from the utility perspective, an argument that is formalized in the work of [2]. Furthermore, in the real-world settings where an adversary, unknown to the data curator, may have access to background information, k-anonymity provides particularly poor privacy guarantees, as the adversary may be able to distinguish among the k individuals with identical quasi-identifiers using the background knowledge. A more detailed discussion of k-anonymity's shortcomings can be found in [128].

3.1.2 Informal Definition

In contrast with k-anonymity, which tries to make one individual's data "blend in" with $k-1$ others, differential privacy, introduced by Dwork, McSherry, Nissim, and Smith [43,49], aims to limit the privacy risk to an individual that arises as a result of their data being used by the company, as compared to the privacy risk that the same individual would have incurred had he not used the service offered by the company. Phrased differently, an analysis or a release of a dataset is differentially private if an attacker can infer approximately the same amount of information about an individual B from the analysis or release of the dataset, whether or not B's data was included in the input on which the analysis or release was based. The meaning of "approximately the same" is quantified using the privacy parameter ϵ.

3.1.3 Formal Definition

More formally,

Definition 1 (ϵ**-differential privacy**). *A randomized algorithm $\mathcal{A}$ is* ϵ-differentially private *if for all datasets D_1 and D_2 differing in at most one individual's data and all $\hat{D} \subseteq Range(\mathcal{A})$:*

$$\Pr[\mathcal{A}(D_1) \in \hat{D}] \leq e^{\epsilon} \cdot \Pr[\mathcal{A}(D_2) \in \hat{D}],$$

where the probabilities are over the coin flips of the algorithm $\mathcal{A}$.

By datasets D_1 and D_2 differing in at most one individual's data we mean that one is a subset of the other, and the larger dataset contains all data from the smaller dataset and data of one more individual. The definition can also be analogously stated for all pairs of datasets D_1 and D_2 that differ in the value of the data of at most one individual or differ in at most one piece of content. We use the original meaning of the definition, and state additional assumptions on D_1 and D_2 separately in each chapter depending on the exact problem being considered.

Subsequent work [47] relaxed the ϵ-differential privacy definition to include a non-zero additive component δ, which allows to ignore events of very low probability.

Definition 2 (**(ϵ, δ)-differential privacy**). *A randomized algorithm $\mathcal{A}$ is* (ϵ, δ)-differentially private *if for all datasets D_1 and D_2 differing in at most one individual's data and all $\hat{D} \subseteq Range(\mathcal{A})$:*

$$\Pr[\mathcal{A}(D_1) \in \hat{D}] \leq e^{\epsilon} \cdot \Pr[\mathcal{A}(D_2) \in \hat{D}] + \delta,$$

where the probabilities are over the coin flips of the algorithm $\mathcal{A}$.

The parameters ϵ and δ are publicly known, and the higher ϵ and δ, the weaker the privacy guarantee provided by the algorithm. There is no agreement on exact values of ϵ and δ that are meaningful in practical contexts as their selection is considered a social question [46], but ϵ can be thought of as a small constant (e.g., $\ln 2, \ln 5$), and δ - as negligible in the number of users whose data is included in the dataset.

3.1.4 The Advantages of the Definition

The privacy guarantee of differential privacy is not an absolute one, and is weaker than the privacy guarantee one may initially hope to achieve, namely, one in which the beliefs of an adversary about an individual prior to seeing the output of algorithm $\mathcal{A}$ are approximately the same as the adversary's beliefs after seeing the output of $\mathcal{A}$. However, such a stringent privacy guarantee cannot be achieved if algorithm $\mathcal{A}$ provides any meaningful aggregate information about its input [43, 50]. For example, suppose an adversary believes that individuals never search for the word "apple", and a search engine states that 20% of its users have searched for "apple" at some point. As a result of such a statement, an adversary's belief about whether or not a particular person B has ever searched for "apple" is radically changed, even though B might have never used the search engine. The differential privacy guarantee, which aims to keep the adversary's beliefs approximately the same whether or not B's data was included in the input to $\mathcal{A}$, is a more sensible and practical guarantee to aim for.

Furthermore, as a definition of what it means to protect privacy, differential privacy is particularly well-matched to the context we aim to address in the thesis (in which companies obtain large amounts of diverse user data as the result of them using their services, and would like to utilize the collected data in order to further improve their services or advance scientific research, while at the same time protecting the privacy of the individual users), for the following reasons:

- The definition measures the privacy guarantee of a particular algorithm, $\mathcal{A}$, which enables reasoning under the assumption that the company is interested in protecting privacy and is evaluating candidate data-mining and data-sharing algorithms from that perspective.

- It measures privacy in terms of the effect that the presence or absence of one individual's data has on the output of the algorithm (which is what becomes available to the adversary). Thus, it measures and "announces" the privacy risk an individual would incur by using the company's service (e.g., using the search engine, participating in the social network), and enables every individual to make a choice of not using the service if he deems that the utility he derives from the service is not worth the privacy risk he is going to incur as a result.

- Unlike black-and-white statement about whether the company protects or does not protect privacy, differential privacy enables a more fine-grained paradigm in which privacy loss can be quantified on a continuous spectrum. This enables both companies and users to trade-off and balance competing objectives of providing better and innovative services while protecting privacy.

- The definition can be applied to any user data, not just data of a particular type, which is especially important given that every day new types of data about individuals become available for collection and analysis. Consider that pre-Internet, collecting several dozen characteristics about an individual was a challenging and time-consuming task, whereas now the possibilities for seamless user data collection are virtually limitless, starting from user personal information provided to the service, to timestamped logs of all actions on the service, to the location, device, and browser used to access the service, to speed with which the user moves between different actions or pieces of content on the service, to combinations and cross-referencing of actions performed on partner services.

- The definition does not make any assumptions about the adversary and the computational, social, and auxiliary data resources available to him. In the world in which computational power doubles every year, and the resources for obtaining additional data and for executing data

linkage attacks are multiplying, making no assumptions about the adversary is an important characteristic for protecting privacy not only from the most advanced adversaries of today, but also from the most sophisticated and resourceful adversaries of tomorrow. Furthermore, the definition protects privacy of user B even if the adversary has coerced all other users of the service except B.

- The definition allows arbitrary post-processing and use of the output of the algorithm without additional privacy risks, a feature that is highly desirable in the real-world, since the data curator may not be able to exhort control over how the results of his analysis or data release are used once they are published, and thus would like to have the privacy guarantees extend to all possible post-publishing analyses.

- The definition adapts well to privacy of multiple individuals and to protecting privacy of multiple simultaneous uses of the same or similar user data by independent entities.

3.2 Known Techniques for Achieving Differential Privacy

3.2.1 Laplace and Exponential Mechanisms

Two specific algorithms, the Laplace mechanism and the Exponential mechanism, are the known building blocks for publishing data or answering queries in a differentially private manner.

3.2.1.1 Laplace Mechanism

The idea of the Laplace mechanism proposed by Dwork et al. [49] is to add properly calibrated random noise to the true answer of the query function being computed on the data. The noise magnitude is calibrated with respect to the query function's *sensitivity*, the maximum possible change in the function value due to the addition or removal of one person's data from the input.

The idea of Laplace mechanism is most naturally applied to histogram queries. Consider an arbitrary domain $\mathcal{D}$ which has been partitioned into r disjoint bins. A histogram query function, $f : \mathcal{D} \to \mathcal{Z}^r$ maps the dataset entries into these bins and reports the number of entries in each bin. The sensitivity $S(f)$ of a function f denotes the maximum possible "change" in the value of f when the inputs differ in one individual's data, i.e.,
$S(f) = \max\{\|f(D_1) - f(D_2)\|_1 : D_1, D_2 \in \mathcal{D}$ differ in at most one individual's data$\}$. Then

Theorem 1. *[49] For all $f : \mathcal{D} \rightarrow \mathcal{R}^r$ the mechanism that given input D publishes $San_f(D) = f(D) + (Y_1, \ldots, Y_r)$, where the Y_i are drawn i.i.d. from $Lap\big(S(f)/\epsilon\big)$, satisfies ϵ-differential privacy.*

$Lap(b)$ denotes the Laplace distribution with scale parameter b, location parameter 0, and variance $2b^2$. In this notation, increasing b flattens out the $Lap(b)$ curve, yielding larger expected noise magnitude and therefore, eventually, better privacy guarantees. Recall the following two properties of the Laplace distribution:

Observation 1. ***[Properties of Laplace distribution]*** *For Laplace distribution with location parameter 0, and scale parameter $b > 0$, denoted by $Lap(b)$, and a random variable X, the cdf $F(x) = Pr[X \leq x]$ satisfies:*

$$\begin{aligned} F(x) &= 1/2 \cdot \exp(x/b), \ if \ x < 0 \\ &= 1 - 1/2 \cdot \exp(-x/b), \ if \ x \geq 0 \end{aligned}$$

Observation 2. ***[Properties of Laplace ratios]*** *Let r be a random $Lap(b)$ noise. Then, $1 \leq \frac{Pr[r<c+1]}{Pr[r<c]} \leq e^{1/b}$ and $e^{-1/b} \leq \frac{Pr[r>c+1]}{Pr[r>c]} \leq 1$.*

Note that the privacy guarantees in Theorem 1 do not depend on r. Further, note that, by definition, the sensitivity depends only on the query function whose output is being published and not on the input dataset. In many practical applications, when the data owner is interested in computing and releasing aggregate statistics over a histogram query function, the sensitivity is low and yields fairly accurate and privacy-preserving output statistics.

3.2.1.2 Exponential Mechanism

The Exponential mechanism proposed by McSherry and Talwar [134] aims to help in privately answering queries for which the addition of random noise to their output makes no sense (e.g., queries with non-numerical output, such as "What is the most frequent word users search for online?"). The idea of the mechanism is to select the output from among all possible query answers at random, with the probability of selecting a particular output being higher for those outputs that are "closer" to the true output.

More formally, let R be the range of possible outputs of the query function f, and let $u_f(D, r)$ be a utility function that measures how good output $r \in R$ is as an answer to the query function f given that the input dataset is D (with higher values of u_f representing better outputs). The

sensitivity $S(u_f)$ is defined as the maximum possible change in the utility function's value u_f due to the addition or removal of one person's data from the input, i.e.,

$$S(u_f) = \max_{D_1, D_2, r \in R} \big\{ \|u_f(D_1, r) - u_f(D_2, r)\|_1 : D_1 \text{ and } D_2 \text{ differ in at most one individual's data} \big\}.$$

Then

Theorem 2. *[134] Let D be the input dataset. The mechanism that chooses output r, with probability proportional to $\exp\big(\frac{\epsilon}{S(u_f)} u_f(D, r)\big)$ satisfies ϵ-differential privacy.*

3.2.2 Other Techniques

Since the introduction of the differential privacy definition [49], and discovery of the two foundational mechanisms for achieving it [49, 134], the search for other algorithms satisfying the definition has been an active area of research [45].

Differentially private algorithms have been proposed for problems such as:

- contingency table releases [21]
- obtaining synthetic databases that are useful for all queries of a particular class (such as predicate queries computing fractional counts) [28]
- computation of count queries satisfying fixed predicates [61]
- collaborative filtering techniques for producing recommendations [133]
- release of degree distributions of social networks [78]
- discovery of frequent itemsets [26]
- aggregation of distributed time-series data [156]
- computation of combinatorial optimization primitives, e.g., k-median and vertex cover [75]
- learning of classifiers via empirical risk minimization [33],

and data analysis frameworks supporting differentially private computations have been developed for traditional [132] and MapReduce [158] computations.

We describe and discuss the details of additional related works in each chapter.

Part II

Algorithms for Sharing and Mining User Data Privately

Chapter 4

Releasing Search Queries and Clicks Privately

There is much to be gained from sharing the search logs with the wider community, and the question of how to publish an anonymized search log was brought to the forefront by a well-intentioned, but privacy-unaware AOL search log release (Section 2.1.1). Since then a series of ad-hoc techniques have been proposed in the literature (Section 2.1.1.6), though none are known to be provably private. The open question to date has been whether there even exists a way to publish search logs in a perturbed fashion in a manner that is simultaneously useful and private. In this chapter, we take a first significant step towards answering that question.

Rather than looking for a privacy-preserving way to publish a search log in its entirety, we focus on a seemingly more attainable goal of releasing a privacy-preserving query click graph. In a query click graph the vertices correspond to both queries and URLs and there is an edge from a query to a URL with weight equal to the number of users who click on that URL given they posed the query. Each query node is labeled by the number of times this query was posed in the log. Similarly, there is an edge from one query to another query with weight equal to the number of users that posed one query and reformulated to another.

While a query click graph is not as rich a dataset as the actual search log, many computations can still be performed on the click graph with results similar to the computations starting with the actual search log [18, 40]. Query suggestions can be derived using common query reformulations. Spelling corrections can be inferred from queries with low click through and high reformulation rates.

Similar queries can be found using common URLs clicked for those queries. Query classification and keyword generation can also be deduced from the query click graph [58].

Our technical contributions are as follows:

- We propose a simple, intuitive, and efficiently implementable algorithm for producing a privacy-preserving query click graph based on a search log (Section 4.3).
- We utilize the formal privacy definition of differential privacy [49] (described in Chapter 3) adapted for the search logs context (Section 4.2), to prove that our proposed algorithm gives rigorous, rather than ad-hoc or intuition-based privacy guarantees (Section 4.4).
- We give a precise characterization of how to set the parameters of the proposed algorithm depending on the privacy guarantees desired (Section 4.5) and analyze the effects of the parameter settings on the characteristics of the publishable query set (Section 4.6.1).
- Although we do not utilize a formal notion of utility, we perform experiments to demonstrate that the query click graph we can produce can be of practical use. We show that the fraction of distinct queries that can be published, as well as the amount of search volume involving those queries, is non-negligible (Section 4.6.1). We then select two applications, keyword generation and studying human fears, and demonstrate that keywords and fears obtained from our query click graph closely resemble those obtained from the original unperturbed data (Section 4.6.2).

The main algorithmic insight of our work is that an intuition that publishing search queries can be done while preserving user privacy by "throwing away tail queries" [1], can be formalized into a privacy-preserving algorithm with rigorous privacy guarantees (Section 4.3). The modifications needed are:

1. limits on the extent to which a particular user's search and click activity is counted towards query and click frequency
2. systematic introduction of some randomness into the decision of whether a particular query is a tail query, by comparing the query's frequency with a fixed threshold after adding some random noise to the true frequency
3. systematic introduction of some randomness into the frequency counts reported, by adding some random noise to the true counts.

Given the queries that are determined to be safe to publish, the ten URLs surfaced are also safe to publish because anyone can pose a query to a search engine and see the top ten links. To publish the number of users who click on a result, we compute the actual number and add random noise.

Our work is the first to take a concrete definition of privacy, differential privacy, and design an algorithm for producing a private query click graph that provably satisfies that definition. Our algorithm also formalizes the "throw away tail queries" intuition into an algorithm with rigorous privacy guarantees, and mathematically quantifies exactly how to choose the threshold and random noise parameters so as to guarantee a desired level of privacy. The algorithm and the insights obtained from its analysis can be applied to improve privacy when sharing and mining user search data [105, 131].

4.1 Related Work

We use the ideas developed in [49], showing that any function with low sensitivity can be computed privately (Section 3.2), for our analysis. The aggregate statistical values reflected in our query click graph have low sensitivity, provided that each user issues a bounded number of queries - a condition which will be enforced by our algorithm.

Randomly sampling a dataset is known to preserve privacy under a definition closely related to differential privacy [32], but only under the assumption that there are few rare values. In the web context, it is more likely that every person's searches are a unique fingerprint to them; thus, randomly sampling the search log breaks privacy.

Prior to our work, no differentially private algorithm was known for efficiently publishing a query click graph. The exponential mechanism due to McSherry and Talwar [134] (Section 3.2.1) and another mechanism by Blum et al. [28] could be adapted to this task in theory but is not feasible in practice, as this mechanism takes time exponential in the size of the output space. More recently, McSherry and Talwar [135] have proposed an algorithm for releasing synthetic queries that holds promise for synthetic data releases.

The work of [69] takes a similar approach to ours, utilizing two thresholds, and addition of noise to query counts, in order to determine which queries can be published. Their work significantly expands on the analysis of the utility of the data, by choosing a concrete measure of algorithm's accuracy, and exploring two additional applications relying on published search log data. Their work provides additional confirmation to our experiments and claims that the subset of the search log data published using our algorithm can be valuable.

Finally, work of [86] aims to help users prevent search engines from building accurate interest profiles of them through a browser extension that issues randomized queries to the search engines. Their approach is interesting and empowering from the user perspective, but does not help address the problem of privacy-preserving search data release.

4.2 Differential Privacy for the Search Logs Context

In order to adapt the differential privacy definition of Section 3.1 to the search logs context we need to choose what we mean when we say that two datasets D_1 and D_2 differ in at most one individual's data. The notion of privacy we would like to preserve is the privacy of users. In the differential privacy context that means we want to ensure that the knowledge an attacker obtains about a particular user from the data release based on a search log is roughly the same, whether or not that user has used the search engine. Hence, when applying the differential privacy definition to the search logs context, we will consider all search logs D_1 and D_2 differing in at most one user's data, i.e., differing in at most one user's entire search history, rather than in one query. Note that a search engine may not always be able to identify when two sets of queries belong to the same user, for example, if those queries were posed from multiple devices, after browser cookies have been wiped, or if the IP addresses are unreliable. Therefore, in practice, our privacy guarantee will hold for user privacy as per search engine's identification of a user.

We use the weaker version of differential privacy, the (ϵ, δ) differential privacy (Definition 2 from Section 3.1.3), as the privacy guarantee we demand from our data release algorithm. We search for algorithms that release a strict subset of the data that was contained in the original search log, rather than algorithms that publish both queries that were contained in the log, as well as additional, "fake" queries that were not present in the log. We show in Lemma 6 (Section 4.8) that among algorithms that do not publish "fake" queries, only the algorithm that does not publish anything satisfies ϵ-differential privacy. Therefore, (ϵ, δ)-differential privacy is, in many ways, the strictest privacy definition we can hope to satisfy while preserving some utility when releasing search log data without introducing any "fake" data.

Hence, a randomized search log data release algorithm A is (ϵ, δ)-*differentially private,* if for all search logs D_1 and D_2 differing in at most one user's search history and all $\hat{D} \subseteq Range(A)$:

$$\Pr[A(D_1) \in \hat{D}] \leq e^{\epsilon} \cdot \Pr[A(D_2) \in \hat{D}] + \delta.$$

There are no hard and fast rules for setting ϵ and δ – it is generally left to the data releaser (recall, however, that these parameters are treated as public. One consideration to take into account when choosing δ is the number of users n participating in the dataset, and aim for δ being negligible in n. Indeed, imagine that the search log consists only of sensitive data, e.g., each user has posed exactly one query, which is the concatenation of their name, date of birth, and social security number. An algorithm that outputs one of these queries uniformly at random satisfies $(0, \frac{1}{n})$-differential privacy. At the same time, the algorithm implies that at least one user's private query is published and therefore, their privacy is compromised. Of course, imagining that the search log consists only of this extremely sensitive queries is a very paranoid view of privacy, but this example argues that the magnitude of $\frac{1}{\text{number of users}}$ is useful to keep in mind when choosing δ.

4.3 Algorithm for Releasing Search Queries and Clicks

We next describe our algorithm for generating a private query click graph. The key components are: determining which queries to publish, together with the number of times the query was posed and, further, determining which URLs to publish, together with the number of times the URL was clicked for each query. Our basic method for accomplishing these tasks utilizes a noisy count: for any statistic x of the data, the *noisy count* of x is $x+$ Lap(b), where Lap(b) denotes a random variable drawn independently from the Laplace distribution with mean zero and scale parameter b.

At a high-level, the algorithm proceeds as follows.

1. **Limit User Activity:** Keep only the first d queries posed by each user and first d_c URL clicks of each user (Line 2 of Algorithm 3).

2. **Queries:** If the noisy count of the number of times a query is posed exceeds a specified threshold, output the query together with its noisy count (Lines 3-5 of Algorithm 3).

3. **URLs:** If a query is safe to publish, then the ten URLs that are surfaced for that query are also safe to publish since anyone can pose the query to a search engine and see the ten results. For each query and ten surfaced URLs for that query, we output the noisy count of the number of times each URL was clicked for that query (Line 6 of Algorithm 3).

Some comments about the algorithm are in order.

- We limit user activity in order to preserve privacy: if a user can contribute an unbounded number of queries and clicks then they can have an unlimited influence on the set of queries

that are published (and the URLs that are clicked). In practice, a typical user does not pose an unlimited number of queries anyway — studies suggest that an average user poses about 34 queries per month [54]. We have flexibility for how we choose the d queries to keep for any user – the first d, random d, etc.

- While the main idea of throwing away tail queries is quite natural and has been previously suggested in the literature [1], we are the first to mathematically quantify exactly how to perform this operation so as to preserve privacy with respect to a rigorous privacy definition. Indeed, our theorems in subsequent sections establish a direct connection between the threshold used for defining a tail query and the resulting privacy guarantees.

- It is crucial to note that every time our algorithm computes the noisy count, the noise should be generated independently from the Laplace distribution. The independence of the random variables used for each query and each query-URL pair is required for the privacy guarantees to hold.

- A small caveat to using the algorithm is that, strictly speaking, its parameters such as K, d, d_c, b, b_q, b_c need to be made publicly known after they are chosen in order to guarantee soundness of the privacy proof.

Algorithm 3 Release-Data

1: **Input:** D - search log, d, d_c - parameters that limit user activity, b, b_q, b_c - noise parameters, K - threshold that defines tail.
2: **Limit-User-Activity**: $D \leftarrow$ Keep the first d queries and the first d_c URL clicks of each user.
3: For each query q, let $M(q, D) =$ number of times q appears in D
4: **Select-Queries**: $Q \leftarrow \{q : M(q, D) + Lap(b) > K\}$
5: **Get-Query-Counts**: For each q in Q, output $\langle q, M(q, D) + Lap(b_q)\rangle$
6: **Get-Click-Counts**: For each URL u in the top ten results for $q \in Q$, output $\langle q, u,$ number of times u was clicked when q was posed $+Lap(b_c)\rangle$.

4.4 Privacy Guarantees

We now state formally the (ϵ, δ)-differential privacy guarantees that our algorithm provides. We then give a sketch of the proof that each of the individual steps preserves privacy and, further, that their composition preserves privacy.

Let K, d, d_c, b, b_q, b_c be the parameters of Algorithm 3 such that $K \geq d$. Define $\alpha = \max(e^{1/b}, 1 + \frac{1}{2e^{(K-1)/b}-1})$ and the multiplicative and additive privacy parameters as $\epsilon_{alg} = d \cdot \ln(\alpha) + d/b_q + d_c/b_c$

and $\delta_{alg} = \frac{d}{2}\exp(\frac{d-K}{b})$.

Theorem 3. *Algorithm 3 is $(\epsilon_{alg}, \delta_{alg})$-differentially private for every pair of search logs differing in one user, where ϵ_{alg} and δ_{alg} are defined as above.*

4.4.1 Proof Overview

In order to prove Theorem 3, we will show that each step of the algorithm is differentially private for appropriate values of ϵ and δ and that their composition is also differentially private.

Lemma 1. ***Select-Queries*** *is $(d \cdot \ln(\alpha), \delta_{alg})$-differentially private if each user is limited to posing at most d queries, where α and δ_{alg} are defined as above.*

Lemma 2. ***Get-Query-Counts*** *is $(d/b_q, 0)$-differentially private.*

Lemma 3. ***Get-Click-Counts*** *is $(d_c/b_c, 0)$-differentially private.*

Lemma 4. *Suppose* ***Select-Queries*** *is (ϵ_1, δ)-differentially private,* ***Get-Query-Counts*** *is $(\epsilon_2, 0)$-differentially private, and* ***Get-Click-Counts*** *is $(\epsilon_3, 0)$-differentially private. Then Algorithm 3 is $(\epsilon_1 + \epsilon_2 + \epsilon_3, \delta)$-differentially private.*

Theorem 3 follows from Lemmas 1, 2, 3, and 4. We next sketch the key ideas in the proofs of the above lemmas.

4.4.2 Privacy for Selecting Queries

We first prove the guarantees for **Select-Queries** when each user can pose at most one query (Lemma 5) and then generalize the proof to hold for d queries per user to obtain Lemma 1.

4.4.2.1 Privacy for Select-Queries with $d = 1$:

Let $\alpha = \max(e^{1/b}, 1 + \frac{1}{2e^{(K-1)/b}-1})$ and $\delta_1 = \frac{1}{2}e^{\frac{1-K}{b}}$. Denote the Select-Queries algorithm by A.

Lemma 5. *If each user can pose at most one query and $K \geq 1$, then Select-Queries satisfies $(\ln(\alpha), \delta_1)$-differential privacy.*

Proof. Let D_1 and D_2 be arbitrary search logs that differ in exactly one query q_* posed by some user such that D_2 is the larger of the two logs. Let $\hat{D} \subseteq Range(A)$ denote an arbitrary set of possible outputs. Then we need to show the following:

$$\Pr[A(D_1) \in \hat{D}] \leq \alpha \Pr[A(D_2) \in \hat{D}] + \delta_1 \tag{4.1}$$

and that

$$\Pr[A(D_2) \in \hat{D}] \leq \alpha \Pr[A(D_1) \in \hat{D}] + \delta_1 \tag{4.2}$$

We consider two different scenarios depending on whether q_* already appears as a query of some user in D_1 or is a new query. For the first case, we do not need the additive parameter (i.e., $\delta_1 = 0$ works) as we can bound the ratio of the probabilities in each expression above. The key idea is to notice that q_* has a slightly higher probability of being released from D_2 compared to D_1 and to argue that the ratio of these probabilities can be bounded. However, for the second case, q_* can never be released from D_1, and hence we cannot bound the ratio of the probabilities. Instead, we make use of the fact that q_* occurs only once in D_2 and use the additive parameter δ_1 to bound the probability that its noisy count exceeds the threshold K.

Recall that our algorithm A only produces a subset of queries contained in D_1 (or D_2). Hence, any set of queries O that contains some query not present in D_1 or D_2 can safely be removed from $\hat{D}$ as the probability that A produces O, with D_1 or D_2 as input, is zero. We also partition $\hat{D}$ into two subsets: $\hat{D}^+$, the query sets in $\hat{D}$ that contain q_* and $\hat{D}^-$, the query sets in $\hat{D}$ that do not contain q_*.

Throughout the proof, we use Observations 1, 2 about the properties of Laplace distribution described in Section 3.2 and Observation 4 from Section 4.8 about the properties of ratios.

Case 1: $q_* \in D_1$

Let $M(q, D)$ be the number of times query q appears in the search log D. Since, by assumption, $q_* \in D_1$, we have $M(q_*, D_1) \geq 1, M(q_*, D_2) = M(q_*, D_1) + 1$.

We first prove inequality (4.1) with $\alpha = e^{1/b}$ and $\delta_1 = 0$ by upper bounding the ratio $\frac{\Pr[A(D_1)\in\hat{D}]}{\Pr[A(D_2)\in\hat{D}]}$, which we denote by R_2^1.

From our partition of $\hat{D}$ into $\hat{D}^+$ and $\hat{D}^-$ and using Observation 4 we have[1]:
$R_2^1 = \frac{\Pr[A(D_1)\in\hat{D}]}{\Pr[A(D_2)\in\hat{D}]} = \frac{\Pr[A(D_1)\in\hat{D}^+]+\Pr[A(D_1)\in\hat{D}^-]}{\Pr[A(D_2)\in\hat{D}^+]+\Pr[A(D_2)\in\hat{D}^-]} \leq \max\left(\frac{\Pr[A(D_1)\in\hat{D}^+]}{\Pr[A(D_2)\in\hat{D}^+]}, \frac{\Pr[A(D_1)\in\hat{D}^-]}{\Pr[A(D_2)\in\hat{D}^-]}\right)$

Consider now the ratio $\frac{\Pr[A(D_1)\in\hat{D}^+]}{\Pr[A(D_2)\in\hat{D}^+]}$. Recall that in our algorithm, the decision to release a particular query is made independently for each query and that D_1 and D_2 differ only in the number of times that q_* occurs in each of them. Hence, for a particular possible output O, s.t. $q_* \in O$: $\frac{\Pr[A(D_1)=O]}{\Pr[A(D_2)=O]} = \frac{\Pr[q_* \text{released by } A(D_1)]}{\Pr[q_* \text{released by } A(D_2)]}$. Generalizing this observation to all outputs $O_i \in \hat{D}^+$ we obtain: $\frac{\Pr[A(D_1)\in\hat{D}^+]}{\Pr[A(D_2)\in\hat{D}^+]} = \frac{\Pr[q_* \text{released by } A(D_1)]}{\Pr[q_* \text{released by } A(D_2)]} = \frac{\Pr[M(q_*,D_1)+Lap(b)>K]}{\Pr[M(q_*,D_2)+Lap(b)>K]} = \frac{\Pr[M(q_*,D_1)+Lap(b)>K]}{\Pr[M(q_*,D_1)+1+Lap(b)>K]}$.

By analogous reasoning with respect to $\hat{D}^-$ we obtain:

[1] Observation 4 holds only for positive denominators. In the event that either or both the denominators are zero, it can be shown that the differential privacy bound continues to hold.

$\frac{\Pr[A(D_1)\in\hat{D}^-]}{\Pr[A(D_2)\in\hat{D}^-]} = \frac{\Pr[M(q_*,D_1)+Lap(b)<K]}{\Pr[M(q_*,D_2)+Lap(b)<K]} = \frac{\Pr[M(q_*,D_1)+Lap(b)<K]}{\Pr[M(q_*,D_1)+1+Lap(b)<K]}$

From these two bounds on the ratios, we bound R_2^1:
$R_2^1 \leq \max\left(\frac{\Pr[M(q_*,D_1)+Lap(b)>K]}{\Pr[M(q_*,D_1)+1+Lap(b)>K]}, \frac{\Pr[M(q_*,D_1)+Lap(b)<K]}{\Pr[M(q_*,D_1)+1+Lap(b)<K]}\right)$, which by Observation 2 implies that

$$R_2^1 = \frac{\Pr[A(D_1)\in\hat{D}]}{\Pr[A(D_2)\in\hat{D}]} \leq \max(1, e^{1/b}) = e^{1/b} \tag{4.3}$$

proving inequality (4.1), as desired.

A similar analysis can be performed to show that

$$\frac{\Pr[A(D_2)\in\hat{D}]}{\Pr[A(D_1)\in\hat{D}]} \leq e^{1/b}, \tag{4.4}$$

yielding the proof of inequality (4.2) with $\alpha = e^{1/b}$ and $\delta_1 = 0$.

Case 2: $q_* \notin D_1, q_* \in D_2$

We now proceed to prove inequality (4.1) with $\alpha = \frac{1}{1-0.5\exp(\frac{1-K}{b})}$ and $\delta_1 = 0$.

First consider outputs that do not contain the new query q_*. By similar reasoning as in Case 1, the probability of obtaining an output O, where $O \in \hat{D}^-$ when starting from the D_2 log differs from the probability of obtaining an output O, when starting from D_1 only in the choice that the algorithm has to make for query q_*. Hence,
$\Pr[A(D_2)\in\hat{D}^-] = \Pr[q_* \text{ was not released by} A(D_2)]\cdot\Pr[A(D_1)\in\hat{D}^-]$, and therefore,

$$\frac{\Pr[A(D_1)\in\hat{D}^-]}{\Pr[A(D_2)\in\hat{D}^-]} = \frac{1}{\Pr[q_*\notin A(D_2)]} = \frac{1}{\Pr[1+Lap(b)<K]} = \frac{1}{1-0.5\exp(\frac{1-K}{b})} \tag{4.5}$$

Since query q_* is not present in D_1, our algorithm would not produce any output containing q_* when given D_1 as input, and thus $\Pr[A(D_1)\in\hat{D}^+] = 0$.

Using the partition of $\hat{D}$ into $\hat{D}^+$ and $\hat{D}^-$ we have:

$$\frac{\Pr[A(D_1)\in\hat{D}]}{\Pr[A(D_2)\in\hat{D}]} = \frac{\Pr[A(D_1)\in\hat{D}^-]}{\Pr[A(D_2)\in\hat{D}^+]+\Pr[A(D_2)\in\hat{D}^-]} \leq \frac{\Pr[A(D_1)\in\hat{D}^-]}{\Pr[A(D_2)\in\hat{D}^-]} \leq \frac{1}{1-0.5\exp(\frac{1-K}{b})} \tag{4.6}$$

proving inequality (4.1), as desired.

It remains to show that in this case, inequality (4.2) is satisfied with $\alpha = 1-0.5\exp(\frac{1-K}{b})$ and $\delta_1 = 0.5\exp(\frac{1-K}{b})$. Observe that

$$\Pr[A(D_2)\in\hat{D}^+] \leq \Pr[q_* \text{ was released}] = \Pr[M(q_*,D_2)+Lap(b_1)>K] = 0.5\exp(\frac{1-K}{b}) \tag{4.7}$$

Combining (4.7) and (4.5), we have:
$\frac{\Pr[A(D_2)\in\hat{D}]}{\Pr[A(D_1)\in\hat{D}]} = \frac{\Pr[A(D_2)\in\hat{D}^+]+\Pr[A(D_2)\in\hat{D}^-]}{\Pr[A(D_1)\in\hat{D}^-]} = \frac{\Pr[A(D_2)\in\hat{D}^-]}{\Pr[A(D_1)\in\hat{D}^-]} + \frac{\Pr[A(D_2)\in\hat{D}^+]}{\Pr[A(D_1)\in\hat{D}^-]} \leq 1 - 0.5\exp(\frac{1-K}{b}) + \frac{0.5\exp(\frac{1-K}{b})}{\Pr[A(D_1)\in\hat{D}]}$,
which completes the proof of inequality (4.2).

Thus from the two cases, depending on whether q_* is an additional occurence of an element already present in D_1 or is an entirely new element to D_1, we established that our algorithm satisfies the $(\ln(\alpha), \delta_1)$-differential privacy, where
$\alpha = \max\big(e^{1/b}, 1 - 0.5\exp(\frac{1-K}{b}), \frac{1}{1-0.5\exp(\frac{1-K}{b})}\big) = \max(e^{1/b}, \frac{1}{1-0.5\exp(\frac{1-K}{b})})$, and $\delta_1 = \frac{1}{2}e^{(\frac{1-K}{b})}$. □

Observation 3 (**Need for δ_1**). *A crucial observation made in the proof of this lemma is that the necessity for δ_1 arises only when the extra query in D_2 is a query that was not previously present in D_1, and we are attempting to upper bound the ratio of $\frac{\Pr[A(D_2)\in\hat{D}]}{\Pr[A(D_1)\in\hat{D}]}$.*

We will use this observation next as we generalize the proof of privacy guarantees to the case where each user can pose at most d queries.

4.4.2.2 Privacy for Select-Queries for arbitrary d:

We next prove Lemma 1. We first show that straight-forward generalization does not work and hence perform a tighter analysis.

4.4.2.2.1 Straight-forward generalization: A natural approach towards this proof is to observe that a search log D_2 that differs from D_1 by at most d queries can be obtained from D_1 by adding the d queries to it, one query at a time. This enables the repeated application of the results of Lemma 5 to obtain:
$\Pr[A(D_1) \in \hat{D}] \leq \alpha \Pr[A(D_1 + q_1) \in \hat{D}] + \delta_1 \leq \alpha(\alpha \Pr[A(D_1 + q_1 + q_2) \in \hat{D}] + \delta_1) + \delta_1 \leq \ldots$
$\leq \alpha^d \Pr[A(D_2) \in \hat{D}] + \delta_1 \frac{\alpha^d - 1}{\alpha - 1}$

However, this approach yields $\delta_{alg} = \delta_1 \frac{\alpha^d - 1}{\alpha - 1}$, which will quickly exceed 1, yielding meaningless privacy guarantees. To avoid the blow-up in the additive component of privacy guarantees, we build on the insights of the Proof of Lemma 5, and especially, on Observation 3 in order to show better guarantees for δ_{alg}.

4.4.2.2.2 Tighter analysis for δ_{alg}: As in the proof of Lemma 5, let D_2 be the larger of the two search logs, containing an additional d queries compared to D_1.

We again split the elements of $\hat{D}$ into two subsets as follows: denote by $\hat{D}^-$ the set of elements of $\hat{D}$ which can be obtained from both D_1 and D_2, and by $\hat{D}^+$ – the set of elements of $\hat{D}$ which can

only be obtained from D_2 (as before, we remove those elements of $\hat{D}$ that cannot be obtained from either D_1 or D_2 from consideration wlog).

Observe that in the proof of Lemma 5, the additive component δ arose only when considering the ratio $\frac{\Pr[A(D_2)\in\hat{D}]}{\Pr[A(D_1)\in\hat{D}]}$ and not when considering the ratio $\frac{\Pr[A(D_1)\in\hat{D}]}{\Pr[A(D_2)\in\hat{D}]}$. We take advantage of this observation by proving the necessary upper bounds by recursively applying Lemma 5 in one case, and by performing a more careful analysis for the need for the additive component in the other case.

Proof of (4.1) with $\alpha_d = \alpha^d, \delta_{alg} = 0$:

Suppose D_2 can be obtained from D_1 by adding queries $z_1, \ldots, z_d$ (some of these may not be distinct). Then $\frac{\Pr[A(D_1)\in\hat{D}]}{\Pr[A(D_2)\in\hat{D}]}$ can be represented as a product of ratios of obtaining an output in $\hat{D}$ when starting from datasets differing in one element as follows:

$$\frac{\Pr[A(D_1)\in\hat{D}]}{\Pr[A(D_2)\in\hat{D}]} = \frac{\Pr[A(D_1)\in\hat{D}]}{\Pr[A(D_1+z_1)\in\hat{D}]} \cdot \frac{\Pr[A(D_1+z_1)\in\hat{D}]}{\Pr[A(D_1+z_1+z_2)\in\hat{D}]} \cdot \ldots \cdot \frac{\Pr[A(D_1+z_1+\ldots+z_{d-1})\in\hat{D}]}{\Pr[A(D_1+z_1+\ldots+z_d)\in\hat{D}]}$$

Applying the above decomposition of the ratio into a product of ratios [2] and the results of intermediate steps (4.3) and (4.6) of Lemma 5 to each of the ratios in the product, we obtain: $\frac{\Pr[A(D_1)\in\hat{D}]}{\Pr[A(D_2)\in\hat{D}]} \leq \alpha^d$, as desired.

Proof of (4.2) with $\alpha_d = \alpha^d, \delta_{alg} = \frac{d}{2}\exp(\frac{d-K}{b})$:

Denote by $x_1, \ldots, x_{n_x}$ those of the additional d queries that are already in D_1 and by $y_1, \ldots, y_{n_y}$ – those queries that are unique to D_2. Note that $\sum_{i=1}^{n_x}\big(M(x_i, D_2) - M(x_i, D_1)\big) + \sum_{i=1}^{n_y} M(y_i, D_2) \leq d$ and $n_x + n_y \leq d$.

By definition of $\hat{D}^+$ one obtains an output in $\hat{D}^+$ when given the search log D_2, only if at least one of the queries in D_2 which was not present in D_1 is chosen for release. Applying the union bound we have:

$$\Pr[A(D_2) \in \hat{D}^+] \leq \sum_{i=1}^{n_y} \Pr[y_i \text{ was chosen for release}] = \sum_{i=1}^{n_y} \Pr[M(y_i, D_2) + Lap(b) > K] =$$

$$= \sum_{i=1}^{n_y} \Pr[Lap(b) > K - M(y_i, D_2)] = \big(\text{since } K \geq d \geq M(y_i, D_2)\big) = \frac{1}{2}\sum_{i=1}^{n_y} \exp\left(\frac{M(y_i, D_2) - K}{b}\right)$$

$$\big(\text{applying the knowledge that } n_y \leq d \text{ and } M(y_i, D_2) \leq d\big) \leq \frac{d}{2}\exp\left(\frac{d-K}{b}\right) \quad (4.8)$$

We now represent the ratio $\frac{\Pr[A(D_2)\in\hat{D}^-]}{\Pr[A(D_1)\in\hat{D}^-]}$ as a product[3] of ratios of obtaining an output in $\hat{D}^-$

[2] As long as $\Pr[A(D_1) \in \hat{D}] \neq 0$, the denominator of all the ratios involved in the product is guaranteed to be non-zero; whereas, if $\Pr[A(D_1) \in \hat{D}] = 0$, then the desired upper bound of α^d holds for it automatically.

[3] The denominator of any of the product terms is 0 only if $\Pr[A(D_1) \in \hat{D}^-] = 0$, in which case the desired differential privacy guarantees follow from (4.8).

when starting from datasets differing in one element:
$\frac{\Pr[A(D_2)\in\hat{D}^-]}{\Pr[A(D_1)\in\hat{D}^-]} = \frac{\Pr[A(D_1+z_1)\in\hat{D}^-]}{\Pr[A(D_1)\in\hat{D}^-]} \cdot \frac{\Pr[A(D_1+z_1+z_2)\in\hat{D}^-]}{\Pr[A(D_1+z_1)\in\hat{D}^-]} \cdot \ldots \cdot \frac{\Pr[A(D_1+z_1+\ldots z_d)\in\hat{D}^-]}{\Pr[A(D_1+z_1+\ldots z_{d-1})\in\hat{D}^-]}$
Recall from (4.4) that if $z_i \in \{D_1 + z_1 + \ldots + z_{i-1}\}$ then $\frac{\Pr[A(D_1+z_1+\ldots z_i)\in\hat{D}^-]}{\Pr[A(D_1+z_1+\ldots z_{i-1})\in\hat{D}^-]} \leq e^{\frac{1}{b}}$
Recall from (4.5) that if $z_i \notin \{D_1 + z_1 + \ldots + z_{i-1}\}$ then $\frac{\Pr[A(D_1+z_1+\ldots z_i)\in\hat{D}^-]}{\Pr[A(D_1+z_1+\ldots z_{i-1})\in\hat{D}^-]} = 1 - 0.5\exp(\frac{1-K}{b})$.
Combining the last three inequalities, we have

$$\frac{\Pr[A(D_2) \in \hat{D}^-]}{\Pr[A(D_1) \in \hat{D}^-]} \leq \prod_{i=1}^{d-n_y} e^{\frac{1}{b}} \prod_{i=1}^{n_y} \big(1 - 0.5\exp(\frac{1-K}{b})\big) \leq e^{\frac{d-n_y}{b}} \leq \alpha^d \tag{4.9}$$

We now use (4.8) and (4.9) to obtain the desired upper bound:
$\frac{\Pr[A(D_2)\in\hat{D}]}{\Pr[A(D_1)\in\hat{D}]} = \frac{\Pr[A(D_2)\in\hat{D}^-]+\Pr[A(D_2)\in\hat{D}^+]}{\Pr[A(D_1)\in\hat{D}^-]} = \frac{\Pr[A(D_2)\in\hat{D}^-]}{\Pr[A(D_1)\in\hat{D}^-]} + \frac{\Pr[A(D_2)\in\hat{D}^+]}{\Pr[A(D_1)\in\hat{D}^-]} \leq \alpha^d + \frac{0.5d\exp(\frac{d-K}{b})}{\Pr[A(D_1)\in\hat{D}^-]} = \alpha^d + \frac{0.5d\exp(\frac{d-K}{b})}{\Pr[A(D_1)\in\hat{D}]}$, hence $\Pr[A(D_2) \in \hat{D}] \leq \alpha^d \Pr[A(D_1) \in \hat{D}] + 0.5d\exp(\frac{d-K}{b})$, as desired. □

4.4.3 Privacy for Noisy Counts

We next show that the steps involving noisy counts (**Get-Query-Counts** and **Get-Click-Counts**) are differentially private. We reduce both these steps to the problem of releasing histograms privately studied by [49] and described in Section 3.2. Intuitively, the privacy guarantee follows from limitations of user activity in Algorithm 3 **Release-Data**, ensuring bounded sensitivity, and the addition of correspondingly calibrated Laplace noise.

For **Get-Query-Counts**, the reduction from Theorem 1 (Section 3.2) is as follows: the domain $\mathcal{D}$ (the search log) is partitioned into bins (distinct queries) according to Q. Function f reports the number of elements in the bin for each bin, i.e., the number of occurences of each query. The datasets differ in one user, who can pose at most d queries, hence $S(f) = d$. Therefore, by Theorem 1 adding $Lap(d/\epsilon)$ noise to each query occurrence count gives ϵ-differential privacy guarantee, and **Get-Query-Counts** is d/b_q-differentially private.

Similarly for **Get-Click-Counts**, each query-URL pair serves as a partition bin (note that the top 10 URLs returned upon searching for a given query are not private if the query is known and hence the partition is known given the set of queries Q). Hence, by Theorem 1, adding $Lap(b_c)$ noise to the true count of the number of clicks on a URL for a query will preserve d_c/b_c-differential privacy.

4.4.4 Privacy for Composition of Individual Steps

We have shown that steps 4–6 of the **Release-Data** algorithm preserve privacy, so it remains to show that limiting the user activity and the composition of the steps preserves privacy and to quantify the effect of applying these algorithms in sequence on the privacy guarantees (Lemma 4).

We note that **Limit-User-Activity** helps to satisfy the condition in Lemma 1 that each user should pose at most d queries and otherwise does not affect the privacy guarantees. To prove Lemma 4, we note that a straight-forward composition does not work, as it would cause the additive privacy parameter to blow up. We exploit the special structure of our algorithm and obtain the desired tighter composition result through a more careful analysis. Our result is similar to the observations of [47, 48] that in the (ϵ, δ) differential privacy context, the parameters are added during composition but we provide it for completeness.

4.4.4.1 Proof of Lemma 4

Proof. Denote the **Select-Queries** algorithm (that operates on the search log input D) by A_1, and **Get-Query-Counts** algorithm (that operates on the search log input D, and queries Q selected for release by $A_1(D)$) by A_2. We show that if A_1 satisfies $(\ln(\alpha_1), \delta)$ differential privacy, and A_2 satisfies $\ln(\alpha_2)$-differential privacy, then the application of A_1 algorithm followed by A_2 satisfies $(\ln(\alpha_1+\alpha_2), \delta)$ differential privacy. The composition with **Get-Click-Counts** follows using a similar argument.

Let D_1 and D_2 be two search logs that differ in one user's search history, i.e., differ in at most d queries and at most d_c clicks. Denote by $\hat{Q}$ the subset of the set of possible outputs $\hat{D}$ restricted to the queries, i.e., ignoring their counts, and by $\hat{C}(Q)$ – the (query, query counts) subset of the set of possible outputs in $\hat{D}$ whose queries are Q. In other words, $\hat{D} = \cup_{Q \in \hat{D}} \hat{C}(Q)$.

As in the previous proofs, wlog, let D_2 be the larger of the two input datasets, and let us exclude from $\hat{D}$ those outputs which are not feasible to achieve from either D_1 nor D_2.

Observe that

$$\Pr[A_2\big(D_1, A_1(D_1)\big) \in \hat{D}] = \sum_{Q \in \hat{Q}} \Pr[A_1(D_1) = Q] \cdot \Pr[A_2(D_1, Q) \in \hat{C}(Q)] \tag{4.10}$$

We now prove the differential privacy guarantees by separately upper bounding the output probabilities when starting from input D_1 relative to starting from input D_2, and vice versa, similar to our approach of separately proving inequalities (4.1) and (4.2) in the proof of Lemma 1.

Upper Bound for $\Pr[A_2(A_1(D_1)) \in \hat{D}]$:

Recall from proof of (4.1) in the proof of Lemma 1 that

$\Pr[A_1(D_1) = Q] \leq \alpha_1 \Pr[A_1(D_2) = Q]$.

By assumption on the privacy guarantees satisfied by A_2 we have

$\Pr[A_2(D_1, Q) \in \hat{C}(Q)] \leq \alpha_2 \Pr[A_2(D_2, Q) \in \hat{C}(Q)]$.

Combining these two inequalities and equation (4.10) we obtain:

$\Pr[A_2\big(D_1, A_1(D_1)\big) \in \hat{D}] \leq \sum_{Q \in \hat{Q}} \alpha_1 \Pr[A_1(D_2) = Q] \cdot \alpha_2 \Pr[A_2(D_2, Q) \in \hat{C}(Q)] =$

$= \alpha_1\alpha_2 Pr[A_2\big(D_2, A_1(D_2)\big) \in \hat{D}]$

Upper Bound for $\Pr[A_2(A_1(D_2)) \in \hat{D}]$:

Analogous to our reasoning in earlier proofs, denote by $\hat{D}^-$ a set of those outputs from $\hat{D}$ possible to obtain from both D_1 and D_2; by $\hat{D}^+$ – those outputs possible to obtain only from D_2. Denote by $\hat{Q}^+$ and $\hat{Q}^-$ the corresponding sets of output queries.

Observe that

$\Pr[A_2\big(D_2, A_1(D_2)\big) \in \hat{D}^+] \leq \Pr[A_1(D_2) \in \hat{Q}^+] \leq$ (from (4.8) in the proof of Lemma 1) $\leq \delta$.

On the other hand, by (4.9) in the proof of Lemma 1

$\Pr[A_1(D_2) \in \hat{Q}^-] \leq \alpha_1 \Pr[A_1(D_1) \in \hat{Q}^-]$,

and by assumption on the privacy guarantees satisfied by A_2 we have

$\Pr[A_2(D_2, Q) \in \hat{C}(Q)] \leq \alpha_2 \Pr[A_2(D_1, Q) \in \hat{C}(Q)]$.

Therefore, $\frac{\Pr[A_2\left(D_2, A_1(D_2)\right) \in \hat{D}]}{\Pr[A_2\left(D_1, A_1(D_1)\right) \in \hat{D}]} = \frac{\Pr[A_2\left(D_2, A_1(D_2)\right) \in \hat{D}^+]}{\Pr[A_2\left(D_1, A_1(D_1)\right) \in \hat{D}^-]} + \frac{\Pr[A_2\left(D_2, A_1(D_2)\right) \in \hat{D}^-]}{\Pr[A_2\left(D_1, A_1(D_1)\right) \in \hat{D}^-]} \leq$

(using definition of $\hat{D}^-$, equation (4.10), Observation 4, and preceding three inequalities)

$\leq \frac{\Pr[A_2\left(D_2, A_1(D_2)\right) \in \hat{D}^+]}{\Pr[A_2\left(D_1, A_1(D_1)\right) \in \hat{D}]} + \max_{Q \in \hat{Q}^-} \left(\frac{\Pr[A_1(D_2) = Q] \cdot \Pr[A_2(D_2, Q) \in \hat{C}(Q)]}{\Pr[A_1(D_1) = Q] \cdot \Pr[A_2(D_1, Q) \in \hat{C}(Q)]} \right)$

$\leq \frac{\delta}{\Pr[A_2\left(D_1, A_1(D_1)\right) \in \hat{D}]} + \alpha_1 \cdot \alpha_2$, and hence,

$\Pr[A_2(D_2, A_1(D_2)) \in \hat{D}] \leq \alpha_1\alpha_2 \Pr[A_2(D_1, A_1(D_1)) \in \hat{D}] + \delta$, as desired. □

4.5 Discussion

The algorithm and analysis leaves several questions open for discussion. How does a data releaser set the various parameters in the analysis above? Is the algorithm really different from just publishing queries with sufficiently large frequency (without the addition of noise)? What frequency queries end up getting published? Why do we use fresh noise in Step 5 **Get-Query-Counts**? What happens if a count is negative after adding noise? Is it possible to release query reformulations? In this section we provide answers to each of these questions in turn.

d	1	5	10	20	40	80	160
K	5.70	31.99	66.99	140.00	292.04	608.16	1264.49
b	0.43	2.17	4.34	8.69	17.37	34.74	69.49

Table 4.1: Optimal choices of the threshold, K and noise, b as a function of d for fixed privacy parameters, $e^{\epsilon} = 10, \delta = 10^{-5}$

Setting parameters. Given the numerous parameters in the theorems just proved, a natural question is how to set them. As mentioned earlier, it is up to the data releaser to choose ϵ, while it is advisable that $\delta < 1/n$, where n is the number of users. What about the remaining parameters? Lemma 1 offers answers for optimally setting[4] the threshold K and noise b when the desired privacy parameters and the limit for the number of queries per user are known: $K = d\left(1 - \frac{\ln(\frac{2\delta}{d})}{\epsilon}\right)$ and $b = \frac{d}{\epsilon}$.

Table 4.1 shows how the optimal choices of threshold K and noise b vary as a function of the number of queries allowed per user, d, for fixed privacy parameters, $e^{\epsilon} = 10$ and $\delta = 10^{-5}$.

Publishing head queries. An important and natural question is why not just publish the queries with frequency larger than an intuitively-pleasing fixed large number? The answer is that any such deterministic algorithm is provably not differentially private [150]. Intuitively, if an adversary knows the number K being used as an exact threshold, and has a guess for someone's private query, he can pose this query to a search engine $K - 1$ times, and observe whether the query is published or not in order to determine if that guess is correct, thereby violating the targeted person's privacy. Beyond the privacy breach, it is not clear how one should even select such a large number. Our approach has the advantage that the threshold value K can be determined purely from the privacy parameters, ϵ and δ, and d. The values K and b are independent of the characteristics of the search log.

Which queries are published. If the algorithm is different than publishing the queries with sufficiently high frequency, it is reasonable to wonder which frequency queries do get published? Consider Figure 4.1 which shows the probability of the query being chosen for release as a function of its frequency, for $d = 20, K = 140$, and $b = 8.69$. The queries whose frequency is above 170 are virtually guaranteed to be released, the queries whose frequency is below 110 are virtually guaranteed not to be released, and the queries whose frequency is between 110 and 170 might or might not be released depending on the random choices of the algorithm. It is clear that since b is smaller than K, **Select-Queries** is close to the intuitive sharp threshold algorithm of only releasing the "head

[4]Assuming we desire to minimize the noise added and that $e^{1/b} \geq 1 + \frac{1}{2e^{(K-1)/b}-1}$, which is the case for value ranges considered.

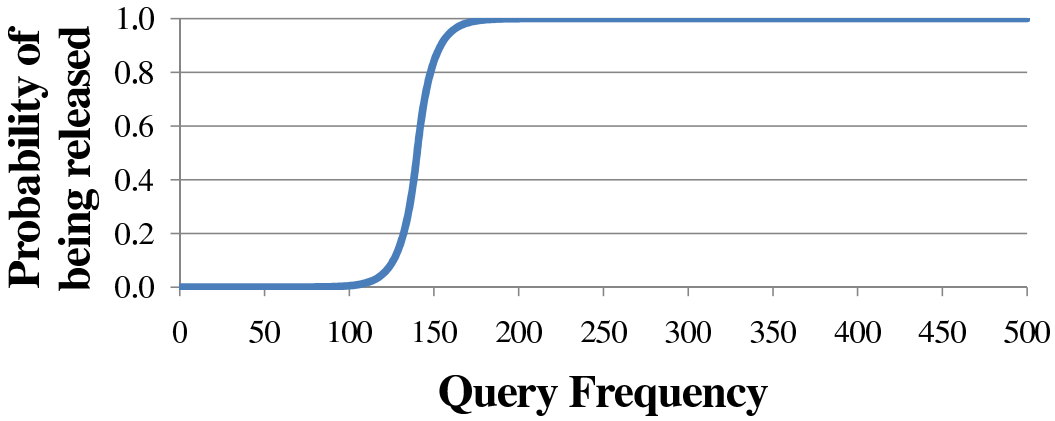


Figure 4.1: Probability of a query being released as a function of its frequency, for $d = 20, K = 140$, and $b = 8.69$

queries", where the "head queries" are defined as those whose frequency is above 170.

Fresh noise. Next we explain why we recompute the noisy query frequency count in Step 5 of **Release-Data**. Technically speaking, Step 5 is not necessary: instead of computing the noisy count of $M(q, D)$, we could release the noisy count that was computed when deciding whether to release query q in Step 4. Re-using the noisy count from Step 4 would lead to all released query occurrence counts being skewed towards the larger side; for instance, there will be no queries whose reported number of occurrences is less than K. On the other hand, skipping Step 5 would improve the ϵ in the differential privacy guarantee of the algorithm by reducing it by d/b_c. It is up to the data releaser to choose the trade-off between more evenly distributed query counts and privacy.

Negative counts. The addition of Laplace random noise might yield lower or higher frequency counts than the true counts, in some cases yielding negative query occurrence or query-click frequency counts. The negative counts released where positive counts are expected are counter-intuitive but do not pose a privacy risk, only a useability risk. If desired, one can perform a post-processing step replacing all negative counts with zeros without impacting privacy (since any post-processing that does not rely on the original data preserves privacy).

Query reformulations. Finally, we revisit the issue of releasing query reformulations, which are a valuable part of the search log, but are not directly handled by our algorithm. In fact, query reformulations can be produced in the specific case when the reformulation is a click on a query suggestion. In such a case, a click on a reformulation is treated as a click on a URL, since a surfaced suggestion is as public as a surfaced URL. To ensure privacy, we could modify Step 2, **Limit User Activity**, to count the first d queries that were typed and treat clicks on reformulations in the

same way as URL clicks. These reformulations would not be as powerful and rich as actual user reformulations since they are limited to what a search engine is already capable of suggesting.

4.6 Experimental Results

There is no doubt that attempting to preserve user privacy when performing a search log data release will take a toll on the utility of the data released, and that an algorithm that aims to satisfy rigorous privacy guarantees will not be able to release datasets that are as useful as the ones obtained through ad-hoc approaches such as merely replacing usernames with random identifiers. However, the decrease in utility is offset by the ability to protect user privacy and the ability to avoid PR disasters and retain user trust in the search engine. In this section, we describe several simple properties of the query and clicks data that can be released by applying our proposed algorithm to the search logs of a major search engine and characterize their dependence on the parameters of the algorithm and other choices that need to be made by the data releaser. The aim of the properties that we study is to initiate a study of utility of the data that can be released for several concrete applications. A much more extensive evaluation of utility using additional applications can be found in the work of [69].

Our experiments suggest that in the absence of other provably private methods for data release, and considering that our approach closely mimics the one that others are anecdotally considering utilizing, our proposed algorithm could serve as a first step towards the eventual goal of performing provably private and useful search log data releases.

Experimental Setup. We obtained the full query logs from a major search engine. The information necessary for our experiments is the session information per user (in order to restrict to d queries per user) and the issued queries together with their clicked URLs. We performed a marginal amount of cleaning of the logs by removing special sets of characters (e.g., extra white spaces or runaway quotes) from the queries. In our comparisons below, we considered queries to match regardless of the case in words.

4.6.1 Published Query Set Characteristics

In our first experiment, we seek to gain insight into the effect of privacy guarantees desired from the algorithm on the following two characteristics of the published query set:

Percent of distinct queries released, i.e., the ratio of the number of distinct queries released to the actual number of distinct queries for all users in the original dataset (without limitations on how many queries the user can pose). Our goal here is to capture how representative is the published query set of the real set in terms of the released queries.

Percent of query impressions released, i.e., the ratio of the number of query impressions released (with each query accounted for as many times as the released noisy count) to the actual number of query impressions for all users in the original dataset. Our goal in this case is to capture how much query volume is published by our algorithm.

4.6.1.1 Effect of Maximum Queries d per User

A crucial step of our algorithm is to limit the number of queries posed per user that we consider for the release to d. Since the optimal choice of d is non-obvious from the perspective of a data releaser, we start by studying the effect of d and fixed privacy parameters on the published query set characteristics when starting from a one week log from October 2007.

We compute the percent of distinct queries and impressions released by Algorithm 3 for different values of d and for parameters $e^\epsilon = 10$ and $\delta = 10^{-5}$, choosing the threshold K and noise b as described in Section 4.5. The results are shown in Figure 4.2. The horizontal axis represents the increasing values of d, the right vertical axis represents the percent of distinct queries released for a given value of d, and the left vertical axis shows the percent of impressions for the respective d.

From Figure 4.2, we observe that although we can only publish a small percent of distinct queries overall, we can cover a reasonbly large percent of impressions. More specifically, the output of our algorithm contains in total at most about 0.75% (for $d = 1$) of the distinct queries present in the original query log. However, the released queries correspond to about 10% – 35% of the search volume, depending on the choice of d, with a maximum of about 34% of volume achieved by d of around 20. The distinct queries released form a tiny fraction of the log because an overwhelming fraction of queries are issued very few times and our algorithm throws away such "tail queries" in order to guarantee privacy. However, we can release all the "frequent queries" and by virtue of their frequency, the volume of the released queries is substantial. Furthermore, although the percent of distinct queries released is small, the absolute number of the distinct queries released is large.

In Figure 4.2, we also observe that the percent of distinct queries released decreases as d increases. This is due to the fact that the threshold K increases slightly more than linearly with d; when d increases, our dataset may contain more distinct queries and larger counts for all previously present

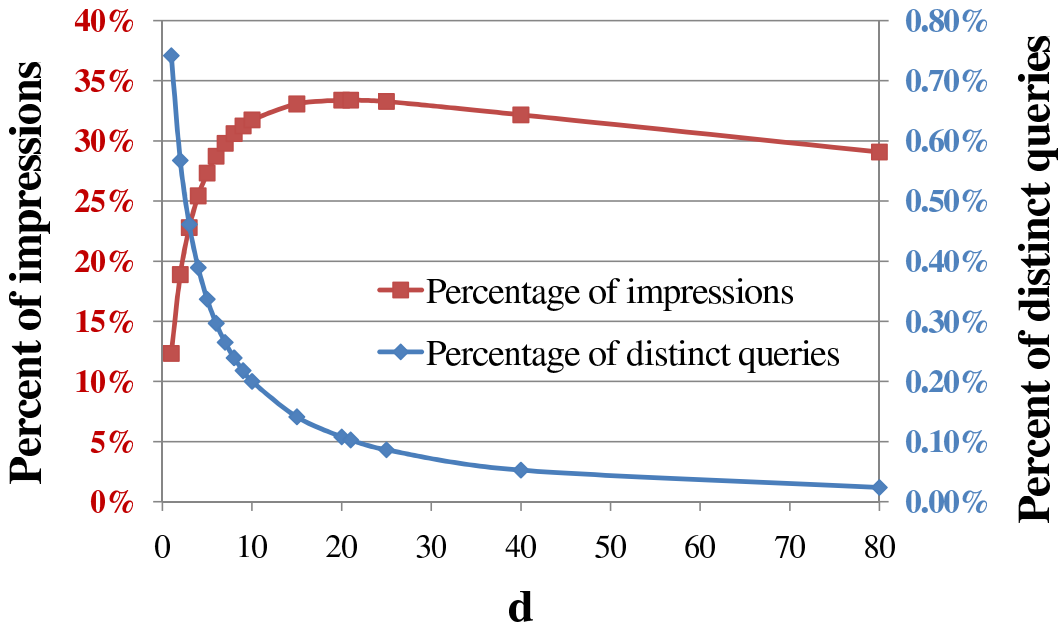


Figure 4.2: Percent of distinct queries and impressions released as a function of d, for fixed privacy parameters, $e^{\epsilon} = 10, \delta = 10^{-5}$ and a one week time period

queries, but such larger count is insufficient to offset the required increase in K to ensure the same privacy guarantees. Finally, we observe that for the one week log, the percent of impressions released initially increases with d, peaks for d around 20 and then decreases. There are two competing forces responsible for this observation: as d gets larger, more queries (and hence impressions) per user are included in the data while at the same time the threshold K needs to be increased in order to maintain the same privacy guarantees.

4.6.1.2 Effect of Time-span of the Log

We next study the effect of the time period of the log considered for release on the size of the released dataset. We repeated the previous experiment with query logs extracted over different time periods (one day, two weeks and one month from October 2007, and also one year) and compared to the output data generated from our log of one week. We plot the percent of released queries (distinct and impressions) in Figures 4.3 and 4.4 respectively. Figure 4.3 demonstrates that the percent of distinct queries released is more or less independent of the time-span of the source query log. On the other hand, the total number of query impressions does depend on the time-span as shown in

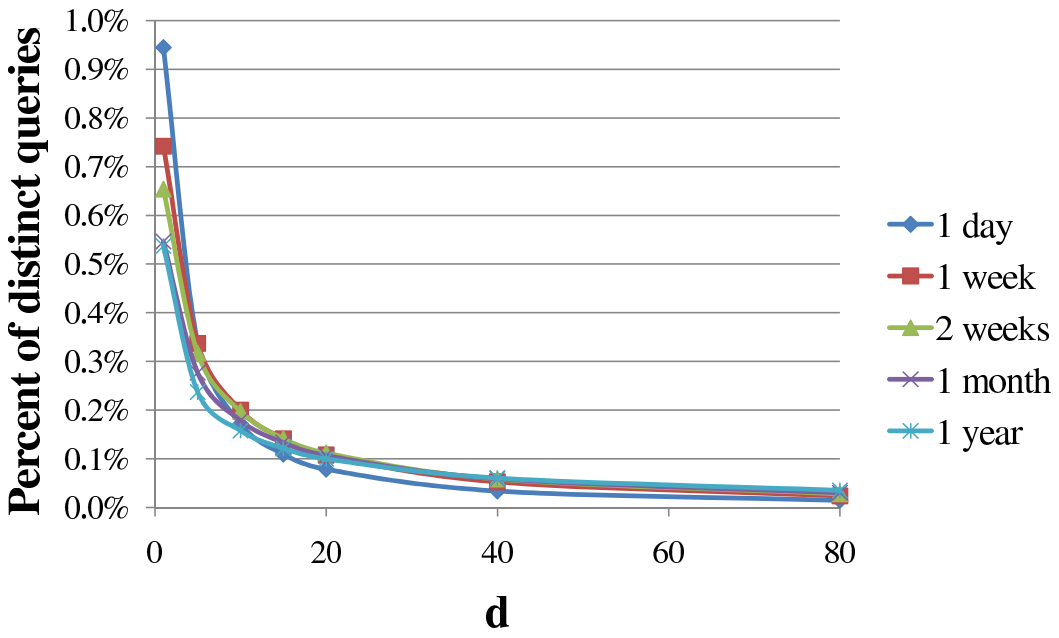


Figure 4.3: Percent of distinct queries released as a function of d for different time periods, with fixed privacy parameters, $e^{\epsilon} = 10, \delta = 10^{-5}$

Figure 4.4. In this case, we observe that the value of d at which the maximum percent of impressions is achieved increases with the length of the time period. This fits well with the intuition for sensible choices of d - as the time-span increases, the number of queries each user is limited to should increase as well.

The absolute number of queries (distinct and impressions) released increases with the increase in time-span of the source log. For example, for $d = 20$, the absolute number of distinct queries released grows 6-fold over one week, 12-fold over two weeks, 24-fold over one month, and 184-fold over one year time-spans of the source logs compared to that of a one day source log. Similarly the absolute number of impressions released grows 7-fold over one week, 15-fold over two weeks, 33-fold over one month, and 325-fold over one year durations compared to that of a one day source log. Thus, for the fixed choices of privacy parameters and d, it may be desirable to start with a query log extracted over a longer period of time, such as one year, to obtain better utility.

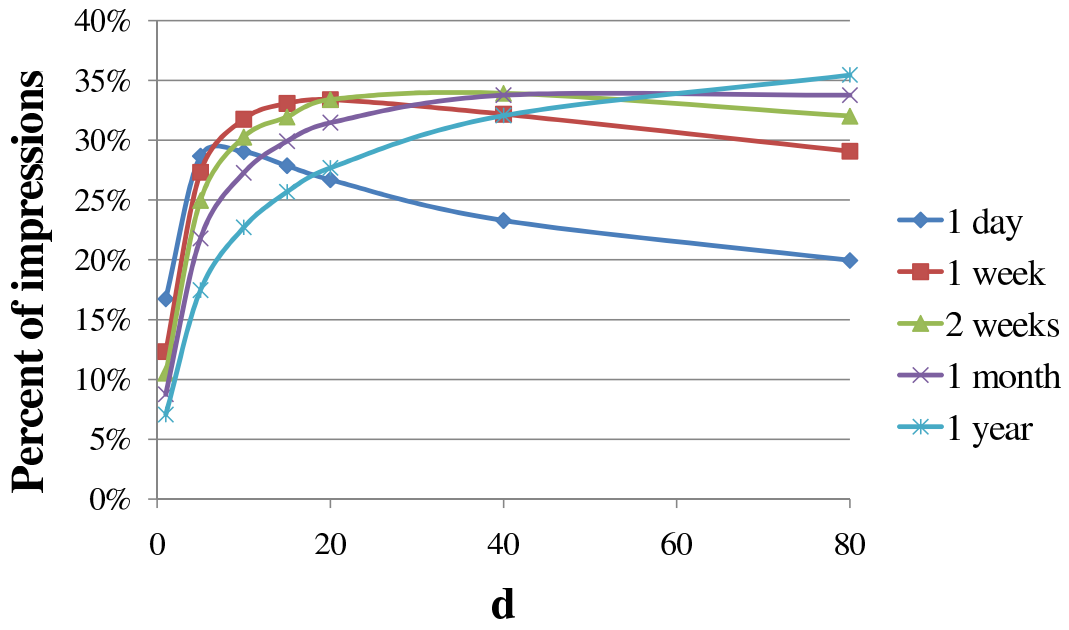


Figure 4.4: Percent of query impressions released as a function of d for different time periods, with fixed privacy parameters, $e^{\epsilon} = 10, \delta = 10^{-5}$

4.6.1.3 Effect of Privacy Parameters

We now turn to studying how the choice of multiplicative and additive privacy parameters used in Algorithm 3 affects the size of the released query set for fixed time period (one week) and $d = 21$[5]. Intuitively, the percent of distinct queries released should increase with less strict privacy requirements. More specifically, the larger the values of ϵ and δ, the larger the portion of the original query log we should be able to release.

We present the percent of distinct queries and impressions as a function of privacy requirements captured by different values of ϵ and δ, in Tables 4.2 and 4.3 respectively. They show that, in general, the percent of queries (distinct and impressions) that we can release increases as δ increases, with the relative increase in the percent of distinct queries being slightly higher for a given ϵ.

[5]Although 21 queries per week per user seems to be a harsh restriction on the amount of queries posed, it turns out that an average user performs even fewer than 21 queries per week [54].

% of distinct queries	$\epsilon = \ln(2)$	$\epsilon = \ln(5)$	$\epsilon = \ln(10)$
$\delta = 10^{-6}$	0.03%	0.06%	0.09%
$\delta = 10^{-5}$	0.03%	0.07%	0.10%
$\delta = 10^{-4}$	0.04%	0.09%	0.12%

Table 4.2: Percent of distinct queries released as a function of privacy parameters, for one week time period and $d = 21$

% of query impressions	$\epsilon = \ln(2)$	$\epsilon = \ln(5)$	$\epsilon = \ln(10)$
$\delta = 10^{-6}$	26.79%	30.92%	32.64%
$\delta = 10^{-5}$	27.55%	31.67%	33.38%
$\delta = 10^{-4}$	28.45%	32.55%	34.23%

Table 4.3: Percent of query impressions released as a function of privacy parameters, for one week time period and $d = 21$

4.6.2 Utility of the Published Dataset

In this section our goal is to study the utility of the dataset output by our algorithm both in terms of the utility of the queries and the utility of the query click graph. First, we give anecdotal examples of queries that could be released and then study the usefulness of these queries for social science research, and the usefulness of the query click graph obtained by the algorithm for an algorithmic application. We call the output of our algorithm *the published or released dataset* for conciseness, even though no data release had occurred as a result of our experiments.

We start with the query log over one year duration (restricted to $d = 21$ queries per user), and run **Release-Data** algorithm to obtain the queries safe to release, their noisy counts, and the noisy query-URL click counts. For each of the queries, we obtain the top 20 most clicked URLs instead of the top 20 URLs returned by the search engine, which is nearly equivalent as almost all users look at only the first page of 10 results [94] and very rarely beyond the second page; hence, these URLs get the most clicks. For simplicity of experiment implementation, we did not limit the number of clicks per user, since most users click on very few results per search query anyway [54]. For determining the queries that are safe to release, we use the privacy parameters $e^{\epsilon} = 10$ and $\delta = 10^{-5}$ (so that the threshold $K = 147.4$ and noise $b = 9.1$). For click counts we use noise $b_c = 0.43$.

Some examples of non-trivial queries released are: “girl born with eight limbs”, “cash register software”, “vintage aluminum christmas trees”, “how to tie a windsor knot”.

Rank	Comorbidity Survey	Original Log	Released Queries
1.	Bugs, mice, snakes	Flying	Flying
2.	Heights	Heights	Heights
3.	Water	Snakes, spiders	Public speaking
4.	Public transportation	Death	Snakes, spiders
5.	Storms	Public speaking	Death
6.	Closed spaces	Commitment	Commitment
7.	Tunnels and bridges	Intimacy	Abandonment
8.	Crowds	Abandonment	The dark
9.	Speaking in public	The dark	Intimacy

Table 4.4: Most common fears, depending on the data source

4.6.2.1 Utility for Studying Human Fears

Since users communicate with a search engine in an uninhibited manner, posing queries containing their most private interests and concerns, the search log could also be an invaluable source of insights for social science research. We take an example proposed by Tancer [178] of using the queries posed by users to obtain insight into human fears and compare the conclusions that can be obtained by studying the queries of the original log vs the released queries.

Tancer [178] suspected that the insight one can gain into human fears through search logs is different than the data obtained through surveys. He compared the results of the National Comorbidity Survey [100], a phone survey where respondents were asked about their fears, with Internet searches he had access to through Hitwise that contained the term "fear of", testing the theory that some people must be searching the Web to understand their fears. He observed that after removing those terms that are not phobia searches (such as "Fear of Clowns", a movie), the ordering of the most frequent fears as reported in the Comorbidity Survey differs from that obtained from searches. Furthermore, there are more social fears in the list of top searches than in the list of top fears from the Comorbidity Survey, perhaps due to people being less likely to admit to a social fear when answering questions posed by another person than when posing queries to an inanimate search engine.

We repeat Tancer's experiment on the search engine data available to us and on the proposed perturbed release of that data, and obtain the ranking of fear frequencies that can be seen in Table 4.4.

The ordering of most popular fears is not fully preserved in the released queries compared to the original searches due to the noise added. However, the set of top nine fear searches obtained is the same in both cases, and is noticeably different from the one reported in the survey. For example,

there is only one social fear in the Comorbidity Survey top list, the fear of speaking publicly, versus four in the "fear of" searches: public speaking, commitment, intimacy, and abandonment. Both the original log and the perturbed query click graph suggest that survey responses may only be reflecting a certain part of what people are afraid of, and suggest a strong direction for further study by social scientists. Hence, the dataset that could be published using our algorithm could have non-trivial utility in identifying directions for future study by social scientists and providing preliminary support for hypotheses.

4.6.2.2 Utility for Keyword Generation

We study the utility of the released query click graph by comparing the performance of an important application that utilizes a query click graph in its original and released forms. The application we study is *keyword generation*: given a business that is interested in launching an online advertising campaign around a concept, suggest keywords relevant to it. The idea of generating additional keywords from a seed set of keywords or URLs is powerful because it enables the advertisers to expand their exposure to users through bidding on a wider set of keywords. We use the algorithm proposed by Fuxman et al. [58] that exploits the query click graph. Their algorithm takes a seed set of URLs about the concept and uses the idea of random walks on the query click graph with the seed set URLs as absorbing states in order to generate more keywords. Typically, random walks are highly sensitive to changes in the graph, and hence, on one hand, it would be surprising if the keyword generation algorithm worked well on the released perturbed query click graph, given how much it has changed from the original query click graph. On the other hand, we would like to understand to what extent it still works.

We compare the keywords generated by this algorithm over the original graph and the released graph for three different seed sets. Each seed set consists of all URLs from the respective domains: 1) *shoes*: shoes.com; 2) *colleges*: Homepage domains of the top ten U.S. universities according to the US News Report ranking from early 2009; 3) *suicide*: six domains associated with depression and suicide (depression.com, suicide.com, and the 4 corresponding ones from WebMD and Wikipedia). The parameters of the keyword generation algorithm are set as in [58] ($\alpha = 0.001, \gamma = 0.0001$). We pruned all query-URL pairs with less than 10 clicks in both the original and the released graphs for efficiency of implementation.

Compared to the query click graph based on the original log, the released private query click graph contains 9.85% of the queries and 20.43% of the edges of the original graph. Nonetheless,

URL seed set		Relevance probability threshold 0.5	0.7	0.9	0.95
colleges	released	3,667	2,403	1,337	883
	original	28,346	17,103	8,287	6,373
shoes	released	2,733	1,620	448	248
	original	19,669	8,418	1,800	1,047
suicide	released	175	116	50	39
	original	3,945	2,517	806	525

Table 4.5: Number of keyword suggestions generated depending on URL seed set and query click graph source. Relevance probability refers to the probability that the keyword belongs to the seed set concept.

as can be seen in Table 4.5, we find that for all three seed sets, the absolute number of keywords generated by the algorithm using the released query click graph is substantial. For the shoes seed set, the number of keywords generated on the basis of the private graph comprises 13.9% to 24.9% of the number of keywords that can be generated using the original graph, depending on the relevance probability threshold used. For the college seed set the number of keywords generated is in the range of 12.9% to 16.1% of the original; and for the suicide seed set it is 4.4% to 7.4%. Moreover, more than 78% of the keyword suggestions obtained from the released graph were also obtained from the original graph, which is an indication of the similar level of relevance of the keywords produced in both cases. We observe greater overlap for more focused concepts (shoes: 93%, suicide: 99%).

We conclude that the released query click graph is still very useful for keyword generation. While one cannot generate as many keyword suggestions on the basis of the released graph as one would on the basis of the original graph[6], the volume of keyword suggestions generated is still substantial.

4.7 Summary and Open Questions

In this chapter, we took a first major step towards a solution for releasing search logs by proposing an algorithm for releasing queries and clicks in a manner that guarantees user privacy according to a rigorous privacy definition.

We have shown that some non-trivial fraction of queries and impressions can be privately released and that the released query click graph can be successfully used for applications such as keyword generation and studies of people's fears. The question of whether this graph would be useful for

[6]It is important to note that the keyword suggestions obtained from the original graph that are not present among those obtained from the released graph, are not necessarily private. Algorithm 3 is conservative and chooses not to release many queries, a large fraction of which are likely not sensitive by themselves.

other applications or whether it could serve as a benchmark log is far from being answered and is further explored with encouraging results in [69].

It is worth noting that as people invent more ways to group similar queries (such as "mom" and "mother"), we could use the grouping techniques to improve the performance of Algorithm 3. It is also possible that a tighter analysis of the algorithm's guarantees could lead to weaker requirements on the threshold K and noise b needed; we mention one possible improvement in the analysis in Section 4.8.

A separate issue is that our algorithm implicitly assumes that users behave in an honest manner. However, there are ways for an attacker to maliciously bring private tail queries already known to them into the head. For instance, since K, b, d are public parameters, an attacker could create, say, $K + 5b$ copies of themselves and in their first d queries issue a private query such as someone else's credit card number. The net effect is that the search engine would publish this private data. We do not have a way to get around such malicious activities and leave this too as a direction for future work.

It seems promising to try using the query click graph to generate a synthesized search log: select a query according to the perturbed frequency distribution that we publish, select clicks according to the perturbed probability a document is clicked, select a reformulation according to the perturbed distribution over reformulations, or select a new query. This procedure would not generate a search log per se, since no timestamps would be published and it is not clear if the sessions would actually be meaningful. We leave open the question of how best to generate a search log from the perturbed data that we publish.

Finally, there are many other aspects to releasing search logs besides the privacy of users. For instance, releasing queries and clicks reveals at a large scale the performance of a search engine. Thus, the log leaks queries where the search engine performs poorly, e.g., abandoned head queries or head queries with few clicks. As another example, search data reveals queries that surface adult content when they should not. So beyond the privacy of users, there are other factors to consider before search data is released. However, our techniques for determining which queries can be released can be of immediate use in existing applications such as Google's autocomplete[7] and Bing's search suggestions[8]. These applications predict and display search queries to a user as they type based on what other users are searching for, and could use our algorithm's approach in determining which suggestions are safe to display.

[7] http://www.google.com/support/websearch/bin/static.py?hl=en&page=guide.cs&guide=1186810&answer=106230
[8] http://onlinehelp.microsoft.com/en-us/bing/ff808490.aspx

4.8 Miscellaneous Technical Details

Lemma 6. *Suppose $A(D)$ is an algorithm that selects which queries to publish from those present in the search log D and A satisfies ϵ-differential privacy. Then $A(D)$ always publishes an empty set of queries.*

Proof. We prove by contradiction. Suppose for some non-empty input search log D_0, the algorithm A publishes a query x from D_0, i.e., $\Pr[A(D_0) = x] > 0$. Now,
while D_i has at least one user u whose search history contains x and $\Pr[A(D_i) = x] > 0$,
let D_{i+1} be the search log obtained by removing that user u and all of his queries from D_i.

We stop after k iterations, for some $k \geq 1$, either because $\Pr[A(D_k) = x] = 0$, or because there are no more occurrences of query x in D_k, in which case, by assumption that our algorithm selects queries to publish from the log and is not allowed to publish "fake" queries, we also have $\Pr[A(D_k) = x] = 0$.

Since, by assumption, A satisfies ϵ-differential privacy, if we set $\hat{D} = \{x\}$ we have:
$\Pr[A(D_0) \in \hat{D}] \leq e^{\epsilon} \Pr[A(D_1) \in \hat{D}] \leq e^{2\epsilon} \Pr[A(D_2) \in \hat{D}] \Pr[A(D_{k-1}) \in \hat{D}] \leq \ldots \leq\leq e^{k\epsilon} \Pr[A(D_k) \in \hat{D}] = 0$. A contradiction, which completes the proof. □

Observation 4. ***[Properties of ratios]*** *For $a, b \geq 0$ and $c, d > 0$: $\frac{a+b}{c+d} \leq \max(\frac{a}{c}, \frac{b}{d})$.*

A small improvement of Theorem 3:
The main result of this chapter (Theorem 3) can be slightly improved by applying a tighter bound in inequality (4.8) used in the proof of Lemma 1. The more careful analysis below shows that Theorem 3 holds for an even smaller δ_{alg} than stated, namely for $\delta_{alg} = 0.5 \exp(-\frac{K}{b})\big(\exp(\frac{d}{b}) + d - 1\big)$. This also implies that when the desired privacy guarantees and the limit of queries per user are known, the threshold K used by the algorithm can be slightly lower than the one we applied in our experiments, namely, $K = \frac{d}{\epsilon}\left(\ln\big(\exp(\epsilon) + d - 1\big) - \ln(2\delta)\right)$.

The improvement can be attained as a result of applying a tighter bound in the last step of inequality (4.8) in the proof of Lemma 1, in which δ_{alg} is determined:
$\frac{1}{2}\sum_{i=1}^{n_y} \exp\left(\frac{M(y_i,D_2)-K}{b}\right) = 0.5 \exp(\frac{-K}{b}) \sum_{i=1}^{n_y} \exp\left(\frac{M(y_i,D_2)}{b}\right) \leq$ (using Lemma 7 below)
$\leq 0.5 \exp(\frac{-K}{b})\left(\exp\big(\sum_{i=1}^{n_y} \frac{M(y_i,D_2)}{b}\big) + n_y - 1\right) \leq 0.5 \exp(\frac{-K}{b})\big(\exp(\frac{d-n_x}{b}) + n_y - 1\big)$
$\leq 0.5 \exp(\frac{-K}{b})\big(\exp(\frac{d}{b}) + d - 1\big)$.

Lemma 7. $\forall x_i > 0, n \geq 2 : \sum_{i=1}^{n} e^{x_i} \leq e^{\sum_{i=1}^{n} x_i} + n - 1$.

Proof. We prove by induction, starting with base case $n = 2$. For the base case to hold, we need to show that for $x > 0, y > 0 : e^x + e^y \leq e^{x+y} + 1$. Indeed,
$1 - e^x + e^{x+y} - e^y = 1 - e^x - e^y(1 - e^x) = (1 - e^x)(1 - e^y) > 0$.

Suppose that the lemma's statement holds for $n = k \geq 2$, and proceed to prove it for $n = k + 1$. $\sum_{i=1}^{k+1} e^{x_i} = \sum_{i=1}^{k} e^{x_i} + e^{x_{k+1}} \leq$ (by inductive assumption for $n = k$) $\leq (e^{\sum_{i=1}^{k} x_i} + k - 1) + e^{x_{k+1}} = e^{\sum_{i=1}^{k} x_i} + e^{x_{k+1}} + k - 1 \leq$ (by inductive assumption for $n = 2$) $\leq (e^{x_{k+1} + \sum_{i=1}^{k} x_i} + 1) + k - 1 = e^{\sum_{i=1}^{k+1} x_i} + k$, which completes the inductive proof. □

Part III

Quantifying Utility-Privacy Trade-offs for Social Data

Chapter 5

Social Recommendations

Making recommendations or suggestions to users in order to increase user engagement is a common and meaningful practice for websites [34, 117]. For instance, YouTube recommends videos, Amazon suggests products, and Netflix recommends movies, in each case with the goal of making as *relevant* a recommendation to the user as possible.

The phenomenal participation of users in social networks, such as Facebook and LinkedIn, has given tremendous hope for designing a new type of user experience, the *social* one. The feasibility of social recommendations has been fueled by initiatives such as Facebook's Open Graph API[1], that explicitly create an underlying graph where people, events, movies, etc., are uniformly represented as nodes, and connections, such as friendship relationships, event participation, interest in a book or a movie, are represented as edges between those nodes. The connections can be established through friendship requests, event RSVPs, and social plug-ins[2], such as the "Like" button.

Web companies are striving to *personalize* recommendations by incorporating the likes and dislikes of an individual's social neighborhood into their recommendation algorithms. Instead of defaulting to generic recommendations of items popular among all users of the site, a social-network aware system can provide recommendations based on what is popular among active friends or friends-of-friends of that particular user. There has been much research and industrial activity to solve two problems: (a) recommending content, products, ads not only based on the individual's prior history but also based on the likes and dislikes of those the individual trusts [9, 27, 127, 191], and (b)

[1] https://developers.facebook.com/docs/reference/api
[2] http://developers.facebook.com/plugins

recommending others whom the individual might trust [82]. In this chapter, we focus on recommendation algorithms based exclusively on graph link-analysis, i.e., algorithms that rely on underlying connections between people and other entities, rather than their individual features, to make recommendations.

However, when designing algorithms based on graph analysis one needs to carefully consider the privacy implications. For instance, a social recommendation algorithm that recommends to you only the products that your friends buy, seems like a perfectly sensible and useful algorithm. However, if you only have one friend, this algorithm would reveal the entire shopping history of that friend – information that he may not have intended to share. Here is another example of how a privacy breach can arise from a social recommendation. Imagine a browser toolbar integrated with your social network that recommends you webpages to visit based on the webpages visited by your friends. Suppose you go to a page X, and the social recommendation algorithm suggests that, based on browsing history of your friends who visited page X, you may also like page Y. If you know which one of your friends likes Y, then the social recommendation enables you to infer that this friend also visited website X. If X's content is sensitive, e.g., related to a medical condition, you infer something your friend may not have wanted to share, resulting in a privacy breach. Finally, a system that uses only *trusted edges* in suggestions may leak information about lack of trust along specific edges, which would also constitute a privacy breach.

In this chapter we present a theoretical study of the privacy/utility trade-offs in personalized graph link-analysis based social recommender systems. There are many different settings in which social recommendations may be used (friend, product, interest recommendations, or trust propagation), each having a slightly different formulation of the privacy concerns (the sensitive information is different in each case). However, all these problems have a common structure – recommendations are made based on a social graph (consisting of people and other entities), where some subset of the edges is sensitive. For clarity of exposition, we ignore scenario specific constraints, and focus on a generic model. Our main contributions are intuitive and precise trade-off results between privacy and utility for a clear formal model of personalized social recommendations, emphasizing impossibility of social recommendation algorithms that are both accurate and private for all users [55].

We consider a graph where all edges are sensitive, and an algorithm that recommends a single node v to some target node u. We assume that the algorithm is based on a *utility function*, satisfying certain natural properties (Section 5.2.4), that encodes the "goodness" of recommending each node in the graph to this target node. We focus on graph link-analysis recommenders; hence, the utility

function must only be a function of the nodes and edges in the graph. Suggestions for graph link-analysis based utility functions include: number of common neighbors, number of weighted paths, and PageRank distributions [87, 122, 163]. We consider an attacker who wishes to deduce the existence of a single edge (x, y) in the graph with n nodes by passively observing a recommendation (v, u). We measure the privacy of the algorithm using ϵ-*differential privacy* - requiring the ratio of the likelihoods of the algorithm recommending (v, u) on the graphs with, and without, the edge (x, y), respectively, to be bounded by e^{ϵ}. We define accuracy of a recommendation algorithm R as the ratio between R's expected utility to the utility achieved by an optimal (non-private) recommender. In this setting:

- We present and quantify a trade-off between accuracy and privacy of any social recommendation algorithm that is based on any general utility function. This trade-off shows a lower bound on the privacy parameter ϵ that must be incurred by an algorithm that wishes to guarantee any constant-factor approximation of the maximum possible utility (Section 5.3).

- We present stronger lower bounds on privacy and the corresponding upper bounds on accuracy for algorithms based on two particular utility functions previously suggested for social recommendations – number of common neighbors and weighted paths [82, 87, 122, 163]. If reasonable privacy is to be preserved when using the common neighbors utility function, only nodes with $\Omega(\log n)$ neighbors can hope to receive accurate recommendations (Section 5.3.3).

- We show how the two well-known differential privacy-preserving mechanisms, Laplace noise addition and the Exponential mechanism (Section 3.2.1), can be used for producing social recommendations based on a known utility vector (Section 5.4.1). We then briefly consider the setting when an algorithm may not know (or be able to compute efficiently) the entire utility vector, and propose and analyze a sampling based linear smoothing algorithm (Section 5.4.2).

- We perform experiments on two real graphs using several utility functions. The experiments compare the accuracy of Laplace and Exponential mechanisms, and the upper bound on achievable accuracy for a given level of privacy, as per our proof. Our experiments suggest three takeaways: (i) For most nodes, the lower bounds imply harsh trade-offs between privacy and accuracy when making social recommendations; (ii) The more natural Laplace algorithm performs as well as Exponential; and (iii) For a large fraction of nodes, the gap between accuracy achieved by Laplace and Exponential mechanisms and our theoretical bound is not significant (Section 5.5).

We now discuss related work and systems, and then formalize our model and problem statement in Section 5.2.

5.1 Related Work

Several papers propose that social connections can be effectively utilized for enhancing online applications [9, 127]. Golbeck [65] uses the trust relationships expressed through social connections for personalized movie recommendations. Mislove et al. [138] attempt an integration of web search with social networks and explore the use of trust relationships, such as social links, to thwart unwanted communication [139]. Approaches incorporating trust models into recommender systems are gaining momentum [199], [140], [173]. In practical applications, the most prominent example of graph link-based recommendations is Facebook's recommendation system that recommends to its users Pages corresponding to celebrities, interests, events, and brands, based on the social connections established in the people and Pages social graph[3]. More than 100,000 other online sites[4], including Amazon[5] and the New York Times, are utilizing Facebook's Open Graph API and social plug-ins. Some of them rely on the social graph data provided by Facebook as the sole source of data for personalization. Depending on the website's focus area, one may wish to benefit from personalized social recommendations when using the site, while keeping one's own usage patterns and connections private – a goal whose feasibility we analyze in this work.

There has been recent work discussing privacy of recommendations, but it does not consider the social graph. Calandrino et al. [31] demonstrate that algorithms that recommend products based on friends' purchases have very practical privacy concerns. McSherry and Mironov [133] show how to adapt the leading algorithms used in the Netflix prize competition to make privacy-preserving movie recommendations. Their work does not apply to algorithms that rely on the underlying social graph between users, as the user-user connections have not been released as part of the Netflix competition. Aïmeur et al. [7] propose a system for data storage for privacy-preserving recommendations. Our work differs from all of these by considering the privacy/utility trade-offs in graph-link analysis based social recommender systems, where the graph links are private.

Bhaskar et al. [26] consider mechanisms analogous to the ones we adapt, for an entirely different problem of making private frequent item-set mining practically efficient, with distinct utility notion, analysis, and results.

[3] http://www.facebook.com/pages/browser.php
[4] http://developers.facebook.com/blog/post/382
[5] https://www.amazon.com/gp/yourstore?ie=UTF8&ref_=pd_rhf_ys

There has been an extensive effort aimed towards publishing anonymized versions of social graphs or synthetic graphs that have similar characteristics to the original [16, 38, 79, 197]. These are not helpful in the social recommendations context as they do not disclose the correspondence of the user to the node in the published graph, and therefore, do not enable graph link-based recommendations for individuals.

5.2 Preliminaries and the Formal Problem Model

We formalize the problem definition and initiate the discussion by establishing notation and describing what a social recommendation algorithm entails. We then adapt the differential privacy definition of Chapter 3 to the social recommendations context. We define the accuracy of an algorithm and formally state the problem of designing a private and accurate social recommendation algorithm. Finally, as we aim for our privacy/utility trade-off work to be applicable for a general class of social recommendation algorithms, we define properties that we expect those algorithms to satisfy.

5.2.1 Social Recommendation Algorithm

Let $G = (V, E)$ be the graph that describes the network of connections between people and entities, such as products purchased. Each recommendation is an edge (i, r), where node i is recommended to the *target node* r. Given graph G, and target node r, we denote the utility of recommending node i to node r by $u_i^{G,r}$, and since we are considering the graph as the sole source of data, the utility is some function of the structure of G. We assume that a recommendation algorithm R is a probability vector on all nodes, where $p_i^{G,r}(R)$ denotes the probability of recommending node i to node r in graph G by the specified algorithm R. We consider algorithms aiming to maximize the expected utility $\sum_i u_i^{G,r} \cdot p_i^{G,r}(R)$ of each recommendation. Our notation defines algorithms as probability vectors, thus capturing randomized algorithms; note that all deterministic algorithms are special cases. For instance, an obvious candidate for a recommendation algorithm would be $\mathcal{R}_{best}$ that always recommends the node with the highest utility (equivalent to assigning probability 1 to the node with the highest utility). Note that no algorithm can attain a higher expected utility of recommendations than $\mathcal{R}_{best}$.

When the graph G and the target node r are clear from context, we drop G and r from the notation – u_i denotes utility of recommending i, and p_i denotes the probability that algorithm R

recommends i. We further define $u_{\max} = \max_i u_i$, and $d_{\max}$ - the maximum degree of a node in G.

5.2.2 Differential Privacy for the Social Recommendations Context

Recall that differential privacy (Section 3.1) is based on the principle that an algorithm preserves privacy of an entity if the algorithm's output is not too sensitive to the presence or absence of the entity's information in the input data set. In our setting of graph link-analysis based social recommendations, we wish to maintain the presence (or absence) of an edge in a graph private. Hence, the differential privacy definition (Definition 1 of Section 3.1.3) in this context can be stated as follows:

Definition 3. *A recommendation algorithm R satisfies ϵ-differential privacy if for any pair of graphs G and G' that differ in one edge (i.e., $G = G' + \{e\}$ or vice versa) and every set of possible recommendations S,*

$$Pr[R(G) \in S] \leq \exp(\epsilon) \times Pr[R(G') \in S] \tag{5.1}$$

where probabilities are over random coin tosses of R.

We show trade-offs between utility and privacy for algorithms making a *single* social recommendation, and restricting our analysis to algorithms making one recommendation allows us to relax the privacy definition. We require Equation (5.1) to hold only for edges e that are not incident to the node receiving the recommendation. This relaxation reflects the natural setting in which the node receiving the single recommendation (the attacker) already knows which nodes in the graph he is connected to, and hence the algorithm only needs to protect the knowledge about the presence or absence of edges that do not originate from the attacker node. While we consider algorithms making a single recommendation throughout, we use the relaxed variant of differential privacy *only* in Sections 5.3.3 and 5.5.

5.2.3 Problem Statement

We now formally define the *private social recommendation problem*. Given utility vectors (one per target node), determine a recommendation algorithm that (a) satisfies the ϵ-differential privacy constraints and (b) maximizes the accuracy of recommendations.

Definition 4 (**Private Social Recommendations**)**.** *Design a social recommendation algorithm R with maximum possible accuracy under the constraint that R satisfies ϵ-differential privacy.*

The remaining question is how to measure the *accuracy* of an algorithm. For simplicity, we focus on the problem of making recommendations for a fixed target node r. Thus, the algorithm takes as input only one utility vector $\vec{u}$, corresponding to utilities of recommending each of the nodes in G to r, and returns one probability vector $\vec{p}$ (which may depend on $\vec{u}$).

Definition 5 (Accuracy). *The accuracy of an algorithm R is defined as* $\min_{\vec{u}} \frac{\sum u_i p_i}{u_{\max}}$.

In other words, an algorithm is $(1-\delta)$-accurate if (1) for every input utility vector $\vec{u}$, the output probabilities p_i are such that $\frac{\sum u_i p_i}{u_{\max}} \geq (1-\delta)$, and (2) there exists an input utility vector $\vec{u}$ such that the output p_i satisfies $\frac{\sum u_i p_i}{u_{\max}} = (1-\delta)$. The second condition is added for notational convenience (so that an algorithm has a well defined accuracy). In choosing the definition of accuracy, we follow the paradigm of worst-case performance analysis from the algorithms literature[6]; average-case accuracy analysis may be an interesting direction for future work.

Recall that $u_{\max}$ is the maximum utility achieved by any algorithm (in particular by $\mathcal{R}_{best}$). Therefore, an algorithm is said to be $(1-\delta)$-accurate if for any utility vector, the algorithm's expected utility is at least $(1-\delta)$ times the utility of the best possible algorithm. A social recommendation algorithm that aims to preserve privacy of the edges will have to deviate from $\mathcal{R}_{best}$, and accuracy is the measure of the fraction of maximum possible utility it is able to preserve despite the deviation. Notice that our definition of accuracy is invariant to rescaling utility vectors, and hence all results we present are unchanged on rescaling utilities.

5.2.4 Properties of Utility Functions and Algorithms

Our goal is to theoretically determine the bounds on maximum accuracy achievable by any social recommendation algorithm that satisfies ϵ-differential privacy. Instead of assuming a specific graph link-based recommendation algorithm, more ambitiously we aim to determine accuracy bounds for a general class of recommendation algorithms. In order to achieve that, we define properties that one can expect most reasonable utility functions and recommendation algorithms to satisfy, and restrict our subsequent analysis to utility functions and algorithms satisfying them.

5.2.4.1 Axioms for Utility Functions

We present two axioms, *exchangeability* and *concentration*, that should be satisfied by a meaningful utility function in the context of recommendations on a social network. Our axioms are inspired by

[6] we assume wlog that the utility of the least useful recommendation is 0; otherwise we would define accuracy as $\min_{\vec{u}} \frac{\sum u_i p_i}{u_{\max} - u_{\min}}$.

work of [122] and the specific utility functions they consider: number of common neighbors, sum of weighted paths, and PageRank based utility measures.

Axiom 1 (**Exchangeability**). *Let G be a graph and let h be an isomorphism on the nodes giving graph G_h, s.t. for target node r, $h(r) = r$. Then $\forall i : u_i^{G,r} = u_{h(i)}^{G_h,r}$.*

This axiom captures the intuition that in our setting of graph link-analysis based recommender systems, the utility of a node i should not depend on the node's identity. Rather, the utility for target node r only depends on the structural properties of the graph, and so, nodes isomorphic from the perspective of r should have the same utility.

Axiom 2 (**Concentration**). *There exists $S \subset V(G)$, such that $|S| = \beta$, and $\sum_{i \in S} u_i \geq \mu \sum_{i \in V(G)} u_i$ for some $\mu = \Omega(1)$ and $\beta = o(n^d)$ for any positive d.*

This says there are some β nodes that together have at least a constant fraction of the total utility. This is likely to be satisfied for small enough β in practical contexts, as in large graphs there are usually a small number of nodes that are very good recommendations for r and a long tail of those that are not. Depending on the setting, β may be a constant, or may be a function growing with the number of nodes. Furthermore, β can be viewed as dependent not only on the utility function but also on μ. For intuition purposes, one may think of μ as a fixed large fraction, say $\mu = 0.9$.

5.2.4.2 Property of Recommendation Algorithms

We present a property, *monotonicity*, that should be satisfied by a meaningful recommendation algorithm:

Definition 6 (**Monotonicity**). *An algorithm is said to be monotonic if $\forall i, j$, $u_i > u_j$ implies that $p_i > p_j$.*

The monotonicity property is a very natural notion for a recommendation algorithm to satisfy. It says that the algorithm recommends a higher utility node with a higher probability than a lower utility node.

In our subsequent discussions, we only consider the class of monotonic recommendation algorithms for utility functions that satisfy the exchangeability axiom as well as the concentration axiom for a reasonable choice of β. In Section 5.3.2.3 we briefly mention how the lower bounds can be altered to avoid this restriction.

A running example throughout the chapter of a utility function that satisfies these axioms and is often successfully deployed in practical settings [82, 122, 163] is that of the *number of common neighbors utility function*: given a target node r and a graph G, the number of common neighbors utility function assigns a utility $u_i^{G,r} = C(i,r)$, where $C(i,r)$ is the number of common neighbors between i and r.

5.3 Privacy Lower Bounds

In this section we show a lower bound on the privacy parameter ϵ for any differentially private recommendation algorithm that (a) achieves a constant accuracy and (b) is based on any utility function that satisfies the exchangeability and concentration axioms, and the monotonicity property. We present tighter bounds for several concrete choices of utility functions in Section 5.3.3.

Theorem 4. *For a graph with maximum degree $d_{\max} = \alpha \log n$, a differentially private algorithm, that satisfies monotonicity and is operating on a utility function that satisfies exchangeability and concentration, and certain technical conditions ($\beta \leq n\mu(1-2\delta)$, and $\delta < 0.5$), can guarantee constant accuracy only if*

$$\epsilon \geq \frac{1}{\alpha}\left(\frac{1}{4} - o(1)\right) \tag{5.2}$$

As an example, the theorem implies that for any utility function that satisfies exchangeability and concentration (with any $\beta = O(\log n)$), and for a graph with maximum degree $\log n$, there is no 0.24-differentially private algorithm that achieves accuracy better than 0.5.

5.3.1 Proof Overview

The building blocks to proving Theorem 4 are Lemmas 8 and 9 that relate the accuracy parameter $1-\delta$ and privacy parameter ϵ first by utilizing exchangeability and monotonicity and then by incorporating the concentration axiom. We first introduce notation used in the Lemmas, state and interpret them, and then provide detailed proofs in Section 5.3.2.

Notation. Let node r be the target of recommendation. Let c be a real number in $(\delta, 1)$, and let V_{hi}^r be the set of nodes $1, \ldots, k$ each of which have utility $u_i > (1-c)u_{\max}$, and let V_{lo}^r be the nodes $k+1, \ldots, n$ each of which have utility $u_i \leq (1-c)u_{\max}$ of being recommended to target node r. Recall that $u_{\max}$ is the utility of the highest utility node. Let t be the number of edge alterations (edge additions or removals) required to turn a node with the smallest probability of

being recommended from the low utility group V_{lo}^r into the node of maximum utility in the modified graph.

The following lemma states the main trade-off relationship between the accuracy parameter $1-\delta$ and the privacy parameter ϵ of a recommendation algorithm:

Lemma 8. $\epsilon \geq \frac{1}{t}\left(\ln(\frac{c-\delta}{\delta}) + \ln(\frac{n-k}{k+1})\right)$

This lemma gives us a lower bound on the privacy guarantee ϵ in terms of the accuracy parameter $1-\delta$. Equivalently, the following corollary presents the result as an **upper bound on accuracy** that is achievable by any ϵ differential privacy preserving social recommendation algorithm:

Corollary 1. $1-\delta \leq 1 - \frac{c(n-k)}{n-k+(k+1)e^{\epsilon t}}$

Consider an example of a social network with 400 million nodes, i.e., $n = 4 \cdot 10^8$. Assume that for $c = 0.99$, we have $k = 100$; this means that there are at most 100 nodes that have utility close to the highest utility possible for r. Recall that t is the number of edges needed to be changed to make a low utility node into the highest utility node, and consider $t = 150$ (which is about the average degree in some social networks). Suppose we want to guarantee 0.1-differential privacy, then we compute the bound on the accuracy $1-\delta$ by plugging in these values in Corollary 1. We get $(1-\delta) < 0.46$. This suggests that for a differential privacy guarantee of 0.1, no algorithm can guarantee an accuracy better than 0.46.

Lemma 8 combined with the concentration axiom with parameters β and μ will yield:

Lemma 9. *For* $(1-\delta) = \Omega(1)$, *and technical conditions (*$\beta \leq n\mu(1-2\delta)$ *and* $\delta < 0.5$*):*

$$\epsilon \geq \frac{\log n - o(\log n)}{t} \tag{5.3}$$

This expression can be intuitively interpreted as follows: in order to achieve good accuracy with a reasonable amount of privacy (where ϵ is independent of n), either the number of nodes, β, that together capture a significant fraction of utility needs to be very large (i.e., $\beta = \Omega(n^d)$), or the number of steps, t, needed to bring up any node's utility to the highest utility needs to be large (i.e., $t = \Omega(\log n)$). Another way to intuitively interpret the requirement that β is large is to say that there are no good utility recommendations to begin with.

As will become clear from the proof of Lemma 9, it is also possible to obtain the same asymptotic bound under different technical conditions, hence they should be viewed as a conceptual representation encoding a reasonable set of values for β, μ, and δ, rather than hard constraints.

Lemma 9 will be used in Section 5.3.3 to prove stronger lower bounds for two well studied specific utility functions, by proving tighter upper bounds on t, which imply tighter lower bounds for ϵ.

We now proceed to prove Lemmas 8 and 9, and then use them in the proof of Theorem 4.

5.3.2 Proof Details

Before jumping into the proof, we explain the intuition for the proof technique for the lower bound on privacy in Lemma 8 using the number of common neighbors utility metric.

5.3.2.1 Proof Sketch for Lemma 8

Let r be the target node for a recommendation. The nodes in any graph can be split into two groups – V^r_{hi}, nodes which have a high utility for the target node r and V^r_{lo}, nodes that have a low utility. In the case of common neighbors utility, all nodes i in the 2-hop neighborhood of r (who have at least one common neighbor with r) can be part of V^r_{hi} and the rest – of V^r_{lo}. Since the recommendation algorithm has to achieve a constant accuracy, it has to recommend one of the high utility nodes with constant probability.

By the concentration axiom, there are only a few nodes in V^r_{hi}, but there are many nodes in V^r_{lo}; in the case of common neighbors, node r may only have 10s or 100s of 2-hop neighbors in a graph of millions of users. Hence, there exists a node i in the high utility group and a node ℓ in the low utility group such that $\Gamma = p_i/p_\ell$ is very large ($\Omega(n)$). At this point, we show that we can carefully modify the graph G by adding and/or deleting a small number (t) of edges in such a way that the node ℓ with the smallest probability of being recommended in G becomes the node with the highest utility in G' (and, hence, by monotonicity, the node with the highest probability of being recommended). By the exchangeability axiom, we can show that there always exist some t edges that make this possible. For instance, for common neighbors utility, we can do this by adding edges between a node i and t of r's neighbors, where $t > \max_i C(i, r)$. It now follows from differential privacy that

$$\epsilon \geq \frac{1}{t} \log \Gamma.$$

5.3.2.2 Detailed Proofs for the General Privacy Lower Bound

Claim 1. *Suppose the algorithm achieves accuracy of $(1 - \delta)$ on a graph G. Then there exists a node x in $V^r_{lo}(G)$, such that its probability being recommended is at most $\frac{\delta}{c(n-k)}$, i.e., $p^G_x \leq \frac{\delta}{c(n-k)}$.*

Proof. In order to achieve $(1-\delta)$ accuracy, at least $\frac{c-\delta}{c}$ of the probability weight has to go to nodes in the high utility group V^r_{hi}, and at most $\frac{\delta}{c}$ of the probability weight can go to nodes in the low utility group V^r_{lo}. Indeed, denote by p^+ and p^- the total probability that goes to high and low utility nodes, respectively, and observe that, by choice of c, V^r_{hi}, and V^r_{lo}: $p^+ u_{\max} + (1-c)u_{\max} p^- \geq \sum_i u_i p_i$. Moreover, if the algorithm achieves accuracy $(1-\delta)$ then by definition of accuracy $\sum_i u_i p_i \geq (1-\delta)u_{\max}$. Furthermore, since p^+ and p^- are probabilities, we have $p^+ + p^- \leq 1$. Combining the last three inequalities, we obtain $p^+ > \frac{c-\delta}{c}, p^- \leq \frac{\delta}{c}$. Since V^r_{lo} contains $n-k$ nodes, that means there exists a node in V^r_{lo} whose probability of being recommended is at most $\frac{p^-}{n-k}$, as desired. □

Proof of Lemma 8. Using the preceding Claim, let x be the node in G_1 that is recommended with utility of at most $\frac{\delta}{c(n-k)}$ by the privacy-preserving $(1-\delta)$-accurate algorithm. And let G_2 be the graph obtained by addition of t edges to G_1 chosen so as to turn x into the node of highest utility. By differential privacy, we have $\frac{p^{G_2}_x}{p^{G_1}_x} \leq e^{\epsilon t}$.

In order to achieve $(1-\delta)$ accuracy on G_2, at least $\frac{c-\delta}{c}$ of the probability weight has to go to nodes in the high utility group, and hence by monotonicity, $p^{G_2}_x > \frac{c-\delta}{c(k+1)}$. Combining the previous three inequalities, we obtain:

$\frac{(c-\delta)(n-k)}{(k+1)\delta} = \frac{\frac{c-\delta}{c(k+1)}}{\frac{\delta}{c(n-k)}} < \frac{p^{G_2}_x}{p^{G_1}_x} \leq e^{\epsilon t}$, hence $\epsilon \geq \frac{1}{t}\big(\ln(\frac{c-\delta}{\delta}) + \ln(\frac{n-k}{k+1})\big)$, as desired. □

Before proceeding to prove Lemma 9, we use the concentration axiom to prove the following claim:

Claim 2. *Suppose the sum of utilities of β nodes satisfying the concentration axiom is $\mu \sum_{i \in V(G)} u_i$, for some $\mu = \Omega(1)$. Then $k \leq \frac{\beta}{\mu(1-c)}$.*

Proof. Recall that by choice of c, k is the number of nodes with utility greater than $(1-c)u_{\max}$. Therefore, $k(1-c)u_{\max} \leq \sum_{i \in V(G)} u_i$.

And recall that by concentration axiom, there exist some β nodes so that the sum of their utilities is at least $\mu \sum_{i \in V(G)} u_i$, for some $\mu = \Omega(1)$. Hence, $\beta u_{\max} \geq \mu \sum_{i \in V(G)} u_i$.

Combining the inequalities of the last two paragraphs, we obtain $k \leq \frac{\beta}{\mu(1-c)}$. □

Proof of Lemma 9. We start with the expression from Lemma 8 and plug in the bound on k from Claim 2.

$\epsilon t \geq \ln(\frac{c-\delta}{\delta}) + \ln(\frac{n-k}{k+1}) \geq \ln(\frac{c-\delta}{\delta}) + \ln(\frac{n-\frac{\beta}{\mu(1-c)}}{\frac{\beta}{\mu(1-c)}+1}) = \ln(\frac{c-\delta}{\delta}) + \ln(\frac{n\mu(1-c)-\beta}{\mu(1-c)+\beta}) = \ln\big(\frac{(c-\delta)(n\mu(1-c)-\beta)}{\delta(\mu(1-c)+\beta)}\big)$

We want to show $\ln\left(\frac{(c-\delta)(n\mu(1-c)-\beta)}{\delta(\mu(1-c)+\beta)}\right) = \ln n - o(\ln n)$, which is equivalent to $\ln n - \ln\left(\frac{(c-\delta)(n\mu(1-c)-\beta)}{\delta(\mu(1-c)+\beta)}\right) = \ln\left(\frac{n\delta(\mu(1-c)+\beta)}{(c-\delta)(n\mu(1-c)-\beta)}\right) = o(\ln n)$, or

$$\forall a > 0\ \exists n_0, \text{s.t. } \forall n > n_0 : |\ln\left(\frac{n\delta(\mu(1-c)+\beta)}{(c-\delta)(n\mu(1-c)-\beta)}\right)| \leq a\ln n. \tag{5.4}$$

Suppose

$$n\mu(1-c)-\beta \geq 0, \tag{5.5}$$

and

$$\left(\frac{n\delta(\mu(1-c)+\beta)}{(c-\delta)(n\mu(1-c)-\beta)}\right) \geq 1 \tag{5.6}$$

then for statement (5.4) and, therefore, for the lemma to hold, we need
$\left(\frac{n\delta(\mu(1-c)+\beta)}{(c-\delta)(n\mu(1-c)-\beta)}\right) \leq n^a$ or
$n\delta(\mu(1-c)+\beta) \leq n^a(c-\delta)(n\mu(1-c)-\beta)$
$\beta(n^a(c-\delta)+n\delta) \leq n^a(c-\delta)n\mu(1-c) - n\delta\mu(1-c)$
$\beta \leq \frac{n^a(c-\delta)n\mu(1-c)-n\delta\mu(1-c)}{n^a(c-\delta)+n\delta} = \frac{(n^a(c-\delta)-\delta)n\mu(1-c)}{n^a(c-\delta)+n\delta}$.

Recall that c can be chosen arbitrarily from $(\delta, 1)$, so let $c = 2\delta$. Then to prove (5.4) we need to show that

$$\forall a > 0\ \exists n_0, \text{s.t. } \forall n > n_0 : \beta \leq \frac{n^a n\mu(1-2\delta) - n\mu(1-2\delta)}{n^a+n} = n\mu(1-2\delta)\frac{n^a-1}{n^a+n} \tag{5.7}$$

and that inequalities (5.5) and (5.6) are satisfied.

Inequality (5.5) is satisfied if $\beta \leq n\mu(1-2\delta)$, which holds by assumption of the lemma.

After substitution of the chosen value of c inequality (5.6) becomes: $\frac{n\delta(\mu(1-2\delta)+\beta)}{\delta(n\mu(1-2\delta)-\beta)} \geq 1$ or $\frac{n(\mu(1-2\delta)+\beta)}{n\mu(1-2\delta)-\beta} \geq 1$, which trivially holds given that (5.5) holds.

To show that inequality (5.7) is satisfied, consider two separate cases $a < 1$ and $a \geq 1$.

Recall that by assumption $\beta = o(n^d)$ for any positive d, meaning that $\forall d > 0, \text{and } \forall z > 0\ \exists n_0^{zd}, \text{s.t. } \forall n > n_0^{zd}, \beta \leq zn^d$.

If $a \geq 1$ then $\exists n_0^a, \text{s.t. } \forall n > n_0^a, \frac{n^a-1}{n^a+n} > \frac{1}{3}$. Let $d = 1, z = \frac{1}{3}\mu(1-2\delta)$. Then $\forall a > 0, n_0 = \max(n_0^a, n_0^{zd})$ we have $\forall n > n_0 : \beta \leq \frac{1}{3}\mu(1-2\delta)n < \frac{n^a-1}{n^a+n}\mu(1-2\delta)n$, as desired by (5.7).

If $a < 1$ then $\exists n_0^a, \text{s.t. } \forall n > n_0^a : n^a + 2n^{1-a} \leq n$, from which it follows that $n^{2a} + 2n \leq n^{1+a}, n^{2a} + 2n + n^{1+a} \leq 2n^{1+a}, n^a(n^a+n) \leq 2n^{1+a} - 2n, n^a \leq \frac{2n^{1+a}-2n}{n^a+n}$. Let $d = a, z = 0.5\mu(1-2\delta)$. Then $\forall a > 0, n_0 = \max(n_0^a, n_0^{zd})$ we have $\forall n > n_0 : \beta \leq 0.5\mu(1-2\delta)n^a \leq 0.5\mu(1-2\delta)\frac{2n^{1+a}-2n}{n^a+n} = n\mu(1-2\delta)\frac{n^a-1}{n^a+n}$, as desired by (5.7). This completes the proof.

It is worth noting that changes in the chosen value of c yield asymptotically the same bounds with slightly different restrictions on β, μ, and δ, hence the conditions provided in the statement of the lemma can be viewed as technical conditions encoding reasonable assumptions, rather than hard constraints needed for the lemma to hold. □

Proof of Theorem 4 (Any utility function). Recall that $d_{\max}$ denotes the maximum degree in the graph. Using the exchangeability axiom, we can show that $t \leq 4d_{\max}$ in any graph. Consider the highest utility node and node with the lowest probability of being recommended, say x and y respectively. These nodes can be *interchanged* by deleting all of x's current edges, adding edges from x to y's neighbors, and doing the same for y. This requires at most $4d_{\max}$ changes. By applying the upper bound on t in Lemma 9 we obtain the desired result. □

5.3.2.3 Non-monotone Algorithms

Our results can be generalized to algorithms that do not satisfy the monotonicity property, assuming that they only use the utilities of nodes (and node names do not matter). We omit the exact lemmas analogous to Lemmas 8 and 9 but remark that the statements and our qualitative conclusions will remain essentially unchanged, with the exception of the meaning of variable t. Currently, we have t as the number of edge additions or removals necessary to *make* the node with the smallest probability of being recommended into the node with the highest utility. We then argue about the probability with which the highest utility node is recommended by using monotonicity. Without the monotonicity property, t would correspond to the number of edge alterations necessary to *exchange* the node with the smallest probability of being recommended and the node with the highest utility. We can then use just the exchangeability axiom to argue about the probability of recommendation. Notice that this requires a slightly higher value of t, and consequently results in a slightly weaker lower bound.

5.3.3 Privacy Lower Bounds for Specific Utility Functions

In this section, we start from Lemma 9 and prove stronger lower bounds for particular utility functions using tighter upper bounds on t.

5.3.3.1 Privacy Bound for Common Neighbors

Theorem 5. *A monotone recommendation algorithm based on the number of common neighbors utility function*[7] *that guarantees any constant accuracy for a target node r has a lower bound on*

[7] satisfying the same technical conditions as in Lemma 9.

privacy given by $\epsilon \geq \frac{1}{\alpha}\big(1 - o(1)\big)$*, if node r's degree is* $d_r = \alpha \log n$*.*

Proof. Consider a graph and a target node r. The crux of the proof is to observe that we can make any node x into the highest utility node by adding $t = d_r + 2$ edges. Indeed, it suffices to add d_r edges from x to all of r's neighbors and additionally add two more edges (one each from r and x) to some node with small utility, to make x the highest utility node. This is because the highest utility node previously has had at most d_r common neighbors with r, and hence had the utility no larger than d_r. Further, adding these edges cannot increase the number of common neighbors to exceed d_r for any other node.

This bound on t combined with Lemma 9 yields the result. □

As we will show in Section 5.5, this is a very strong lower bound. Since a significant fraction of nodes in real-world graphs have small d_r (due to a power law degree distribution), we can expect no algorithm based on common neighbors utility to be both accurate on most nodes and satisfy differential with a reasonable ϵ. Moreover, this is contrary to the commonly held belief that one can eliminate privacy risk by connecting to a few high degree nodes.

Consider an example to understand the consequence of this theorem of a graph on n nodes with maximum degree $\log n$. Any algorithm that makes recommendations based on the common neighbors utility function and achieves a constant accuracy is *at best*, 1.0-differentially private. Specifically, for example, such an algorithm cannot guarantee a 0.999-differential privacy on this graph.

5.3.3.2 Privacy Bound for Weighted Paths

A natural extension of the common neighbors utility function and one whose usefulness is supported by the literature [122], is the weighted path utility function, defined as:
$\mathbf{score}(s, y) = \sum_{l=2}^{\infty} \gamma^{l-2} |paths^{(l)}_{(s,y)}|$, where $|paths^{(l)}_{(s,y)}|$ denotes the number of length l paths from s to y. Typically, one would consider using small values of γ, such as $\gamma = 0.005$, so that the weighted paths score is a "smoothed version" of the common neighbors score.

Theorem 6. *A monotone recommendation algorithm based on the weighted paths utility function*[8] *with* $\gamma < \frac{1}{\rho d_{\max} + 2}$ *that guarantees constant accuracy for a target node r has a lower bound on privacy given by* $\epsilon \geq \frac{1}{\alpha}\left(\frac{1}{\rho - 3 - \sqrt{\rho^2 - 8\rho + 8}} - o(1)\right)$*, if the degree of r,* $d_r = \alpha \log n$*,* $\rho \geq 4 + 2\sqrt{2}$*, and* $0.5(\rho - 2 - \sqrt{\rho^2 - 8\rho + 8}) d_r (d_{\max} + 1) < n - 2$*.*

[8] satisfying the same technical conditions as in Lemma 9.

For example, if $\rho = 7$, the theorem implies $\epsilon \geq \frac{1}{\alpha}\big(\frac{1}{3} - o(1)\big)$; $\rho = 8.5 \implies \epsilon \geq \frac{1}{\alpha}\big(\frac{1}{2} - o(1)\big)$; $\rho = 9.4 \implies \epsilon \geq \frac{1}{\alpha}\big(\frac{1}{1.8} - o(1)\big)$; $\rho = 14.2 \implies \epsilon \geq \frac{1}{\alpha}\big(\frac{1}{1.4} - o(1)\big)$; $\rho = 24.1 \implies \epsilon \geq \frac{1}{\alpha}\big(\frac{1}{1.2} - o(1)\big)$.

This is a stronger lower bound on ϵ than that of Theorem 4. Moreover, as $\gamma \to 0$, (and correspondingly, $\rho \to \infty$), or in other words, as the score function becomes more and more similar to the number of common neighbors, we get essentially the same bound as in Theorem 5. Hence the same example as before suggests roughly that for nodes with at most logarithmic degree, a recommendation algorithm with constant accuracy cannot guarantee anything better than constant differential privacy.

Proof of Theorem 6. Unlike in the proof of Theorem 5 for the common neighbors utility function, we cannot trivially upper bound t with d_r or even $d_{\max}$ here. The highest utility node in the original graph, say x, has at most $d_{\max}$ edges, and one could make any other node, say y have at least as much utility as x by adding the exact same edges as the ones outgoing from x. However, there could be other nodes that would now have a larger utility than both x and y since the new edges have been added and may have changed the number of paths of various lengths. Therefore, we need a more careful analysis to obtain an upper bound on t in the case of weighted neighbors utility function.

Let y be the node with the smallest probability of being recommended. We rewire the original graph G into G' as follows to make y into the highest utility node in G'. Connect both r and y to b nodes (other than r and y themselves), chosen in such a way so that these b nodes and the d_r nodes r is already connected to have no common neighbors (this can be done if $(b+d_r)d_{\max} < n-2-b-d_r$). Additionally, connect y to all of r's d_r neighbors. We have thus added $t = d_r + 2b$ edges.

Observe that utility of node y is now $u_y^{G'} \geq d_r + b$. We now bound from above the utility of any other node in the rewired graph G'.

All nodes in the new graph have degree at most $d_{\max}+2$, except nodes r and y, which have degrees of at most $d_r + b$ and $d_{\max} + d_r + b$, respectively. Therefore, the number of paths of length l for $l \geq 3$ from node r to any other node is at most $(d_r + b)(d_{\max} + d_r + b)(d_{\max} + 2)^{l-3}$. The number of paths of length 2 from r to any node except y is at most d_r due to how the b nodes to connect r and y to were chosen. Therefore, for any node z other than y, the utility $u_z^{G'} \leq d_r + \sum_{l=3}^{\infty} \gamma^{l-2}(d_r+b)(d_{\max}+d_r+b)(d_{\max}+2)^{l-3} = d_r + (d_r+b)(d_{\max}+d_r+b)\gamma\sum_{l=3}^{\infty}\gamma^{l-3}(d_{\max}+2)^{l-3} = d_r + (d_r+b)(d_{\max}+d_r+b)\gamma\sum_{l=0}^{\infty}\gamma^{l}(d_{\max}+2)^{l}$. Thus as long as $\gamma < \frac{1}{d_{\max}+2}$, then $u_z^{G'} \leq d_r + (d_r+b)(d_{\max}+d_r+b)\frac{\gamma}{1-\gamma(d_{\max}+2)}$.

For y to be the maximum utility node in the new graph G', we need for all nodes z in the graph: $u_y^{G'} > u_z^{G'}$. We now show that it will be the case if we choose $b = 0.5(\rho - 4 - \sqrt{\rho^2 - 8\rho + 8})d_r$.

Indeed, let $b = 0.5\big(\rho - 4 - \sqrt{\rho^2 - 8\rho + 8}\big)d_r$, denote $s = \sqrt{\rho^2 - 8\rho + 8}$, and recall the assumption

that $\gamma < \frac{1}{\rho d_{\max}+2}$. Then
$u_z^{G'} - d_r \leq (d_r + b)(d_{\max} + d_r + b)\frac{\gamma}{1-\gamma(d_{\max}+2)} < (d_r + b)(d_{\max} + d_r + b)\frac{\frac{1}{\rho d_{\max}+2}}{1-\frac{1}{\rho d_{\max}+2}(d_{\max}+2)} =$
$= 0.5(\rho - 2 - s)d_r\big(d_{\max} + 0.5(\rho - 2 - s)d_r\big)\frac{1}{(\rho-1)d_{\max}} \leq 0.5(\rho - 2 - s)d_r\big(d_{\max} + 0.5(\rho - 2 - s)d_{\max}\big)\frac{1}{(\rho-1)d_{\max}} = 0.5(\rho-2-s)d_r\big(1 + 0.5(\rho - 2 - s)\big)\frac{1}{(\rho-1)} = 0.5(\rho-2-s)d_r\big(0.5(\rho - s)\big)\frac{1}{(\rho-1)} =$
$0.25(\rho-2-s)d_r\big(\rho - s\big)\frac{1}{(\rho-1)} = (\rho^2 - \rho s - 2\rho + 2s - \rho s + s^2)\frac{0.25d_r}{(\rho-1)} =$
$= (\rho^2 - \rho s - 2\rho + 2s - \rho s + \rho^2 - 8\rho + 8)\frac{0.25d_r}{(\rho-1)} = (2\rho^2 - 2\rho s + 2s - 10\rho + 8)\frac{0.25d_r}{(\rho-1)} =$
$= (\rho^2 - \rho s + s - 5\rho + 4)\frac{0.5d_r}{(\rho-1)} = (\rho-1)(\rho-4-s)\frac{0.5d_r}{(\rho-1)} = (\rho-4-s)0.5d_r = b \leq u_y^{G'} - d_r$

Therefore, if $b = 0.5\big(\rho-4-\sqrt{\rho^2-8\rho+8}\big)d_r$, then for any node z in the new graph G', $u_y^{G'} > u_z^{G'}$.

Observe that to obtain the new graph, we have added $t = d_r + 2b = (\rho - 3 - \sqrt{\rho^2-8\rho+8})d_r$ edges. Substituting this value of t into Lemma 9, we obtain the desired lower bound on the privacy parameter, $\epsilon \geq \frac{\log n - o(\log n)}{t} = \frac{1}{\alpha}\left(\frac{1}{\rho-3-\sqrt{\rho^2-8\rho+8}} - o(1)\right)$. □

The bound of Theorem 6 can be significantly strengthened for particular nodes and graphs, which we omit for reasons of consistency with Theorems 4 and 5 and simplicity of interpretation. For example, it is clear from the proof that if we bound γ in terms of $d_{\max}$ and d_r, rather than only $d_{\max}$, the bound on γ may be made weaker, or, correspondingly, the bound on ϵ can be made stronger for nodes where $d_r << d_{\max}$. Furthermore, our proofs assumed a worst-case scenario, by bounding the number of paths of length l assuming that each node has a maximum possible number of outgoing edges and all those edges meaningfully contribute to the path from r to z. For practical graphs that is not the case. We will use a node-specific and graph-specific version of Corollary 1 in Section 5.5 when we assess the practical implications of Theorems 5 and 6.

5.4 Privacy-preserving Recommendation Algorithms

In this section we articulate how the known approaches to preserving privacy, the Laplace noise addition and the Exponential mechanism (Section 3.2.1), can be applied in the setting of social recommendations. We describe how, when given an input vector $\vec{u}$ of utilities, to recommend a node in a privacy-preserving manner, achieving ϵ-differential privacy, where the desired privacy guarantee of ϵ is specified by the designer. We will show experimentally in Section 5.5 that although one can, perhaps, hope to develop slightly better algorithms than the known ones for this particular problem setting, our utility loss lower bound from Section 5.3 suggests that any improvement in utility achieved would be fairly incremental. In Section 5.4.2 we describe and analyze a sampling-based approach towards making recommendations for the case when the entire utility vector may

not be known to the algorithm and only efficient sampling from it is possible.

5.4.1 Privacy-preserving Algorithms for Known Utility Vectors

Assume that given a graph and a target node, our algorithm has access to (or can efficiently compute) the utilities u_i for all other nodes in the graph. Recall that our goal is to compute a vector of recommendation probabilities p_i such that (a) $\sum_i u_i \cdot p_i$ is maximized, and (b) differential privacy is satisfied. Also recall that maximum accuracy is achieved by $\mathcal{R}_{best}$, the algorithm always recommending the node with the highest utility u_{max}. However, any ϵ-differentially private monotonic algorithm applied to a utility function that satisfies exchangeability aiming to achieve non-zero accuracy must recommend every node, even the ones that have zero utility, with a non-zero probability [150]. Therefore, in order to guarantee privacy, we need to search for algorithms that will ensure that every node has a chance of being recommended, and, in order to maximize accuracy, will give the higher utility nodes as high a chance of being recommended as possible. The Exponential mechanism and Laplace noise addition (Section 3.2) are two approaches towards producing this probability vector that is, in some way, a "smoothed version" of the utility vector.

The Exponential mechanism (Section 3.2.1) creates a smooth probability distribution from the utility vector and samples from it.

Definition 7. ***Exponential mechanism for social recommendations:*** *Given nodes with utilities* $(u_1, \ldots, u_i, \ldots, u_n)$*, algorithm* $A_E(\epsilon)$ *recommends node* i *with probability* $\frac{\exp(\frac{\epsilon}{S(u)} u_i)}{\sum_{k=1}^{n} \exp(\frac{\epsilon}{S(u)} u_k)}$*, where* $\epsilon > 0$ *is the privacy parameter, and* $S(u)$ *is the sensitivity of the utility function computed as* $S(u) = \max_r \max_{G,G':G=G'+e} ||\vec{u}^{G,r} - \vec{u}^{G',r}||_1$.

The proof that $A_E(\epsilon)$ guarantees ϵ differential privacy follows from the privacy of Exponential mechanism (see Theorem 2 due to McSherry and Talwar [134] in Section 3.2.1, using a slightly different meaning of sensitivity: [134] use sensitivity defined as a change in the function when applied to inputs that differ in a single value, whereas we consider sensitivity as a change in the function when applied to inputs that differ by an edge.)

Unlike the Exponential mechanism, the Laplace mechanism (Section 3.2.1) in this context more closely mimics the optimal mechanism $\mathcal{R}_{best}$. It first adds random noise drawn from a Laplace distribution, and, then, like the optimal mechanism, picks the node with the maximum noise-infused utility.

Definition 8. ***Laplace mechanism for social recommendations:*** *Given nodes with utilities* $(u_1, \ldots, u_i, \ldots, u_n)$*, algorithm* $A_L(\epsilon)$ *first computes a modified utility vector* $(u'_1, \ldots, u'_n)$ *as follows:*

$u_i' = u_i + x$, where x is a random variable chosen from the Laplace distribution with scale $(\frac{S(u)}{\epsilon})$ independently at random for each i. Then, $A_L(\epsilon)$ recommends node z whose noisy utility is maximal among all nodes, i.e., $z = \arg\max_i u_i'$.

The proof that $A_L(\epsilon)$ guarantees ϵ differential privacy follows from the privacy of Laplace mechanism when publishing histograms (see Theorem 1 due to [49] described in Section 3.2.1): each node can be treated as a histogram bin and u_i' is the noisy count for the value in that bin. Since $A_L(\epsilon)$ is effectively doing post-processing by releasing only the name of the bin with the highest noisy count, the algorithm remains private.

A_L as stated does not satisfy monotonicity; however, it satisfies it in expectation, which this is sufficient for our purposes, if we perform our comparisons between mechanisms and apply the bounds to A_L's expected, rather than one-time, performance.

As we will see in Section 5.5, in practice, A_L and A_E achieve very similar accuracies. The Laplace mechanism may be a bit more intuitive of the two, as instead of recommending the highest utility node it recommends the node with the highest noisy utility. It is natural to ask whether the two are isomorphic in our setting, which turns out not to be the case, as we show in Section 5.7.1 by deriving a closed form expression for the probability of each node being recommended by the Laplace mechanism as a function of its utility when $n = 2$ and comparing it with the probability of each node being recommended by the Exponential mechanism given the same utilities.

5.4.2 Sampling and Linear Smoothing for Unknown Utility Vectors

Both the differentially private algorithms we have just discussed assume the knowledge of the entire utility vector, an assumption that cannot always be made in social networks for various reasons. Firstly, computing, as well as storing the utility of n^2 pairs may be prohibitively expensive when dealing with graphs of several hundred million nodes. Secondly, even if one could compute and store them, these graphs change at staggering rates, and therefore, utility vectors are also constantly changing.

We now propose a simple algorithm that assumes no knowledge of the utility vector; it only assumes that sampling from the utility vector can be done efficiently. We show how to modify any given recommendation algorithm A, which is μ-accurate but not provably private, into an algorithm $A_S(x)$ that guarantees differential privacy, while still preserving, to some extent, the accuracy of A.

Definition 9. *Given an algorithm $A = (p_1, p_2, \ldots, p_n)$, which is μ-accurate, algorithm $A_S(x)$ recommends node i with probability $\frac{1-x}{n} + xp_i$, where $0 \leq x \leq 1$ is a parameter.*

Intuitively, $A_S(x)$ corresponds to flipping a biased coin, and, depending on the outcome, either sampling a recommendation using A or making one uniformly at random.

Theorem 7. *$A_S(x)$ guarantees $\ln(1 + \frac{nx}{1-x})$-differential privacy and $x\mu$ accuracy.*

Proof. Let $p_i'' = \frac{1-x}{n} + xp_i$. First, observe that $\sum_{i=1}^n p_i'' = 1$, and $p_i'' \geq 0$, hence $A_S(x)$ is a valid algorithm. The utility of $A_S(x)$ is $U(A_S(x)) = \sum_{k=1}^n u_k p_k'' = \sum_{k=1}^n (\frac{1-x}{n})u_k + \sum_{k=1}^n xp_k u_k \geq x\mu u_{\max}$, where we use the facts that $\sum_k u_k \geq 0$ and $\sum p_k u_k \geq \mu u_{\max}$ by assumption on A's accuracy. Hence, $U(A_S(x))$ has accuracy $\geq x\mu$.

For the privacy guarantee, note that $\frac{1-x}{n} \leq p_i'' \leq \frac{1-x}{n} + x$, since $0 \leq p_i \leq 1$. These upper and lower bounds on p_i'' hold for *any* graph and utility function. Therefore, the change in the probability of recommending i for any two graphs G and G' that differ in exactly one edge is at most:

$$\frac{p_i(G)}{p_i(G')} \leq \frac{x + \frac{1-x}{n}}{\frac{1-x}{n}} = 1 + \frac{nx}{1-x}.$$

Therefore, A_S is $\ln(1 + \frac{nx}{1-x})$-differentially private, as desired. □

Note that to guarantee ϵ-differentially privacy for $A_S(x)$, we need to set the parameter x so that $\ln(1 + \frac{nx}{1-x}) = \epsilon$, namely $x = \frac{\exp(\epsilon)-1}{\exp(\epsilon)+n-1}$.

5.5 Experiments

In this section we present experimental results on two real-world graphs and for two particular utility functions. We compute accuracies achieved by the Laplace and Exponential mechanisms, and compare them with the theoretical upper bound on accuracy (Corollary 1) that any ϵ-differentially private algorithm can hope to achieve. Our experiments suggest three takeaways: (i) For most nodes, our bounds suggest that there is an inevitable harsh trade-off between privacy and accuracy when making social recommendations, yielding poor accuracy for most nodes under reasonable privacy parameter ϵ; (ii) The more natural Laplace mechanism performs as well as the Exponential mechanism; and (iii) For a large fraction of nodes, the accuracy achieved by Laplace and Exponential mechanisms is close to the best possible accuracy suggested by our theoretical bound.

5.5.1 Experimental Setup

We use two publicly available social networks – Wikipedia vote network (G_{WV}) and Twitter connections network (G_T). While the edges in these graphs are not private, we believe that these graphs

exhibit the structure and properties typical of other private social networks.

The Wikipedia vote network (G_{WV}) [121] is available from Stanford Network Analysis Package[9]. Some Wikipedia users are administrators, who have access to additional technical features. Users are elected to be administrators via a public vote of other users and administrators. G_{WV} consists of all users participating in the elections (either casting a vote or being voted on), since inception of Wikipedia until January 2008. We convert G_{WV} into an undirected network, where each node represents a user and an edge from node i to node j represents that user i voted on user j or user j voted on user i. G_{WV} consists of 7,115 nodes and 100,762 edges, and has the maximum degree of $1,065$.

The second data set we use (G_T) is a sample of the Twitter connections network, obtained from [168]. G_{WV} is directed, as the "follow" relationship on Twitter is not symmetrical; consists of $96,403$ nodes, $489,986$ edges, and has the maximum degree of $13,181$.

Similar to Section 5.3.3 we use two particular utility functions: the number of common neighbors and weighted paths (with various values of γ), motivated both by literature [122] and evidence of their practical use by many companies [82], including Facebook [170] and Twitter [167]. For the directed Twitter network, we count the common neighbors and paths by following edges out of target node r, although other interpretations are also possible.

We select the target nodes for whom to solicit recommendations uniformly at random (10% of nodes in G_{WV} and 1% of nodes in G_T). For each target node r, we compute the utility of recommending to it each of the other nodes in the network (except those r is already connected to), according to the two utility functions. We approximate the weighted paths utility by considering paths of length up to 3. We also omit from further consideration a negligible number of the nodes that have no non-zero utility recommendations available to them. Then, fixing a desired privacy guarantee, ϵ, given the computed utility vector $\vec{u}^r$, and assuming we will make one recommendation for r, we compute the expected accuracy of ϵ-private recommendation for r. For the Exponential mechanism, the expected accuracy follows from the definition of $A_E(\epsilon)$ directly; for the Laplace mechanism, we compute the accuracy by running $1,000$ independent trials of $A_L(\epsilon)$, and averaging the utilities obtained in those trials. Finally, we use Corollary 1 to compute the theoretical upper bound we derived on accuracy achievable by any ϵ privacy-preserving recommendation algorithm. Note that in our experiments, we can compute exactly the value of t to use in Corollary 1 for a particular $\vec{u}^r$, and we are also not constrained by the technical conditions of Lemma 9, as Corollary 1 does not depend on it.

[9] http://snap.stanford.edu/data/wiki-Vote.html

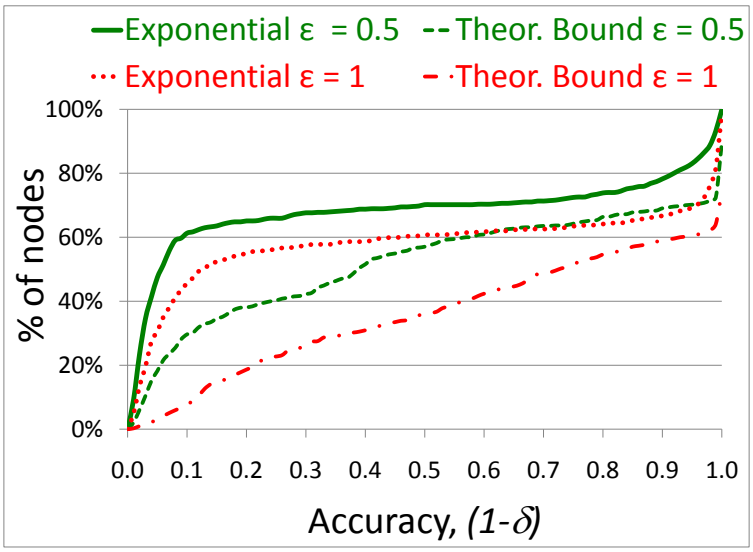
5.5.2 Results

We verified in all experiments that the Laplace mechanism achieves nearly identical accuracy as the Exponential mechanism. For readability, we include only the accuracy of Exponential mechanism in all figures, as the two curves are indistinguishable on the charts.

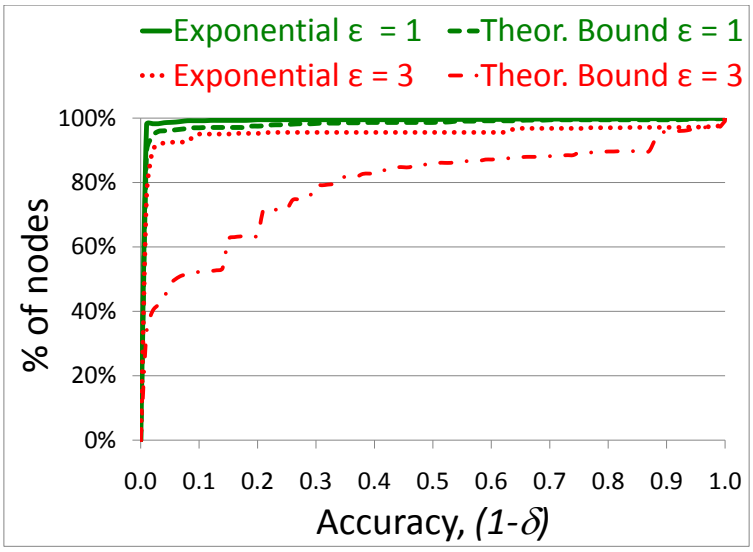
We now experimentally illustrate the best accuracy one can hope to achieve using an ϵ privacy-preserving recommendation algorithm as given by our theoretical bound of Corollary 1. We compare this bound to the accuracy of the Exponential mechanism. In the following Figures 5.1(a), 5.1(b), 5.2(a), and 5.2(b), we plot accuracy $(1 - \delta)$ on the x-axis, and the fraction of target nodes that receive recommendations of accuracy $\leq (1 - \delta)$ on the y-axis (a visualization similar to CDF plots).

5.5.2.1 Common Neighbors Utility Function

Figures 5.1(a) and 5.1(b) show the accuracies achieved on G_{WV} and G_T, resp., under the common neighbors utility function. As shown in Figure 5.1(a), for some nodes in G_{WV}, the Exponential mechanism performs quite well, achieving accuracy of more than 0.9. However, the number of such nodes is fairly small – for $\epsilon = 0.5$, the Exponential mechanism achieves less than 0.1 accuracy for 60% of the nodes. When $\epsilon = 1$, it achieves less than 0.6 accuracy for 60% of the nodes and less than 0.1 accuracy for 45% of the nodes. The theoretical bound proves that any privacy preserving algorithm on G_{WV} will have accuracy less than 0.4 for at least 50% of the nodes, if $\epsilon = 0.5$ and for at least 30% of the nodes, if $\epsilon = 1$.



(a) On Wiki vote network



(b) On Twitter network

Figure 5.1: Accuracy of algorithms using # of common neighbors utility function for two privacy settings. X-axis is the accuracy $(1-\delta)$ and y-axis is the % of nodes receiving recommendations with accuracy $\leq 1-\delta$

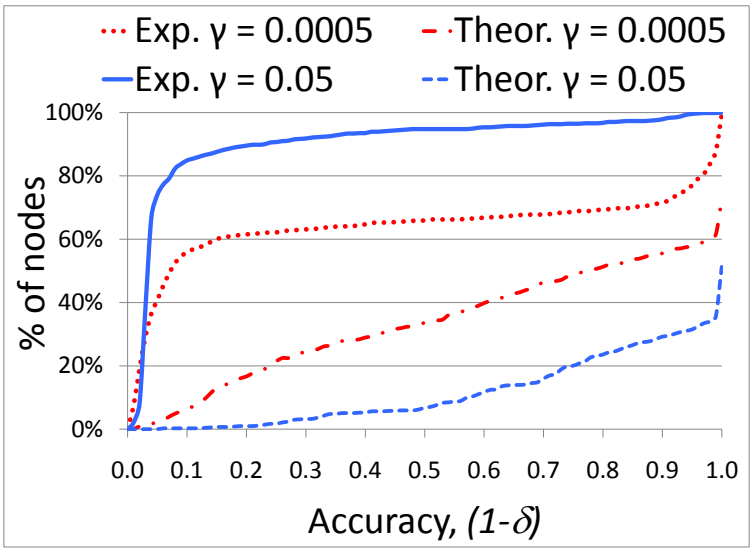
The performance worsens drastically for nodes in G_T (Figure 5.1(b)). For $\epsilon = 1$, 98% of nodes will receive recommendations of accuracy less than 0.01, if the Exponential mechanism is used. Moreover, the poor performance is not specific to the Exponential mechanism. As can be seen from the theoretical bound, 95% of the nodes will necessarily receive less than 0.03-accurate recommendations, no matter what privacy-preserving algorithm is used. Compared to the setting of $\epsilon = 1$, the performance improves only marginally even for a much more lenient privacy setting of $\epsilon = 3$ (corresponding to one graph being $e^3 \approx 20$ times more likely than another): if the Exponential mechanism is used, more than 95% of the nodes still receive an accuracy of less than 0.1; and according to the theoretical bound, 79% of the nodes will necessarily receive less than 0.3-accurate recommendations, no matter what the algorithm.

This matches the intuition that by making the privacy requirement more lenient, one can hope to make better quality recommendations for more nodes; however, this also pinpoints the fact that for an overwhelming majority of nodes, the Exponential mechanism and any other privacy preserving mechanism can not achieve good accuracy, even under lenient privacy settings.

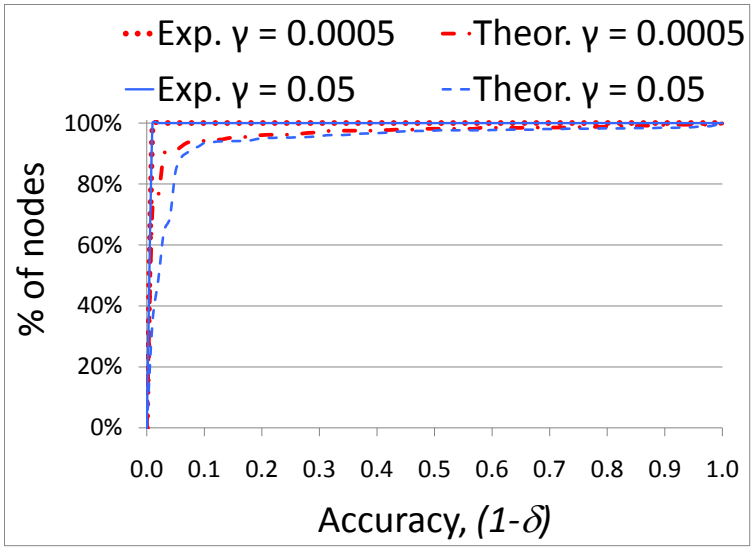
5.5.2.2 Weighted Paths Utility Function

We show experimental results with the weighted paths utility function on G_{WV} and G_T in Figures 5.2(a) and 5.2(b), respectively. As expected based on Theorem 6, we get a weaker theoretical bound for a higher parameter value of γ. Moreover, for higher γ, the utility function has a higher sensitivity, and hence worse accuracy is achieved by the Exponential and Laplace mechanisms.

The main takeaway is that even for a lenient $\epsilon = 1$, the theoretical and practical performances are both very poor (and worse in the case of G_T). For example, in G_{WV}, when using the Exponential mechanism (even with $\gamma = 0.0005$), more than 60% of the nodes receive accuracy less than 0.3. Similarly, in G_T, using the Exponential mechanism, more than 98% of nodes receive recommendations with accuracy less than 0.01. Even for a much more lenient setting of desired privacy of $\epsilon = 3$ (Figure 5.3)), the Exponential mechanism still gives more than 98% of the nodes the same extremely low accuracy of less than 0.01.



(a) Accuracy on Wiki vote network using # of weighted paths as the utility function, for $\epsilon = 1$.



(b) Accuracy on Twitter network using # of weighted paths as the utility function, for $\epsilon = 1$.

Figure 5.2: Accuracy of algorithms using weighted paths utility function. X-axis is the accuracy $(1 - \delta)$ and the y-axis is the % of nodes receiving recommendations with accuracy $\leq 1 - \delta$

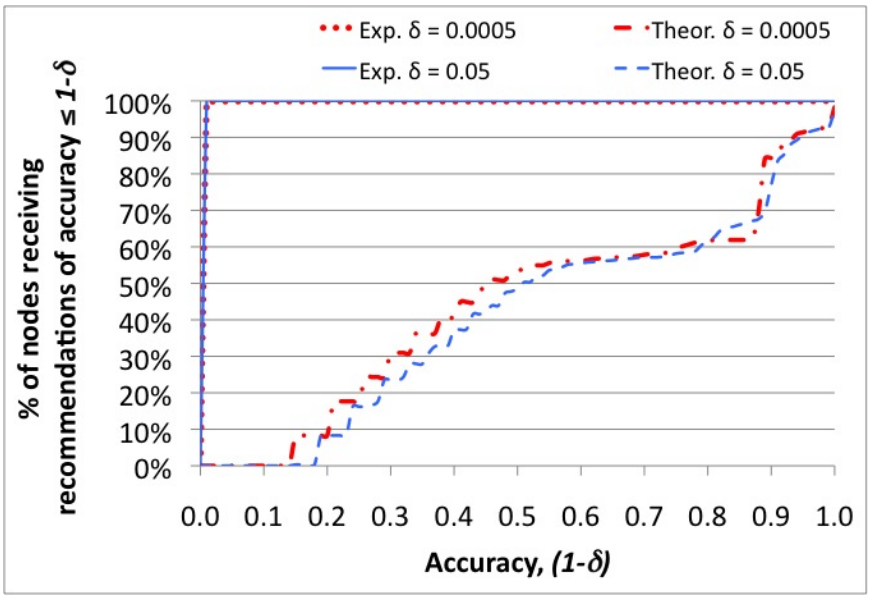


Figure 5.3: Accuracy on Twitter network using # of weighted paths as the utility function, for $\epsilon = 3$.

Even if one is able to come up with more accurate mechanisms than Exponential and Laplace, our theoretical bounds quite severely limit the best accuracy **any** privacy-preserving algorithm can hope to achieve for a large fraction of target nodes. Even for the lenient privacy setting of $\epsilon = 3$, at most 52% of the nodes in G_T can hope for an accuracy greater than 0.5 if $\gamma = 0.05, 0.005$, or 0.0005, and at most 24% of the nodes can hope for an accuracy greater than 0.9. These results show that even to ensure a weak privacy guarantee, the utility accuracy is severely compromised.

Our findings throw into serious doubt the feasibility of developing graph link-analysis based social recommendation algorithms that are both accurate and privacy-preserving for many real-world graphs and utility functions.

5.5.2.3 The Least Connected Nodes

Finally, in practice, it is the least connected nodes that are likely to benefit most from receiving high quality recommendations. However, our experiments suggest that the low degree nodes are also the most vulnerable to receiving low accuracy recommendations due to needs of privacy-preservation: see Figure 5.4 for an illustration of how accuracy depends on the degree of the node. This further suggests that, in practice, one has to make a choice between accuracy and privacy.

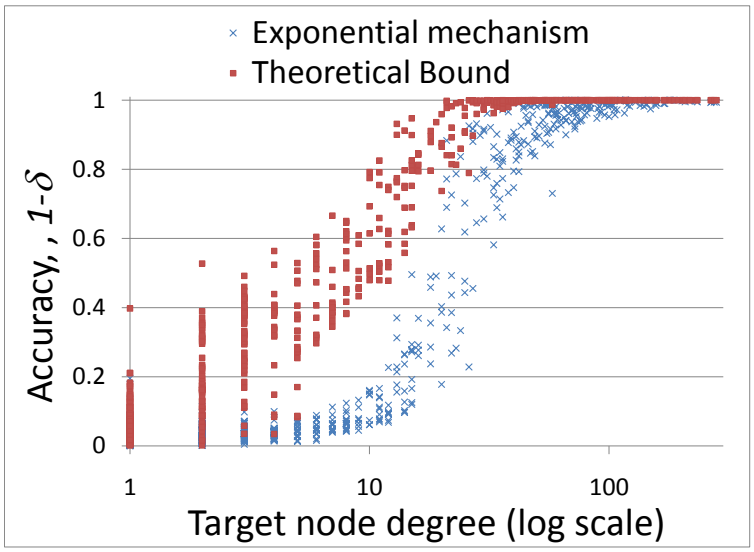


Figure 5.4: Accuracy achieved by $A_E(\epsilon)$ and predicted by Theoretical Bound as a function of node degree. X-axis is the node degree and the y-axis is accuracy of recommendation on Wiki vote network, using # common neighbors as the utility function for $\epsilon = 0.5$.

5.6 Summary and Open Questions

Several interesting questions remain unexplored. While we have analyzed privacy/utility trade-offs for two particular utility functions, it would be nice to extend our analysis to others, including those that are not purely graph-based, e.g., [17]. Also, although we have shown how to make privacy-preserving social recommendations on static data, social networks fairly rapidly change over time. Dealing with such temporal graphs and understanding their trade-offs would be very interesting.

Another interesting setting to consider is the case when only certain edges are sensitive. For example, in particular settings, only people-product connections may be sensitive but people-people connections are not, or users are allowed to specify which edges are sensitive. We believe our lower bound techniques could be suitably modified to consider only sensitive edges.

We have presented a theoretical and experimental analysis of the privacy/utility trade-ofs in personalized graph link-analysis based social recommender systems. We have shown that even when trying to make a single social recommendation the trade-offs are harsh, i.e., there is a fundamental

limit on the accuracy of link-based privacy-preserving recommendations. Although the trade-offs may be less harsh in settings where only some edges are sensitive, for other types of utility functions than the ones we considered, or for the weaker privacy definition of (ϵ, δ)-differential privacy, our analysis suggests that, at least for some utility functions, it may be more promising to look for non-algorithmic solutions for enabling social recommendations that preserve privacy. For example, a promising direction is development of systems that allow users to effortlessly specify which of their connections, purchases, and likes are sensitive, and which they are comfortable sharing with the service aiming to improve recommendations using social data, and then using algorithms that rely only on the approved connections.

5.7 Miscellaneous Technical Details

5.7.1 Comparison of Laplace and Exponential Mechanisms

Although we have observed in Section 5.5 that the Exponential and Laplace mechanisms perform comparably and know anecdotally that the two are used interchangeably in practice, the two mechanisms are not equivalent.

To show that, we compute the probability of each node being recommended by each of the mechanisms when $n = 2$, using the help of the following Lemma:

Lemma 10. *Let u_1 and u_2 be two non-negative real numbers and let X_1 and X_2 be two random variables drawn independently from the Laplace distribution with scale $b = \frac{1}{\epsilon}$ and location 0. Assume wlog that $u_1 \geq u_2$. Then*

$$Pr[u_1 + X_1 > u_2 + X_2] = 1 - \frac{1}{2}e^{-\epsilon(u_1 - u_2)} - \frac{\epsilon(u_1 - u_2)}{4e^{\epsilon(u_1 - u_2)}}$$

To the best of our knowledge, this is the first explicit closed form expression for this probability (the work of [144] gives a formula that does not apply to our setting)[10].

Proof. Let $\phi_X(u)$ denote the characteristic function of the Laplace distribution, it is known that $\phi_X(u) = \frac{1}{1+b^2u^2}$. Moreover, it is known that if X_1 and X_2 are independently distributed random variables, then

$$\phi_{X_1+X_2}(u) = \phi_{X_1}(u)\phi_{X_2}(u) = \frac{1}{(1 + b^2u^2)^2}$$

[10]We thank Sergey Melnik of Progmestars for contributing the technical ideas used in the proof.

Using the inversion formula, we can compute the pdf of $X = X_1 + X_2$ as follows:

$$f_X(x) = F'_X(x) = \frac{1}{2\pi} \int_{-\infty}^{\infty} e^{-iux} \phi_X(u) du$$

For $x > 0$, the pdf of $X_1 + X_2$ is $f_X(x) = \frac{1}{4b}(1 + \frac{x}{b})e^{-\frac{x}{b}}$ (adapting formula 859.011 of [42]) and the cdf is $F_X(x) = 1 - \frac{1}{4}\epsilon e^{-\epsilon x}(\frac{2}{\epsilon} + x)$.

Hence $Pr[u_1 + X_1 > u_2 + X_2] = Pr[X_2 - X_1 < u_1 - u_2] = 1 - \frac{1}{4}\epsilon e^{-\epsilon(u_1-u_2)}(\frac{2}{\epsilon} + (u_1 - u_2)) = 1 - \frac{1}{2}e^{-\epsilon(u_1-u_2)} - \frac{\epsilon(u_1-u_2)}{4e^{\epsilon(u_1-u_2)}}$ □

It follows from Lemma 10 and the definition of the mechanisms in Section 5.4.1 that when $n = 2$, and the node utilities are u_1 and u_2 (assuming $u_1 \geq u_2$ wlog), the Laplace mechanism will recommend node 1 with probability $1 - \frac{1}{2}e^{-\epsilon(u_1-u_2)} - \frac{\epsilon(u_1-u_2)}{4e^{\epsilon(u_1-u_2)}}$, and the Exponential mechanism will recommend node 1 with probability $\frac{e^{\epsilon u_1}}{e^{\epsilon u_1}+e^{\epsilon u_2}}$. The reader can verify that the two are not equivalent through value substitution.

Chapter 6

Social Graph Visibility

A major part of the value of participating in an online social network for a user lies in the ability to leverage the structure of the *social network graph*. For example, in the case of LinkedIn, an online network of professionals, each connection signifies a professional relationship between two individuals, such as having worked together in the past. If an individual has access to their connections, and their connections' connections, and so on, this extended network enables professional networking at an unprecedented scale, through ability to find potential collaborators, clients, employers, and subject experts and to be introduced to them through a chain of mutually trusting individuals [11]. Similarly, in the case of Facebook, an ability to assess a relation between oneself and others in the social graph (measured, for example, using a weighted combination of the number of common friends, friends-of-friends, and so on), could enable more trusting transactions between otherwise unacquainted individuals, e.g., for the purposes of subletting an apartment, online dating, or weighing the applicability of a product review. In all these cases, the larger the local snapshot of the social network that an individual can access, the more useful the social network is to him.

On the other hand, the graph link information is a valuable asset for the social network owner, one that they have a business interest to protect from malicious adversaries and competitors. An illustrative example of the value of link information to social network owners and users is an ongoing battle between Facebook and Google focused on preventing the competitor's service from seamlessly importing the friends or connections the user has established in their online service [62, 92, 166]. Hence, in online social networks, a user is typically permitted only limited access to the link structure. For example, a LinkedIn user can only see the profiles and friend lists of his friends and the profiles of friends of friends. The limit on the extent of access to the link structure is in effect not only in

the user interface, but also in the API functionality made available to third-party developers.

In this chapter we consider a privacy threat to a social network in which the goal of an attacker is to obtain knowledge of a significant fraction of the links in the network. We focus on a particular threat model, in which an attacker, whose goal is to ascertain a significant fraction of the links in a network, obtains access to parts of the network by gaining access to the accounts of some select users. This is done either maliciously by breaking into user accounts or by offering each user a payment or a service in exchange for their permission to view their neighborhood of the social network. Both scenarios are realistic and common in practice. Online social networks, such as LiveJournal and Facebook, regularly experience successful account hijacking attempts [36, 99, 113]. Users routinely voluntarily grant access to their friends list in exchange for services when using applications developed by third parties on the developer platforms provided by the social networks (for example, as of 2011, more than 95% of Facebook users have used at least one application built on Facebook Platform[1]). Although the third-party developer may not be able to store and re-purpose the obtained snippets of social graph according to the terms of service limitations, such restrictions are difficult to enforce in practice.

We formalize the typical social network interface and the information about graph links that it provides to its users in terms of lookahead. We classify possible strategies that an adversary can use for choosing users whose accounts to target in order to obtain local snapshots of the graph, such as targeting users with the most connections, those who are likely to give the most incremental gain, or arbitrary ones. For each of the strategies of user targeting, using a real-world social network, we analyze experimentally the difficulty of obtaining a large portion of the social network graph depending on lookahead. For two of the strategies, we provide an explicit mathematical analysis for the relationship between lookahead and fraction of users whose accounts need to be compromised in order to obtain a large portion of the social graph, assuming that the social network is formed using the preferential attachment graph model.

Our analysis is the first step towards helping social network owners make quantitative trade-offs between utility they provide to the users through increasing the lookahead and the privacy threat this lookahead choice poses to their business from competitors [126].

In Section 6.1, we discuss related work on privacy in social networks and models of social network graphs. Section 6.2 lays out a formal model of the kind of attacks we consider and the goal of the attacker. We present experimental results of the success of different attack strategies on both simulated and real world social network graphs in Section 6.3, and present a rigorous theoretical analysis

[1] https://www.facebook.com/note.php?note_id=171817642885051

of the attacks' potential for success in Section 6.4. We conclude in Section 6.5 with recommendations of actions for web service providers that would preserve user privacy.

6.1 Related Work

There has been much recent interest in anonymized social network graph releases. Backstrom et. al. [16] consider a framework where a social network owner announces the intention to release an anonymized version of a social network graph, i.e., a copy where true usernames are replaced with random ids but the network structure is unchanged, and the goal of an attacker is to uniquely identify the node that corresponds to a real world entity in this anonymized graph. They show that, if given a chance to create as few as $\Theta(\log(n))$ new accounts in the network prior to its anonymized release, an attacker can efficiently recover the connections between any $\Theta(\log^2(n))$ nodes chosen a-priori. This is achieved by first finding the new accounts that the attacker inserted into the network and working through the connections established between the attacker's accounts and the chosen targets to identify the targets. In [80], the authors experimentally evaluate how much background information about the structure of the neighborhood of an individual would be sufficient for an attacker to uniquely identify the individual in such an anonymized graph. In [197] the emphasis is on protecting the types of links associated with individuals in an anonymized release. Simple edge-deletion and node-merging algorithms are proposed to reduce the risk of sensitive link disclosure. [198] and [124] pursue the question of privacy as it relates to social networks from various other perspectives.

The problem we study is different from the anonymized social network data release and re-identification work of [16, 79, 148, 197]. Rather than searching for ways to release the social network while protecting the privacy of individuals in it or analyzing the barriers for such a release, we consider the scenario in which the social network owner would like to protect the privacy of its business by keeping the entire structure of the graph hidden from any one entity while providing utility to its users by giving them partial access to the link structure of the graph. This distinction of whose privacy we are trying to protect - the social network owner's rather than the users', is also the reason that in this chapter we do not utilize the definition of differential privacy.

There has been considerable theoretical work in modeling the structure and evolution of the web graph and social networks, some of which we utilize in our theoretical analysis. In [20] and [116] the preferential attachment model and the copying model are introduced as generative models for the web graph. Many variations and extensions of these models have been proposed, such as [29]

and [37]. It has been observed that social networks are subject to the small-world phenomenon [104] and models such as [189] have been proposed to account for it. The model of [120] aims to account for all of the commonly found patters in graphs. The common theme in this research is a search for a random process that models how users establish links to one another. The various models succeed to differing extents in explaining certain properties of the web graph and social networks observed in practice.

In the attack strategies that we consider, the effectiveness of the strategies is likely to depend on the underlying social graph and the degree distribution of its nodes, which is commonly known to be close to power law [41, 188]. In our theoretical analysis of the effectiveness of an attack, we use the configuration model of [24] and [6] that guarantees a power law distribution. Unlike the evolutionary models such as preferential attachment, this model does not consider the process by which a network comes to have a power law degree sequence; rather, it takes the power law degree distribution as a given and generates a random graph whose degree distribution follows such a power law (specifics of graph generation according to this model are described in Section 6.3.1.1). We could also use the preferential attachment or copying models for analysis, but a static model such as [24] or [6] suffices for our purpose and allows for simpler analysis.

6.2 Preliminaries and the Formal Problem Model

We now describe the problem statement formally. We first define the primary goal of the privacy attack considered and a measure of the extent to which an adversary achieves this goal (Section 6.2.1); then discuss the knowledge of social networks available to users, and thus adversaries (Section 6.2.2); finally, we list possible attack strategies (Section 6.2.3).

We view a social network as an undirected graph $G = (V, E)$, where the nodes V are the users and the edges E represent connections or interactions between users. Even though some online social networks, such as LiveJournal, allow one-directional links, many others, and especially those where the link information is sensitive and subject to privacy considerations, such as LinkedIn and Facebook, require mutual friendship. In those networks links between users are naturally modeled as undirected edges, and thus we focus on undirected graphs in our discussion and analysis.

6.2.1 Attack Goal and Effectiveness Measure

We consider privacy attacks whose primary goal is to discover the link structure of the network. Knowledge of the entire network is superior to knowledge of connections of a subset of individual users because it allows seamless application of commonly used graph mining algorithms, such as computation of the shortest path between two people, clustering, or study of diffusion processes. We measure an attack's effectiveness using the notion of *node coverage,* or simply *coverage*, which measures the amount of network graph structure exposed to the attacker.

Definition 10 (**Node Coverage**)**.** *The fraction of nodes whose entire immediate neighborhood is known.*

One may also consider measuring an attack's effectiveness using a notion of *edge coverage,* defined in one of the following ways:

1. *The fraction of edges known to the attacker among all edges that exist in the graph.* This notion of edge coverage does not account for the attacker's knowledge about non-existing edges, and, therefore, is not a comprehensive view of an attacker's knowledge.

2. *Among all pairs of users, the fraction of pairs between which the attacker knows whether or not an edge exists.*
   As will become clear in the following sections, our definition of node coverage is more sensible for the attack strategies we consider and implies the knowledge of edge coverage under this definition. Thus, throughout the chapter we will use node coverage as the primary measure of an attack's effectiveness.

6.2.2 The Network through a User's Lens

An online social network has the flexibility to choose the extent to which links in the network are made visible to its users through the service's interface, and this choice may depend both on how sensitive the links themselves are and on how protective the service is about the links being obtained by other entities. For example, LinkedIn allows a user to see all edges incident to oneself, as well as all edges incident to one's friends. We formalize such choices in the social network's interface using the notion of *lookahead* that, intuitively, measures the distance in the graph that is visible to a user beyond his own connections. We say that the social network has lookahead of 0 if a user can see exactly who he links to; it has lookahead 1 if a user can see exactly the friends that he links to as well as the friends that his friends link to; and so on. In general,

Definition 11 (**Lookahead**). *We say that the social network has lookahead ℓ if a user can see all of the edges incident to the nodes within distance ℓ from him.*

Using this definition, LinkedIn has lookahead 1. In terms of node coverage, a lookahead of ℓ means that each node covers all nodes within distance ℓ from it; nodes at distance $\ell + 1$ are *seen* (i.e., their existence is known to the user), but not *covered* (i.e., their connections are not known to the user).

There are other variations on the type of access that a user can have to the social graph structure. For example, some networks allow a user to see the shortest path between himself and any other user, some display the path only if it is relatively short, some only display the length of the shortest path, and others let the user see the common friends he has with any other user. For simplicity, we do not incorporate these additional options into our model, but observe that the presence of any of them reduces the difficulty of discovering the entire link structure, thereby strengthening our results.

In addition to the connection information, a typical online social network also provides a search interface, where people can search for users by username, name or other identifying information such as email or school or company affiliation. The search interface returns usernames of all users who satisfy the query, often with the numbers of friends of those users, i.e., the degrees of the nodes corresponding to those users in the social network graph, G. LinkedIn is an example of a social network that allows such queries and provides degree information.

We formalize the most common aspects of social network interfaces that may be leveraged by attackers to target specific user accounts using the following functions:

- *neighbors(username, password, ℓ)*: Given a username with proper authentication information, return all users within distance ℓ and all edges incident to those users in the graph G;
- *exists(username)*: Given a username, return whether the user exists in the network;
- *degree(username)*: Given a username, return the degree (number of friends) of the user with that username. Note that *degree(username)* implies *exists(username)*;
- *userlist()*: Return a list of all usernames in the network.

In the above, only *neighbors()* requires authentication information, all other functions are publicly available. A social network might expose some or all of these functions to its users. For example, LinkedIn provides *neighbors(username, password, ℓ)* for $\ell = 0$ or 1, but not for $\ell > 1$; it also provides *exists(username)* and *degree(username)*. Most social networks do not expose *userlist()*

directly; however, an attacker may be able to generate a near complete user list through other functionalities provided by the network such as fuzzy name search or public profiles.

A particular network may expose only a subset of the above functions and even if all functions are available, their costs may vary greatly. Therefore, when we discuss attack strategies in the next section we list the functions required by each strategy, and when we evaluate and compare strategies there is a trade-off between the effectiveness of an attack and the complexity of the available interface it requires.

6.2.3 Possible Attack Strategies

We say that a node is ***covered***, if and only if the attacker knows precisely which nodes it is connected to and which nodes it is not connected to. We call the users whose accounts the attacker had gained access to ***bribed*** users. Thus, each time the attacker gains access to or bribes a user account, he immediately covers all nodes that are at a distance of no more than the lookahead ℓ enabled by the social network.

In order to understand the privacy/utility trade-offs when choosing the lookahead, we need to study not only how an attack's success or attained node coverage varies depending on lookahead, but also how it varies based on the power of the attacker to select nodes to bribe. We now list the strategies an attacker can use for bribing nodes in decreasing order of information needed for the attacker to be able to implement them, and study the success of attacks following these strategies for various settings of the lookahead both experimentally and theoretically in Sections 6.3 and 6.4.

Benchmark-Greedy: From among all users in the social network, pick the next user to bribe as the one whose perspective on the network will give the largest possible amount of new information. More formally, at each step the attacker picks the node covering the maximum number of nodes not yet covered. For $\ell \leq 1$ this can be implemented if the attacker can access the degrees of all users in the network. However, for $\ell > 1$ it requires that for each node the attacker has access to all usernames covered by that node, which is not a primitive that we consider available to the attacker. Thus this strategy serves as a benchmark rather than as an example of a feasible attack – it is the optimal bribing algorithm that is computationally feasible when given access to the entire graph G. Note that by reduction to set cover, finding the optimal bribing set for a given G is NP-hard, thus the best polynomial-time (computationally feasible) approximation algorithm is the greedy algorithm described.

Requires: G;

Heuristically Greedy: Pick the next user to bribe as the one who can offer the largest possible amount of new information, according to some heuristic measure. The heuristic measure is chosen so that the attacker does not need to know G to evaluate it. In particular, we consider the following strategy:

- **Degree-Greedy**: Pick the next user to bribe as the one with the maximum "unseen" degree, i.e., its degree according to the *degree(username)* function minus the number of edges incident to it already known by the adversary.
  Requires: *neighbors(username, password, ℓ), degree(username), userlist();*

Highest-Degree: Bribe users in the descending order of their degrees.
Requires: *neighbors(username, password, ℓ), degree(username), userlist();*

Random: Pick the users to bribe at random. Variations could include picking the users uniformly at random, with probability proportional to their degrees, and so on. In particular, we study one strategy in this category:

- **Uniform-Random**: Pick the users to bribe uniformly at random.
  Requires: *neighbors(username, password, ℓ), userlist();*

Crawler: This strategy is similar to the Heuristically Greedy strategy, but the attacker chooses the next node to bribe only from the nodes already seen (within distance $\ell+1$ of some bribed node). We consider one such strategy:

- **Degree-Greedy-Crawler**: From among all users already seen, pick the next user to bribe as the one with the maximum unseen degree.
  Requires: *neighbors(username, password, ℓ), degree(username)*;

Note that the **Degree-Greedy-Crawler** and **Uniform-Random** strategies are very easily implementable in practice on most social networks, since they do not require any knowledge of nodes that are not within the neighborhood visible to the attacker.

6.3 Experiment-based Analysis

We present experimental results from the application of the strategies from Section 6.2.3 to both synthetic and real world social network data. At a high level, our experiments explore the fraction, f, of nodes that need to be bribed by an attacker using the different bribing strategies in order to

achieve $1 - \varepsilon$ node coverage of a social network with lookahead ℓ. Our experimental results show that the fraction of users an attacker needs to bribe in order to acquire a fixed coverage decreases exponentially with increase in lookahead. In addition, this fraction corresponds to a fairly small number of users from the perspective of practical attack implementation, indicating that several of the attack strategies from Section 6.2.3 are feasible to implement in practice and will achieve good results, especially for lookaheads ≥ 2.

We implemented and evaluated the following five strategies, ordered in the decreasing order of complexity of the social network interface needed for them to become feasible: **Benchmark-Greedy** (abbreviated as **Benchmark**); **Degree-Greedy** (abbrev. as **Greedy**); **Highest-Degree** (abbrev. as **Highest**); **Uniform-Random** (abbrev. as **Random**); **Degree-Greedy-Crawler** (abbrev. as **Crawler**).

6.3.1 Results on Synthetic Data

6.3.1.1 Generating Synthetic Graphs

In order to measure the effectiveness of the different attack strategies, we generate random graphs with power-law degree distributions and apply our strategies to them. Following the motivation of Section 6.1, we use the configuration model of [6] to generate the graphs. The model essentially generates a graph that satisfies a given degree distribution, picking uniformly at random from all such graphs.

More specifically, let n be the total number of nodes in G, α ($2 < \alpha \leq 3$, [35]) be the power law parameter; and $d_{\min}$ and $d_{\max}$ be the minimum and maximum degree of any node in the graph, respectively. First, we generate the degrees of all the nodes $d(v_i), i = 1, \ldots, n$ independently according to the distribution $\Pr[d(v_i) = x] = C/x^{\alpha}, d_{\min} \leq x \leq d_{\max}$, where C is the normalizing constant. Second, we consider $D = \sum d(v_i)$ minivertices which correspond to the original vertices in a natural way and generate a random matching over D. Finally, for each edge in the matching, we construct an edge between corresponding vertices in the original graph. As a result, we obtain a random graph with a given power-law degree distribution. The graph is connected almost surely [64]. The graph has a few multi-edges and self-loops that we remove in our experiments, without affecting the power law degree distribution.

Furthermore, following the practice of [137], we cap $d_{\max}$, the maximum number of connections that a user may have, at $\sqrt{n}$, reflecting the fact that in a large enough social network, a single person, even a very social one, cannot know a constant fraction of all users.

We denote the fraction of nodes bribed by f, the number of nodes bribed by $k = fn$, and the coverage achieved by $1 - \varepsilon = \frac{number\ of\ nodes\ covered}{n}$.

6.3.1.2 Comparison of Strategies

We analyze the relative performance of five of the strategies proposed in Section 6.2.3 on random power-law graphs with $n = 100,000$ nodes, $\alpha = 3$, and $d_{\min} = 5$. We run each strategy on 10 power-law graphs generated as described in Section 6.3.1.1, with the aim of achieving coverage of 0.5 through 0.99. For each strategy, we average across the experimental runs the fraction of nodes that need to be bribed with that strategy in order to achieve the desired coverage. This gives us f as a function of $1 - \varepsilon$ for each strategy. We present the results for lookaheads 1 and 2 in Figure 6.1.

The experimental results show that **Benchmark** has the best performance, i.e., to achieve a fixed coverage of $1 - \varepsilon$, **Benchmark** needs to bribe fewer nodes than any other strategy. However, as mentioned previously, **Benchmark** is not feasible to implement in practice because it requires knowledge of the entire graph structure, and so it can only serve as a benchmark upper bound on how good any given strategy can be.

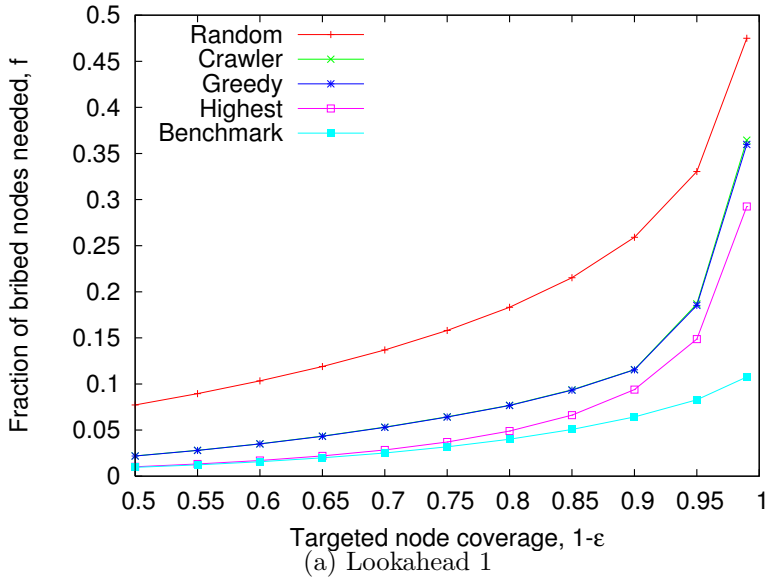
Some of the other observations we make are that **Highest** and **Benchmark** perform almost equally well when the desired coverage is less than 90%. However, the performance of **Highest** deteriorates as the lookahead increases and desired coverage increases.

Somewhat surprisingly, we find that **Greedy** performs worse than **Highest** while **Greedy** and **Crawler** perform equally well. Not surprisingly, **Random** performs the worst out of all the strategies.

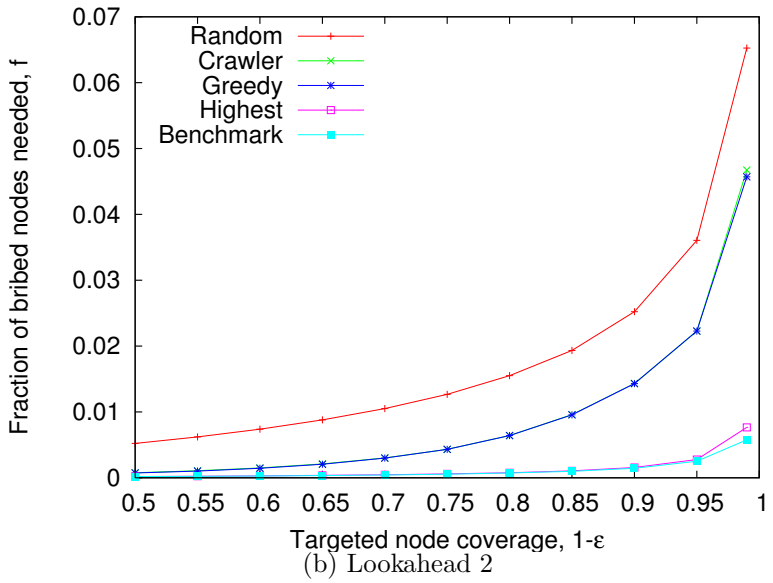
We choose the following three strategies to analyze in more detail and show that they can pose serious threats to link privacy: **Highest** and **Crawler** as a measure of performance of somewhat sophisticated yet still implementable strategies; and **Random** as the most easily implementable attack strategy that can serve as a lower bound on how well other strategies can work.

6.3.1.3 Dependence on the Number of Users

We analyze how performance of a bribing strategy changes with an increase in the number of nodes in the graph. We observe that the number of nodes k that need to be bribed using the **Highest** strategy in order to achieve a fixed coverage of $1 - \varepsilon$ is linear in the size of the network, for various values of ε. We illustrate it in Figure 6.2 for lookahead of 2. Since **Highest** has the best performance among all the suggested realistically implementable strategies, this implies that k is linear in n for



(a) Lookahead 1



(b) Lookahead 2

Figure 6.1: **Comparison of attack strategies on synthetic data.** Fraction of nodes that needs to be bribed depending on the coverage desired and bribing strategy used, for lookaheads 1 and 2. $n = 100,000$, $\alpha = 3$, and $d_{\min} = 5$. The lines for **Crawler** and **Greedy** are nearly identical and hence hardly distinguishable.

other strategies as well. However, it is worth observing that the slope of the linear function is very small, for all ε not very close to 1. As discussed in the next section, this makes all of the strategies a realistic threat at lookaheads greater than 1.

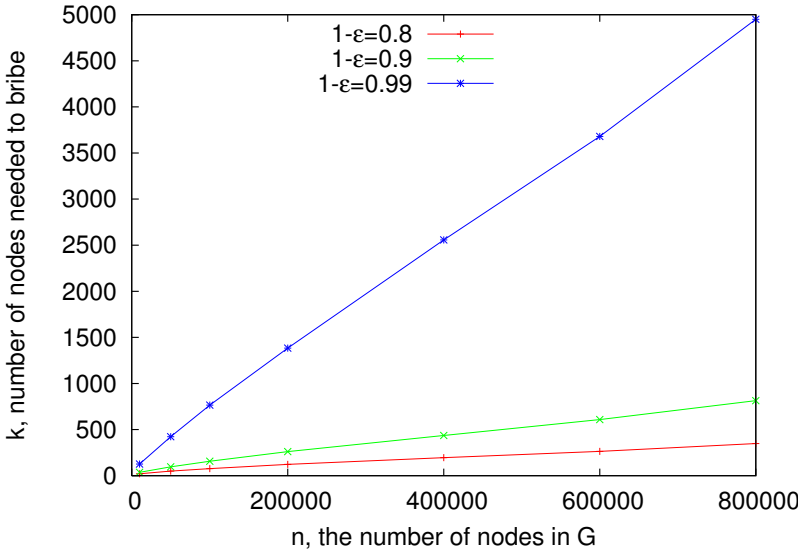


Figure 6.2: **Number of nodes that need to be bribed for graphs of size n using Highest with lookahead 2 for coverage** $0.8, 0.9, 0.99$**.**

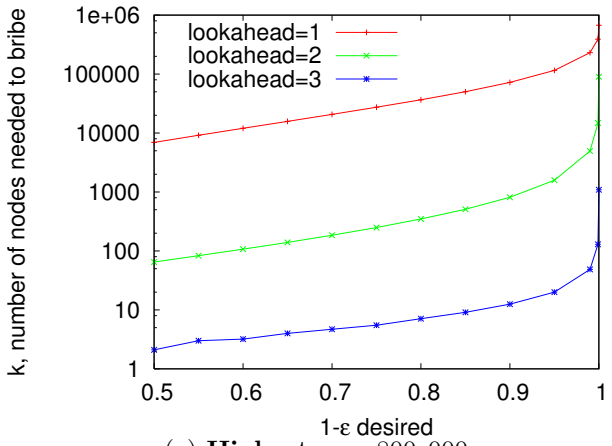
6.3.1.4 Dependence on Lookahead

The performance of all strategies substantially improves with increase in lookahead. Consider, for example, the performance of the **Highest** strategy, plotted in Figure 6.3 (a), and also detailed in Table 6.1.

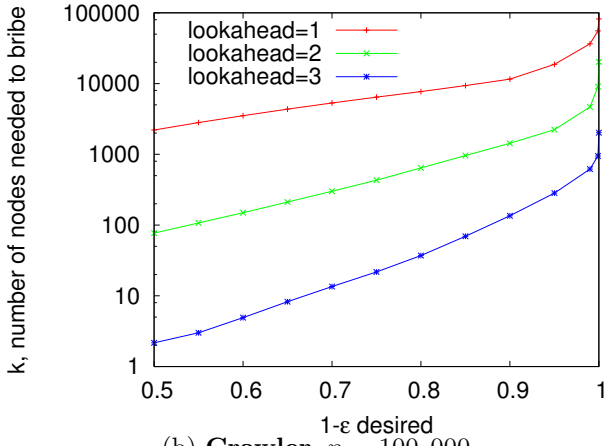
1-ε	f_1/f_2	f_2/f_3
0.7	112.3	39.3
0.8	105.0	49.1
0.9	88.6	65.1
0.95	73.1	79.0
0.99	46.6	101.7

Table 6.1: Factors of improvement in performance of **Highest** strategy with increases in lookahead. f_i - fraction of nodes that needs to be bribed to achieve $1 - \varepsilon$ coverage when lookahead is i.

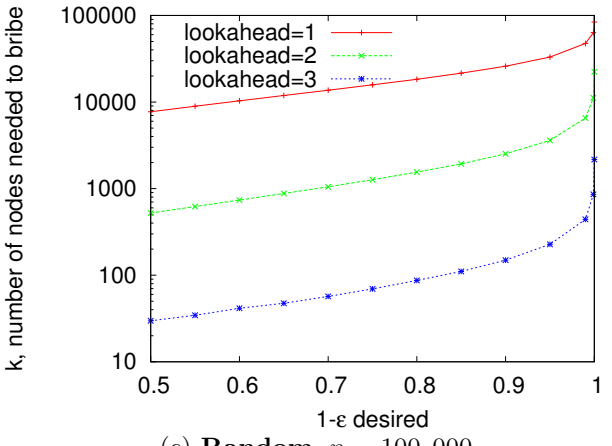
With each increase in lookahead, the number of nodes k that need to be bribed in order to achieve the same $1 - \varepsilon$ coverage decreases by two orders of magnitude. In an $800,000$-user social



(a) **Highest**, $n = 800,000$



(b) **Crawler**, $n = 100,000$



(c) **Random**, $n = 100,000$

Figure 6.3: **Effect of lookahead on attack difficulty on synthetic data.** The number of nodes needed to bribe to achieve $1 - \varepsilon$ coverage with various lookaheads, using **Highest** and **Crawler** strategies, respectively. The y axis is log scale.

network, **Highest** needs to bribe 36,614 users in order to achieve a 0.8 coverage in a network with lookahead 1, but in the network of the same size with lookahead 2 **Highest** needs to bribe 348 users to achieve the same coverage, and only 7 users, if the lookahead is 3. In other words, the fraction of nodes that need to be bribed to achieve fixed coverage decreases exponentially in the lookahead, making the **Highest** strategy attack a feasible threat at lookahead 2 in social networks with under 1 million users, and a feasible threat at lookahead 3 in social networks with as many as 100 million users.

We observe a similar exponential decrease with increase in lookahead in the number of nodes that need to be bribed for **Crawler** (Figure 6.3 (b)) and for **Random** (Figure 6.3 (c)).

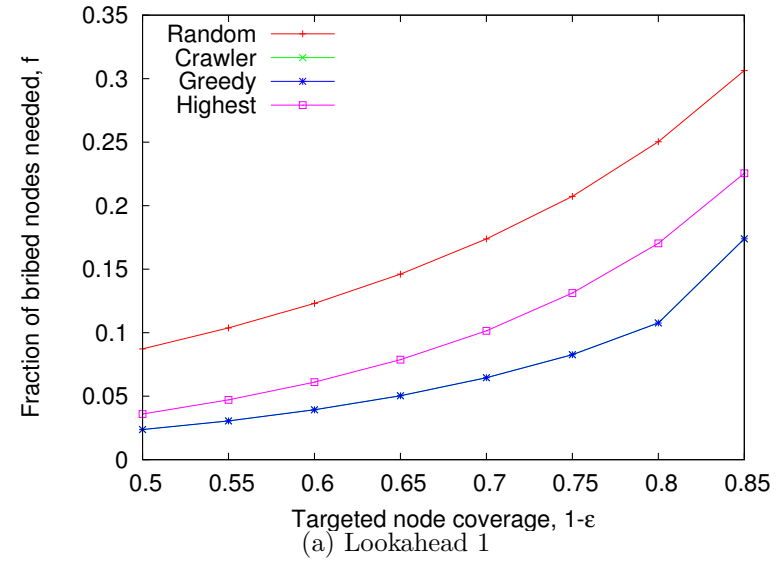
6.3.2 Results on Real Data

As we felt that attacking accounts of LinkedIn users with a goal of recovering the network's structure would be inappropriate as a research exercise, we used the LiveJournal friendship graph, whose link structure is readily available, instead as a proxy. We crawled LiveJournal using the friends and friend-of listings to establish connections between users and extracted a connected component of 572,949 users.

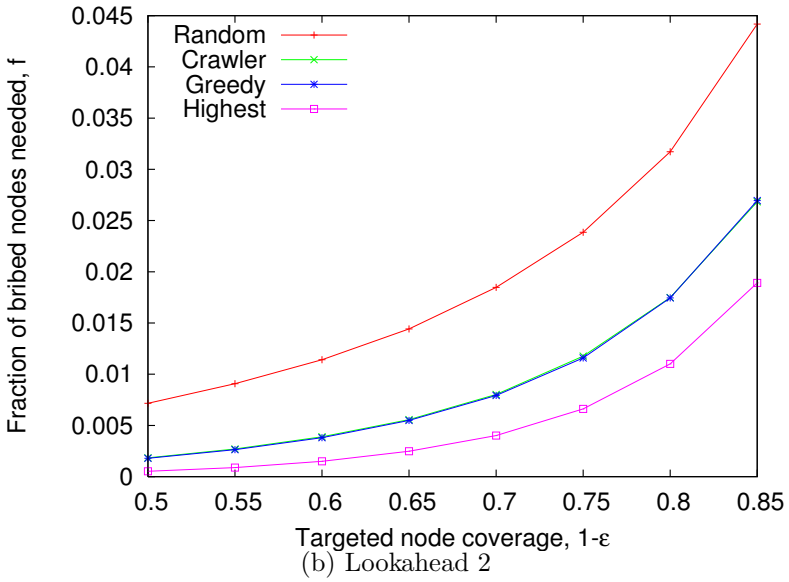
The obtained LiveJournal graph has an average degree of 11.8, $d_{\min} = 1, d_{\max} = 1974, \alpha = 2.6$. The obtained $d_{\max}$ is higher than the one assumed by our synthetic model, but the LiveJournal graph contained only 12 nodes with degrees higher than $\sqrt{572,949}$.

6.3.2.1 Comparison of Strategies

Analogous to our discussion in Section 6.3.1.2 we compare the performance of the different bribing strategies on the LiveJournal graph at lookaheads of 1 and 2 in Figures 6.4 (a) and (b). The relative performance of the different strategies is the same as on the synthetic data, with the exception of **Highest** performing worse than **Crawler** and **Greedy** at lookahead 1. The **Crawler** and **Greedy** strategies also perform better on real data than on the synthetic data. Our intuition is that these differences are due to the disparities between properties of the graphs generated using the theoretical model and the real social network. The real social network graphs tend to contain a larger number of triangles than the graphs generated using the theoretical model (i.e., in practice, conditioned on edges (a, b) and (b, c), the edge (a, c) is more likely than random [189]), with this local property likely leading to the **Crawler** and **Greedy** strategies being more effective.



(a) Lookahead 1

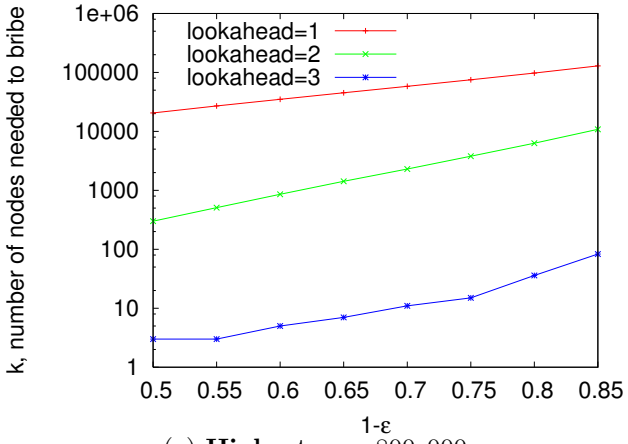


(b) Lookahead 2

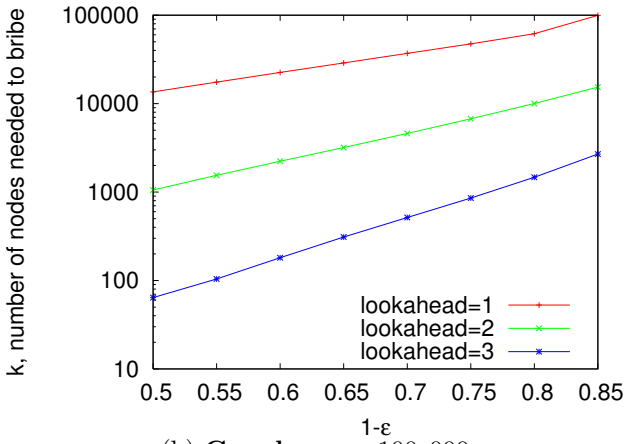
Figure 6.4: **Comparison of attack strategies on LiveJournal data.** Fraction of nodes that needs to be bribed depending on the coverage desired and bribing strategy used, for lookaheads 1 and 2. The lines for **Crawler** and **Greedy** are nearly identical.

6.3.2.2 Dependence on Lookahead

Furthermore, as on the synthetic data, the number of nodes that need to be bribed in order to achieve fixed coverage of LiveJournal decreases exponentially with an increase in lookahead (see Figure 6.5).



(a) **Highest**, $n = 800,000$



(b) **Crawler**, $n = 100,000$

Figure 6.5: **Effect of lookahead on attack difficulty on LiveJournal data.** The number of nodes needed to bribe to achieve $1 - \varepsilon$ coverage with various lookaheads, using **Highest** and **Crawler** strategies, respectively. The y axis is log scale.

These experiments also confirm our hypothesis that while none of the strategies are a truly feasible threat at lookahead 1, some of them become feasible at lookahead 2, and all of them become feasible

at lookahead 3. For example, in order to obtain 80% coverage of the 572,949-user LiveJournal graph using lookahead 2 **Highest** needs to bribe 6,308 users, and to obtain the same coverage using lookahead 3 **Highest** needs to bribe 36 users – a number of users that is sufficiently small given the size of the network, and thus, feasible to bribe in practice.

6.4 Theoretical Analysis for Random Power Law Graphs

In this section we provide a theoretical analysis of the performance of two of the bribing strategies from Section 6.2: **Uniform-Random** and **Highest-Degree**. We analyze the fraction of nodes an attacker needs to bribe to reach a constant node coverage with high probability for a power law social network graph drawn from the configuration model described in Section 6.3.1.1. We carry out the analysis for power law graphs; for configuration models with other degree distributions, our analysis technique still applies, but the result depends on the specific degree distribution.

We use the same notation as in Section 6.3: n is the number of nodes in the network; m is the number of edges; $d_{\min}$ is the minimum degree of a node; $d_{\max}$ is the maximum degree; $2 < \alpha \leq 3$ is the power law parameter; C is the normalizing constant for the degree distribution so that $\sum_{d=d_{\min}}^{d_{\max}} Cd^{-\alpha} = 1$; the target node coverage is $1 - \varepsilon$; k is the number of bribed nodes, and $f = \frac{k}{n}$ is the fraction of the total number of nodes that are bribed.

6.4.1 Analysis of Lookahead $\ell = 1$

We first answer a simpler question: if in each trial the attacker covers a node randomly with probability proportional to its degree (all trials being independent), after how many trials will the attacker have covered $(1-\varepsilon)n$ distinct nodes? Once we have the analysis for randomly covered nodes, we refine it and take into account the effect of the bribing strategy being used on the probabilities of nodes being covered.

6.4.1.1 Analysis of Covering Proportional to Node Degree

If all nodes had an equal probability of being covered, the question would reduce to occupancy and coupon collector problems [142]. Schelling [186] studied an instance of the weighted coupon collector problem in which the probability of sampling each coupon is explicitly given. However, in our problem, not only do we need to consider the weighted random choices of coupon collection, but also the random realization of the graph.

Lemma 11. *Suppose in each trial we cover a node randomly with probability proportional to its degree, independently of previous trials. Then for $\varepsilon_0 < 1$, after $\frac{-\ln \varepsilon_0}{d_{\min}} 2m$ trials, with high probability, the number of distinct nodes covered is at least $n(1-\varepsilon-o(1))$, where $\varepsilon = \sum_{d=d_{\min}}^{d_{\max}} \varepsilon_0^{\left(1-o(1)\right)\frac{d}{d_{\min}}} Cd^{-\alpha}$.*

Proof of Lemma 11. We prove the lemma by calculating the expected number of trials that result in covering nodes of degree i for each $d_{\min} \le i \le d_{\max}$, and then applying the result of the Occupancy problem (Section 6.6) to determining expected number of distinct nodes of degree i covered in those trials. Since both quantities are sharply concentrated around their expectation, the result will hold with high probability.

We first compute the expected number of trials covering nodes of degree i if the total number of trials is $\frac{-\ln \varepsilon_0}{d_{\min}} 2m$.

Denote by c_i the fraction of nodes in the graph of degree i. Since we cover nodes with probability proportional to their degree, the probability that a node of degree i is covered in a particular trial is $\frac{ic_i n}{2m}$ (recall that $2m$ is the total sum of degrees of nodes in the graph). Hence, out of $\frac{-\ln \varepsilon_0}{d_{\min}} 2m$ the expected number of trials covering nodes of degree i is $\frac{-\ln \varepsilon_0}{d_{\min}} 2m * \frac{ic_i n}{2m} = -\frac{i}{d_{\min}} c_i n \ln \varepsilon_0$. Moreover, by Chernoff bound [142], there are at least $-(1-o(1))\frac{i}{d_{\min}} c_i n \ln \varepsilon_0$ such trials with high probability.

Observe that if the the trial is covering a node of degree i, all nodes with degree i have an equal probability of being covered in that trial. Thus if we want to compute the expected number of *distinct* nodes of degree i that are covered, we have a classic occupancy problem with the number of balls being constrained to those $-(1-o(1))\frac{i}{d_{\min}} c_i n \ln \varepsilon_0$ trials covering nodes of degree i and the number of bins being $c_i n$. Therefore, using Lemma 12 (Section 6.6), the expected number of distinct nodes of degree i covered is at least

$$c_i n\left(1 - \exp\big(-\frac{-(1-o(1))\frac{i}{d_{\min}} c_i n \ln \varepsilon_0}{c_i n}\big)\right) = \left(1 - \varepsilon_0^{\left(1-o(1)\right)\frac{i}{d_{\min}}}\right) c_i n$$

and by sharp concentration, the number of such nodes is at least $\left(1 - \varepsilon_0^{\left(1-o(1)\right)\frac{i}{d_{\min}}} - o(1)\right) c_i n$ with high probability.

In total, after $\frac{-\ln \varepsilon_0}{d_{\min}} 2m$ trials, the number of nodes that is not covered is at most

$$\sum_{d_{\min} \le i \le d_{\max},} \big(\varepsilon_0^{\left(1-o(1)\right)\frac{i}{d_{\min}}} + o(1)\big) c_i n = \sum_i \big(\varepsilon_0^{\left(1-o(1)\right)\frac{i}{d_{\min}}}\big) c_i n + o(n).$$

In the power law random graph model, $c_i = Ci^{-\alpha} + o(1)$ with high probability, therefore, the number of nodes that is not covered is at most $\sum_i \big(\varepsilon_0^{\left(1-o(1)\right)\frac{i}{d_{\min}}}\big) Ci^{-\alpha} n + o(n)$, i.e., we cover at

least $n(1-\varepsilon-o(1))$ distinct nodes with high probability, as desired. □

In total, with high probability we miss at most $\sum_{d_i} c_i n\varepsilon_0^{d_i/d_{\min}} + o(n)$ nodes after $\frac{-\ln \varepsilon_0}{d_{\min}} 2m$ trials. In the power law random graph model, $c_i = Cd_i^{-\alpha} + o(1)$ with high probability, therefore, we miss at most $\sum_{d=d_{\min}}^{\sqrt{n}} Cd^{-\alpha} n\varepsilon_0^{d/d_{\min}} + o(n)$, i.e., we collect at least $n(1-\varepsilon-o(1))$ nodes. □

It is easy to see that ε is always smaller than ε_0. Table 6.2 gives some asymptotic values of ε for chosen values of ε_0; for example, when $\alpha = 3$ and $d_{\min} = 5$, $\varepsilon = 0.4$ gives $\varepsilon_0 = 0.534$.

6.4.1.2 Effect of Bribing Strategy on Coverage

We now compute the number of nodes an attacker needs to bribe when lookahead = 1, in order to obtain the coverage of $1-\varepsilon$ of the graph, depending on the bribing strategy used by the attacker. Recall that when lookahead is 1, the attacker covers a node only if it is a direct neighbor of a bribed node. Thus, it is not surprising that when $\ell = 1$, the power of the bribing strategy is correlated with the sum of degrees of nodes bribed by the strategy.

Theorem 8. *Suppose distinct nodes $b_1, b_2, \ldots, b_k$ selected using an arbitrary strategy are bribed. Denote the sum of their degrees by $D = \sum_{i=1}^{k} d(b_i)$. If lookahead = 1, and $D = \frac{-\ln \varepsilon_0}{d_{\min}} 2m$ for some $\varepsilon_0 < 1$, then the node coverage attained is at least $1-\varepsilon-o(1)$ with high probability, where $\varepsilon = \sum_{d=d_{\min}}^{d_{\max}} \varepsilon_0^{\left(1-o(1)\right)\frac{d}{d_{\min}}} Cd^{-\alpha}$.*

Proof. Pick a node b to bribe using any strategy. Consider one edge of the bribed node, the other endpoint of the edge can be any node and the probability of it being a particular node v is $d(v)/2m$ if we randomize over all graphs with the given degree sequence using the Principle of Deferred Decisions [142]. Therefore, if we bribe a node with degree d and cover all its neighbors, it is equivalent to having made d trials covering nodes with probability proportional to their degree, as in the setup of Lemma 11. And if we bribe distinct nodes $b_1, b_2, \ldots, b_k$ and cover all their neighbors, it is equivalent to having made $D = \sum_{i=1}^{k} d(b_i)$ such trials.

Moreover, in the set-up of bribing nodes, not every trial covers a node v with probability proportional to its degree: if v was already covered in a previous trial, the probability of covering it again decreases, whereas if it was not covered in a previous trial, the probability of covering it increases with each new trial. Therefore, the expected number of distinct nodes covered according to Lemma 11 is a lower bound on the actual number of distinct nodes covered, which completes the proof. □

Theorem 8 establishes the connection between the total degree of bribed nodes (regardless of the strategy for choosing nodes to bribe) and the attained node coverage. In order to complete the analysis of particular bribing strategies it remains to analyze the total degree of k nodes bribed by that strategy.

We first analyze the strategy of bribing nodes uniformly at random without replacement.

Corollary 2. *If an attacker bribes* $\frac{-\ln \varepsilon_0}{d_{\min}} n$ *nodes picked according to the* **Uniform-Random** *strategy, then he covers at least* $n(1-\varepsilon-o(1))$ *nodes with high probability, where* $\varepsilon = \sum_{d=d_{\min}}^{d_{\max}} \varepsilon_0^{\left(1-o(1)\right)\frac{d}{d_{\min}}} C d^{-\alpha}$.

Proof. In any graph, a node chosen uniformly at random has expected degree $\bar{d} = 2m/n$, and bribing k nodes yields expected total degree $D = 2mk/n$. Plugging this expected total degree into Theorem 8 we obtain the corollary. □

Next we analyze the **Highest-Degree** strategy.

Corollary 3. *If an attacker bribes* $\left(-\frac{\ln \varepsilon_0}{d_{\min}} + \frac{1}{(\frac{d_{\max}}{d_{\min}})^{\alpha-2}-1}\right)^{\frac{\alpha-1}{\alpha-2}} n$ *nodes picked according to the* **Highest-Degree** *strategy, then he covers at least* $n(1-\varepsilon-o(1))$ *nodes with high probability, where* $\varepsilon = \sum_{d=d_{\min}}^{d_{\max}} \varepsilon_0^{\left(1-o(1)\right)\frac{d}{d_{\min}}} C d^{-\alpha}$, *provided that* $2 < \alpha \leq 3$ *and* $d_{\max} > d_{\min}$.

Proof. To apply Theorem 8, we compute the expected total degree of the nodes with degree d and higher. Denote the number of such nodes by k. In our calculations, we focus on the expectations, observing that when n is large and $k = \Theta(n)$, the actual values are tightly concentrated around expectations. Then, in the power law random graph model, the constant C is chosen in such a way that the total number of nodes is n: $\sum_{x=d_{\min}}^{d_{\max}} C x^{-\alpha} n = n$, and the expected number of nodes of degree d and higher is k: $\sum_{x=d}^{d_{\max}} C x^{-\alpha} n = k$.

When n is large, we can use integration to approximate the sum, and thus get the following system of equations:

$$\begin{cases} \int_{x=d_{\min}}^{d_{\max}} C x^{-\alpha} dx = 1 \\ \int_{x=d}^{d_{\max}} C x^{-\alpha} dx = \frac{k}{n} \end{cases} \quad \begin{cases} \frac{C}{1-\alpha} x^{1-\alpha}\big|_{x=d_{\min}}^{d_{\max}} = 1 \\ \frac{C}{1-\alpha} x^{1-\alpha}\big|_{x=d}^{d_{\max}} = \frac{k}{n} \end{cases} \quad \begin{cases} C = \frac{1-\alpha}{d_{\max}^{1-\alpha} - d_{\min}^{1-\alpha}} = \frac{\alpha-1}{d_{\min}^{1-\alpha} - d_{\max}^{1-\alpha}} \\ d_{\max}^{1-\alpha} - d^{1-\alpha} = \frac{k(1-\alpha)}{nC} = \frac{k(d_{\max}^{1-\alpha} - d_{\min}^{1-\alpha})}{n} \end{cases}$$

Suppose $d_{\max} \geq t d_{\min}$ for some $t \geq 1$. Then $d^{1-\alpha} = d_{\max}^{1-\alpha} - \frac{k(d_{\max}^{1-\alpha} - d_{\min}^{1-\alpha})}{n} = \frac{k}{n} d_{\min}^{1-\alpha} + (1-\frac{k}{n}) d_{\max}^{1-\alpha} \geq \frac{k}{n} d_{\min}^{1-\alpha} - \frac{k}{n} d_{\max}^{1-\alpha} \geq \frac{k}{n}\left(d_{\min}^{1-\alpha} - (t d_{\min})^{1-\alpha}\right) = \frac{k}{n} d_{\min}^{1-\alpha}(1-t^{1-\alpha})$, from which it follows that for $\alpha \geq 2$:

$$d^{2-\alpha} \geq (\frac{k}{n})^{\frac{2-\alpha}{1-\alpha}} d_{\min}^{2-\alpha} (1-t^{1-\alpha})^{\frac{2-\alpha}{1-\alpha}} \tag{6.1}$$

The sum of degree of all nodes is:
$\sum_{x=d_{\min}}^{d_{\max}} Cx^{-\alpha}nx \approx \int_{x=d_{\min}}^{d_{\max}} Cx^{1-\alpha}ndx = \frac{nC}{2-\alpha}x^{2-\alpha}|_{d_{\min}}^{d_{\max}} = \frac{nC}{2-\alpha}(d_{\max}^{2-\alpha} - d_{\min}^{2-\alpha}) = 2m$

The sum of degrees of the k nodes whose degree is at least d is:
$D = \sum_{x=d}^{d_{\max}} Cx^{-\alpha}nx \approx \int_{x=d}^{d_{\max}} Cx^{1-\alpha}ndx = \frac{nC}{2-\alpha}x^{2-\alpha}|_{d}^{d_{\max}} = \frac{nC}{2-\alpha}(d_{\max}^{2-\alpha} - d^{2-\alpha})$. Combining with the previous equation regarding the sum of all degrees we have: $\frac{D}{2m} \approx \frac{d_{\max}^{2-\alpha} - d^{2-\alpha}}{d_{\max}^{2-\alpha} - d_{\min}^{2-\alpha}} = \frac{d^{2-\alpha} - d_{\max}^{2-\alpha}}{d_{\min}^{2-\alpha} - d_{\max}^{2-\alpha}}$. Using inequality (6.1), we obtain:

$$\frac{D}{2m} \geq \frac{(\frac{k}{n})^{\frac{2-\alpha}{1-\alpha}} d_{\min}^{2-\alpha}(1-t^{1-\alpha})^{\frac{2-\alpha}{1-\alpha}} - d_{\max}^{2-\alpha}}{d_{\min}^{2-\alpha} - d_{\max}^{2-\alpha}} = (\frac{k}{n})^{\frac{2-\alpha}{1-\alpha}}(1-t^{1-\alpha})^{\frac{2-\alpha}{1-\alpha}} - \frac{(1-(\frac{k}{n})^{\frac{2-\alpha}{1-\alpha}}(1-t^{1-\alpha})^{\frac{2-\alpha}{1-\alpha}})d_{\max}^{2-\alpha}}{d_{\min}^{2-\alpha} - d_{\max}^{2-\alpha}}.$$

By choice of t, we have $\frac{d_{\max}^{2-\alpha}}{d_{\min}^{2-\alpha} - d_{\max}^{2-\alpha}} \leq \frac{d_{\max}^{2-\alpha}}{(\frac{d_{\max}}{t})^{2-\alpha} - d_{\max}^{2-\alpha}} = \frac{1}{t^{\alpha-2}-1}$. Hence,
$\frac{D}{2m} \geq (\frac{k}{n})^{\frac{2-\alpha}{1-\alpha}}(1-t^{1-\alpha})^{\frac{2-\alpha}{1-\alpha}} - \frac{1-(\frac{k}{n})^{\frac{2-\alpha}{1-\alpha}}(1-t^{1-\alpha})^{\frac{2-\alpha}{1-\alpha}}}{t^{\alpha-2}-1} = (\frac{k}{n})^{\frac{2-\alpha}{1-\alpha}}\left((1-t^{1-\alpha})^{\frac{2-\alpha}{1-\alpha}} + \frac{(1-t^{1-\alpha})^{\frac{2-\alpha}{1-\alpha}}}{t^{\alpha-2}-1}\right) - \frac{1}{t^{\alpha-2}-1}$. Using Lemma 13 (Section 6.6) we obtain:

$$\frac{D}{2m} \geq (\frac{k}{n})^{\frac{\alpha-2}{\alpha-1}} - \frac{1}{t^{\alpha-2}-1} \geq (\frac{k}{n})^{\frac{\alpha-2}{\alpha-1}} - \frac{1}{(\frac{d_{\max}}{d_{\min}})^{\alpha-2}-1}$$

Therefore, bribing $\left(-\frac{\ln \varepsilon_0}{d_{\min}} + \frac{1}{(\frac{d_{\max}}{d_{\min}})^{\alpha-2}-1}\right)^{\frac{\alpha-1}{\alpha-2}} n$ nodes according to the **Highest-Degree** strategy suffices to obtain $n(1-\varepsilon-o(1))$ nodes with high probability,

□

Corollaries 2 and 3 only give lower bounds on the attained node coverage, but our simulation results in Section 6.4.1.3 indicate that the analysis is fairly tight.

From the corollaries, it is clear that when $\ell = 1$, in order to cover a certain fraction of the nodes, an attacker needs to bribe much fewer nodes when using the **Highest-Degree** bribing strategy than when using the **Uniform-Random** bribing strategy. For example, when $\alpha = 3$, if an attacker bribes an f fraction of the nodes with the **Uniform-Random** strategy, then he only needs to bribe an $\approx f^2$ fraction of the nodes using the **Highest-Degree** strategy to attain the same coverage. Moreover, the smaller the powerlaw parameter α of the network, the fewer nodes need to be bribed using the **Highest-Degree** strategy to attain the same coverage. On the other hand, the bad news for an attacker targeting a social network that provides only lookahead of $\ell = 1$ is that even if he has the power to choose the highest degree nodes for an attack, a linear number of nodes will need to be bribed in order to cover a constant fraction of the whole graph (since the number of nodes needed to bribe is linear in n in both Corollaries). Hence, lookahead $\ell = 1$ is fairly protective of

privacy.

6.4.1.3 Validating Theoretical Analysis With Simulation

We validate our theoretical estimates of Section 6.4.1 by simulation. Our theoretical analysis shows that in order to achieve a certain fixed node coverage, the number of nodes needed to bribe is linear in the total number of nodes in the social network, i.e., f is a constant with varying n. This matches and confirms our simulation results from Section 6.3.1.3.

Next we check whether the f values predicted by Corollaries 2 and 3 match simulation results (see Table 6.2[2]). We observe that the f values obtained through simulation are smaller than those predicted in Corollaries 2 and 3. This is because Theorem 8, on which Corollaries 2 and 3 rely, gives **a lower bound** on the number of covered nodes. There are two factors responsible for the underestimation of the coverage attained in our theoretical analysis: (1) the different trials cover uncovered nodes with higher probability; (2) we did not count the bribed nodes as covered. The second factor responsible for the underestimation is more severe when the number of bribed nodes is not negligible in comparison to the number of covered nodes, which is especially true in the case of the **Uniform-Random** strategy. We can remedy this by taking into consideration the bribed nodes and refining our analysis. Using the same parameters as in Table 6.2, for $\varepsilon = 0.4, 0.2, 0.1$, the refined predicted fs for the **Uniform-Random** bribing strategy are $0.110, 0.204, 0.305$ respectively, which are closer to the simulation results, indicating that our theoretical analysis is fairly tight.

		Uniform-Random		**Highest-Degree**	
ε	ε_0	f_p	f_s	f_p	f_s
0.4	0.534	0.125	0.103	0.016	0.015
0.2	0.309	0.235	0.183	0.055	0.045
0.1	0.173	0.350	0.259	0.123	0.090

Table 6.2: **Theoretical estimates vs simulation results.** We compute f for varying ε for two bribing strategies. f_p is the estimate of the fraction of nodes needed to bribe according to Corollaries 2 and 3. f_s is the fraction needed to bribe obtained experimentally through simulation. We use $\alpha = 3$ and $d_{\min} = 5$.

6.4.2 Heuristic Analysis of Lookahead $\ell > 1$

Performing an exact analysis of performance of bribing strategies when lookahead $\ell > 1$ is challenging, so we perform a heuristic analysis in order to understand the influence of an increase in

[2]We omit the $\frac{1}{(\frac{d_{\max}}{d_{\min}})^{\alpha-2}-1}$ term of Corollary 3 in our computation as it is negligible when $n \to \infty, d_{\max} >> d_{\min}$.

lookahead on the number of nodes needed to bribe at least directionally. In practice, the trade-off can be made more precise by running simulations measuring performance of strategies on the social network graph that the social network owner wishes to protect, depending on lookahead ℓ and the strategy used.

Our heuristic analysis shows that in order for the attacker to get $(1-\varepsilon)$ coverage using the **Uniform-Random** strategy, the fraction of nodes f that the attacker needs to bribe is $\approx \frac{-\ln \varepsilon_0}{d_{\min} b^{\ell-1}}$, where ε and ε_0 satisfy the equation in Lemma 11 and $b = \Theta\big(\ln(d_{\max})\big)$. When using the **Highest-Degree** strategy, the fraction of nodes the attacker needs to bribe is $\approx \big(\frac{-\ln \varepsilon_0}{d_{\min} b^{\ell-1}}\big)^{\frac{\alpha-1}{\alpha-2}}$. Details of the analysis that leads to these estimates can be found in Section 6.4.2.1.

The heuristic analysis shows that the fraction of nodes needed to bribe in order to achieve constant coverage decreases exponentially with increase in lookahead ℓ. For example, when lookahead $\ell = \Theta(\frac{\ln n}{\ln \ln d_{\max}})$, bribing a constant number of nodes is sufficient to attain coverage of almost the entire graph, making link privacy attacks on social networks with lookahead $\ell > 1$ truly feasible.

6.4.2.1 Details of Heuristic Analysis of Lookahead $\ell > 1$

Denote by B the set of bribed nodes; by $N_i(B)$ the set of nodes whose shortest distance to B is exactly i. Our goal is to estimate the number of covered nodes given the bribed nodes when lookahead is ℓ, which is equivalent to the number of nodes within distance $\leq \ell$ from B, denoted by $D_\ell(B) = |\bigcup_{0\leq i\leq \ell} N_i(B)|$. Then, in order to achieve coverage of $(1-\varepsilon)$, we need to bribe $f = |B|/n$ fraction of nodes, where B is such that $D_\ell(B) = (1-\varepsilon)n$.

Suppose that for $i \leq \ell-1$, $N_i(B)$ is small enough such that $\bigcup_{0\leq i\leq \ell-1} N_i(B)$ is a forrest rooted at B, i.e., there are no loops between nodes belonging to $N_i(B)$s in the network. Under this assumption, $|N_\ell(B)|$ is much larger than all $|N_i(B)|$s ($i < \ell$), so we can use $|N_\ell(B)|$ as an approximation to $D_\ell(B)$.

To compute $|N_\ell(B)|$, we first study the expansion rate from N_i to N_{i+1} for $1 \leq i \leq \ell-2$, denoted by $b_i = |N_{i+1}(B)|/|N_i(B)|$. We then apply Lemma 11 to compute $|N_\ell(B)|$ given $|N_{\ell-1}(B)|$, and use the results of Corollaries 2 and 3 to estimate $|N_1(B)|$.

6.4.2.1.1 Estimating b_i: Under the no-loop assumption, b_i can be estimated as the expected average degree of nodes in $N_i(B)$ decreased by 1 (in order to exclude the edges coming from $N_{i-1}(B)$). Note that nodes in $N_i(B)$ are not chosen uniformly at random; rather, they are chosen with probability proportional to their degrees because of the random realization of the graph. Therefore, the probability that such a node has degree x is proportional to $xCx^{-\alpha}$, and consequently the expected average degree of these nodes is

$E[d_{avg}] = \frac{\sum_{x=d_{\min}}^{d_{\max}} xxCx^{-a}}{\sum_{x=d_{\min}}^{d_{\max}} xCx^{-a}} \approx \frac{\int_{x=d_{\min}}^{d_{\max}} x^{2-a}dx}{\int_{x=d_{\min}}^{d_{\max}} x^{1-a}dx} \geq \frac{\int_{x=d_{\min}}^{d_{\max}} x^{-1}dx}{\int_{x=d_{\min}}^{d_{\max}} x^{1-a}dx} = (2-\alpha)\frac{\ln(d_{\max})-\ln(d_{\min})}{d_{\max}^{2-\alpha}-d_{\min}^{2-\alpha}} = (\alpha-2)\frac{\ln(d_{\max})-\ln(d_{\min})}{d_{\min}^{2-\alpha}-d_{\max}^{2-\alpha}} \geq (\alpha-2)d_{\min}^{\alpha-2}\big(\ln(d_{\max})-\ln(d_{\min})\big)$.

Hence, the expansion rate $b_i = (\alpha-2)d_{\min}^{\alpha-2}\ln(\frac{d_{\max}}{d_{\min}}) - 1 = \Theta(\ln(d_{\max}))$ and it is independent of i.

6.4.2.1.2 Estimating $|N_\ell(B)|$: When $b|N_{\ell-1}(B)|$ is large, we can no longer use the no-loop assumption to estimate $|N_\ell(B)|$. There still are $b|N_{\ell-1}(B)|$ edges incident to $N_{\ell-1}(B)$ but now some of these edges may share the same endpoints. In this case, the set-up is the same as in Lemma 11 and so in order to compute the number of distinct nodes in $N_\ell(B)$ we can apply the result of Lemma 11, i.e., if $b|N_{\ell-1}(B)| = \frac{-\ln\varepsilon_0}{d_{\min}}2m$, then $|N_\ell(B)| \approx n(1-\varepsilon)$.

6.4.2.1.3 Estimating $|N_1(B)|$: Using Corollary 2 we know that if we bribe k nodes using the **Uniform-Random** strategy, then the expected total degree is $2mk/n$. Under the no-loop assumption, this implies $|N_1(B)| = 2m|B|/n$.

Similarly, using Corollary 3 we know that for **Highest-Degree** strategy, $|N_1(B)| \approx 2m(\frac{|B|}{n})^{\frac{\alpha-2}{\alpha-1}}$.

6.4.2.1.4 Combining the Estimates to Complete Heuristic Analysis: We now complete the heuristic analysis of the fraction of nodes that need to be bribed to achieve $1-\varepsilon$ coverage when lookahead is ℓ. Suppose we bribe a small number $|B|$ nodes, so that $\bigcup_{0\leq i\leq \ell-1} N_i(B)$ is a forrest, and only for $i=\ell$ the nodes start repeating significantly. Then, $|N_{\ell-1}(B)| \approx b^{\ell-2}|N_1(B)|$. Moreover, under the forrest assumption, $D_\ell(B) \approx |N_\ell(B)|$ and hence to achieve $1-\varepsilon$ coverage we need $|N_{\ell-1}(B)| = \frac{-\ln\varepsilon_0}{bd_{\min}}2m$. Combining these approximations we obtain that we need $|N_1(B)| = \frac{-\ln\varepsilon_0}{b^{\ell-1}d_{\min}}2m$.

Hence, for **Uniform-Random** strategy, we need to bribe $|B| = \frac{-\ln\varepsilon_0}{b^{\ell-1}d_{\min}}n$ nodes; and for **Highest-Degree** strategy, we need to bribe $|B| = \big(\frac{-\ln\varepsilon_0}{b^{\ell-1}d_{\min}}\big)^{\frac{\alpha-1}{\alpha-2}}n$ nodes.

We have made several crude approximations in this analysis, especially around the assumption that nodes at distances less than $\ell-1$ from B form a forrest. However, we have mitigated the effect of this assumption by using $|N_\ell|$ rather than $\sum_{i=0}^{\ell}|N_i(B)|$ to estimate $D_\ell(B)$, and by applying the occupancy problem based estimation for computing N_ℓ, based on the largest seed of nodes than all other N_is. The heuristic analysis illustrates that the inverse exponential dependance between lookahead ℓ and the fraction of nodes that need to be bribed to achieve constant coverage. For real-world applications, the exact dependance can be established experimentally, based on the particular

network the social network owner wishes to protect, and their assumptions of the attacker's ability to choose bribing strategies.

6.5 Summary

In this chapter, we provided a theoretical and experimental analysis of the vulnerability of a social network to a certain kind of privacy attack, in which an attacker aims to obtain knowledge of a significant fraction of links belonging to a social network by gaining access to local snapshots of the graph through accounts of individual users. We described several strategies for carrying out such attacks, and analyzed their potential for success as a function of the lookahead permitted by the social network's interface. We have shown that the number of user accounts that an attacker needs to subvert in order to obtain a fixed portion of the link structure of the network decreases exponentially with increase in lookahead chosen by the social network owner. We conclude that social networks owners interested in protecting their social graphs ought to carefully balance the trade-offs between the social utility offered by a large lookahead and the threat that such a lookahead poses to their business, and our analysis can serve as a starting point towards evaluating these trade-offs.

We showed that as a rule of thumb, the social network owners concerned about protecting their social graph may want to refrain from permitting lookaheads higher than 2 in their interface. They may also consider decreasing their vulnerability through other restrictions, such as not displaying the exact number of connections of each user or by varying the lookahead available to particular users depending on their trustworthiness.

6.6 Miscellaneous Technical Details

Lemma 12 (**Occupancy Problem** (Theorem 4.18 in [142]))**.** *Let Z be the number of empty bins when m balls are thrown randomly into n bins. Then $E[Z] = n(1-\frac{1}{n})^m \leq n\exp(-\frac{m}{n})$ and for $\lambda > 0$, $\Pr[|Z - E[Z]| \geq \lambda] \leq 2\exp\big(-\frac{\lambda^2(n-0.5)}{n^2 - E[Z]^2}\big)$.*

Lemma 13. *If $2 < a \leq 3$ and $t > 1$ then $Z = (1-t^{1-a})^{\frac{2-a}{1-a}} + \frac{(1-t^{1-a})^{\frac{2-a}{1-a}}}{t^{a-2}-1} \geq 1$.*

Proof. $Z = (1-t^{1-a})^{\frac{2-a}{1-a}} + \frac{(1-t^{1-a})^{\frac{2-a}{1-a}}}{t^{a-2}-1} = (1-t^{1-a})^{\frac{2-a}{1-a}}\left(1 + \frac{1}{t^{a-2}-1}\right) = (1-t^{1-a})^{\frac{2-a}{1-a}}\left(\frac{t^{a-2}}{t^{a-2}-1}\right) = \left(\frac{t^{a-1}-1}{t^{a-1}}\right)^{\frac{a-2}{a-1}}\left(\frac{t^{a-2}}{t^{a-2}-1}\right) = \frac{(t^{a-1}-1)^{\frac{a-2}{a-1}}}{t^{a-2}-1}$

We prove that $\frac{(t^{a-1}-1)^{a-2}}{(t^{a-2}-1)^{a-1}} \geq 1$ from which it follows that $Z \geq 1$ for the chosen values of a and t.

Let's find the extreme values of $(t^{a-1}-1)^{a-2}-(t^{a-2}-1)^{a-1}$ by computing the zeros of the expression's derivative.

$\ln(t^{a-1}-1)\cdot(t^{a-1}-1)^{a-2}\ln t\cdot t^{a-1}-\ln(t^{a-2}-1)\cdot(t^{a-2}-1)^{a-1}\ln t\cdot t^{a-2}=0$

$\ln t=0$, or $t=0$ or $\ln(t^{a-1}-1)\cdot(t^{a-1}-1)^{a-2}\cdot t-\ln(t^{a-2}-1)\cdot(t^{a-2}-1)^{a-1}=0$

Note that $t>t^{a-2}-1$, and $t^{a-1}-1>t^{a-2}-1$, therefore $(t^{a-1}-1)^{a-2}\cdot t\geq(t^{a-1}-1)^{a-2}\cdot(t^{a-2}-1)\geq(t^{a-2}-1)^{a-1}$ and thus the last equation has no solutions.

Hence, it remains for us to check that $\frac{(t^{a-1}-1)^{a-2}}{(t^{a-2}-1)^{a-1}}\geq 1$ for $a\to 2$ and $a=3$, which is, indeed, the case, for $t>1$. □

Chapter 7

Contributions and Open Questions

In today's digital online world, massive amounts of data on individual users is amassed on a daily basis. The growth of this data and the ability of online service providers to mine and share it provides great opportunities for innovation that can in turn improve user experience. However, finding ways to support this innovation while protecting individual privacy is a truly multidisciplinary challenge facing modern society, involving a diverse set of stakeholders with competing interests and sets of expertise. The role of algorithmic research in this debate is to envision and develop a foundation for dealing with the challenge and to lead the way in shaping the debate through elucidation of capabilities, limitations, and quantitative analyses of trade-offs.

In this thesis, we have explored examples of privacy violations, proposed privacy-preserving algorithms, and analyzed the trade-offs between utility and privacy when mining and sharing user data for several concrete problems in search and social networks. We suggested practical algorithms and provided quantitative and actionable analyses of the trade-offs for the problems considered.

From the algorithmic perspective, the main challenge for future work in the field is to remove the remaining barriers to adoption of approaches satisfying rigorous privacy guarantees, such as differential privacy, to real-world settings. Open questions for future research include:

- expand the toolkit of privacy-preserving algorithms by building primitives for the core data-mining operations used today
- where possible, improve the existing algorithms to achieve better utility, and to respect exogenous constraints such as data accuracy or consistency
- make the application of privacy-preserving techniques and privacy/utility trade-off analyses

straightforward for people with no special training in privacy, working under tight time-constraints

- make the algorithms and analyses applicable to a rapidly expanding variety of input data
- make the algorithms useful and practical for scenarios when data-mining and sharing are happening on a continuous basis
- develop an infrastructure and support for users to exercise an individualized choice on the privacy risk they are willing to incur when using a service, and ability to adjust the service's performance and cost accordingly
- be able to provide justification for the choice of parameter values in the privacy framework.

Beyond the algorithmic advances, if we are to achieve a better balance in the trade-off between privacy and innovation, progress needs to be made in integrating those into the societal discourse taking place around issues of privacy. The main challenges are to:

- build a foundation for enabling the discourse to move away from a binary perspective on privacy (either something is private or it is not) towards a more fine-grained quantifiable gradation of the extent to which a particular service's data-sharing and mining practices preserve privacy
- communicate to users the protections offered by parameterized privacy-preserving algorithms and empower them to make informed decisions about their use of a service based on the privacy guarantees it provides
- make the benefits and dangers of data-mining and sharing transparent and quantifiable for all participants of the ecosystem
- find opportunities for integration into legal and regulatory frameworks for measuring risk and compliance of companies' data mining and sharing practices with respect to their promises.

Few topics today arouse as much heated discussion as issues of user privacy. As individuals' online presence and activities rapidly expand, the question of how we balance innovation and open platforms with privacy concerns will only grow. There is a great opportunity for algorithmic research not only to help advance computational solutions for these challenges but also contribute to developing a more constructive and granular societal discourse around privacy. This thesis focused on making practical and constructive strides in that direction.

Bibliography

[1] E. Adar. User 4xxxxx9: Anonymizing query logs. In *Query Log Analysis: Social And Technological Challenges Workshop at WWW*, 2007.

[2] C. C. Aggarwal. On k-anonymity and the curse of dimensionality. In *VLDB*, pages 901–909, 2005.

[3] G. Aggarwal, T. Feder, K. Kenthapadi, R. Motwani, R. Panigrahy, D. Thomas, and A. Zhu. Anonymizing tables. In *ICDT*, pages 246–258, 2005.

[4] E. Agichtein, E. Brill, and S. Dumais. Improving web search ranking by incorporating user behavior information. In *SIGIR*, pages 19–26, 2006.

[5] R. Agrawal, A. Halverson, K. Kenthapadi, N. Mishra, and P. Tsaparas. Generating labels from clicks. In *WSDM*, pages 172–181, 2009.

[6] W. Aiello, F. Chung, and L. Liu. A random graph model for power law graphs. *IEEE Symposium on Foundations of Computer Science*, 2000.

[7] E. Aïmeur, G. Brassard, J. M. Fernandez, and F. S. Mani Onana. Alambic: a privacy-preserving recommender system for electronic commerce. *Int. J. Inf. Secur.*, 7(5):307–334, 2008.

[8] American Academy of Matrimonial Lawyers. Big surge in social networking evidence says survey of nation's top divorce lawyers. http://www.aaml.org/about-the-academy/press/press-releases/e-discovery/big-surge-social-networking-evidence-says-survey-, February 10, 2010.[1].

[1]The contents of all hyperlinks referenced in this thesis were archived in June 2012, and are available at http://theory.stanford.edu/~korolova/Thesis/References

[9] R. Andersen, C. Borgs, J. T. Chayes, U. Feige, A. D. Flaxman, A. Kalai, V. S. Mirrokni, and M. Tennenholtz. Trust-based recommendation systems: an axiomatic approach. In *WWW*, pages 199–208, 2008.

[10] L. Andrews. Facebook is using you. *The New York Times*, February 4, 2012.

[11] E. Arcaute and S. Vassilvitskii. Social networks and stable matchings in the job market. In *WINE*, pages 220–231, 2009.

[12] M. Arrington. AOL proudly releases massive amounts of private data. *TechCrunch*, August 6, 2006.

[13] M. Arrington. Being Eric Schmidt (on Facebook). *TechCrunch*, February 1, 2010.

[14] M. Arrington. Facebook COO: 175 million people log into Facebook every day. *TechCrunch*, October 10, 2010.

[15] L. Backstrom. Anatomy of Facebook. Facebook data blog, http://www.facebook.com/note.php?note_id=10150388519243859, November 21, 2011.

[16] L. Backstrom, C. Dwork, and J. M. Kleinberg. Wherefore art thou r3579x?: anonymized social networks, hidden patterns, and structural steganography. In *WWW*, pages 181–190, 2007.

[17] L. Backstrom and J. Leskovec. Supervised random walks: predicting and recommending links in social networks. In *WSDM*, pages 635–644, 2011.

[18] R. Baeza-Yates and A. Tiberi. Extracting semantic relations from query logs. In *KDD*, pages 76–85, 2007.

[19] J. Bar-Ilan. Access to query logs - an academic researcher's point of view. In *Query Log Analysis: Social And Technological Challenges Workshop at WWW*, 2007.

[20] A. Barabasi and R. Albert. Emergence of scaling in random networks. *Science*, 286(5439):509–512, 1999.

[21] B. Barak, K. Chaudhuri, C. Dwork, S. Kale, F. McSherry, and K. Talwar. Privacy, accuracy and consistency too: A holistic solution to contingency table release. In *PODS*, pages 273–282, 2007.

[22] M. Barbaro and T. Z. Jr. A face is exposed for AOL searcher No. 4417749. *The New York Times*, August 9, 2006.

[23] R. M. Bell and Y. Koren. Lessons from the Netflix prize challenge. *SIGKDD Explor. Newsl.*, 9(2):75–79, December 2007.

[24] E. A. Bender and E. R. Canfield. The asymptotic number of labeled graphs with given degree sequences. *Journal of Combinatorial Theory, Series A*, 24(3):296–307, 1978.

[25] J. Bennett and S. Lanning. The Netflix prize. In *KDD Cup and Workshop in conjunction with KDD*, 2007.

[26] R. Bhaskar, S. Laxman, A. Smith, and A. Thakurta. Discovering frequent patterns in sensitive data. In *KDD*, pages 503–512, 2010.

[27] R. Bhattacharjee, A. Goel, and K. Kollias. An incentive-based architecture for social recommendations. In *RecSys*, pages 229–232, 2009.

[28] A. Blum, K. Ligett, and A. Roth. A learning theory approach to non-interactive database privacy. In *STOC*, pages 609–618, 2008.

[29] B. Bollobás, C. Borgs, J. T. Chayes, and O. Riordan. Directed scale-free graphs. In *SODA*, pages 132–139, 2003.

[30] d. boyd and K. Crawford. Six provocations for big data. In *A Decade in Internet Time: Symposium on the Dynamics of the Internet and Society*, September 21, 2011.

[31] J. A. Calandrino, A. Kilzer, A. Narayanan, E. W. Felten, and V. Shmatikov. "You might also like:" Privacy risks of collaborative filtering. In *IEEE Symposium on Security and Privacy*, pages 231–246, 2011.

[32] K. Chaudhuri and N. Mishra. When random sampling preserves privacy. In *CRYPTO*, pages 198–213, 2006.

[33] K. Chaudhuri, C. Monteleoni, and A. D. Sarwate. Differentially private empirical risk minimization. *Journal of Machine Learning Research*, 12:1069–1109, 2011.

[34] P.-Y. Chen and S. Wu. Does collaborative filtering technology impact sales? Empirical evidence from Amazon.com. *Social Science Research Network*, July 8, 2007.

[35] F. Chung and L. Lu. *Complex Graphs and Networks*. American Mathematical Society, Aug. 2006.

[36] G. Cluley. 600,000+ compromised account logins every day on Facebook, official figures reveal. Sophos Naked Security Blog, http://nakedsecurity.sophos.com/2011/10/28/compromised-facebook-account-logins, October 28, 2011.

[37] C. Cooper and A. M. Frieze. A general model of web graphs. *Random Struct. Algorithms*, 22(3):311–335, 2003.

[38] G. Cormode, D. Srivastava, T. Yu, and Q. Zhang. Anonymizing bipartite graph data using safe groupings. *VLDB J.*, 19(1):115–139, 2010.

[39] N. Craswell, R. Jones, G. Dupret, and E. Viegas, editors. *Proceedings of the workshop on Web Search Click Data (WSCD)*. ACM, 2009.

[40] N. Craswell and M. Szummer. Random walks on the click graph. In *SIGIR*, pages 239–246, 2007.

[41] G. Csányi and B. Szendrői. Structure of a large social network. *Phys. Rev. E*, 69(3), 2004.

[42] H. B. Dwight. *Tables of integrals and other mathematical data*. The Macmillan Company, 4th edition, 1961.

[43] C. Dwork. Differential privacy. In *ICALP*, pages 1–12, 2006.

[44] C. Dwork. An ad omnia approach to defining and achieving private data analysis. In *Lecture Notes in Computer Science*, volume 4890, pages 1–13. Springer, 2008.

[45] C. Dwork. Differential privacy: A survey of results. In *Theory and Applications of Models of Computation*, pages 1–19, 2008.

[46] C. Dwork. A firm foundation for private data analysis. *Commun. ACM*, 54(1):86–95, 2011.

[47] C. Dwork, K. Kenthapadi, F. McSherry, I. Mironov, and M. Naor. Our data, ourselves: Privacy via distributed noise generation. In *EUROCRYPT*, pages 486–503. Springer, 2006.

[48] C. Dwork and J. Lei. Differential privacy and robust statistics. In *STOC*, pages 371–380, 2009.

[49] C. Dwork, F. McSherry, K. Nissim, and A. Smith. Calibrating noise to sensitivity in private data analysis. In *Theory of Cryptography*, pages 265–284, 2006.

[50] C. Dwork and M. Naor. On the difficulties of disclosure prevention in statistical databases or the case for differential privacy. *Journal of Privacy and Confidentiality*, 2(1):93–107, 2010.

[51] P. Eckersley. How unique is your web browser? In *Privacy Enhancing Technologies*, pages 1–18, 2010.

[52] P. Eckersley. A primer on information theory and privacy. Electronic Frontier Foundation https://www.eff.org/deeplinks/2010/01/primer-information-theory-and-privacy, January 26, 2010.

[53] B. Edelman. Facebook leaks usernames, user ids, and personal details to advertisers. http://www.benedelman.org/news/052010-1.html, May 20, 2010.

[54] D. Fallows. Search engine users. Pew Internet and American Life Project http://www.pewinternet.org/Reports/2005/Search-Engine-Users.aspx, January 23, 2005.

[55] K. FC. The fundamental limits of privacy for social networks. MIT Technology Review Physics arXiv Blog, http://www.technologyreview.com/view/418819/the-fundamental-limits-of-privacy-for-social, May 5, 2010.

[56] Federal Trade Commission. FTC charges deceptive privacy practices in Google's rollout of its Buzz social network. Press Release, http://www.ftc.gov/opa/2011/03/google.shtm, March 30, 2011.

[57] Federal Trade Commission. Facebook settles FTC charges that it deceived consumers by failing to keep privacy promises. Press Release, http://ftc.gov/opa/2011/11/privacysettlement.shtm, November 29, 2011.

[58] A. Fuxman, P. Tsaparas, K. Achan, and R. Agrawal. Using the wisdom of the crowds for keyword generation. In *WWW*, pages 61–70, 2008.

[59] G. Gates. Facebook privacy: A bewildering tangle of options. *The New York Times*, May 12, 2010.

[60] C. Ghiossi. Explaining Facebook's spam prevention systems. The Facebook Blog, https://blog.facebook.com/blog.php?post=403200567130, June 29, 2010.

[61] A. Ghosh, T. Roughgarden, and M. Sundararajan. Universally utility-maximizing privacy mechanisms. In *STOC*, pages 351–360, 2009.

[62] S. Gillmor. Facebook's glass jaw. *TechCrunch*, May 17, 2008.

[63] J. Ginsberg, M. H. Mohebbi, R. S. Patel, L. Brammer, M. S. Smolinski, and L. Brilliant. Detecting influenza epidemics using search engine query data. *Nature*, 457(7232):1012–1014, 02 2009.

[64] C. Gkantsidis, M. Mihail, and A. Saberi. Conductance and congestion in power law graphs. *SIGMETRICS Perform. Eval. Rev.*, 31(1):148–159, June 2003.

[65] J. Golbeck. Generating predictive movie recommendations from trust in social networks. In *ICTM*, pages 93–104, 2006.

[66] S. A. Golder and M. W. Macy. Diurnal and seasonal mood vary with work, sleep, and daylength across diverse cultures. *Science*, 333(6051):1878–1881, 2011.

[67] P. Golle. Revisiting the uniqueness of simple demographics in the US population. In *WPES: Proceedings of the 5th ACM workshop on Privacy in Electronic Society*, pages 77–80, 2006.

[68] Google Politics and Elections. Super Tuesday: Who's ahead and what's at stake? https://plus.google.com/114401727024677849167/posts/bTqc4msv3WA, March 6, 2012.

[69] M. Götz, A. Machanavajjhala, G. Wang, X. Xiao, and J. Gehrke. Publishing search logs - a comparative study of privacy guarantees. *IEEE Transactions on Knowledge and Data Engineering*, 24(3):520–532, March 2012.

[70] K. Greene. Who benefits from AOL's released search logs? MIT Technology Review, http://www.technologyreview.com/blog/editors/17312, August 15, 2006.

[71] J. V. Grove. Just married: Groom changes Facebook relationship status at the altar. http://mashable.com/2009/12/01/groom-facebook-update, December 1, 2009.

[72] R. Grover. Relationships and happiness. Facebook data blog, http://www.facebook.com/note.php?note_id=304457453858, February 14, 2010.

[73] S. Guha, B. Cheng, and P. Francis. Challenges in Measuring Online Advertising Systems. In *Proceedings of the Internet Measurement Conference (IMC)*, November 2010.

[74] S. Guha, B. Cheng, and P. Francis. Privad: Practical Privacy in Online Advertising. In *Proceedings of the 8th Symposium on Networked Systems Design and Implementation (NSDI)*, March 2011.

[75] A. Gupta, K. Ligett, F. McSherry, A. Roth, and K. Talwar. Differentially private combinatorial optimization. In *SODA*, pages 1106–1125, 2010.

[76] K. Hafner. Researchers yearn to use AOL logs, but they hesitate. *The New York Times*, August 23, 2006.

[77] J. Harper. It's modern trade: Web users get as much as they give. *The Wall Street Journal*, August 7, 2010.

[78] M. Hay, C. Li, G. Miklau, and D. Jensen. Accurate estimation of the degree distribution of private networks. In *ICDM*, pages 169–178, 2009.

[79] M. Hay, G. Miklau, D. Jensen, D. F. Towsley, and P. Weis. Resisting structural re-identification in anonymized social networks. *PVLDB*, 1(1):102–114, 2008.

[80] M. Hay, G. Miklau, D. Jensen, P. Weis, and S. Srivastava. Anonymizing social networks. *University of Massachusetts Amherst Technical Report No. 07-19*, 2007.

[81] M. Helft. Marketers can glean private data on Facebook. *The New York Times*, October 23, 2010.

[82] W. Hess. People you may know, 2008. http://whitneyhess.com/blog/2008/03/30/people-you-may-know.

[83] K. Hill. Social media background check company ensures that job-threatening Facebook photos are part of your application. *Forbes*, June 20, 2011.

[84] K. Hill. Facebook keeps a history of everyone who has ever poked you, along with a lot of other data. *Forbes*, September 27, 2011.

[85] T. Hogg. Inferring preference correlations from social networks. *Electronic Commerce Research and Applications*, 9(1):29 – 37, 2010.

[86] D. C. Howe and H. Nissenbaum. TrackMeNot: Resisting surveillance in web search. In I. Kerr, V. Steeves, and C. Lucock, editors, *Lessons from the Identity Trail: Anonymity, Privacy, and Identity in a Networked Society*, chapter 23, pages 417–436. Oxford University Press, Oxford, UK, 2009.

[87] Z. Huang, X. Li, and H. Chen. Link prediction approach to collaborative filtering. In *JCDL*, pages 141–142, 2005.

[88] N. Hunt. Netflix prize update. The Netflix Blog, http://blog.netflix.com/2010/03/this-is-neil-hunt-chief-product-officer.html, March 12, 2010.

[89] M. Ingram. News flash: Yes, Facebook is selling you to advertisers. GigaOM, http://gigaom.com/2011/12/23/news-flash-yes-facebook-is-selling-you-to-advertisers, December 23, 2011.

[90] M. Ingram. Being tracked by Google isn't bad it's actually good. GigaOM, http://gigaom.com/2012/03/02/being-tracked-by-google-isnt-bad-its-actually-good, March 2, 2012.

[91] L. Italie. Divorce lawyers: Facebook tops in online evidence in court. Associated Press, http://www.usatoday.com/tech/news/2010-06-29-facebook-divorce_N.htm, June 29, 2010.

[92] A. Jesdanun. Google, Facebook in stalemate over social data. *USA Today*, May 24, 2008.

[93] T. Joachims. Optimizing search engines using clickthrough data. In *KDD*, pages 133–142, 2002.

[94] T. Joachims, L. Granka, B. Pang, H. Hembrooke, and G. Gay. Accurately interpreting clickthrough data as implicit feedback. In *SIGIR*, pages 154–161, 2005.

[95] C. Y. Johnson. Project 'Gaydar'. *The Boston Globe*, September 20, 2009.

[96] M. Jones. Protecting privacy with referrers. Facebook Engineering's Notes, http://www.facebook.com/notes/facebook-engineering/protecting-privacy-with-referrers/392382738919, May 24, 2010.

[97] R. Jones, R. Kumar, B. Pang, and A.Tomkins. "I know what you did last summer": query logs and user privacy. In *CIKM*, pages 909–914, 2007.

[98] R. Jones, R. Kumar, B. Pang, and A. Tomkins. Vanity fair: Privacy in querylog bundles. In *CIKM*, pages 853–862, 2008.

[99] M. Kelly. Facebook security: Fighting the good fight. The Facebook Blog, http://blog.new.facebook.com/blog.php?post=25844207130, August 7, 2008.

[100] R. Kessler, M. Stein, and P. Berglund. Social Phobia Subtypes in the National Comorbidity Survey. *Am J Psychiatry*, 155(5):613–619, 1998.

[101] J. Kincaid. Senators call out Facebook on instant personalization, other privacy issues. *TechCrunch*, April 27, 2010.

[102] J. Kincaid. Live blog: Facebook unveils new privacy controls. *TechCrunch*, May 26, 2010.

[103] M. Kirkpatrick. Why Facebook's data sharing matters. ReadWriteWeb, http://www.readwriteweb.com/archives/why_facebooks_data_sharing_matters.php, January 13, 2012.

[104] J. Kleinberg. Navigation in a small world. *Nature*, 406(845), 2000.

[105] R. Knies. Making the web more user-friendly. Microsoft Research News, http://research.microsoft.com/en-us/news/features/www2009-042109.aspx, April 22, 2009.

[106] Y. Koren. Collaborative filtering with temporal dynamics. In *KDD*, pages 447–456, 2009.

[107] Y. Koren and R. M. Bell. Advances in collaborative filtering. In *Recommender Systems Handbook*, pages 145–186. Springer, 2011.

[108] Y. Koren, R. M. Bell, and C. Volinsky. Matrix factorization techniques for recommender systems. *IEEE Computer*, 42(8):30–37, 2009.

[109] A. Korolova. Privacy violations using microtargeted ads: A case study. *Journal of Privacy and Confidentiality*, 3(1):27–49, 2011.

[110] A. Korolova, K. Kenthapadi, N. Mishra, and A. Ntoulas. Releasing search queries and clicks privately. In *WWW*, pages 171–180, 2009.

[111] A. Korolova, R. Motwani, S. U. Nabar, and Y. Xu. Link privacy in social networks. In *CIKM*, pages 289–298, 2008.

[112] D. Kravets. Judge approves $9.5 million Facebook 'Beacon' accord. *Wired Magazine*, March 17, 2010.

[113] B. Krebs. Account hijackings force LiveJournal changes. *The Washington Post*, January 20, 2006.

[114] B. Krishnamurthy and C. E. Wills. On the leakage of personally identifiable information via online social networks. In *Proceedings of the 2nd ACM workshop on Online social networks (WOSN)*, pages 7–12, 2009.

[115] R. Kumar, J. Novak, B. Pang, and A. Tomkins. On anonymizing query logs via token-based hashing. In *WWW*, pages 629–638, 2007.

[116] R. Kumar, P. Raghavan, S. Rajagopalan, D. Sivakumar, A. Tomkins, and E. Upfal. Random graph models for the web graph. In *FOCS*, pages 57–65, 2000.

[117] R. Lawler. How Google plans to change the way you watch TV. GigaOM, http://gigaom.com/video/google-schmidt-tv, August 29, 2011.

[118] M. Learmonth. 'Power Eye' lets consumers know why that web ad was sent to them. Advertising Age, http://adage.com/digital/article?article_id=144557, June 21, 2010.

[119] K. LeFevre, D. J. DeWitt, and R. Ramakrishnan. Incognito: Efficient full-domain k-anonymity. In *SIGMOD*, pages 49–60, 2005.

[120] J. Leskovec, D. Chakrabarti, J. M. Kleinberg, and C. Faloutsos. Realistic, mathematically tractable graph generation and evolution, using Kronecker multiplication. In *PKDD*, pages 133–145, 2005.

[121] J. Leskovec, D. Huttenlocher, and J.Kleinberg. Predicting positive and negative links in online social networks. In *WWW*, pages 641–650, 2010.

[122] D. Liben-Nowell and J. Kleinberg. The link prediction problem for social networks. In *CIKM*, pages 556–559, 2003.

[123] G. Linden. A chance to play with big data. Geeking with Greg, http://glinden.blogspot.com/2006/08/chance-to-play-with-big-data.html, August 4, 2006.

[124] K. Liu, K. Das, T. Grandison, and H. Kargupta. Privacy-preserving data analysis on graphs and social networks. In H. Kargupta, J. Han, P. Yu, R. Motwani, and V. Kumar, editors, *Next Generation Data Mining*. CRC Press, 2008.

[125] S. Lohr. A $1 million research bargain for Netflix, and maybe a model for others. *The New York Times*, September 21, 2009.

[126] G. Loukides and A. Gkoulalas-Divanis. Privacy challenges and solutions in the social web. *Crossroads*, 16(2):14–18, Dec. 2009.

[127] H. Ma, I. King, and M. R. Lyu. Learning to recommend with social trust ensemble. In *SIGIR*, pages 203–210, 2009.

[128] A. Machanavajjhala, J. Gehrke, D. Kifer, and M. Venkitasubramaniam. l-diversity: Privacy beyond k-anonymity. In *ICDE*, page 24, 2006.

[129] A. Machanavajjhala, A. Korolova, and A. D. Sarma. Personalized social recommendations - accurate or private? *PVLDB*, 4(7):440–450, 2011.

[130] M. Madden and A. Smith. Reputation management and social media. http://www.pewinternet.org/Reports/2010/Reputation-Management.aspx, May 26, 2010.

[131] P. Marks. Noise could mask web searchers' ids. *New Scientist*, March 7, 2009.

[132] F. McSherry. Privacy integrated queries: an extensible platform for privacy-preserving data analysis. In *SIGMOD*, pages 19–30, 2009.

[133] F. McSherry and I. Mironov. Differentially private recommender systems: building privacy into the Netflix prize contenders. In *KDD*, pages 627–636, 2009.

[134] F. McSherry and K. Talwar. Mechanism design via differential privacy. In *FOCS*, pages 94–103, 2007.

[135] F. McSherry and K. Talwar. Private Communication, 2008.

[136] Microsoft. Beyond search: Semantic computing and internet economics workshop. http://research.microsoft.com/en-us/events/beyondsearch2009/default.aspx, June 10-11, 2009.

[137] M. Mihail, A. Saberi, and P. Tetali. Random walks with lookahead on power law random graphs. *Internet Mathematics*, 3(2), 2007.

[138] A. Mislove, K. P. Gummadi, and P. Druschel. Exploiting social networks for internet search. In *HotNets*, pages 79–84, 2006.

[139] A. Mislove, A. Post, K. P. Gummadi, and P. Druschel. Ostra: Leverging trust to thwart unwanted communication. In *NSDI*, pages 15–30, 2008.

[140] M. Montaner, B. López, and J. L. de la Rosa. Opinion-based filtering through trust. In *In Proceedings of the Sixth International Workshop on Cooperative Information Agents (CIA)*, pages 164–178, 2002.

[141] M. A. Moreno, D. A. Christakis, K. G. Egan, L. N. Brockman, and T. Becker. Associations between displayed alcohol references on Facebook and problem drinking among college students. *Arch Pediatr Adolesc Med*, 166(2):157–163, 2012.

[142] R. Motwani and P. Raghavan. *Randomized Algorithms*. Cambridge University Press, 1995.

[143] J. Mullock, S. Groom, and P. Lee. International online behavioural advertising survey 2010. Osborne Clarke, www.osborneclarke.com/~/media/Files/publications/import/en/international-online-behavioural-advertising.ashx, May 20, 2010.

[144] S. Nadarajah and S. Kotz. On the linear combination of Laplace random variables. *Probab. Eng. Inf. Sci.*, 19(4):463–470, 2005.

[145] A. Narayanan. About 33 bits. http://33bits.org/about, 2008.

[146] A. Narayanan, E. Shi, and B. I. P. Rubinstein. Link prediction by de-anonymization: How we won the Kaggle social network challenge. In *International Joint Conference on Neural Networks (IJCNN)*, pages 1825 –1834, 2011.

[147] A. Narayanan and V. Shmatikov. Robust de-anonymization of large sparse datasets. In *IEEE Symposium on Security and Privacy*, pages 111–125, 2008.

[148] A. Narayanan and V. Shmatikov. De-anonymizing social networks. In *IEEE Symposium on Security and Privacy*, pages 173–187, 2009.

[149] Nielsen Research. Ads with friends: Analyzing the benefits of social ads. http://blog.nielsen.com/nielsenwire/online_mobile/ads-with-friends-analyzing-the-benefits-of-social-ads, March 6, 2012.

[150] K. Nissim. Private data analysis via output perturbation. In *Privacy-Preserving Data Mining: Models and Algorithms*, pages 383–414. Springer, 2008.

[151] Office of the Information and Privacy Commissioner Ontario. Tie for top award - two winners exemplify innovative privacy research in two diverse areas. Canada Newswire, http://www.newswire.ca/en/story/818761/tie-for-top-award-two-winners-exemplify-innovative-privacy-research-in-two-diverse-areas.

[152] P. Ohm. Netflix's impending (but still avoidable) multi-million dollar privacy blunder. Paul Ohm's Blog, https://freedom-to-tinker.com/blog/paul/netflixs-impending-still-avoidable-multi-million-dollar-privacy-blunder, September 21, 2009.

[153] N. O'Neill. Barry Diller: we spend every nickel we can on Facebook. Interview to CNN Money http://allfacebook.com/barry-diller-we-spend-every-nickel-we-can-on-facebook_b15924, July 26, 2010.

[154] K. Purcell, J. Brenner, and L. Rainie. Search engine use 2012. Pew Internet & American Life Project, http://www.pewinternet.org/Reports/2012/Search-Engine-Use-2012.aspx, March 9, 2012.

[155] F. Radlinski and T. Joachims. Query chains: learning to rank from implicit feedback. In *KDD*, pages 239–248, 2005.

[156] V. Rastogi and S. Nath. Differentially private aggregation of distributed time-series with transformation and encryption. In *SIGMOD*, pages 735–746, 2010.

[157] A. Roth. What your Twitter profile reveals about you. Foundations of data privacy course blog, http://privacyfoundations.wordpress.com/2011/09/20/what-your-twitter-profile-reveals-about-you, September 20, 2011.

[158] I. Roy, S. T. V. Setty, A. Kilzer, V. Shmatikov, and E. Witchel. Airavat: Security and privacy for MapReduce. In *NSDI*, pages 297–312, 2010.

[159] T. Ryan and G. Mauch. Getting in bed with Robin Sage. Black Hat USA, https://media.blackhat.com/bh-us-10/whitepapers/Ryan/BlackHat-USA-2010-Ryan-Getting-In-Bed-With-Robin-Sage-v1.0.pdf, July 28, 2010.

[160] R. Salakhutdinov, A. Mnih, and G. Hinton. Restricted Boltzmann machines for collaborative filtering. In *ICML*, pages 791–798, 2007.

[161] P. Samarati and L. Sweeney. Generalizing data to provide anonymity when disclosing information. In *PODS*, page 188, 1998.

[162] S. Sandberg. The role of advertising on Facebook. The Facebook Blog, http://blog.facebook.com/blog.php?post=403570307130, July 6, 2010.

[163] P. Sarkar, D. Chakrabarti, and A. W. Moore. Theoretical justification of popular link prediction heuristics. In *COLT*, pages 295–307, 2010.

[164] B. Schnitt. Responding to your feedback. The Facebook Blog, http://blog.facebook.com/blog.php?post=379388037130, April 5, 2010.

[165] E. Schrage. Facebook executive answers reader questions. The New York Times Bits Blog, http://bits.blogs.nytimes.com/2010/05/11/facebook-executive-answers-reader-questions, May 11, 2010.

[166] S. Shankland. Facebook blocks contact-exporting tool. CNET, http://news.cnet.com/8301-30685_3-20076774-264/facebook-blocks-contact-exporting-tool, July 5, 2011.

[167] M. Siegler. Like Facebook, Twitter starts using algorithms to bulk up social graph. *TechCrunch*, July 30, 2010.

[168] A. Silberstein, J. Terrace, B. F. Cooper, and R. Ramakrishnan. Feeding frenzy: selectively materializing users' event feeds. In *SIGMOD*, pages 831–842, 2010.

[169] R. Singel. Netflix spilled your Brokeback Mountain secret, lawsuit claims. *Wired*, December 17, 2009.

[170] J. Smith. Facebook starts suggesting "People you may know". *TechCrunch*, March 26, 2008.

[171] B. Stone. Ads posted on Facebook strike some as off-key. *The New York Times*, March 3, 2010.

[172] S. Su. User survey results: Which ads do Facebook users like most (and least)? http://www.insidefacebook.com/2010/06/15/facebook-users-survey-results-ads, June 15, 2010.

[173] G. Swamynathan, C. Wilson, B. Boe, K. Almeroth, and B. Y. Zhao. Do social networks improve e-commerce?: a study on social marketplaces. In *WOSP*, pages 1–6, 2008.

[174] L. Sweeney. Uniqueness of simple demographics in the U.S. population. In *Carnegie Mellon University, School of Computer Science, Data Privacy Lab White Paper Series LIDAP-WP4*, 2000.

[175] L. Sweeney. k-anonymity: A model for protecting privacy. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 10(5):557–570, 2002.

[176] B. Szoka. Privacy MythBusters: No, Facebook doesn't give advertisers your data! http://techliberation.com/2010/07/06/privacy-mythbusters-no-facebook-doesnt-give-advertisers-your-data, July 6, 2010.

[177] G. Takács, I. Pilászy, B. Németh, and D. Tikk. Scalable collaborative filtering approaches for large recommender systems. *Journal of Machine Learning Research*, 10:623–656, 2009.

[178] B. Tancer. *Click: What Millions of People Are Doing Online and Why it Matters*. Hyperion, 2008.

[179] B. Taylor. The next evolution of Facebook platform. Facebook Developer Blog, http://developers.facebook.com/blog/post/377, April 21, 2010.

[180] V. Toubiana, A. Narayanan, D. Boneh, H. Nissenbaum, and S. Barocas. Adnostic: Privacy preserving targeted advertising. In *17th Annual Network and Distributed System Security Symposium (NDSS)*, 2010.

[181] TRUSTe. TRUSTe launches new privacy index measuring consumer privacy insights and trends. http://www.truste.com/about-TRUSTe/press-room/news_truste_launches_new_trend_privacy_index, February 13, 2012.

[182] J. Turow, J. King, C. J. Hoofnagle, A. Bleakley, and M. Hennessy. Americans reject tailored advertising and three activities that enable it. *Social Science Research Network*, September 29, 2009.

[183] H. R. Varian and H. Choi. Predicting the present with Google trends. *Social Science Research Network*, April 2, 2009.

[184] J. Vascellaro. Google agonizes on privacy as ad world vaults ahead. *The Wall Street Journal*, August 10, 2010.

[185] T. Vega. Ad group unveils plan to improve web privacy. *The New York Times*, October 4, 2010.

[186] H. von Schelling. Coupon collecting for unequal probabilities. *Am. Math. Monthly*, 61:306–311, 1954.

[187] S. L. Warner. Randomized response: A survey technique for eliminating evasive answer bias. *Journal of the American Statistical Association*, 60(309):pp. 63–69, 1965.

[188] S. Wasserman and K. Faust. *Social Network Analysis*. Cambridge University Press, Cambridge, USA, 1994.

[189] D. J. Watts and S. H. Strogatz. Collective dynamics of small-world networks. *Nature*, 393(6684):440–442, 1998.

[190] R. Wauters. Googlers buy more junk food than Microsofties (and why Rapleaf is creepy). *TechCrunch*, March 22, 2011.

[191] J. Weng, C. Miao, A. Goh, Z. Shen, and R. Gay. Trust-based agent community for collaborative recommendation. In *Proceedings of the fifth international joint conference on Autonomous agents and multiagent systems (AAMAS)*, pages 1260–1262, 2006.

[192] B. Womack. Facebook advertisers boost spending 10-fold, COO says. *Bloomberg*, August 3, 2010.

[193] J. Wortham. The Facebook resisters. *The New York Times*, December 13, 2011.

[194] L. Xiong and E. Agichtein. Towards privacy-preserving query log publishing. In *Query Log Analysis: Social And Technological Challenges Workshop in WWW*, 2007.

[195] J. Yan, N. Liu, G. Wang, W. Zhang, Y. Jiang, and Z. Chen. How much can behavioral targeting help online advertising? In *WWW*, pages 261–270, 2009.

[196] T. Zeller Jr. AOL technology chief quits after data release. *The New York Times*, August 21, 2006.

[197] E. Zheleva and L. Getoor. Preserving the privacy of sensitive relationships in graph data. In *Proceedings of the International Workshop on Privacy, Security and Trust in KDD*, 2007.

[198] B. Zhou and J. Pei. Preserving privacy in social networks against neighborhood attacks. In *ICDE*, pages 506–515, 2008.

[199] C.-N. Ziegler and G. Lausen. Analyzing correlation between trust and user similarity in online communities. In *ICTM*, pages 251–265, 2004.

[200] M. Zimmer. "But the data is already public": on the ethics of research in Facebook. *Ethics and Information Technology*, 12(4):313–325, 2010.

[201] M. Zuckerberg. From Facebook, answering privacy concerns with new settings. *The Washington Post*, May 24, 2010.