ABSTRACT

Title of dissertation: PREDICTION, EVOLUTION AND PRIVACY IN SOCIAL AND AFFILIATION NETWORKS

Elena Zheleva, Doctor of Philosophy, 2011

Dissertation directed by: Professor Lise Getoor
Department of Computer Science

In the last few years, there has been a growing interest in studying online social and affiliation networks, leading to a new category of inference problems that consider the actor characteristics and their social environments. These problems have a variety of applications, from creating more effective marketing campaigns to designing better personalized services. Predictive statistical models allow learning hidden information automatically in these networks but also bring many privacy concerns. Three of the main challenges that I address in my thesis are understanding 1) how the complex observed and unobserved relationships among actors can help in building better behavior models, and in designing more accurate predictive algorithms, 2) what are the processes that drive the network growth and link formation, and 3) what are the implications of predictive algorithms on the privacy of users who share content online.

The majority of previous work in prediction, evolution and privacy in online social networks has concentrated on the single-mode networks which form around user-user links, such as friendship and email communication. How-

ever, single-mode networks often co-exist with two-mode affiliation networks in which users are linked to other entities, such as social groups, online content and events. I study the interplay between these two types of networks and show that analyzing these higher-order interactions can reveal dependencies that are difficult to extract from the pair-wise interactions alone. In particular, I present my contributions to the challenging problems of collective classification, link prediction, network evolution, and preserving privacy in social and affiliation networks. I evaluate my models on real-world data sets from well-known online social networks, such as Flickr, Facebook, Dogster and LiveJournal.

PREDICTION, EVOLUTION AND PRIVACY IN SOCIAL AND AFFILIATION NETWORKS

by

Elena Zheleva

Dissertation submitted to the Faculty of the Graduate School of the
University of Maryland, College Park in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
2011

Advisory Committee:
Professor Lise Getoor, Chair/Advisor
Professor Jennifer Golbeck
Professor David Jacobs
Professor Jordan Boyd-Graber
Professor Min Wu

Acknowledgments

I am grateful to all the people who have helped me in completing my Ph.D. studies and dissertation.

First and foremost, I would like to thank my advisor, Prof. Lise Getoor, for giving me the opportunity to explore my research interests and to work on intellectually stimulating projects. She is the reason why I took interest in machine learning and social networks, and she has taught me the importance of setting high standards for my work. She has been an invaluable resource as an individual and a professional, and she has helped me in putting structure in my ideas and work as a whole. I cannot imagine a better advisor to work with. She has been very approachable and easily available for discussions, and I have always felt that she has my best interest in mind.

Besides being fortunate to have an amazing advisor, I am also grateful for the opportunity to work with passionate and knowledgeable research collaborators - Aleksander Kolcz, Jen Golbeck, Ugur Kuter, Hossam Sharara, Natasa Milic-Frayling, Eduarda Mendes Rodrigues, John Guiver, Sunita Sarawagi, and Evimaria Terzi. I have learned from each and every one of you, and it has been a pleasure working with you.

My friends and fellow computer science researchers have given me peer and friendship support, as well as perspectives on the various choices I had to make along the way. I am most indebted to Michael Schatz who besides always being there to bounce ideas off of and listen to me, also stepped in and helped

me at a crucial moment in my work with his expertise and experience in integrating code from various sources. I would also like to thank Olga Nikolova, Evdokia Nikolova, Nataliya Guts and Grecia Lapizco-Encinas, brilliant young women who were more or less at the same stages in their Ph.D. studies as me. They have helped in keeping my spirits high on numerous occasions. In this list are also my colleagues from the LINQS research group who gave me the feeling that I was a part of a community with common goals. Many thanks to the successfully graduated Indrajit Bhattacharya, Rezarta Islamaj, Prithviraj Sen and Mustafa Bilgic for setting an example for me, and to the rest of the group, especially Galileo Namata for his help with GAIA.

I am thankful for having friends whose emotional support has provided me with the comfort of a close social environment. Thank you, Stela, Toni, Vesi, Ina, Ani, Lidia, Antonina, Flo, Ilya, Marko, Gustavo, the members of the folk-dancing group Zharava and many others.

Last but not least, I thank my family for encouraging me to pursue whatever career path I desire, and for supporting me unconditionally in the process. I especially thank my mother Milka and my brother Stanimir – without you, this journey would have not been possible and worthwhile.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

In the last few years, with the myriads of social media and social network websites appearing online, there has been a renewed and growing interest in understanding social phenomena rising from people's interactions and affiliations [3, 38]. These websites have thousands, and even millions of users which voluntarily submit personal information in order to benefit from the services offered, such as maintaining friendships, blogging, sharing photos, music, articles and so on. This rich information can be used in a variety of ways to study peoples personal preferences, patterns of communication and flow of information.

Social network analysis (SNA) as a field started at the end of the nineteenth century and it analyzes the network of connections between individuals in order to evaluate and quantify an individual's role in a group or community [44]. One of the central ideas in it are that people develop their identities by interacting with other people which makes studying their social environment very important. People's personal attributes and roles often correlate with the ones

of the people with whom they associate, e.g., teenagers are likely to be friends with other teenagers, graduate students often collaborate with their research advisors, company employees often write emails to their direct managers. Nowadays, SNA involves collecting massive amounts of data from multiple sources, analyzing the data to identify relationships and mining it for new information [3].

Data mining, machine learning and statistics researchers have been using statistical methods for decades to model and study interesting patterns in data [109, 18, 51], and there has been an increasing interest in developing methods that work specifically for network data. Traditionally, machine learning has studied data instances which are independent and identically distributed. With the growth of the World Wide Web and the emergence of online social networks, there has been an increasing interest in developing algorithms which can take advantage of the dependencies between data instances. At the intersection of statistical methods with network analysis, come the fields of link mining and statistical relational learning. Link mining is a part of data mining which studies descriptive or predictive models of data which take advantage of the explicit links between data instances [46, 149]. Statistical relational learning (SRL) [47] is a part of artificial intelligence and machine learning which models uncertainty in rich, relational domains, often by combining probabilistic graphical models [69] with first-order logic. Common tasks in link mining and SRL include link-based classification, ranking, entity resolution, group detection, and link prediction, to mention a few. Goldenberg et al. [50] provide a survey on statistical network models.

Most of the work in link mining for online social networks has focused on

Figure 1.1: A hypothetical Facebook profile.

studying the network around actor-actor links such as friendships. However, social media data is usually rich and in addition to the links between users, it includes links of users to other types of objects, such as groups, events, community pages and preferred content. We call these links affiliation links, and the contribution of our work is in overlaying social and affiliation networks, and developing algorithms for analyzing the two. We will argue that studying both explicit and hidden user characteristics in this setting can give us a more complete profile of social network users and lead to the development of better predictive algorithms. Before we introduce the specific problems which we address in this thesis, we present the data model used throughout.

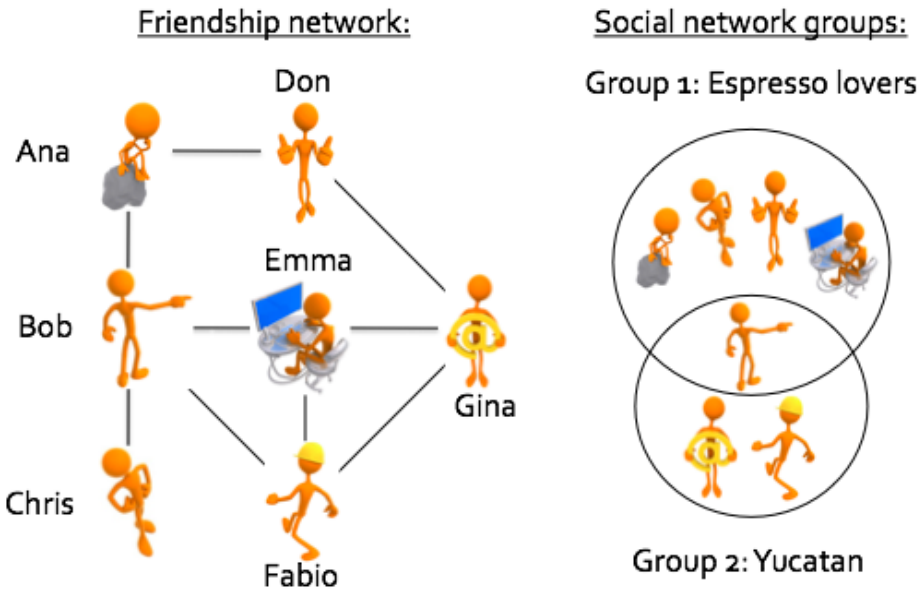


Figure 1.2: A toy social and affiliation network.

1.1 Data model

In the context of this thesis, when we refer to social networks, we generally mean online social networks. This includes online sites such as Facebook, Flickr, LinkedIn, etc., where individuals can link to, or "friend," each other, and which allow rich interactions such as joining communities or groups of interest, or participating in discussion forums. These sites often also include online services which allow users to create profiles and share their preferences and opinions about items, such as tagging articles and postings, commenting on photos, and rating movies, books or music. Thus, we view a social network as a multi-modal graph in which there are multiple kinds of entities, including people, groups, and items, but where at least one type is an individual and the links between individuals represent some sort of social tie. Each node of an individual has a profile, and profiles can have personal attributes, such as age, gender, political affiliation,

etc.

We concentrate on networks which have two types of commonly occurring links - user-user links, and user-group links. More formally, we represent the social network as a graph $G = (V, E_v, H, E_h)$, where V is a set of n nodes which represent user profiles, such as the one in Figure 1.1. Each node can have a set of attributes $v.A$. An edge $e_v(v_i, v_j) \in E_v$ is a *social link* and represents a relationship between the nodes v_i and v_j such as friendship. Relationships can be of different types (such as in a multiplex network), and there can be more than one relationship between a pair of users. We use H to denote both formal online groups and other online content for which users have preference, such as photos, movies, fan pages, etc. We refer to H as affiliation groups. An edge $e_h(v_i, h_j) \in E_h$ represents an *affiliation link* of the membership of node v_i to affiliation group h_j. Social links, affiliation links and groups also can have attributes, $e_v.A$, $e_h.A$ and $h.A$, respectively. We also define P to be a set of real-world entities which represent actual people.

As a running example, we consider the social network presented in Figure 1.2. It consists of seven profiles which describe a collection of individuals (Ana, Bob, Chris, Don, Emma, Fabio, and Gina), along with their friendship links and their affiliation groups of interest. Users are linked by a friendship link, and in this example they are reciprocal. There are two groups that users can participate in: the "Espresso lovers" affiliation group and the "Yucatan" affiliation group. These individuals also have personal attributes on their profiles: name, age, gender, zip code and political views (see Figure 6.3 on page 151). User-group

affiliations can also be represented as a bipartite graph, such as the ones in Figure 6.4 (page 167) and Figure 6.5(a) (page 168).

1.2 Organization

In this thesis, we present our work in the area of statistical modeling for social and affiliation networks which spans three related areas. The first part, on predictive modeling, looks at the problems of inferring the personal attributes and latent preferences of users. The second part looks at models for the growth and evolution of these two types of networks. While predictive statistical models allow learning hidden information automatically in these networks, they also bring many privacy concerns because of the potentially sensitive nature of personal data. We discuss some of the privacy issues with predicting personal information using social and affiliation networks in the third part of our thesis. Next, we give a brief overview of the three parts of the thesis. The introduction to each of the parts gives a broader perspective, and puts each work in the context of related work.

Accurate predictive algorithms are highly desired as they allow social media providers to create more complete user profiles, and they can be used for a variety of applications, from ad targeting to offering personalized services. In Part I of my thesis, I propose algorithms for inferring personal attributes of users considering two scenarios. In the first scenario, the social network data contains some users for which the attribute values, e.g. gender, political views, location,

are known, and the algorithm has to infer the values for the other users. In the second scenario, we are interested to infer a user attribute that is not directly observable for any of the users, e.g., users' taste in music, books or music, or whether a user is shy. I call this type of attribute latent user characteristics.

Supervised learning, or classification, refers to learning a machine learning model from labeled training data in order to infer the labels of unknown, testing data. For inferring personal attributes, we study probabilistic graphical models for relational classification in networks. In Chapter 2 we look at social and affiliation networks in which friendship links and group membership can be used to infer hidden attributes in a collective inference framework. We explored different ways of using the social groups as either node features or to construct a higher-order Markov Random Field (MRF) structure [156]. The bottleneck in applying higher-order MRFs to a domain with many overlapping large cliques is the complexity of inference which is exponential in the size of the largest clique. To circumvent the slow inference problem, we use graph-cut based methods to achieve fast approximate inference results. Our results on a Facebook dataset suggest that our higher-order MRF models are capturing the important structural dependencies in the networks and improve model accuracy.

Sometimes, we are interested in learning user characteristics which emerge from aggregating data patterns across users and for which there is no labeled data. Besides studying classification for predicting missing or unknown user attributes in the supervised setting, we have studied probabilistic graphical models for discovering latent user characteristics in an unsupervised setting. In Chap-

ter 3, we present statistical models for describing patterns of song listening in online music communities [157] and for discovering the latent taste and mood of users. First, we adapt a topic model, namely the Latent Dirichlet Allocation (LDA) model [20] to capture music taste from listening activities across users and identify both the groups of songs associated with the specific taste and the groups of listeners who share the same taste. Second, we define a graphical model that takes into account listening sessions and captures the listening moods of users in the community. Our session model leads to groups of songs and groups of listeners with similar behavior across listening sessions and enables faster inference when compared to the LDA model. Our experiments with the data from an online media site demonstrate that the session model is better in terms of the perplexity compared to two other models: the LDA-based taste model that does not incorporate cross-session information and a baseline model that does not use latent groupings of songs.

A common property of online social and affiliation networks is that they are dynamic and change over time – new users join the network and link to other users, new groups are formed and users form affiliations to these groups. Newman et al. give an overview of the research literature on the structure and dynamics of networks [119]. Part II of the thesis studies the processes of evolution and link formation which is crucial in understanding what drives the users' engagement, and consequently, the growth of online social and affiliation networks.

In Chapter 4, we address the problem of modeling social network generation which explains both link and group formation [158]. Recent studies on

online social network evolution propose generative models which capture the statistical properties of real-world networks related only to node-to-node link formation. We propose a novel model which captures the co-evolution of online social and affiliation networks. We provide surprising insights into group formation based on observations in several real-world networks, showing that users often join groups for reasons other than their friends. Our experiments show that the co-evolution model is able to capture both the newly observed and previously studied network properties. This work is the first to propose a generative model which captures the statistical properties of these complex networks. The proposed model facilitates controlled experiments which study the effect of actors' behavior on the network evolution, and it allows the generation of realistic synthetic datasets.

A related link mining task is predicting which users are likely to form a link, which is referred to as link prediction. Link prediction can be posed as a supervised classification task which considers node and structural attributes as features. Chapter 5 presents how social and affiliation networks can be overlaid to perform better link prediction, and proposes a feature taxonomy for link prediction in this setting [155]. We show that when there are tightly-knit groups in a social network, the accuracy of link prediction models can be improved. This is done by making use of the likely structural equivalence of participants in the groups. Our experiments on a trio of interesting real-world social networks demonstrate significantly higher prediction accuracy (between 15% and 30% more accurate) as compared to using more traditional features such as de-

scriptive node attributes and structural features.

While discovering hidden knowledge in social networks is a compelling application of machine learning and graph mining algorithms, this knowledge often relies on potentially sensitive data. As social networks are becoming ubiquitous, the implications to people's privacy are not well understood. Privacy in social networks is a very young field which studies what constitutes an unauthorized intrusion in social networks and how to protect sensitive user information. In Chapter 6 I identify the research problems in privacy in social networks and give an overview of the research literature on this topic [154]. Part III discusses our contributions to the field which are in defining two privacy problems and potential ways to deal with them. Chapter 7 presents the problem of sensitive attribute inference in online social networks [153], and Chapter 8 – the problem of sensitive link re-identification in anonymized network data [152].

Our overview of the research literature on privacy in social networks appears as a book chapter in Social Network Data Analytics [3]. In it, we identify and formally define the possible privacy breaches and describe the privacy attacks that have been studied. We present definitions of privacy in the context of privacy mechanisms and anonymization together with existing anonymization techniques.

The problem of sensitive attribute inference occurs naturally in online social networks where people have different privacy preferences. While many social media websites allow users to hide their personal profiles from the public in order to address privacy concerns, this is often not sufficient for guarding personal

information. In our work, we have shown how an adversary can exploit an online social network with a mixture of public and private user profiles to predict the private attributes of users [153]. We map this problem to a relational classification problem and we propose practical models that use friendship and group membership information (which is often not hidden) to infer sensitive attributes. The key novel idea is that in addition to friendship links, groups can be carriers of significant personal information. We show that on several well-known social media sites, we can easily and accurately recover the information of private-profile users. To circumvent this problem, social network users need to be informed of their privacy risks and social networks need to provide users with means to protect their data online. This work is complementary to Chapter 2 because it uses groups for classification though it does not consider higher-order probabilistic graphical models.

The other problem we have studied is how a data provider can preserve the privacy of sensitive relationships in graph data [152]. We refer to the problem of inferring sensitive relationships from anonymized graph data as link re-identification. We propose five different anonymization strategies, which vary in terms of the amount of data removed (and hence their utility) and the amount of privacy preserved. We assume the adversary has an accurate predictive model for links, and we show experimentally the success of different link re-identification strategies under varying structural characteristics of the data.

In summary, my thesis proposes new methods for overlaying networks formed around social and affiliation links of users. The contributions of the thesis

are in studying the interplay between these two types of networks and showing that analyzing these higher-order interactions can reveal dependencies that are difficult to extract from the pair-wise interactions alone. We present the benefits of overlaying the networks in a variety of settings, including predictive modeling, evolution and privacy.

Part I

Prediction in Social and Affiliation Networks

Creating a successful, self-sustaining social media service is a challenge because of the complexity of social interactions that ensue once the service is in place. A broad range of issues related to this problem have been addressed in the literature on social networks, e-commerce, recommendations, rating, collaborative filtering, and personalization. Predictive and descriptive models for inferring personal attributes of users are a centerpiece in many personalized online services. In Chapter 2, we propose a classification framework for social and affiliation network data which contains users for which the attribute values, e.g. gender, political views, location, are observed, and others are unobserved, and we are interested in predicting the unobserved attributes [156]. In Chapter 3, we study the problem of inferring latent user characteristics that are not directly observable for any of the users [157]. The work in both thesis chapters relies on probabilistic graphical models. Before we present each of the two scenarios, we provide context for our work by discussing research related to our approaches.

Collective classification

In the last decade, there has been a growing interest in supervised classification that relies not only on the object attributes but also on the attributes of the objects it is linked to, some of which may be unobserved [47]. Link-based or collective classification breaks the assumption that data comprises of i.i.d. instances and it can take advantage of autocorrelation, the property that makes the class labels of linked objects correlated with each other. Social networks are one

of the domains where link-based classification can be applied because personal attributes and roles of connected people are often correlated. For example, political affiliations of friends tend to be similar, students tend to be friends with other students, etc. Comprehensive reviews of link-based classification can be found in the works by Sen et al. [131] and Bhagat et al. [16].

In a typical classification setting, there are data instances (or nodes) with two types of attributes - a class and features. The class is an attribute of the nodes that we are interested to predict, e.g., gender. Features are other attributes of the nodes, e.g. weight, height, and hair length, which are given as an input to the classifier. The nodes with observed class labels comprise the training data, and they can be used to train the classifier to distinguish between the possible class labels. What distinguishes link-based classification from traditional classification is that some of the nodes' features are based on the features or class labels of neighboring nodes in the network.

Sen et al. define collective classification as a combinatorial problem in which given the class labels of some nodes in the network, the task is to infer the class labels of the rest of the nodes. They identify two types of approaches to collective classification. The first type relies on local conditional classifiers to perform approximate inference. Iterative classification algorithms [92, 115] and Gibbs sampling-based algorithms [95, 101] fall into this category. Iterative classifiers rely on attributes of the nodes, some of which are observed and based on the labels of neighboring node, to infer an initial labeling of the unlabeled node and then use this labeling to update the node features. This process continues it-

eratively until the assignment of labels stabilizes or the iterations reach a pre-specified threshold. Some models use a simplified version of Gibbs sampling which also relies on local classifiers and iterates through each node and estimates its probability distribution conditioned on its neighbor labels.

The second type of approach to collective classification is based on defining a global optimization function. One type of model which relies on a global optimization function is a pairwise Markov Random Field (MRF) [137], an undirected probabilistic graphical model. In particular, each actor's attribute in the social network corresponds to a random variable in the MRF, and each actor-actor link is considered as a pairwise dependency between two random variables. Inference on the MRF can be used for classification of the missing attributes in the data.

Bhagat et al. [16] recognize a third category of methods for node classification in social networks which propagate node labels using random walks. The assumption in these models is that the probability of a node label is equal to the probability that a random walk starting from the node in question will end at another node in the network with that label.

Most of the work on collective classification for social networks concentrates on the network formed around the pairwise social links. In Chapter 2 we propose a framework for collective classification for inferring node attributes in the presence of group affiliations which induce dependencies which go beyond pairwise. Our method makes use of the higher-order dependencies induced by groups affiliations to build a higher-order MRF, and it distinguishes groups that

are relevant to classification based on group features such as size and homogeneity. Our work in Chapter 7 is similar in that it uses groups for classification in a privacy scenario though it does not consider higher-order probabilistic graphical models.

Latent variable statistical models

User experience in social media involves rich interactions with the media content and other participants in the community. In order to support such communities, it is important to understand the factors that drive the users' engagement. Unlike supervised learning, uncovering latent characteristics requires extracting knowledge from unlabeled data. In many cases, unsupervised learning involves clustering objects together based on their similarity. In social networks, clusters can be found based on attribute and/or structural information. In fact, attributes, roles and affiliations of people are sometimes caused by the presence of such hidden clusters or groups in the data [116, 64, 6].

One way to model hidden clusters is through latent variable models. They have been of particular interest to researchers who study large text corpora. Latent semantic analysis techniques provide a powerful means of identifying underlying topics as clusters of terms derived from document-word co-occurrences [32, 60]. Steyvers and Griffiths [135], as well as Blei and Lafferty [19] have written recent overviews of topic models. Goldenberg et al. also present topic models in the broader context of statistical network models [50].

One popular latent variable model is the Latent Dirichlet Allocation model

(LDA) [20]. It has been introduced to capture statistical properties of text documents in a collection and provide a compact document representation in terms of underlying topics. More precisely, the method assumes that each document is a mixture of latent topics and uses a three-level hierarchical graphical model to characterize the statistical relations among terms and documents, resulting in topics that are represented as clusters of words. We describe the model in more detail in Section 3.3.1.

The LDA model has gained popularity due to its simple but powerful structure, and it has been applied to other domains besides topic modeling. Zhang et al. [151] propose an LDA-based model for identifying latent structures in large networks, using topological features as the only input. They apply the model to identify communities in large social networks. A similar model for analyzing graph data is described by Henderson & Eliassi-Rad [58]. Based on LDA, Bhattacharya and Getoor derive a new model for entity resolution in relational domains [17].

There are other generative models that combine latent variable modeling and social network modeling in a single framework. One of the first models in this space is the stochastic blockmodel for directed graphs of Wang and Wong [142]. Stochastic blockmodels assume that the observed actor-actor links in social networks can be explained by latent communities which can be discovered [42]. Airoldi et al. [6] study mixed-membership stochastic blockmodels for clustering of relational data. It is assumed that the cluster assignment is related to the node attribute value in question. The infinite relational model discovers latent concepts

in relational data where the number of such concepts is not known a priori [64]. A related approach is the one by Neville and Jensen [116]. It discover hidden clusters or groups in the data which influence the attributes of the group members using a spectral clustering method based on node links in the data.

Some of the work in this space concentrates on citation and collaboration networks. The mixed-membership model of Erosheva et al. [39] discovers latent clusters based on the document abstracts and the citation network of documents. The Author-Topic model of Rosen-Zvi et al. assumes that the distribution of latent topics in a document are a mixture of the topic distributions of the document's authors [129]. The Author-Role-Topic model, proposed by McCallum et al. [99], discovers discussion topics in threaded conversations, conditioned on sender-recipient interactions. The Group-Topic model [100] discovers latent groups in a network and clusters of associated topics based on text. Dietz et al. [33] present a model for predicting citation influences. The Topic-Link LDA model of Liu et al. [89] combines the LDA model [20] with the mixed-membership model of Erosheva et al. [39] to discover latent topics.

In our work, we use hierarchical probabilistic graphical models to derive latent user characteristics in music communities. In particular, we represent the song-listening activities in terms of *latent tastes* and *latent listening moods* of the community that are derived from the logs of media usage. For the latent tastes characterization we adapted the LDA model to the song-listening activities. Every instance of song listening is modeled as a finite mixture over the underlying set of tastes which, in turn, correspond to the clusters of songs derived from the

listening patterns. For listening moods, we increased the complexity of the model by incorporating session information. As a result, we arrive at a novel hierarchical graphical model that exploits additional structure in the data and identifies latent moods as clusters of songs that emerge from the song-listening sessions across the community.

Chapter 2

Collective Classification with Groups

A common assumption in social network analysis is that one can infer a lot about people from their social environment. A useful task in this type of analysis is collective classification where the goal is to infer hidden attributes of the nodes, and the classification algorithm considers not only the local features of each node in the network but also the features and the class labels of its network neighbors [131]. In a social network, where nodes represent actors, the actor-actor links are used to boost the accuracy of local classifiers or even provide classification labels in the absence of local features. While most collective classification algorithms take advantage of the statistical dependencies induced by the actor-actor links, very little work has been on using actor groups of size larger than two.

Online affiliation networks contain information about groups that actors have formed over time. Unlike the clusters resulting from automatic graph clustering of the social network which make certain assumptions about what constu-

tutes a cluster, online groups are observed affiliations in which the actors have participated. They provide a clustering of the actors that is completely data driven and perhaps more informative than inferring groups based on actor-actor links. Affiliation networks have been shown to have a strong signal for predicting actor attributes [153].

The goal of our work is to provide a principled approach to classification using the available data in a model which overlays information from the social network and the affiliation network. We investigate the use of higher-order Markov Random Field models which exploit the structure of both social and affiliation networks to perform better classification. Our contributions include identifying an approach for defining higher-order MRFs based on multi-modal social networks and proposing a model selection method informed by the existing structure in the network.

Relational data, such as social networks, can be modeled as a pairwise Markov Random Field [137]. In particular, each actor's attribute in the social network is a random variable in the MRF, and each actor-actor link is considered as a pairwise dependency between two random variables. Inference on the MRF can be used for classification of the missing attributes in the data. To the best of our knowledge, MRFs which use not only the dependencies coming from the observed friendship links but also from the observed affiliations have not been applied to classification tasks in social networks. Yet, the affiliation network structure provides rich dependencies which go beyond pairwise.

One way of including information from the affiliation network is to intro-

duce a clique for each group. This approach has a number of challenges. First, in online social networks both the number and the size of groups can be very large, and inference on a dense network can be prohibitively slow. Therefore, it becomes extremely important to learn which groups should participate in the MRF structure. Second, many MRFs rely on approximate inference algorithms which have to be tailored to the domain of interest in order to perform well.

Within the computer vision community, there has been a growing body of work on higher-order MRFs [68]. For example, in image analysis, segmentation is an important task in which given pixel information, such as color and location, an algorithm aims to classify each pixel to one of a number of classes, e.g., tree, sky, ground. Rather than taking the picture as a vector of pixels, pairwise MRFs encode structural information by considering the dependencies between neighbouring pixels. This has been shown to improve classification because classes of neighboring pixels are often dependent on each other. However, pairwise MRFs tend to make mistakes on pixels that are on the edges of image segments, e.g., the border pixels separating tree and sky. Incorporating longer-range dependencies between the pixels leads to better solutions. Higher-order MRFs take care of such dependencies by considering overlapping segments from different segmentation algorithms as cliques [68]. Recently, the computer vision community has developed inference algorithms that are extremely efficient and can find optimal solutions for a class of models in polynomial time. We discuss them in Section 2.3.

2.1 Preliminaries

Online social networks, such as Facebook, Flickr, LinkedIn, etc, allow individuals to create a profile and link, e.g. "friend" each other, or "affiliate" by joining groups of interest. They include online services which allow users to set their preferences to online content, such as tagging articles, commenting on photos and rating movies, to mention a few.

2.1.1 Data model and graphical model

We distinguish between two types of graphs: 1) the data graph, which we refer as the network G, and 2) the graph of random variables which represents the graphical model. The social and affiliation network data $G = (\mathbf{V}, \mathbf{E}_v, \mathbf{H}, \mathbf{E}_h)$ consists of n actors $\mathbf{V}$ with attributes $\mathbf{V}.\mathbf{A}$, and two types of commonly occurring links - actor-actor links, $\mathbf{E_v}$, and actor-group links, $\mathbf{E_h}$. The groups can be overlapping and of various sizes. The graphical model consists of a vector of discrete random variables $\mathbf{X} = \{X_1, X_2, ..., X_n\}$ for the actor attribute we aim to classify. Each variable X_i can take on a number of class labels. A Markov Random Field model is an undirected graphical model which represents a family of probability distributions for a random variable vector $\mathbf{X}$ given by

$$Pr(\mathbf{x}) = \frac{1}{Z} exp(-\sum_{c \in C} \phi_c(\mathbf{x}_c))$$

where C is a set of cliques, ϕ_c is the potential function for clique c, and Z is the normalizing constant, known as the partition function. $E(\mathbf{x}) = \sum_{c \in C} \phi_c(\mathbf{x}_c)$ corresponds to the Gibbs energy of a possible variable assignment. Each clique consists of a fully connected set of variables which are statistically dependent. A potential function is a function which assigns a positive real number to each possible variable vector assignment in the clique, and we discuss specific potential functions in Section 2.2.3. In pairwise MRFs, the clique potentials are over pairs of variables whereas higher-order MRFs can have cliques of arbitrary size.

2.1.2 Problem description

Given a network G in which the values of attribute a are given for some observed nodes $\mathbf{V}_o$, we would like to find the hidden attribute for the rest of the nodes in the network, $\mathbf{V}_h$. We concentrate on the case where the group memberships and friendship links are given for all nodes, and there are no other node attributes. The incentive for this is to evaluate the worth of the dependencies expressed in the network structure alone.

To make this problem more concrete, we construct the graphical model. First, we partition the random variables into $\mathbf{X}_o$ and $\mathbf{X}_h$, corresponding to the nodes $\mathbf{V}_o$ and $\mathbf{V}_h$. We would like to find the most probable assignment of $\mathbf{X}_h$, given the assignment of $\mathbf{X}_o$. This corresponds to the maximum a posteriori (MAP)

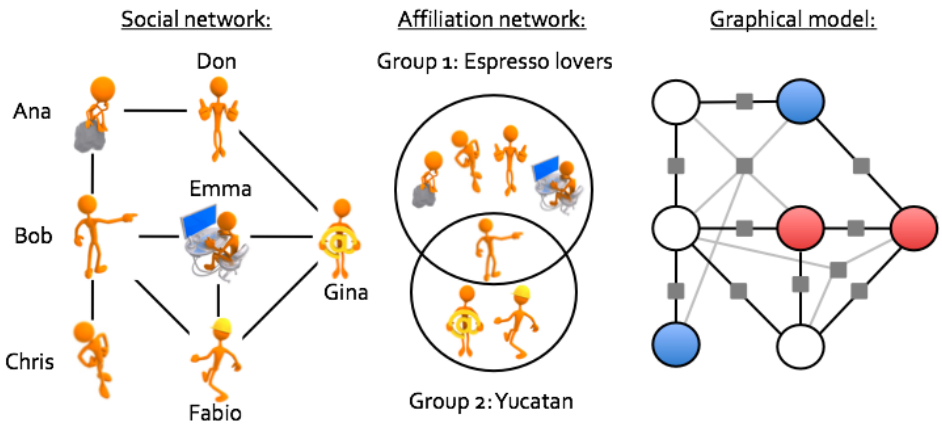


Figure 2.1: A toy social and affiliation network and its corresponding higher-order MRF represented as a factor graph.

estimation of $\mathbf{X}_h$:

$$\hat{\mathbf{X}}_h = \underset{\mathbf{x_h}}{\operatorname{argmax}}\, Pr(\mathbf{X}_h|\mathbf{x}_o) = \underset{\mathbf{x_h}}{\operatorname{argmin}}\, E(\mathbf{x_h}, \mathbf{x}_o)$$

Next we discuss how to construct the cliques in the graphical model.

2.2 Graph structure and potentials

Our solution first selects the MRF structure as discussed in Section 2.2.1, then it bootstraps the MRF model by computing informative node potentials described in Section 2.2.2. Then, using the node and clique potentials (in Section 2.2.3), it performs efficient MRF inference which we overview in Section 2.3.

2.2.1 MRF structure

There are different ways in which one can construct the graphical model by incorporating the structure of the network data. Here, we propose four different constructions. The most naïve way is to include all the friendship links as pairwise dependencies in the MRF. Each link $e_v(v_i, v_j)$ in the data corresponds to a clique of size two, $c(x_i, x_j)$. This incorporates the idea of homophily in social networks, or the tendency of individuals to associate with similar others, by making class labels of friends dependent on each other. This creates the pairwise Markov Random Field model, *pMRF*.

Another possibility is to include the affiliation network by treating each social group $h(\mathbf{V}^*)$ as a clique in the MRF $c(\mathbf{X}^*)$, where the random variables in the clique, $\mathbf{X}^*$, correspond to the group members $h(\mathbf{V}^*)$. This leads to our second model, the higher-order MRF with all groups, *hoMRF-AG*.

While including all groups may be an enticing idea, some of the social groups are more informative about certain actor attributes than others, e.g. women may be more likely to join a social group for breastfeeding advice than men. Following this idea, we look at group properties and select informative groups, which leads to our third model, higher-order MRF with selected groups (*hoMRF-SG*). We select the set of informative groups in the network based on their observed properties, such as size and entropy of the nodes with observed class labels. Our last model constructs the MRF by using both the pairwise dependencies from the friendship links and the higher-order cliques from selected social groups

(*hoMRF-SG-AL*).

Figure 2.1 shows an example social and affiliation network, together with its corresponding graphical model. The graphical model is presented as a factor graph to make the cliques (grey rectangles) over which the potentials are defined explicit. There are 7 actors with 9 friendship links and 2 social groups. The two social groups correspond to the two cliques of size 3 and 5 in the graphical model. Each link has a potential associated with it as well. The class labels of some of the actors are observed (shaded circles in the graphical model), while the labels of others are unknown (unshaded circles).

2.2.2 Node potentials

Each node in the MRF is a clique of size one, and it has a unary node potential. For each $X \in \mathbf{X}_h$, we compute the potential for each class value to be the negative log likelihood of the class value according to a linear classifier. The classifier, such as logistic regression or Naïve Bayes, uses the friendship links and group memberships as node features. Besides computing the node potentials, this classifier provides the baseline method in our experiments.

2.2.3 Clique potentials

Possible potentials for cliques of size larger than one are functions of the counts of class labels, such as majority and sum, negative/reciprocal entropy. We adopt the Robust P^n Potts clique potential of Kohli et al.[68] because it is intuitive

and it allows efficient inference. This potential is defined as:

$$\phi_c(\mathbf{x}_c) = \begin{cases} \gamma_k + \frac{N_i(\mathbf{x}_c)}{Q}(\gamma_{max} - \gamma_k) & \text{if } N_i(\mathbf{x}_c) \leq Q \\ \gamma_{max} & \text{else} \end{cases}$$

where γ_k is the minimum possible potential value if all labels in the clique are the same, and γ_{max} is the maximum possible potential value if the number of node labels that are different from the majority class label, $N_i(\mathbf{x}_c)$, is larger than a pre-specified threshold, called truncation ratio Q. For pairwise MRFs, this potential simplifies to the Potts potential. The intuition behind the Robust P^n Potts potential is that it allows disagreement between class labels inside each clique to a certain extent. Besides being intuitive, this potential is important for efficient inference using graph-cut based methods which we discuss next.

2.3 Inference and energy minimization

Exact inference in higher-order MRF models is exponential in the size of the largest clique. There are a number of approximate inference algorithms, e.g., belief propagation, variational inference, MCMC-based techniques, which aim to alleviate the complexity burden [63]. In the computer vision community, graph cut based methods have gained popularity because they have a polynomial complexity when assuming certain potential functions, such as Robust P^n Potts potential, and they work efficiently in practice [68]. Kohli et al. [68] compare the running

time and accuracy of tree-reweighted message passing (TRW-S) [70] with move-making inference algorithms which use graph cuts for models with large cliques. They find that the move-making algorithms are faster and yield better solutions.

A move-making algorithm starts from an initial solution and it makes a series of moves leading to lower energy solutions. At each step, it searches for the best possible move within its allowed range and then makes that move. The algorithm converges when it reaches a state from which it cannot find a lower energy solution.

Two move-making inference algorithms are α-expansion and α-β swap [21]. A move can be encoded as a vector of binary variables $\mathbf{t}$, one for each unobserved random variable in the hoMRF, $X_i \in \mathbf{X_h}$. In an α-expansion move, each random variable X_i either retains its current label if $t_i = 1$, or changes it to α if $t_i = 0$. In an α-β swap move, each random variable $X_i \in \mathbf{X_h}$ with a current label of α or β can either retain/change to a label of α if $t_i = 0$, or retain/change to a label of β if $t_i = 1$. One iteration of the algorithm searches through the space of possible move vectors to find the one that would lead to the lowest energy solution and then it makes that move.

Finding the optimal move vector for both the expansion and swap algorithms can be computed in polynomial time using graph cuts. For details, we refer the reader to Kohli et al. [68]. According to the same authors, the best ordering of moves is an ongoing research topic.

2.4 Experiments

2.4.1 Data description

For our evaluation, we studied a dataset from the social network Facebook[1], available for research purposes [78]. Facebook allows users to communicate with each other, to form undirected friendship links and participate in groups and events. The dataset contains all $1,225$ profiles of first-year students in a small college who share at least one interest group with another first-year student according to their Facebook profiles. The interest groups are the favorite books, music and movies of the users. There are $2,932$ groups, and the largest one has 290 members. There are $51,389$ friendship links in the data. The attribute we are trying to predict is the gender of each student. Half of the students are female, so a random guess would achieve an accuracy of 50%.

2.4.2 Experimental setup

We provide results for two-fold cross validation. The node potentials are computed using the java version of the liblinear logistic regression classifier [150]. For the move-making inference, we adapt the implementation of Kohli et al. [68]. For selecting the groups to be included as cliques in the MRF, we vary the allowed size, entropy and percent of observed nodes per group. First we performed a coarse-grained search through the space of parameters by setting the minimum group size to $\{2, 4, 6, 10\}$, maximim group size to $\{10, 50, 290\}$, maxi-

[1] At `http://www.facebook.com`.

mum entropy of the nodes with observed class labels to $\{0, 0.5, 0.7, 0.9, 1\}$, and the minimum percentage of nodes with observed class labels in the group to $\{0, 0.5\}$. This yields a space of 120 experiment points, e.g. point $(10, 290, 0, 0.5)$ means all groups of size between 10 and 290 with entropy of 0 and at least 50% of node labels observed. To obtain further improvement, we performed a fine-grained search around the parameters that yielded the best accuracy in the coarse-grained search.

We set γ_{max} to 10, γ_k to 0 for all possible labels, and the truncation ratio Q to 0.3 after some limited exploration of the parameter space. We set the node potentials to the negative log probabilities of the class labels coming out of the linear classifier. In the case of probability of 0, we set it to 10 (which is close to the negative log of the smoothed out probability). We report on three types of node features: friendship link vector, group membership vector and a vector which includes both. We compare the results for the linear classifier (*LR*), the pairwise MRF (*pMRF*) and the variants of the higher-order MRF: *hoMRF-AG*, *hoMRF-SG* and *hoMRF-SG-AL*.

2.4.3 Results

Table 2.1 summarizes the results from our experiments. The baseline linear classifier which uses the friendship link vector as features to classify nodes yielded an accuracy of 64.06%. Using the group memberships, this accuracy increases to 71.67%. Using both types of features, the accuracy is the highest,

Table 2.1: Accuracy of the logistic regression baseline (LR), the pairwise MRF (*pMRF*), the higher-order MRF with all groups as cliques (*hoMRF-AG*), with selected groups as cliques (*hoMRF-SG*) and with selected groups and all friendship links as cliques (*hoMRF-SG-AL*).

FEATURES	LR	PMRF	HOMRF-AG	HOMRF-SG	HOMRF-SG-AL
Friendships	64.06%	64.31%	69.13%	69.22%	69.22%
Groups	71.67%	71.83%	69.80%	74.12%	74.53%
Both	75.75%	75.84%	69.63%	77.39%	78.37%

75.75%. Our observations on the comparison between the linear classifier and our proposed models can be summarized as follows:

1) Using all groups naïvely as the cliques in the hoMRF (*hoMRF-AG*) improves performance only when the node potentials are bootstrapped with the friendship links as features alone. In the other two cases, where the node potentials use the group memberships as features, *hoMRF-AG* is not able to exploit the affiliation network structure further and it even hurts performance.

2) *pMRF* improves accuracy only marginally ($0.09 - 0.25\%$) compared to the baseline.

3) Adding selected groups as cliques in the MRF increases the prediction accuracy in all cases ($1.64 - 5.16\%$). Moreover *hoMRF-SG* consistently outperforms *LR* for the different folds of the cross validation. This means that the higher-order MRF is able to exploit the affiliation structure twice, once as features in the node potentials, and a second time by using informative groups as cliques. We report on the group selection experiment with the highest average accuracy in the *hoMRF-SG* column of Table 2.1.

4) *hoMRF-SG-AL* which adds the friendship links as pairwise cliques to

hoMRF-SG did not improve the accuracy of *hoMRF-SG* when using friendship links as features. However, when the node potentials were using group memberships or both types of features, the accuracy of *hoMRF-SG-AL* is higher than *hoMRF-SG*, by 0.41% and 0.98%, respectively.

The common theme in the parameter values for the best performing *hoMRF-SG* is that the selection criteria based on entropy are irrelevant to accuracy and that very small groups are uninformative. More concretely, for the friendship link features, the experiment points of group selection which yielded the highest accuracy were $(5, \{30, 40, 50\}, any, 0)$. In the strictest case (one with smallest number of groups as cliques), $(5, 30, 0, 0)$, with only 290 cliques out of the $2,932$ groups, it is possible to achieve 5.16% improvement from the baseline. Similarly, the experiment points with the highest accuracy (74.12%) for the group membership features is $(6, \{40, 50\}, any, 0)$. In its strictest case, $(6, 40, 0, 0)$, this includes an average of 201 cliques out of the $2,932$ groups. Lastly, the highest accuracy using both types of features in the *hoMRF-SG* was at experiment points $(8, 30, any, 0)$. In its strictest case, $(8, 30, 0, 0)$, this includes an average of only 100 cliques. Learning the best parameters from data is left for future work.

We also experimented with setting weights for the clique potentials based on the feature weights of the linear classifier since the node features and the graphical model cliques have a one-to-one correspondence. However, this did not provide any increase in accuracy.

The approximate inference in the hoMRF is very fast, and it takes less than 2 seconds to run on our dataset using a machine with 3.2 GHz processor and 3

Gb of RAM.

2.5 Conclusion

This is a preliminary study on the application of higher-order MRFs to classification in social and affiliation networks. We used recent advances in the computer vision community to ensure fast and accurate approximate inference results. In this study, we were relying on the given, noisy structure of the data to find the graphical model structure using feature selection criteria based on the group properties. In our future work, we would like to explore principled structure learning algorithms which incorporate the knowledge of the existing structure in the network data in various ways. In addition, we would like to apply this method to other real-world and synthetic datasets, in order to understand its properties better.

Chapter 3

Discovering Latent User Characteristics

Next, we present our work on discovering latents user characteristics in social media in the context of online music communities [157]. With broad proliferation of online social networks around media content, there is an increased interest in analyzing interactions among users and characterizing their behavior in terms of the individuals' and community preference for specific types of content. Among the popular and ever-growing social media sites centered around music are Last.fm, Zune Social, Flotones, JamNow, Haystack, Midomi, Sellabound, MySpace, Mercora radio, iLike, MusoCity, Sonific, and iJigg. Many of them include features that encourage social interactions by providing personalized recommendations to influence media selection of individuals. Furthermore, they offer community-based recommendations and interfaces for browsing and searching for available content.

For such complex systems, it is important to develop techniques that can be used to describe and study processes that drive the observed user engagement. Such methods need to be able to handle large-scale data logs from social media services and, therefore, produce effective representations of media consumption in order to enable efficient processing. In this chapter we use the example of music listening to demonstrate how that objective can be achieved. We illustrate an effective representation of usage data that can be applied to enhance individual user's experience, e.g., by recommending songs for the user's playlist that are relevant for the current music-listening session. Considering the large number of users and songs, such contextual recommendations require highly compact data representations.

Selecting a suitable song descriptor is an important initial step. We observe that many media services provide a static taxonomy of media types or *genre*. Such taxonomies serve as the means for individuals to express their interests and find adequate media. They provide media categories that are commonly adopted by the user community and, thus, could be used to characterize user's song-listening behavior, e.g., as a probability distribution over clusters of same-genre songs. The genre also captures an essential aspect of the song-listening process: while a person may not necessarily wish to repeat the same song, the person is likely to choose the next song to play from the same or a related genre.

On the other hand, even basic genre taxonomies may have a large number of categories and lead to sparse and ineffective representations of listening patterns. Thus, we aim to create a compact representation of media listening that retain

the essential statistical properties and relations among data. For that purpose we choose to derive generative probabilistic models based on the logs of song-listening and control the number of the underlying media clusters.

The contributions of our work are:

- A systematic approach to characterizing social media processes that drive music listening patterns
- A novel graphical model which provides a compact representation of the media based on listening sessions
- A model that has better predictive properties and enable faster inference than other known models.

More precisely, we define graphical models with latent variables that are intuitive and appropriate for modeling song listening. The first model captures the collective *music taste* as a set of tastes or media preferences that a particular community develops. We use them to characterize song listening by an individual user as a finite mixture of the underlying tastes. The second model captures the *listening moods* across listening sessions of the users in the community. In such a model, an instance of song listening by a user is described as a finite mixture of the underlying *set of listening moods*. In both cases we can vary the model parameters and explore the effect that different number of derived tastes and moods have on the model quality. In particular, we demonstrate the computational efficacy and compare the perplexity of the two models.

Our work is the first to utilize a hierarchical graphical model to incorpo-

rate listening moods based on session information. By applying the models to half a million song-listening instances from the Zune Social[1] music community, we demonstrate a clear advantage of using a more refined model to achieve both better perplexity for the co-occurrence of genres in sessions and higher computational efficiency. Although we introduced and evaluated it in the context of song listening, the same model can be applied to a broad range of scenarios, from browsing sessions on YouTube or Flickr to characterizing the sentiment and topics of blog-posting within given periods of time.

In the following Section 6.4 we provide background on graphical models. We then discuss the social media context in Section 3.2. In Section 7.3 we describe the data and define the hierarchical graphical models. In Section 8.5 we present experimental results and then reflect on broader implications of our work in Section 7.5. We conclude with a summary of our contributions and directions for further work.

3.1 Background

Here we provide context for our work by discussing related research on user modeling and song recommendations. Then we provide background information that is pre-requisite for the models we explore.

[1] `http://social.zune.net/`.

3.1.1 Related work

User modeling

An individual's taste and mood are two factors that are likely to influence consumption of media and social interactions. Thus, characterizing them in an effective manner is invaluable for personalizing retrieval, classification, and recommendation of media content. However, the variability and subjective nature of these notions makes it difficult to describe them in a systematic way. Nonetheless, there have been efforts to characterize mood as a property of songs and the effects they may have on listeners.

Feng et al. [43] attempt to detect mood of songs from their acoustical features such as tempo and articulation. Liu et al. [83] use intensity, timbre and rhythm instead. Hu & Downie [61] study the relationship between mood and music genre, and mood and artists. In all these cases, the researchers proposed taxonomies of mood types. Feng et al. [43] define four mood labels: *happiness*, *sadness*, *anger*, and *fear* for training a music classifier. Liu et al. [83] use a mood model that characterizes emotions along two dimensions, *energy* and *stress*. They define four mood quadrants: *contentment*, *depression*, *exuberance*, and *anxious/frantic* and use them as labels for mood detection in music using a framework based on Gaussian mixture models. Hu and Downie [61] derive a set of five mood clusters from the *All Music Guide*[2] mood repository to examine the correlation between music genre and mood and artist and mood.

[2]At `http://www.allmusic.com`

The results of this approach are of limited utility because comprehensive, generally accepted, and universally applicable taxonomies for taste and mood do not exist and are difficult to conceive. That would require an in-depth understanding of human emotions, mapping out a wealth of human relations to the external world, and providing a reference scale to measure the intensity of emotions that could be applied in an objective manner.

In our approach, we derive a *latent mood* rather than *a priori* specifying the mood as a property of the music. We use the terms *music taste* and *listening mood* to describe the users' affinity to listen to specific groups of songs as observed from the listening patterns of the whole community. For listening moods we derive the song clusters from the media selection within and across listening sessions, where a session is determined by a threshold of idle time, i.e., a pause between two consecutive songs.

Song recommendations

Ragno et al. [126] address the problem of recommending songs to the user based on a *seed song* that the user has listened to, with the aim to generate a complete playlist that fits the user preferences. It is assumed that the user wishes to listen to songs that are, in some sense, similar to the seed song. In [126] the authors use multiple radio broadcast streams to determine song proximity and define a graph representing the song-similarity. Automatic playlists are generated through random walks of this graph starting on a given seed song. There are many other approaches for automatic playlist generation (e.g., [122, 124]). In

[122], Pampalk et al. use audio similarity and feedback from users, in the form of accepting or skipping a song recommendation, to define a set of heuristics for playlist generation. In [124], Platt et al. learn a Gaussian Process kernel to predict user playlists using music metadata such as genre or style as input.

The recent work on recommender systems by Stern et al. [134] proposes a probabilistic rating model which combines collaborative filtering and item metadata for predicting items that may be of interest to a given user. Marlin et al. [97, 98] also use graphical models for the task of rating prediction. Hoffman et al. [59] propose a probabilistic model which uses audio features to predict song tags.

3.1.2 Preliminary concepts

Graphical models and factor graphs

Factor graphs are a useful way of representing probabilistic graphical models [90]. They consist of two types of nodes representing *variables* and *factors*, respectively. Figure 3.3 and Figure 3.4 show examples of factor graphs with standard notation where variables are represented as round nodes and factors as square nodes. In a probabilistic model, the factors refer to probabilistic distributions, deterministic functions, or constraints. Graphically, the factor nodes connect only to variable nodes that are arguments of the factor. The factors are multiplied together to give an overall distribution function. In this sense, a factor graph is a visual representation of the dependency structure among variables in the overall distribution. In case of generative models, for example, we aim to ex-

plain the observed data and typically arrive at a rich dependency structure where latent and observed variables are generated from parent variables via a factor. In Section 7.3 we describe in detail the generative processes inherent in our listening taste and mood models and demonstrate how both the generative process and the joint probability distribution can be directly read off the corresponding factor graphs.

Factor graphs utilize additional notation that simplifies the visual representation such as *plates* (see for example [18]) which represent replicated parts of the model, and *gates* [107] which represent parts of the model that are switched on or off depending on the value of a random variable. Plates are shown as rectangles with a solid boundary line, and gates are shown as dashed rectangles, with the gating variable attached to the rectangle rather than to the variables inside. The factors inside the gate are switched on or off by the value of the gating variable.

3.1.3 Inference in factor graphs

While useful for visualizing relationships and conditional independence among variables, factor graphs are particularly important as a framework for describing message-passing algorithms for performing inference. In this chapter we make use of a message-passing algorithm for approximate inference called variational message passing (VMP) [144]. This is one of a class of algorithms that are given a unified treatment in [105].

These algorithms typically make use of a fully factorized approximation of

Figure 3.1: The two-level genre taxonomy of Zune Social. Genres have sub-genres. Examples of sub-genres are shown only for the genres *Rock* and *Classical*.

User	Media item	Timestamp
u1	m1	07-11-09 08:12:06
u1	m2	07-11-09 08:15:42
u2	m5	07-11-09 08:16:41
u1	m3	07-11-09 08:20:17
u2	m4	07-11-09 08:20:33
u2	m1	07-11-09 18:56:43
u2	m1	07-12-09 23:54:45
u2	m2	07-12-09 23:58:21
u2	m1	07-13-09 00:01:12
u1	m4	07-15-09 07:54:04
u1	m5	07-15-09 07:56:43

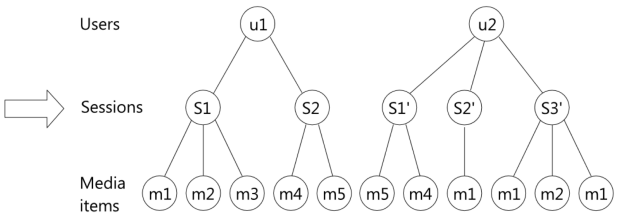


Figure 3.2: Log data for two users and their corresponding music-listening sessions and media items.

the joint probability distribution; i.e. a factorization of each factor itself into univariate factors. For each factor in the graph, the algorithm will calculate outgoing messages from the factor to each variable; each message is in the form of a univariate distribution over the target variable, and is calculated from the factor itself and all the incoming distribution messages via an update equation which minimizes a local divergence measure. The factorized approximation to the factor is given by the product of the outgoing messages.

These message-passing algorithms are fast and also have the benefit that calculations are local, so complex models can be pieced together with reusable building blocks — the Dirichlet and Discrete factors in (Figure 3.3 and Figure 3.4) are two such building blocks, as are the message update equations to deal with plates and gates. Infer.NET [106], which we use to perform inference in our models, is a framework which makes good use of these considerations to provide a variety of message-passing algorithms for graphical models.

3.2 Social media context

In this section we motivate the work through the example of a specific social media service.

3.2.1 Social media description

For the purposes of our study we consider the Zune Social music community and analyze the data set that comprises 14 weeks of usage logs. For each registered user the Zune Social service maintains a user profile with a list of songs that the user has listened to on the Zune device or via Zune software installed on a personal computer.

The Zune Social community members can rate songs, establish friendship links, and recommend songs to each other. Songs are classified using a two-level genre taxonomy. Figure 3.1 shows all 17 top level genre categories and the second level categories for two specific genres, *Rock* and *Classical*. The full taxonomy can

be found on the Zune Social website.

Our objective is to capture users' listening affinities as reflected in the data logs. Thus, we make a concerted effort to clean the usage logs of accidental data access and playing of songs. For each user we consider only those instances where the user listened to a song and rated it positively. This set could be easily expanded using different heuristics. For example, one could include songs that have no ratings but are listened to multiple times by the user. Analysis of our data shows that, on average, the rated songs are listened to 3.62 times. In comparison, the average/mean across all the songs is only 2.26 times.

We assume that the users listen to songs during listening sessions and we employ a simple segmentation technique to specify the session boundaries. We study the distribution of time intervals between the start times of consecutive songs played by the same user. We identify the peaks and use them as thresholds for determining the start of the new session. We observed a few prominent peaks in the distribution. One of the peaks corresponds to the average song length (3.67 minutes).

3.2.2 Terminology and data representation

Let $U = \{u_1, ..., u_n\}$ represent a set of users and $M = \{m_1, ..., m_k\}$ represent a set of media items that the users can listen to. A media item can be a song genre, an artist or a particular song. For ease of representation and without loss of generality, we will refer to a media item as a song. Each song-listening in-

stance (u, m, t) represents user u listening to song m at time t. In order to define listening sessions, we define an *interval* as the time difference between the start times of two consecutive songs for the same user. Alternatively, one can define an interval as the time difference between the end time of one song and the start time of the next song but we chose the former definition because we did not have information of the song end times in our data. A *session* $S = (m_1, ..., m_l)$ is then a sequence of l songs that the user u has listened to, such that the interval between every two consecutive songs m_i and m_{i+1} is below a specified threshold $p_{threshold}$. The playlist $\mathbf{S}_u$ of each user includes a sequence of song-listening sessions $\mathbf{S}_u = (S_1, ..., S_{t_u}) = (m_1, ..., m_N)$. Note that, for the same user, a song can be repeated both in the same session, and across sessions. We also assume that there are latent media clusters $C = \{c_1, ...c_n\}$ which explain the co-occurrence patterns of songs that users play, and they provide a soft clustering of the media items M. Thus, for each cluster c_i, there is a distribution ψ_i over the media items M.

Figure 3.2 shows an example of the data model. The table shows the log of two users $u1$ and $u2$ who have listened to 5 media items at different time points. The log data is visualized as a tree, showing the segmentation into sessions based on the time interval threshold. This threshold can be predefined or learned from data. This example shows some patterns: session $S2$ of user $u1$ is the same as session $S1'$ of user $u2$, and session $S1$ of user $u1$ is similar to session $S3'$ of user $u2$. The goal of our work is to find and characterize such patterns.

3.3 Statistical models

Here we describe in detail the *taste model*, and also our *session model* which extends the taste model and captures the listening mood across song-listening sessions.

3.3.1 Taste model

Following the LDA model [20], we define a probabilistic graphical model that represents consumption of media as a distribution over a set of latent media clusters, referred to as 'tastes.' Each taste media cluster is represented as a distribution over the songs. The model generates each song m in the user's playlist $\mathbf{S}_u$ by picking one of the media clusters c, and then picking a song from that media cluster's mixture ψ. We refer to this model as the *taste model* because each media cluster represents a particular taste. It is a direct adaptation of the LDA model.

A factor graph of the model is shown in Figure 3.3 where the rectangles indicate plates of users, songs of a user, and media clusters. For each user, and each song in the user's playlist, the variable c switches on a particular media cluster, and switches off the others. Algorithm 1 describes the generation of a playlist $\mathbf{S}_u$ for each user u.

$Dir(\alpha)$ is an exchangeable Dirichlet prior, i.e., all pseudo-counts are identical and given by the parameter α. $\theta(u) \sim Dir(\alpha)$ is the parameter vector for a user-dependent Discrete distribution over media clusters. $Dir(\beta)$ is also an exchangeable Dirichlet prior and $\psi(c) \sim Dir(\beta)$ is the parameter vector for a

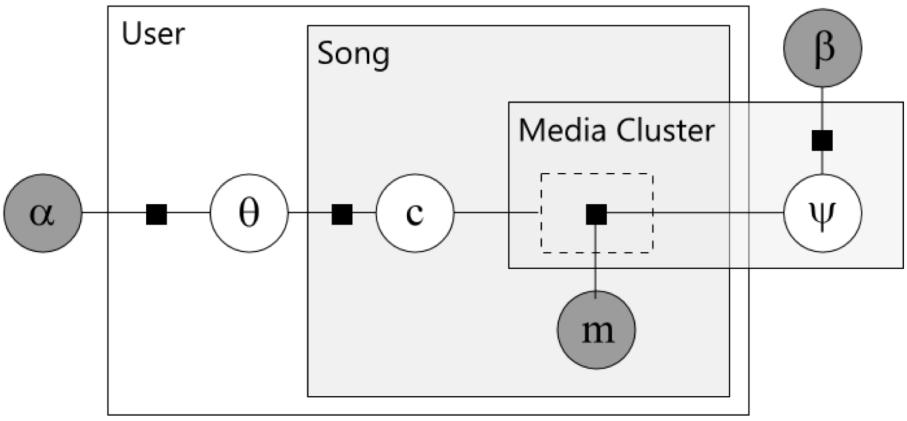


Figure 3.3: Factor graph of the Taste model.

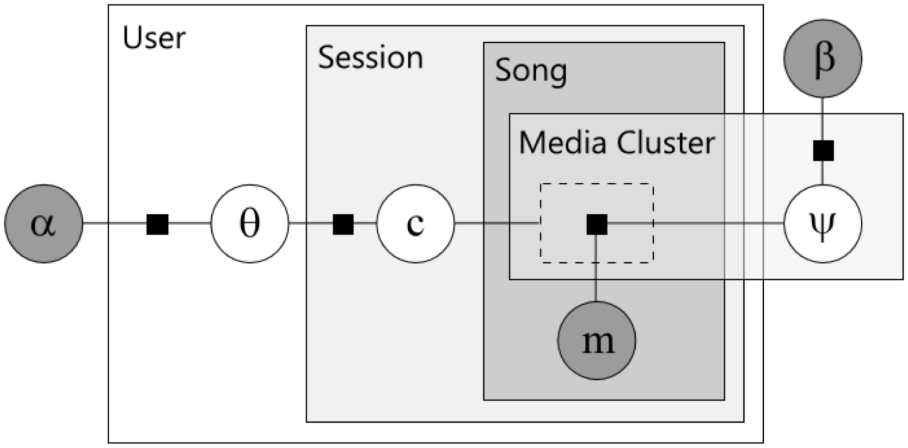


Figure 3.4: Factor graph of the Session model.

cluster-dependent Discrete distribution over songs.

The number of media clusters K is fixed in advance but this constraint can be alleviated as discussed by Blei et al. [20]. According to this model, the joint probability distribution of the distributions ψ over songs, the distributions θ over clusters, the cluster choice c for each user and song, and the songs in user u's playlist $\mathbf{S}_u = (S_1, ..., S_t(u)) = (m_1, ..., m_N)$, is:

$$p(m, c, \psi, \theta|\alpha, \beta) = \prod_{u=1}^{n} p(\theta_u|\alpha) \prod_{j=1}^{N} p(m_{uj}|\psi(c_{uj}))p(c_{uj}|\theta_u)) \prod_{k=1}^{K} p(\psi_k|\beta).$$

We then observe $(m_1, ..., m_N)$ and perform Bayesian inference to recover the posterior marginal distributions of ψ and θ.

Algorithm 1 Taste model

for each media cluster k **do**
  Choose a distribution over songs, $\psi_k \sim Dir(\beta)$
end for
for each user u **do**
  Choose a distribution over media clusters, $\theta_u \sim Dir(\alpha)$
  for each song in the user's playlist $\mathbf{S}_u$ **do**
    Choose a media cluster $c_{uj} \sim Discrete(\theta_u)$
    Observe song $m_{uj} \sim Discrete(\psi(c_{uj}))$
  end for
end for

3.3.2 Session model

We use the session model to detect music-listening *moods* as exhibited in song-listening sessions. Mood is a latent variable in the session model. The model assumes that each user is represented as a distribution over different moods, and for each session, there is a latent mood which guides the choice of songs. A factor graph of the model is shown in Figure 3.4. Here, the media cluster c represents the mood as a mixture of songs.

The session model assumes that $\psi(c)$ for each mood c is picked from $Dir(\beta)$. Algorithm 2 describes the generation of each user's playlist $\mathbf{S}_u$.

Algorithm 2 Session model

for each media cluster k **do**
  Choose a distribution over songs $\psi_k \sim Dir(\beta)$
end for
for each user u **do**
  Choose a distribution over media clusters $\theta_u \sim Dir(\alpha)$
  for each session $S_i \in \mathbf{S}_u$ **do**
    Choose a media cluster $c_{ui} \sim Discrete(\theta_u)$
    for each song in the session **do**
      Observe $m_{uij} \sim Discrete(\psi(c_{ui}))$
    end for
  end for
end for

The joint distribution is:

$$p(m, c, \psi, \theta|\alpha, \beta) = \prod_{u=1}^{n} p(\theta_u|\alpha) \prod_{i=1}^{t_u} p(c_{ui}|\theta_u) \prod_{j=1}^{l_i} p(m_{uij}|\psi(c_{ui})) \prod_{k=1}^{K} p(\psi_k|\beta)$$

When there is one song per session (each song in the playlist has its own session), then the session and taste models are equivalent. As the number of songs per session grows, inference for the session model gets faster than inference on the taste model because it has fewer random variables. In other words, the cluster variable c is picked only once per session and it remains the same for all the songs in the session, whereas in the taste model, c is picked every time a song is generated.

The *session model* embodies the finer level structure in the data. Just as the LDA model, the session model can be applied to a corpus of documents and capture word pattern on the sub-document level. For example, by constraining words within chunks of the document, e.g., paragraphs, to belong to the same

topic, we begin to identify topic patterns associated with paragraphs. Again, an important advantage is the simplified inference and, consequently, the ability to process large document collections efficiently.

3.4 Evaluation

We present results for the problem of playlist generation and discuss the characteristics of the media clustering approach by visualizing the genres per cluster, comparing the discovered latent clusters with the genre taxonomy, investigating the sensitivity of the clustering to the number of pre-specified clusters, and measuring the time performance of the models. We represent each song-listening instance in terms of the corresponding song genre. Since each song can belong to one or more music genres $g \in G$, for each song-listening instance, there are multiple genre instances. We use this media representation to study the connection between the latent media clusters that correspond to listening mood and taste and the song genres. Furthermore, we can explore the usefulness of our models for generating song playlists of individual users. We do that by predicting the genre of the song that the user may want to hear next during the listening session, considering the few seed songs that the user has already listened to. By identifying the desired genre we provide a good foundation for selecting specific songs to present to the user.

3.4.1 Data sample

We train and evaluate the models using a sample of $2,014$ users who have listened to songs that belong to 84 different music genres. From the 14 weeks of data, we use the first two months as training data to learn the parameters of each model and the rest as the test data. Considering the song-listening instances in the training data we arrive at $239,425$ genre instances and $14,703$ sessions using a time interval threshold of 30 minutes and no restriction on the number of songs per session. The test data contains $248,631$ genre instances in $5,079$ sessions which contain at least 11 genres. We control the minimum number of genres per session in order to allow testing the session model with 5 and 10 seed songs. The sample includes all users who have joined the Zune Social service in the studied period, and whose playlists include between 120 and 200 different music artists.

3.4.2 Inference

We implemented the statistical models using Infer.NET, an efficient, general-purpose inference engine for graphical models [106]. Since exact inference is not possible in the taste and session models, we used variational message passing [144] for learning the parameters of each model.

We fixed $\beta = 0.5$ and $\alpha = \frac{1.5}{K}$. β was set to give the best performance for the baseline test model (see Section 3.4.3), and the same value was used for the taste and session models. The value of α was set based on limited manual optimization with respect to the taste model and adopted for the session model as well.

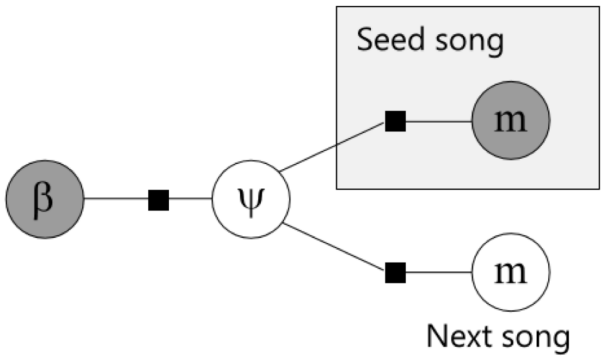


Figure 3.5: Factor graph of the baseline, unigram model.

3.4.3 Results for playlist generation

We evaluated the proposed session model by comparing its performance in terms of model perplexity to that of the taste model on the task of playlist generation for a song-listening session. Besides these two models, we consider a *unigram model* as a simple baseline model that does not consider latent media clusters and learns each session distribution over genres independently. First, we present the unigram model in more detail and then we describe the experimental setup and results.

Baseline test model

In the unigram model the genres in each music-listening session are drawn independently from a single discrete distribution that describes the session. A factor graph of the model is shown in Figure 3.5. Algorithm 3 shows the generative process.

Here, $Dir(\beta)$ is an exchangeable Dirichlet prior and ψ is the parameter vector for a Discrete distribution over songs. During inference, it learns the distribu-

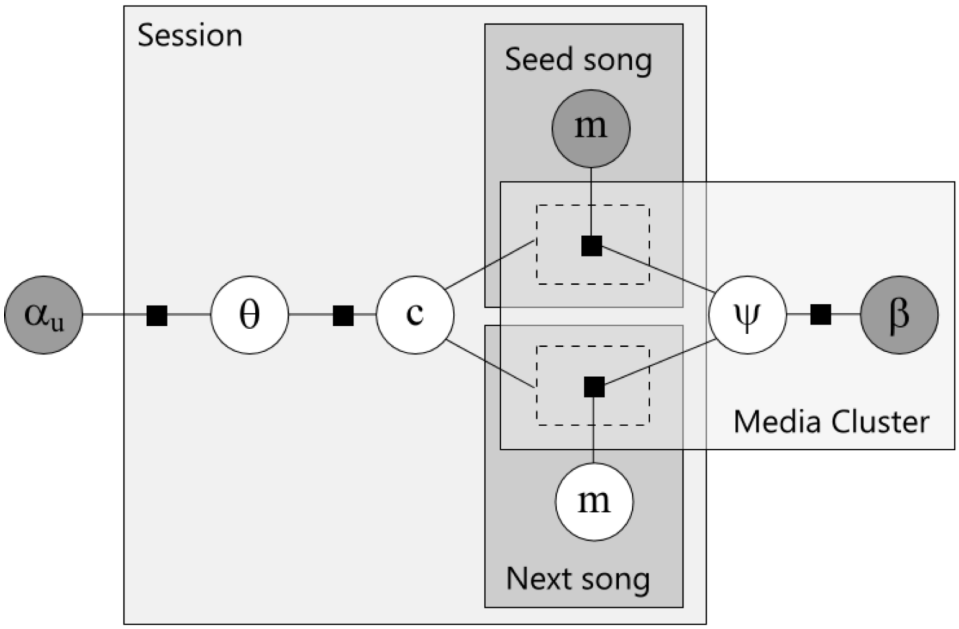


Figure 3.6: Factor graph of the test model for evaluating the session and taste models.

Algorithm 3 Unigram model

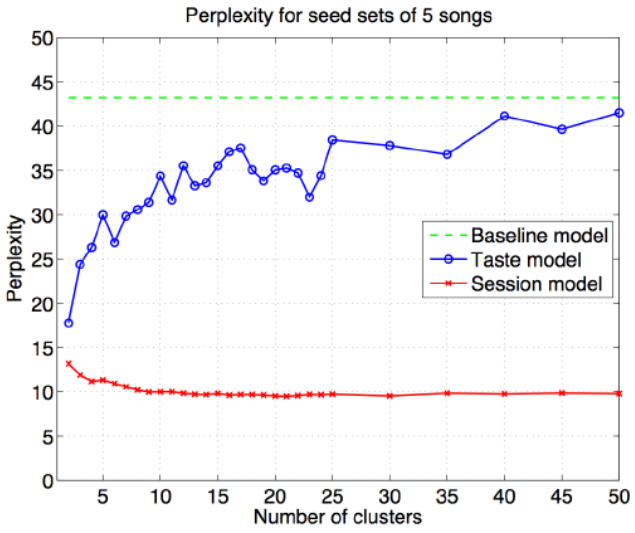
1: **for** each session $S_i \in$ **S do**
2: Choose $\psi_i \sim Dir(\beta)$
3: **for** each song in the session **do**
4: Observe $m_{ij} \sim Discrete(\psi_i)$
5: **end for**
6: **end for**

tion over genres based on the seed songs in the session and uses it to predict the genres of the remainder of the songs in the session.

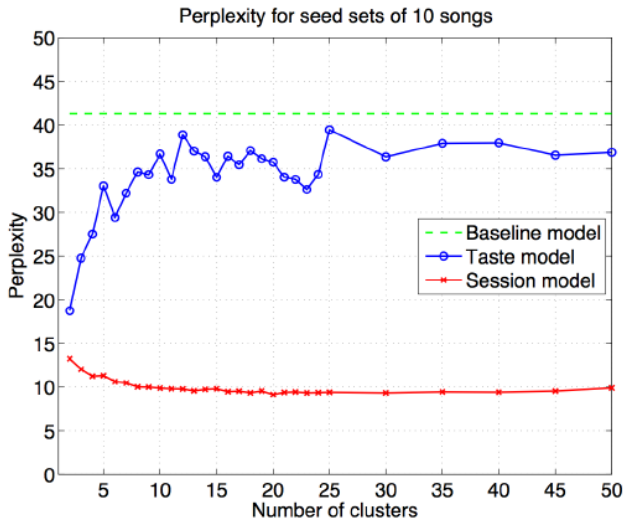
The joint distribution for session S_i is

$$p(\psi_i, m|\beta) = p(\psi_i|\beta) \prod_{j=1}^{l_i} p(m_{ij}|\psi_i)$$

This model assumes that sessions are independent of each other and, unlike the



(a)



(b)

Figure 3.7: Comparison of the perplexity of each model for session genres after observing a) 5 seed genres and b) 10 seed genres.

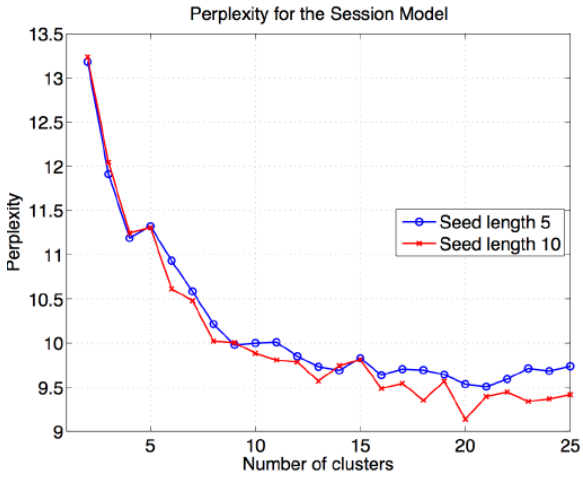


Figure 3.8: Session model perplexity for session genres after observing 5 or 10 seed genres.

taste and session models, it does not consider latent media clusters.

Test model

The taste and session models learn the posterior distributions for their parameters from the training data. These posteriors are used as priors in the testing phase. In the testing phase, the model "observes" the first few seed songs, in our case 5 or 10 songs in a test session, it infers the posteriors of the model parameters, and then finds the likelihood of the genres for the rest of the session songs.

Figure 3.6 shows a factor graph of the test setup for the session and taste models. In the test setup, α_u is the pseudo-count vector of the posterior Dirichlet distribution for θ_u from the training model, where u is the user whose listening session is used as a test. Similarly, for each cluster, β is the pseudo-count vector of the posterior Dirichlet distribution for the ψ of that cluster, derived from the training model. Performing inference on the test model then finds the posterior

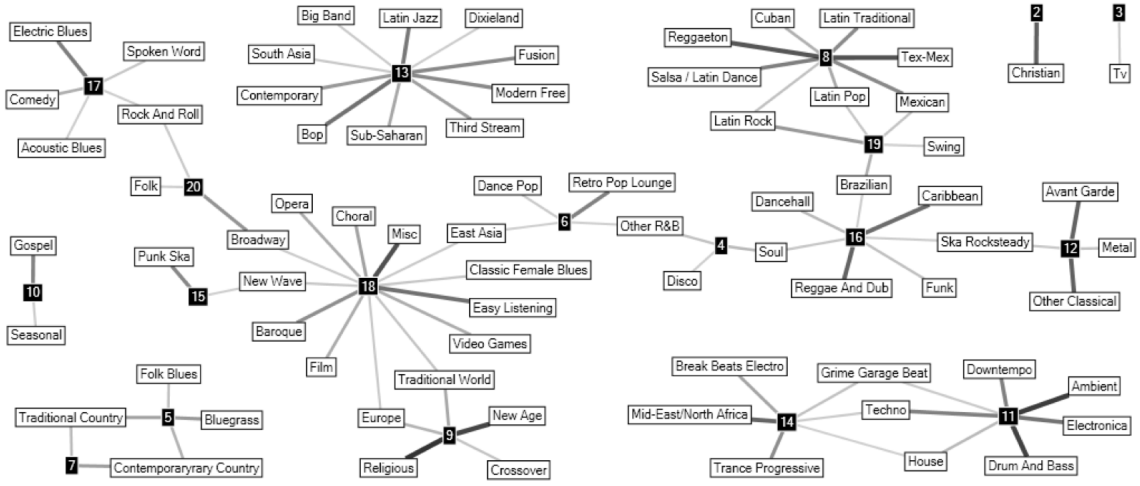


Figure 3.9: Resulting media clusters for the session model. Line thickness signifies cluster affiliation strength.

Dirichlet distributions for θ, the session's distribution over clusters, and ψ, the cluster's distribution over genres, based on a few seed songs (*Seed song* plate in Figure 3.6). Then the log-likelihood is calculated for the genres of the remaining session songs.

Performance metric

In order to assess which model explains the co-occurrence of song genres in listening sessions better, we compare the perplexities of the three models. Perplexity is an entropy-based score assigned to a probabilistic model and commonly used to evaluate topic models such as LDA [20]. It captures how well a model trained on observed data predicts unobserved data. The lower the perplexity of a model, the better its predictive power. We report on the perplexity of each model on the test data:

$$Perplexity = \exp(\sum_{u=1}^{n} \sum_{S \in S_u} \sum_{i=seed+1}^{size(S)} \frac{1}{G} \ln(p(m_i|\psi(c_{ui}))))$$

Computing the perplexity involves finding the log-probabilities of genres in each test session, excluding the seed song genres, and averaging over the number of genre instances G.

Results

Figure 3.7 shows the perplexity scores for the three models: baseline, taste and session models. The session model has consistently lower perplexity than both the baseline and the taste model for the number of clusters between 2 and 50. That means it models better than the other two the patterns of co-occurring genres within the same music-listening session. The lowest perplexity of the session model occurs at 21 clusters for 5 seed songs (9.51), and at 20 clusters for 10 seed songs (9.14), while the lowest perplexity of the taste model occurs at 2 clusters (with perplexity of 18.74 for 5 seed songs, and 17.77 for 10 seed songs). The baseline model perplexity is 43.22 and 41.32 for 5 and 10 seed songs, respectively, and it is constant since it does not assume any latent clusters. These results imply that for the problem of playlist generation, it is better to consider the local patterns across sessions, as captured by the session model, rather than global patterns characterized by the taste model.

Figure 3.8 shows the results for the session model in more detail. It shows

that the predictive power of the model increases as we increase the number of clusters up to $20 - 21$ clusters, depending on the number of seed songs.

3.4.4 Characterizing latent media clusters

We can visualize the affinity of genres to clusters by looking at the distribution of each media cluster over the genre categories. Figure 3.9 shows how genres are associated with listening mood clusters produced by the session model. In the graph we show connecting edges only if the normalized Dirichlet posterior of a genre in the media cluster is more than 0.25. The thickness of the edge reflects the strength of the genre affiliation with the cluster.

We observe that some latent clusters of genre resemble the groupings of genre in the taxonomy shown in Figure 3.1. Indeed, media clusters 8 and 11 have similar genre grouping as the top genre categories $Latin$ and $Electronic/Dance$, respectively. On the other hand, the media cluster 6 comprises a mixture of high-level genres: $Electronic/Dance$, $R\&B$, Pop and $World$.

Comparing latent clusters with taxonomy

In Section 3.4.4 we showed that, in some cases, the collection of genres associated with a listening mood corresponds to one of the top-level genres from the Zune Social taxonomy. For other moods that is not the case. Here, we examine how close a media clustering is to the genre taxonomy, i.e., we estimate how well the static genre taxonomy reflects the listening patterns that emerge from the users' behavior in the social media. The taxonomy itself can be considered as a

collection of clusters where two sub-genres are in the same cluster if and only if they have the same parent genre.

Similarity metric

To compare two media clusterings, we employ a similarity metric based on the *Mallows distance* [96, 161]. This measure is well-suited for comparing clusterings in which the clusters are soft and exchangeable, i.e., it is not known beforehand which pairs of clusters to compare. Zhou et al. [161] discuss the advantages of this measure over other measures for clustering similarity, such as *pair counting*, *set matching* and *variation of information*. The Mallows distance measures the difference between two multivariable probability distributions, and it can be interpreted as an optimal cluster matching scheme between two clusterings C_1 and C_2:

$$Mallows(C_1, C_2) = \min_{w_{k,j}} \sum_{k=1}^{K} \sum_{j=1}^{J} w_{k,j} \sum_{i=1}^{N} |p_{i,k} - q_{i,j}|$$

with the constraints that $w_{k,j} \geq 0$, $\sum_{k=1}^{K} w_{k,j} = \beta_j$, $\sum_{j=1}^{J} w_{k,j} = \alpha_k$ for all k, j. To compute the Mallows distance, one has to solve an optimization problem using linear programming. It yields a global optimum which is unique.

In our case, the computation involves the pseudo-counts for the media cluster posteriors. For each genre, we normalize across clusters to get $p_{i,k}$ where i is a genre index and k is a cluster index. Similarly for $q_{i,j}$. Then, we find the to-

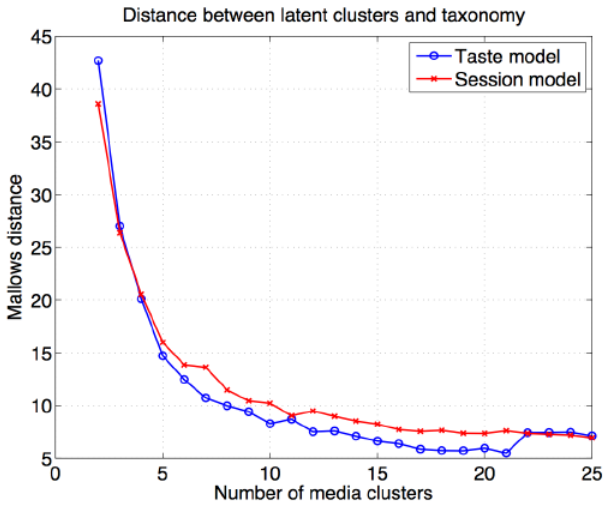


Figure 3.10: Mallows distance between the genre taxonomy and the clusterings found by the taste and session models.

tal count for each Dirichlet and normalize across clusters to get the α_k and β_j. For the optimization part, we apply linear programming using Microsoft Solver Foundation[3].

Cluster comparison results

Figure 3.10 shows that, as the number of clusters increases, the similarity between the genre clusters derived by the session or taste model and the Zune genre taxonomy increases as well. For a range of cluster numbers, the Zune genre taxonomy is slightly more similar to clusters resulting from the taste model than from the session model. However, for both models the resulting genre clusters are different from the original genre taxonomy. Thus, the clusters provide alternative groupings of genre categories that reflect the usage of mobile media and the preferences of the community, as confirmed by the perplexity results in Sec-

[3] `http://code.msdn.microsoft.com/solverfoundation`.

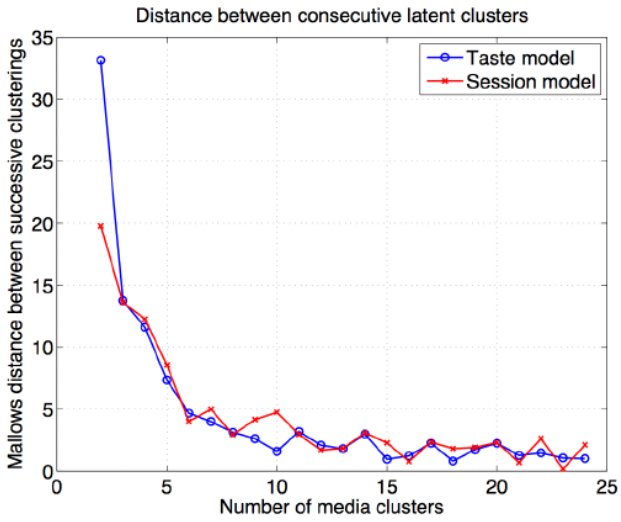


Figure 3.11: Sensitivity of the models to the pre-specified number of clusters.

tion 3.4.3.

3.4.5 Sensitivity to number of clusters

In this section, we conduct a simple experiment to investigate how sensitive the models are to the pre-specified number of clusters. For that, we look at the similarity between clusterings that correspond to successive numbers of clusters. For example, we measure whether a clustering with 15 media clusters is very different from a clustering with 16 clusters. It is of interest to know how the similarity between them changes and whether the clusterings converge. We use the Mallows distance as the similarity score. The larger the Mallows distance between two successive clusterings, the more sensitive the clustering model is to increasing the pre-specified number of clusters.

Figure 3.11 shows that when we increase the number of clusters, the sensitivities of both the taste and session models decrease, i.e. the clusterings become

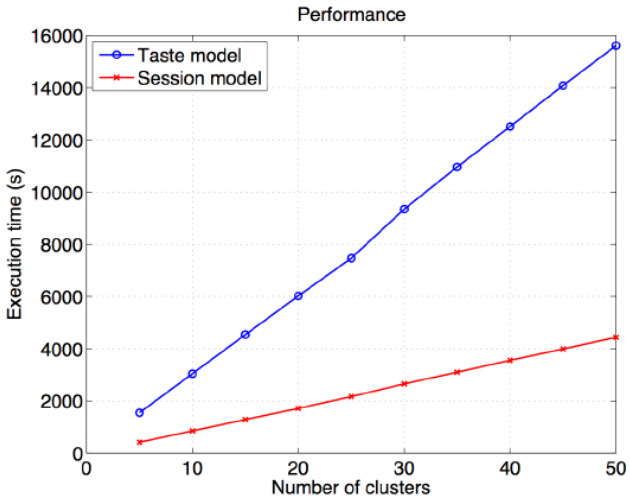


Figure 3.12: Model training time.

more similar to each other. However, for low numbers of clusters, the clusterings are very different from each other. For example, the distance between the clusterings produced for 2 and 3 clusters is 33.2 for the taste model, and 19.5 for the session model.

3.4.6 Time performance of the models

One of the important aspects of statistical models is the computational time required to train the models. Our comparison of the taste and session models confirms that training of the session model is faster. As expected, for both models the training time increases linearly with the number of clusters. However, the rate of increase differs. On our data sample, inference using the session model is 3.7 times faster than for the taste model as Figure 3.12 shows.

3.5 Discussion

Reflecting on the experimental results, we consider possible application scenarios. In music communities such as Zune Social or Last.fm, our approach can be used to enrich user experience. Through media clustering, the service can provide song recommendations based on the collective community tastes and listening moods. As we have shown, the session model can facilitate the playlist completion based on previous listening sessions or several songs that the user has just listened to. Indeed, this can be presented as an improved *shuffle feature* offering a selection of song snippets as short previews during a listening session. The shuffle could adapt based on the user's mood. Furthermore, as an added benefit to identifying media clusters, our models produce groupings of individuals with shared tastes and moods. This information can be leveraged to suggest new friendship ties between users in the social media community.

From the perspective of the service architecture and optimization, clustering media content can contribute to improved load balancing and more efficient content access. Since social media services can involve millions of users on a daily basis, it can be beneficial to distribute service requests across several servers based on appropriate media clusters.

From research point of view, it would be interesting to study user interpretations of the discovered media clusters. It would be valuable to investigate whether latent media clusters, representing for example moods, correspond to different experiences that the users may be able to articulate.

3.6 Conclusion

In this thesis chapter we presented a novel and improved statistical model for finding latent user characteristics in consuming social media content. By taking into account information about the listening sessions of individual users in a music community, we arrive at a new, session-based hierarchical graphical model that has lower perplexity and a shorter training time than alternative approach based on the standard LDA model.

Using the data from the Zune Social music community, we show how generative probabilistic models enable us to capture latent variables that drive the consumption of media. In particular, we adapted the LDA model to capture the taste in music and we define a session based model that captures the user mood in listening sessions. Thus, an instance of song listening can be represented as a finite mixture of the underlying tastes that have been discovered through statistical modeling. Similarly, a song listening within a session can be modeled with respect to the latent moods that the session model generates. Both taste and mood are essentially media clusters that are identified from the statistical analysis of the media usage.

In Zune Social the songs are classified using a fixed two-level taxonomy of music genres. We use genre to characterize the individual songs, and the resulting taste and mood media clusters are represented as genre distributions. In our analysis we conclude that both the taste and mood-based clusterings derived from usage data differ from the static taxonomy. Thus, they offer alternative

genre taxonomies, informed by the community listening patterns. Furthermore, we show that the resulting clusters can be used for playlist generation. The service can thus recommend songs based on a few songs that the user has already listened to.

Our future work will focus on refinements of the session model to capture additional aspects of song listening. One such aspect is listening 'saturation' that requires extending the model to include a 'decay factor.' We also intend to explore application and evaluation of the session model in contexts other than online media consumption.

Part II

Social and Affiliation Network Growth

Besides predicting user characteristics for better personalization of online services, there is a growing interest in understanding what drives the users' engagement in online social media, and consequently, the growth and evolution of online social and affiliation networks [23, 103, 120, 119, 133]. In this part of the thesis, we propose two models for overlaying social and affiliation networks in order to understand social network growth. Chapter 4 looks at the global changes in the network structure. We present a generative co-evolution model for social and affiliation networks which captures the statistical properties of real-world networks [158]. Chapter 5 looks at the related problem of link prediction which uses local properties of the social and affiliation network to predict new social links between users [155].

The evolution of social and affiliation networks exhibits a number of properties previously studied in the literature. We describe some of them in Section 4.2.2, as well as novel properties that we have discovered. Albert and Barabaási give an overview of the statistical mechanics of evolving networks [128], and McGlohon et al. [103] provide a more recent survey on the statistical properties of online social networks. The majority of literature on analyzing network properties has focused on friendship networks, or actor-actor networks in general. Studying the static snapshots of graphs has led to discovering properties such as the 'small-world' phenomenon [143] and the power-law degree distributions [10, 40]. Time-evolving graphs have also attracted attention recently, where interesting properties have been discovered, such as shrinking diameters, and

edge densification [77].

There have been a number of network evolution models proposed to capture statistical properties of social networks. For a survey, one can consult the work by Chakrabarti and Faloutsos [23], as well as by Myra Spiliopoulou [133]. The survey on statistical network models by Goldenberg et al. [50] also covers the history of network evolution models. For example, unlike the random graph model of Erdős and Rényi [121], the preferential attachment model proposed by Barabasi et al. [10] captures power-law degree distributions. The forest fire model [77] also captures the power-law degree distribution together with densification and shrinking diameters over time. A more recently proposed, microscopic evolution model [76] is based on properties observed in large, temporal network data, providing insight into the node and edge arrival processes. Another recent model, the butterfly model [102], concentrates on capturing the evolution of connected components in a graph. In Chapter 4, we extend the microscopic evolution model by including processes of forming and joining groups of interest.

There are studies that describe the relationship between friendship links and group formation properties [8, 108]. They show that the probability of a user joining a group increases with the number of friends already in the group [8], and that higher degree nodes tend to belong to a higher number of groups [108]. Group detection is a related problem (for a survey, see [46]). Its goal is to find new communities based on node features and structural attributes. Unlike group detection work, our work concentrates on unraveling the process according to

which existing groups were formed. Another closely related work to ours is the work by Lattanzi and Sivakumar [75]. They develop a model for the evolution of affiliation networks according to which a social network results from a folding of an affiliation network, i.e., a pairwise social tie between two nodes exists if and only if they share a common affiliation. The resulting networks meet a number of desired properties, such as power-law degree distributions, edge densification and shrinking diameter. In contrast, our work concentrates on an affiliation network which is formed around user-defined online groups in which, as we show in Section 4.1.4, social ties between users affiliated with the same group are rare.

Prior to developing our co-evolution model [158], there was no model that captured the evolution of social and affiliation networks in online communities. In Chapter 4, we present this generative model which captures a number of global statistical properties in these types of networks. Some of the properties have been known in the research literature [8, 10, 77, 76, 108], and we also discover and discuss other novel properties.

Besides capturing the global network properties, understanding the dynamic nature of networks involves studying local node properties for predicting which pairs of users are likely to form social links. In Chapter 5, we look at this problem which is known as link prediction. Lü and Zhou [91], as well as Al Hasan and Zaki [53], have written recent surveys on the topic of link prediction. Lü and Zhou distinguish between similarity-based algorithms, maximum likelihood methods and probabilistic model approaches. Similarly, Al Hasan and Zaki categorize approaches into feature-based link prediction, Bayesian probabilistic

models, probabilistic relational models and linear algebraic methods.

In general, link-prediction algorithms process a set of features in order to learn and predict whether it is likely that two nodes in the data are linked. Sometimes, these features are hand-constructed by analyzing the problem domain, the attributes of the actors, and the relational structure around those actors [2, 52, 80, 127]. Other times, they are automatically generated, i.e., the prediction algorithm first learns the best features to use and then predicts new links [125]. Next, we discuss the existing work that is most relevant to the link-prediction problem in multi-relational social networks.

Closest to our work are link prediction techniques which rely on intelligent feature construction [2, 52, 62, 80, 127]. The constructed features not only include the attributes of the actors, but also the characteristics of the structure. Most of this work examines co-authorship and citation networks [52, 80, 125, 127]. Some of the approaches use machine learning techniques for classification [52, 74, 125, 138], and others rely on ranking the feature values [2, 80, 127].

Link prediction methods can use a similarity function. For example, Adamic and Adar [2] use this type of method to predict friendships amongst students. They gather data from university student websites and mailing lists, and construct a vector of descriptive features for each student such as website text, in-links, out-links, and mailing lists the students belong to. Their approach uses descriptive features.

It has been shown that there is "useful information contained in the network topology alone" [80]. Liben-Nowell and Kleinberg use a variety of structural fea-

tures such as shortest path, (a variant of) number of friends, number of common friends, jaccard coefficient, and more elaborate structural features based on all paths between two nodes in co-authorship networks. Their experiments compare the link-prediction accuracy of each feature in isolation. They rank the node-pairs by each feature value and pick the top pairs as their predicted links. Their results suggest that simple features such as *number of common friends* coefficient perform well compared to others.

Rattigan and Jensen [127] recognize that the extremely large class skew associated with the link-prediction task makes it very challenging. They look at a related problem, anomalous link discovery, in which instead of discovering new links, they are interested in learning properties of the existing links. They use structural features in co-authorship networks and rank the most and least likely collaborations based on an expensive structural feature, the Katz score. Another work that uses link prediction for anomaly discovery is the work of Huang and Zeng [62], in which they rank anomalous emails in the Enron dataset.

The work described so far uses descriptive and structural attributes in isolation. Hasan et al. [52] use both. Their work studies classification for link prediction based on hand-constructed features in co-authorship networks. They report prediction accuracy (F score), precision, and recall results from a range of classifiers such as decision trees, k-nearest neighbor, multilayer perceptron, and support-vector machines. we study link prediction in richer social network settings and we explore the use of group features and alternate representations.

The link-prediction problem has also been studied in the domain of citation

networks for scientific publications [125]. The authors posed the link-prediction problem as a binary classification problem, and used logistic regression to solve it. Their features are database queries such as *most cited author*, and thus they are similar to both the descriptive and structural features we have discussed so far. Their work describes a statistical learning approach for feature generation. In particular, it extends the traditional *Inductive Logic Programming (ILP)* to reason about probabilities, and uses this extension to learn new features from the problem domain both statistically and inductively. The experiments in this work suggest that the ratio of existing to non-existing links in the test data mattered, and the fewer non-existing link examples were included, the better the precision-recall curve was. However, testing with more non-existing link examples would give a better estimate of the probability of a randomly picked pair of nodes in the network to be classified correctly. Another statistical learning approach to link prediction was presented by Taskar et al. [138]. The authors use relational Markov networks to define a probabilistic model over the entire link graph. Their features are both descriptive and relational. They apply their method to two domains: linked university websites and a student online social network.

Another automated feature-generation method has been presented by Kubica et al. [74] who described a learning method for the task of *friend identification* which is similar to anomalous link discovery. Their method, called cGraph, learns an approximate graph model of the actual underlying link data given noisy linked data. Then, this graph model is used to predict pairwise friendship information regarding the nodes in the network. The types of features that they use

are descriptive, structural and group.

Link completion is a problem related to link prediction. Given the arity of a relationship and all but one entity participating in it, the goal is to predict the missing entity, as opposed to classifying the missing link itself. Goldenberg et al. [49] present a comparison of several classification algorithms such as Naive Bayes, Bayesian networks, cGraph (mentioned above), logistic regression, and nearest neighbor. This study used several real-world datasets from different domains, including co-authorship networks, and data collected from the Internet Movie Database site. It suggested that logistic regression performs well in general in the datasets above; in our study on real-world social networks, logistic regression usually performed worse than decision-tree classifier in terms of accuracy.

There has also been interest in learning group features in social networks. Kubica et al. [73] describe a group-detection algorithm that uses descriptive features and links. First, they perform clustering based on the descriptive features (clustering) and find the groups. They allow group overlap and assume that group memberships are conditionally independent of each other given descriptive features. Then, their algorithm assigns a probability of a link between two actors based on the similarity of their groups, and it can answer ranking queries similar to the ones in the anomalous link discovery work. One of the issues with the proposed algorithm is that it is slow [74].

The work of Friedland and Jensen [45] studies the problem of identifying groups of actors in a social network that exhibit a common behavior over time.

The authors focused on networks of employees in large organizations, and investigated the employee histories to identify the employees who worked together intentionally from those who simply shared frequently occurring employment patterns in the industry.

In Chapter 5, we study link prediction in social and affiliation network settings where there are closely-knit groups. We explore the use of descriptive, structural and group features, as well as alternate network representations, and present a taxonomy for link prediction. Our results suggest that there is a significant increase in link prediction accuracy when the affiliation network is overlaid with the social network.

Chapter 4

Co-evolution Model

Many of the existing online social networks have millions of users, and allow complex interactions through linking to friends, public messaging, photo commenting, participating in groups of interest, and many others. Studies have been performed to characterize and explain the behavior of users, and most of them concentrate on modeling how users join the network and form links to each other. Little is known about how different types of interaction influence each other. In this chapter, I address the problem of modeling social network generation explaining both link and group formation.

In social networks, users are linked to each other by a binary relationship such as friendship, co-working relation, business contact, etc. Each social network often co-exists with a two-mode affiliation network, in which users are linked to groups of interest, and groups are linked to their members. In our study we use three large datasets from online social and affiliation networks, and discover a number of interesting properties. The datasets were from Flickr, Live-

Journal and YouTube, collected by Mislove et al. [108].

Using the newly observed and previously studied statistical properties of these networks, we propose a generative model for social and affiliation networks. The model explains the complex process of forming the networks, and captures a number of affiliation network properties which have not been captured by a model before: power-law group size distribution, large number of singletons (group members without friends in the group), power-law relation between the node degree and the average number of group affiliations, and exponential distribution of the number of group affiliations for nodes of a particular degree. Our findings are important for understanding the evolution of real-world networks and suggest that the process is more complex than a naïve model in which groups are added to a fully evolved social network. They also show that users join groups for different reasons and having friends in the group is often not necessary. This suggests that information spreads in the network through channels other than the friendship links, and this observation has implications on information diffusion and group recommendation models.

In addition, this model can be used for synthetic network generation. This is an important application because real-world network datasets are often proprietary and hard to obtain. Controlling network parameters allows the generation of datasets with different properties which can be used for thorough exploration and evaluation of network analysis algorithms.

Our contributions include the following:

- We discover a number of new properties in social and affiliation networks.
- We propose the first generative model for network evolution which captures the properties of both real-world social and affiliation networks.
- We provide a thorough evaluation of our model which shows its flexibility for synthetic data generation.

Because we study the evolution of graphs over time, we introduce the notation $e_v(v_i, v_j, t)$ to denote the social link that v_i and v_j form at time t and $e_h(v_i, h_j, t)$ to denote the affiliation link between user v_i and group h_j formed at time t, when this user becomes a member of the group. There are a number of reasons why groups are formed. For example, groups can exist because of a common interest, such as philately or book-reading clubs; they can be based on common business relation, such as an employing company, or they can be based on common personal traits, such as geographic location. What is common between the groups that we study is that users have voluntarily chosen to be parts of them, as opposed to clustered together by a group detection algorithm.

4.1 Observations

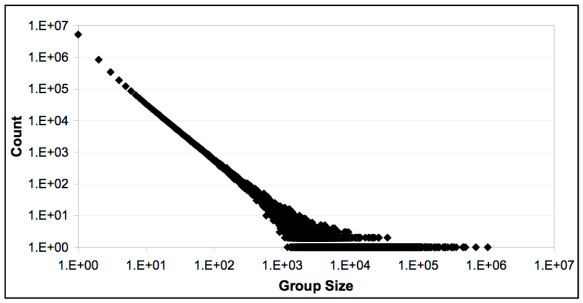
Though affiliation groups constitute a major part of many social networks, very little work in the literature focuses on analyzing group memberships and evolution. In this section, we analyze different affiliation networks and try to characterize some properties of affiliation groups that are consistent across various datasets. For our analysis, we used three large real-world datasets from

LiveJournal, Flickr and YouTube.

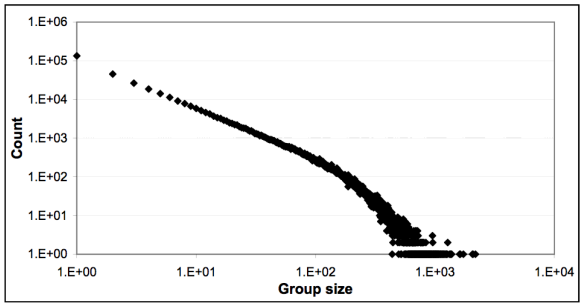
LiveJournal is a popular blogging website whose users form a social network through friendship links. Users also form affiliation links to various 'communitites,' which are groups of users with similar interests. We used a LiveJournal dataset with over 5.2 million users, 72 million links, and over 7.4 million affiliation groups. The second dataset is from Flickr, a photo-sharing website based on a social network with friendships and family links. Groups in Flickr are also formed on the basis of common interest. The Flickr dataset contains over 1.8 million users, 22 million links, and around $100,000$ groups. The third dataset is from YouTube, a popular video-sharing website with an underlying social network based on users' contacts. Users also form an affiliation network by joining social groups where they can post and discuss videos. The YouTube dataset contains over 1.1 million users, 4.9 million links and around $30,000$ groups. The full dataset descriptions can be found in the work of Mislove et al. [108]. Now, we describe the observations that we discovered by analyzing the datasets, and we relate them to previously observed properties.

4.1.1 Group size distribution

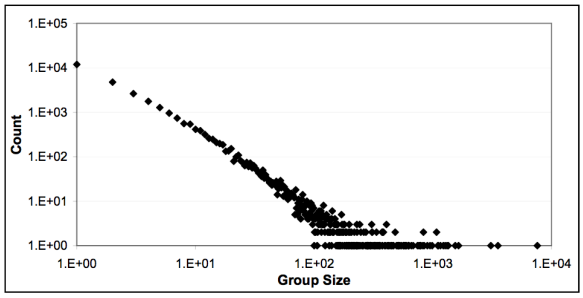
We begin by characterizing the relationship between the size of the affiliation group and its frequency of occurrence. The main observation is that, analogous to the degree distribution, the group size distribution follows a power law, with a large number of small groups and a smaller number of large ones. This has



(a) LiveJounral



(b) Flickr



(c) Youtube

Figure 4.1: Distribution of the number of groups of a particular size on log-log scale.

also been observed by Mislove et al. [108]. The results are illustrated in Figure 4.1.

4.1.2 Node degree vs. average number of group affiliations

Looking at the relationship between the degree of a node and the number of its group affiliations, we observe that the nodes of lower degree tend to be members of fewer number of groups than the nodes with higher degree. However, the relation starts declining after a certain point, yielding lower number of

group memberships for very high degree nodes. The relationship is illustrated in Figure 4.2, where the x-axis represents the node degree and the y-axis represents the average number of group affiliations for nodes with that degree. The nodes in the declining part represent a very small portion of the overall number of nodes ($< 1\%$ of the size of the network in all cases), which is why we fitted only the increasing part of the data points to a function. We compared against over 55 different distributions including logistic, Dagum and Laplace, using EasyFit [1], a software for distribution fitting. A power-law relation was the best fit according to the Kolmogorov-Smirnov ranking coefficient.

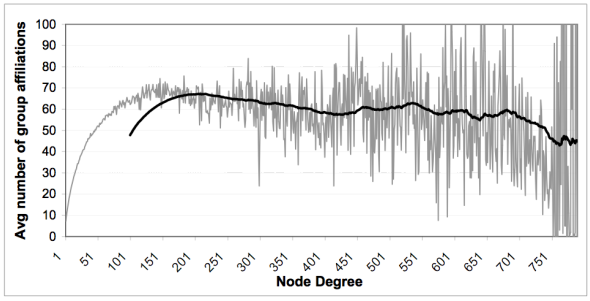
4.1.3 Distribution of the number of group affiliations

The previous observation was about the average number of group affiliations for nodes with different degrees. Here, we look at the actual distribution of the number of group affiliations with respect to the node degree. It turns out that the number of group affiliations for nodes of a certain degree k follows an exponential distribution. Figure 4.3 reports on $k = 50$ for LiveJournal and Flickr, and on $k = 25$ for YouTube but this was true for other degrees as well.

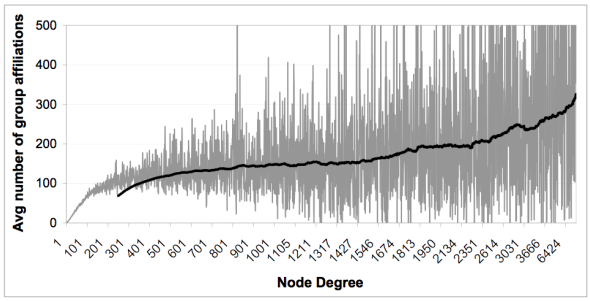
4.1.4 Properties of group members

According to Backstrom et al. [8], nodes are more likely to join groups in which they have more friends. However, it turns out that, in our datasets, there is a large portion of group members without friends in the group ($singletons$),

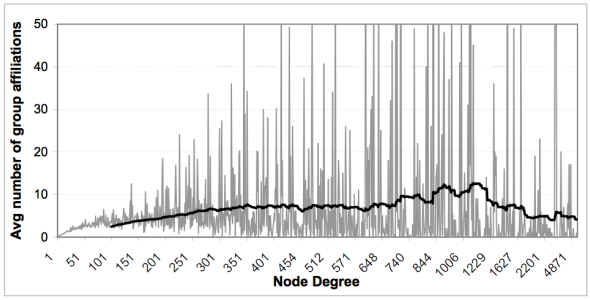
[1] At `http://www.mathwave.com`



(a) LiveJounral

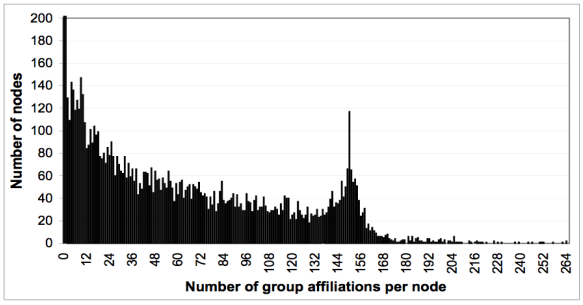


(b) Flickr

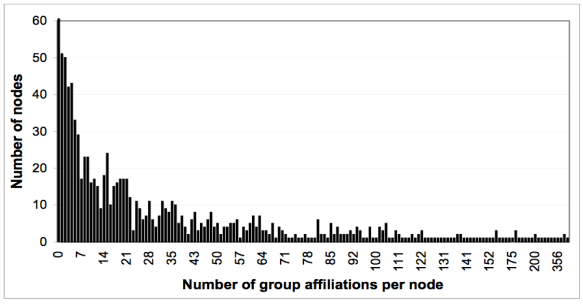


(c) Youtube

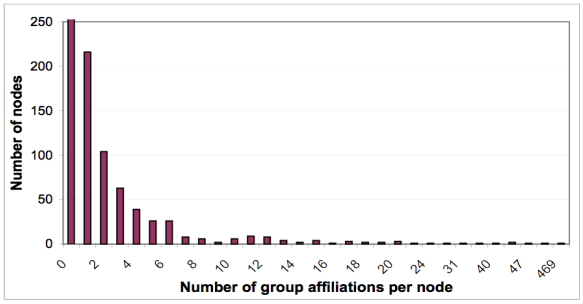
Figure 4.2: Node degree vs. average number of group affiliations



(a) LiveJounral - Degree = 50



(b) Flickr - Degree = 50



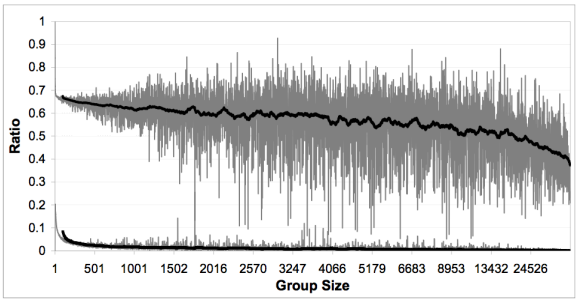
(c) Youtube - Degree = 25

Figure 4.3: Distribution of the number of group affiliations for nodes with specific node degrees.

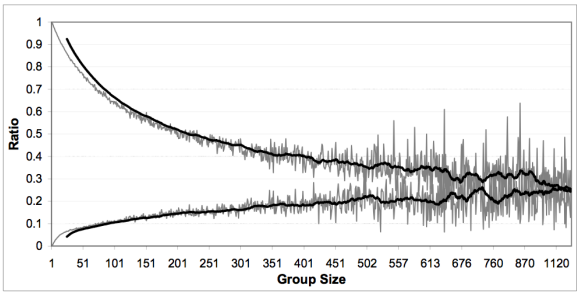
meaning that they did not join the group because of a friend. This is surprising because it shows that users join groups for various reasons, friendship being only one of them.

We measure the maximum node degree within groups of various sizes in our datasets. For all groups of a given size, we measure the average maximum degree per group and the average number of singletons (nodes with no friends within this group) as a percent of the group size. The results show a large number of singletons overall, especially in small groups, indicating that a large percentage of the members of a specific group do not have any friends within this group. This conclusion was confirmed by analyzing the average maximum degree per group. It turned out that the friends of the maximum-degree node within a group do not constitute a large percentage of the group size, even in small groups.

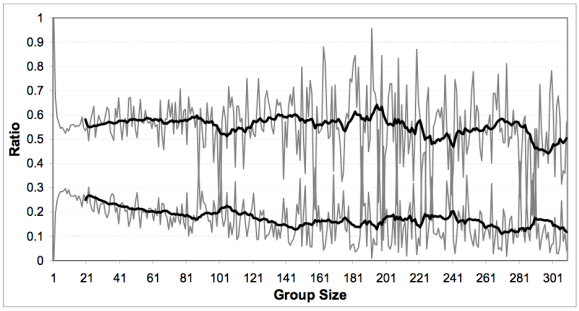
The numbers are illustrated in Figure 4.4, where the $upper$ series shows the average ratio of the number of singletons to the group size, and the $lower$ series represents the average ratio of the maximum degree to the group size. This result shows that the larger the group a user belongs to, the more likely it is for him/her to have a friend in the group. For example, in Flickr, 76% of the members of groups of size 50 are singletons, while for groups of size 500, this number drops to 29%.



(a) LiveJounral



(b) Flickr



(c) Youtube

Figure 4.4: Ratio of the average number of singletons to the group size (upper series) and ratio of the maximum degree to the group size (lower series).

4.2 Co-evolution properties and model

A model which describes the evolution of a social network together with the evolution of an affiliation network needs to capture a number of simple events, as well as statistical properties of both networks. Here, we present the events of our co-evolution model and desired properties, some of which have been presented in other work. Then, we present our co-evolution model, which extends the node arrival and link formation processes of the microscopic evolution model [76] to dynamic social and affiliation networks.

4.2.1 Events

The possible events that our model allows are:

- a node joins the network and links to someone
- a new group is formed with one member
- a node joins an existing group
- a new link is formed between existing users

4.2.2 Desired properties

A co-evolution model needs to capture properties of both social and affiliation networks. Here, we show three types of properties: properties of the social network alone, properties of the affiliation network alone, and properties of both.

Properties of the social network. The properties are:

- *power law degree distribution* - the node degrees are distributed according to a power law with a heavy tail. This property has been observed in many other studies.
- *network densification* - the density of the network increases with time [77].
- *shrinking diameter* - the effective diameter of the network decreases as more nodes join the network [77].

Properties of the affiliation network. The model also needs to capture the following affiliation network property:

- *power law group size distribution* - the group sizes are distributed according to a power law with a heavy tail.

Properties involving both the social and affiliation networks. These properties describe the relationship between a social network and an affiliation network:

- *large number of singletons* - many nodes do not have any friends inside the groups they are affiliated with.
- *power-law relation between the node degree and the average number of group affiliations* - see Section 4.1.2.
- *exponential distribution of the number of group affiliations for a particular node degree* - see Section 4.1.3.

4.2.3 Co-evolution model

We now propose a co-evolution model which captures the discussed desired properties. Our model is undirected, and it has two different sets of parameters: one is concerned with the evolution of the social network, and the other determines the factors of development of the affiliation network. We also present a naïve model which assumes that the evolution of the affiliation network is independent of the evolution of the social network. Both models utilize the microscopic evolution model [76] for generating the social network because that model is based on observing the temporal properties of large social networks. We present its main components first.

Microscopic evolution model. The main ideas behind the microscopic evolution model are that nodes join the social network following a node arrival function, and each node has a lifetime a, during which it wakes up multiple times and forms links to other nodes. These are the set of parameters needed for the microscopic evolution model: $N(.)$ is the node arrival function, λ is the parameter of the exponential distribution of the lifetime, and α, β are the parameters of the power law with exponential cut-off distribution for the node sleep time gap. Further details of the model can be found in the paper by Leskovec et al. [76]. We utilize these parts:

Node arrival. New nodes $V_{t,new}$ arrive at time t according to a pre-defined arrival process $N(.)$.

Lifetime sampling. At arrival time t, v samples lifetime a from $\lambda.e^{-\lambda.a}$: v be-

comes inactive after time $t_{end}(v) = t + a$.

First social linking. v picks a friend w with probability proportional to degree(w) and forms edge $e_v(v, w, t)$.

Sleep time sampling. v decides on a discrete sleep time δ by sampling from $\frac{1}{Z}.(\delta^{-\alpha}).e^{-\beta.\text{degree(v)}.\delta}$. If the node is scheduled to wake up before the end of its lifetime ($t + \delta \leq t_{end}(v)$), then it is added to the set of nodes $V_{t+\delta}$ that will wake up at time $t + \delta$.

Social linking. At wake up time t, v creates an edge $e_v(v, w, t)$ by closing a triad two random steps away (i.e., befriends a friend w of a friend).

Naïve model. Before we present our model, we present a naïve model which assumes that the evolutions of the social network and the affiliation network are two independent processes. As a first step, it creates the social network using the model of Leskovec et al. [76]. Then, it generates and populates groups in such a way that their sizes follow a power-law distribution with an exponent k. Algorithm 4 presents the naïve model in detail. We use this model as a baseline.

Co-evolution model. In this model, the affiliation network evolution co-occurs and depends on the social network evolution. When a node wakes up, besides linking to another node, it also decides on a number of groups to join. With probability τ, it creates a new group, else, it joins an existing group. There are two mechanisms by which it picks a group to join. In the first one, it joins the group of one of its friends. In the second one, it picks a group at random. Algorithm 5 presents the co-evolution model in detail.

Here, we present the parameters of the affiliation network evolution part in

Algorithm 4 Naïve model

Set of nodes $V = \emptyset$
for each time period $t \in T$ **do**
    Set of active nodes at time t, $V_t = \emptyset$
end for
for each time period $t \in T$ **do**
    Node arrival. $V = V \cup V_{t,new}$
    for each new node $v \in V_{t,new}$ **do**
        Lifetime sampling
        First social linking
    end for
    for each node $v \in V_t$ **do**
        Social linking
    end for
    for each node $v \in V_t \cup V_{t,new}$ **do**
        Sleep time sampling
    end for
end for
Set of groups $H = \emptyset$.
for i=1:number of groups **do**
    Group creation. New group h_i is created and its size s is sampled from s^{-k}. $H = H \cup \{h_i\}$.
    for j=1:s **do**
        Group joining. Pick a random node $v \in V$ and form an affiliation link to it $e_h(v, h_i, null)$.
    end for
end for

Algorithm 5 Co-evolution model

Set of nodes $V = \emptyset$
Set of groups $H = \emptyset$
for each time period $t \in T$ **do**
  Set of active nodes at time t, $V_t = \emptyset$
end for
for each time period $t \in T$ **do**
  Node arrival. $V = V \cup V_{t,new}$
  for each new node $v \in V_{t,new}$ **do**
    Lifetime sampling
    First social linking
  end for
  for each node $v \in V_t$ **do**
    Social linking
    Affiliate linking. v determines n_h, the number of groups to join, sampled from an exponential distribution $\lambda' e^{-\lambda' n_h}$ with a mean $\mu' = \frac{1}{\lambda'} = \rho.\text{degree(v)}^{\gamma}$.
    for $i = 1 : n_h$ **do**
      if $rand() < \tau$ **then**
        Group creation. v creates group h, and forms edge $e_h(v, h, t)$. $H = H \cup \{h_i\}$.
      else
        Group joining. v forms edge $e_h(v, h, t)$. Group h is picked through a friend with probability p_v; otherwise, or if no friends' groups are available, it joins a random group with prob. proportional to the size of h.
      end if
    end for
  end for
  for each node $v \in V_t \cup V_{t,new}$ **do**
    Sleep time sampling
  end for
end for

more detail. The first parameter, ρ, represents a tuning parameter that controls the density of the affiliation links in the network. The second parameter, γ, is the exponent of the power law that relates node degree with number of group affiliations. The last parameter to our model, τ, represents the probability by which an actor creates a new group at each time point. All our parameter values range over the interval $[0, 1]$ except ρ which ranges between 0 and the average number of group affiliations per node. We provide some guidelines for picking the right parameter values in the experiments section.

As noted in Section 4.2.2, the relationship between node degree and average number of affiliations is a power-law relation. Even though one can vary the exponent γ of this function, for simplicity, we fixed its value to 0.5, utilizing a square root function to compute this average.

It is also worth noting that other, more sophisticated techniques can be utilized in both social and affiliation aspects of the model that might be able to capture stronger correlation between the evoultion of both kinds of networks. One possible modification for the social link creation is considering random steps but with group bias, such that the probability of choosing a node u to close the triad is proportional to the number of groups the two nodes share. Another possible modification is to specify the number of groups a node will join in advance using the estimated power-law function. A disadvantage of such approach is that the approximated degree is hard to compute because it depends on the expected value of a function which changes with the degree. A thorough investigation of the different alternatives is left as future work.

In the group joining step of the algorithm, a node decides to join a group and it has two choices for picking that group. One is through a friend, and the second one is by picking a random group with probability proportional to the size of that group. It follows the first choice with some probability p_v, else it resorts to the second one. The intuition behind this is that some nodes in each group are singletons while others have friends in it. The second choice is also based on the observation that the size of the groups follows a power-law distribution; on the principle of "rich get richer," groups with larger size should have a larger probability of getting picked.

There are many options for computing the probability p_v such as making it a constant or dependent on the node degree. One can test which one is most appropriate in the presence of temporal data for affiliation networks. Since such data is hard to obtain, we try different possibilities in our model. It turns out that using a constant for p_v yields a relationship between the group size and the singleton ratio that decreases at first but then stabilizes around $1 - p_v$ at higher group sizes. In contrast, what we had observed initially was a relationship which decreases with increasing group sizes (see Figure 4.4). When we use a p_v which is correlated with the degree, then we observe a relationship closer to the desired one. In particular, we compute:

$$p_v = \begin{cases} \eta * \text{degree(v)} & \text{if } \eta * \text{degree(v)} < 1 \\ 1 & \end{cases} \tag{4.1}$$

though other functions of the degree may be more appropriate. The parameter η represents the friends' influence on the actor's decision to join a group; i.e. the likelihood of an actor joining one of the groups of his/her friends increases by increasing the value of η. The main intuition behind using a degree-correlated probability is the fact that as a node has more friends, the probability that one of its friends belongs to one of the larger size groups increases. Thus, utilizing the friendship bias parameter η actually increases its chances of joining this larger size group of its friend, thus leading to the decreasing relationship noted in the observations.

4.3 Experiments

We present three sets of experiments. The first set observes the properties of data, generated by our co-evolution model, and the second set shows that the model is able to produce a dataset, very similar to one of the real-world datasets. We also present results for the naïve model which adds groups on top of a social network, showing that this model is not able to produce the real-world affiliation network properties.

4.3.1 Synthetic data

In our first set of experiments, we vary the parameters of the model in order to generate a few synthetic datasets. Then, we check whether each dataset has the properties described in Section 4.2.2.

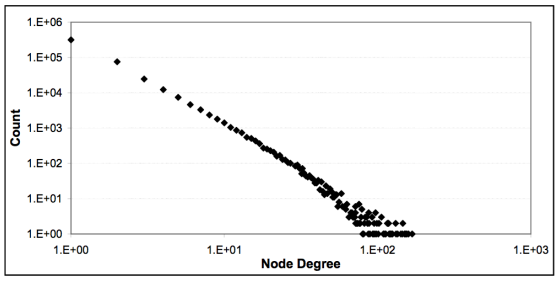


Figure 4.5: Degree distribution in a synthetic network

We have fixed the parameters of the social evolution part throughout this set of experiments, and varied the parameters of the affiliation part of the network. We assume an exponential node arrival function, to achieve higher growth rate in our generated network, which is in accordance with what Leskovec et al. [76] showed in some social networks, such as Flickr. However, other arrival functions can also be utilized within our model. The other parameters of the social evolution aspect were fixed as reported by Leskovec et al. for Flickr data: $\lambda = 0.0092$, $\alpha = 0.84$, and $\beta = 0.002$. We also fix the value of the second parameter to the affiliation model, γ, to 0.5.

We first illustrate the results for the social network generated using the specified parameters. The model was run for 400 time steps, resulting in a network with 140,158 actors and 245,043 social links. The degree distribution in the resulting network follows a power-law, as Figure 4.5 shows. The network densification property also holds, as illustrated in Figure 4.6 which represents the number of nodes and number of edges at each time point on a log-log scale.

In order to test the affiliation aspect of our evolution model, we investigated the effect of each parameter in the model on the properties of the resulting affil-

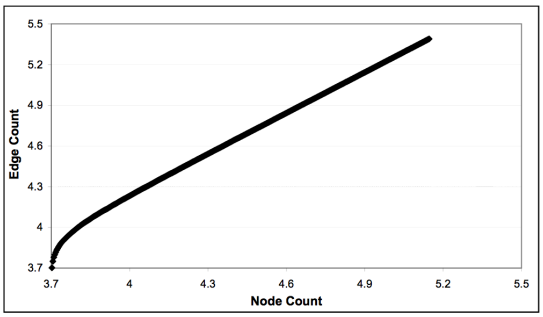


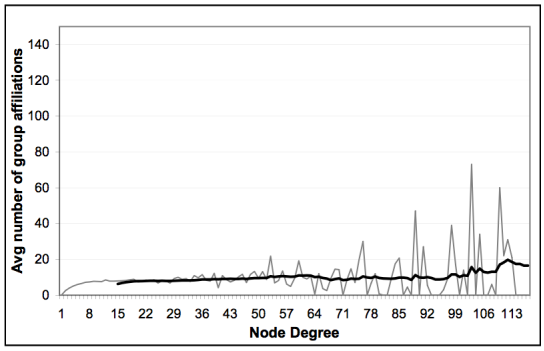
Figure 4.6: Densification in a synthetic network

Table 4.1: Number of affiliation links with varying ρ

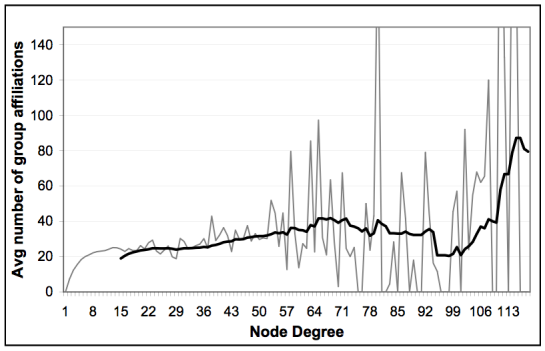
ρ	**Affiliation Count**
3	285,536
10	2,411,710
20	4,771,072

iation network. We start with our first parameter ρ, which represents a tuning factor of the affiliation links' density. The main properties that are affected by varying the value of ρ are the total number of affiliations and the distribution between the node degree and average number of group affiliations. As illustrated in Figure 4.7, we can note that the general power distribution persists among different values of ρ, but the main effect is the scale of the distribution; as increasing the value of ρ, more affiliation links are created, and correspondingly increasing the average number of group affiliations per node. Theoretically, the values for this parameter can vary from 0, where no affiliation links are created in the network, to the maximum number of groups, where fully connected affiliation network emerges. Practical values for ρ varies between 0 and 25. The total number of affiliation links for each value of ρ is reported in Table 4.1.

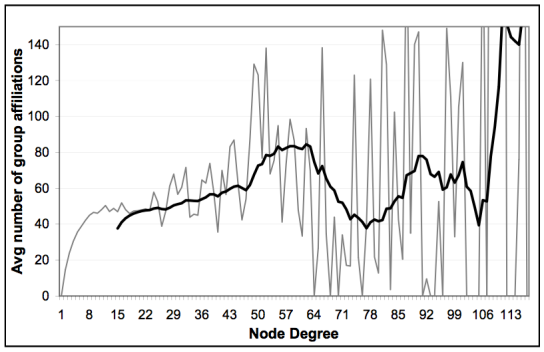
Our next parameter, τ, represents the probability with which a node creates



(a) $\rho = 3$

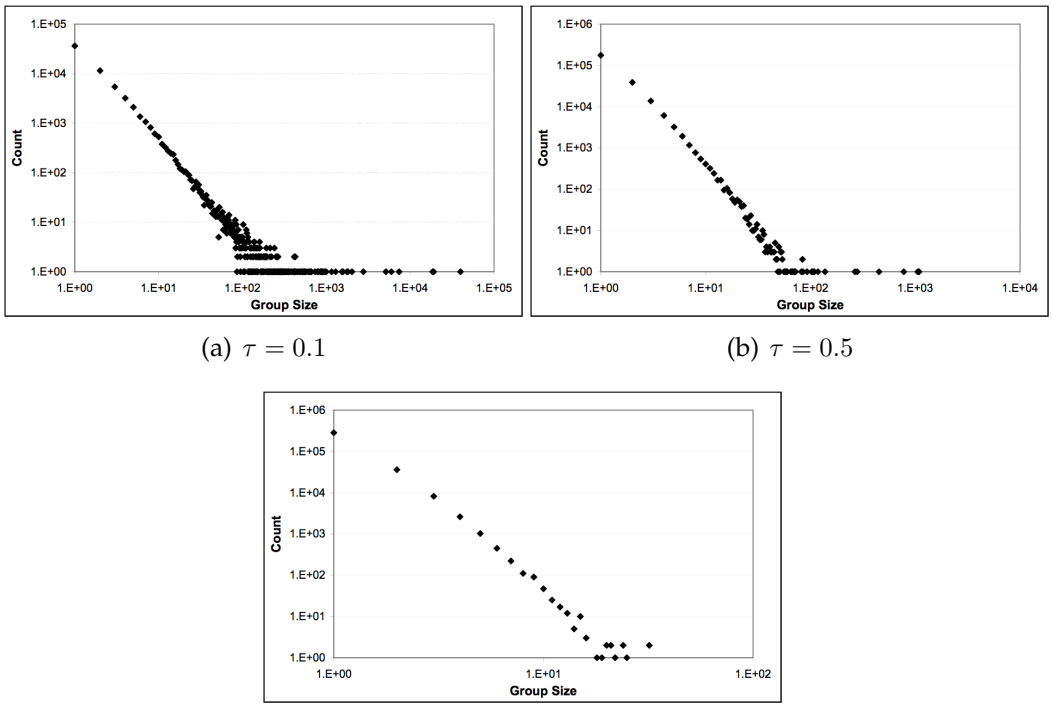


(b) $\rho = 10$



(c) $\rho = 20$

Figure 4.7: Degree vs. average number of group affiliations with varying ρ.



(a) $\tau = 0.1$ (b) $\tau = 0.5$

(c) $\tau = 0.9$

Figure 4.8: Group size distribution with varying τ

a new group. This parameter directly affects the number of groups in the resulting network, as well as the group size distribution. As illustrated in Figure 4.8, we note that although the power law distribution of the group sizes holds for various values of τ (which is one of the desired properties), the maximum group size decreases significantly with increasing the value of τ. This decline in the maximum group size is caused by the fact that for higher values of τ, nodes tend to create new groups more often than joining existing ones, which leads to the existence of a large number of groups with relatively small sizes. This conclusion is also clear in the results illustrated in Table 4.2, where the resulting number of groups in the network and the maximum group size vary significantly with changing the parameter value.

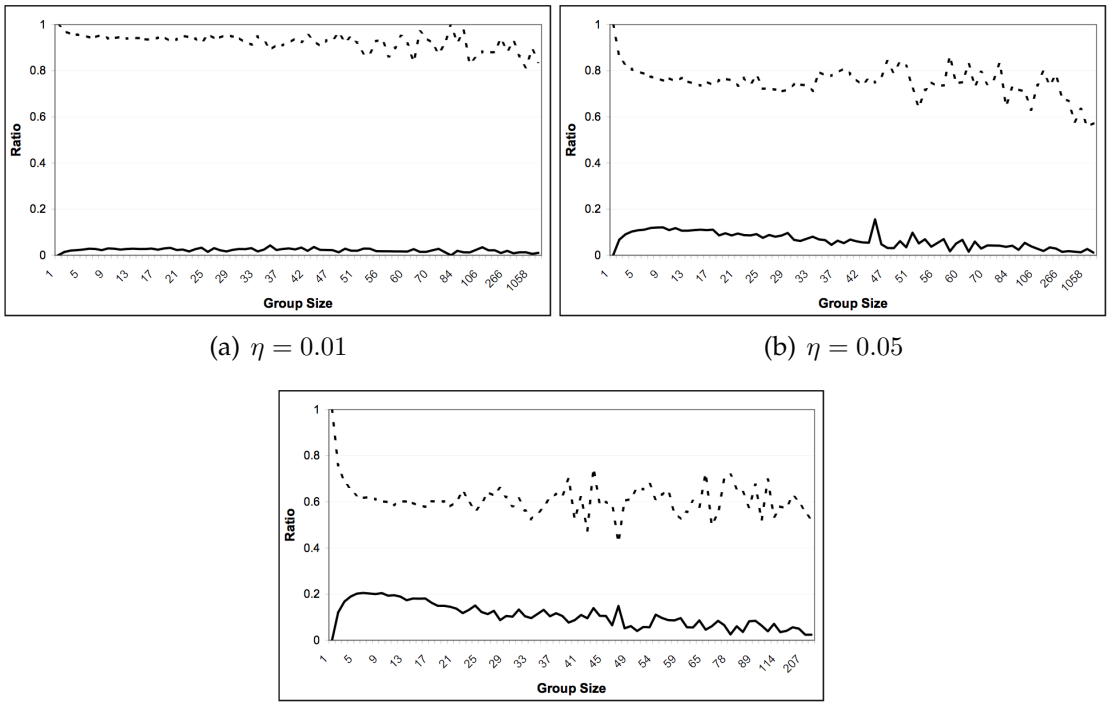
Table 4.2: Number of groups with varying τ

τ	**Groups Count**	**Max Group Size**
0.1	66,887	39,753
0.5	245,143	560
0.9	332,437	32

Table 4.3: Statistics of a real network (Flickr) vs. a synthetic one with $\rho = 2.5$, $\gamma = 0.5$, $\eta = 0.1$, $\tau = 0.03$.

	Real network	**Synthetic network**
No. of users	1,846,198	1,707,475
No. of groups	103,648	88,749
No. of affiliations	8,529,435	7,813,910
Avg. no. of affiliations per user	4.62	4.58
Ratio of groups to users	0.0561	0.052

Finally, we investigate the parameter on which p_v depends, η. η represents the extent to which friends influence the decision of a node to join groups. The outcome of increasing the value of this parameter is a decreasing number of *singletons* and an increasing relative degree of the nodes within different groups. As illustrated in Figure 4.9, we can easily note that the general distribution captures the desired properties and the observations in real data. The value of η is highly dependent on the social network structure properties, such as the average node degree in the social network and the desired influence of friends on node's decision. For instance, if we have a value of $\eta = 0.1$ in a setting where the expected value for the average node degree is around 10, then we expect to see high percentage of nodes in the network being affected by their friends.



(a) $\eta = 0.01$ (b) $\eta = 0.05$

(c) $\eta = 0.09$

Figure 4.9: Group size vs. member attributes with varying η (dashed line: % ratio of singletons to group size, solid line: % ratio of maximum degree to group size).

4.3.2 Real data

In this set of experiments, we look for the model parameters that will produce a network similar to one of the real-world datasets we have used in the observations of Section 4.1. We searched for parameters that will produce an affiliation network resembling the actual one of Flickr since the social network evolution parameters for Flickr have already been reported by Leskovec et al. [76]. In order to get an initial seed of the search space for the evolution parameters of the affiliation network, we analyze the affiliation network properties of Flickr as observed in Section 4.1. A summary of the affiliation network statistics of Flickr is given in Table 4.3.

The Flickr dataset is characterized by a relatively small number of groups in comparison to the number of users, where the actual ratio between the group count and the user count is 0.056. As a result, we expect to have a small value of τ close to this ratio. On the other hand, the average number of group affiliations per user in the real dataset is 4.62, and we assign this value to ρ. Finally, as observed in Figure 4.4, the average percentage of singletons in each group is lower than the average for the other datasets, indicating more friendship bias, thus increasing the value of η.

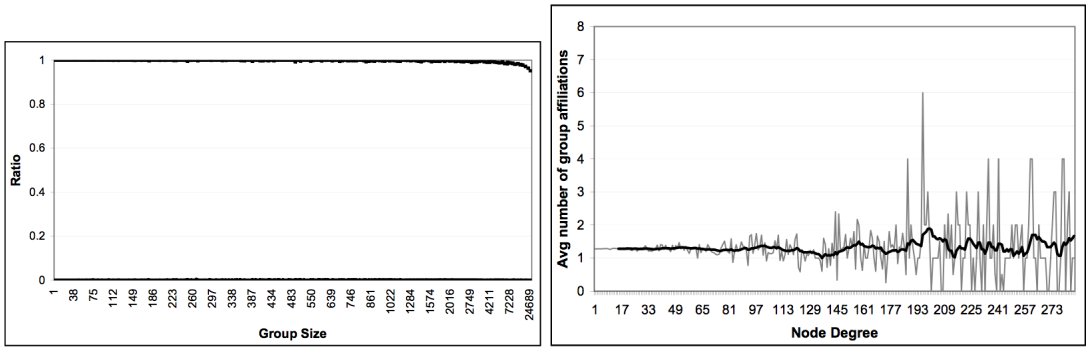
There are other factors to consider when specifying the affiliation network evolution parameters, such as the rate of node arrival and the probabilistic nature of the node's lifetime and sleep time gaps. For example, in Flickr's case, the exponential node arrival rate means that more nodes are created at later times. In this case, the distribution parameters should be a bit lower than the desired ones because many nodes will join towards the end of the evolution process but they will not have time to create many links and affiliations. By utilizing all these pieces of information to guide the parameter search, we were able to generate a network that has similar attributes to Flickr's, illustrated in table Figure 4.3. We argue that using a similar procedure for parameter selection can result in generating synthetic networks that have many of the properties of a real one.

4.3.3 Comparison with the naïve model

In this set of experiments, we were interested to learn whether we can produce the desired network properties by utilizing the naïve evolution model. The model can clearly capture the social network properties since the process of creating it is the same as in our co-evolution model. In terms of the affiliation network properties, we used the naïve model to produce a social network similar to Flickr, as described in the previous experiment. Then we created the desired number of groups and picked the size of each one from a power-law distribution with the parameters observed in Flickr. Each group was populated by picking random users from the social network. As a result, the naïve model is able to capture the group size distribution. However, Figure 4.10(a) shows that it is not able to capture the average number of singletons and the average maximum degree as a percent of the group size. By picking random members, almost all members in each group end up being singletons (except for groups with very large sizes), and the average maximum degree is close to 0. Figure 4.10(b) shows that the model is also not able to capture the relation between degree and average number of group affiliations for nodes with lower degrees. The naïve model generates a relation between them which is closer to linear than a power law.

4.4 Conclusions

We presented a generative model for creating social and affiliation networks. The model captures important statistical properties of these networks, and pro-



(a) Average number of singletons (dashed line) and average maximum degree (solid line)

(b) Degree vs. avg number of affiliation groups

Figure 4.10: The affiliation properties produced by the naïve model

vides new insights into the evolution of networks with both social and affiliation links. It shows that groups can be formed for various reasons and friendship links are not the only propagators of influence. We believe that this observation not only affects the design of network evolution models but it may have broader implications on other mechanism designs, such as group recommendation, information diffusion and viral marketing strategies.

Chapter 5

Link Prediction

Besides studying network evolution at a macro level, we are interested in understanding network dynamics at a micro level. In this chapter, we investigate the power of combining friendship and affiliation networks for the task of predicting new social links using local node properties. We use the notion of structural equivalence, when two actors are similar based on participating in equivalent relationships, which is fundamental to finding groups in social networks. Our approach is an attempt to bridge approaches based on structural equivalence and community detection, where densely connected groups of actors are clustered together into communities. We show how predictive models, based on descriptive, structural, and group features, perform surprisingly well on challenging link-prediction tasks.

We validate our results on a trio of social media websites describing friendships and family relationships. We show that our models are able to predict links accurately, in this case friendship relationships, in held-out test data. This is typ-

ically a very challenging prediction problem. With our results, we also hope to motivate further research in discovering closely-knit groups in social networks and using them to improve link-prediction performance.

Our link-prediction approach can be applied in a variety of domains. The important properties of the data that we use are that there are actors, links between them and closely-knit groups such as families, housemates or officemates. In some data, groups are given; in other datasets, it may be necessary to first cluster the nodes in a meaningful manner. For example, in email communication networks, such as Enron [62, 67], groups could be cliques of people that email each other frequently. In the widely studied co-authorship networks [11, 52, 80, 117, 118, 125, 127], affiliation groups may be cliques of authors that collaborate on many papers together. In these domains, the link-prediction task translates to finding people who are likely to communicate with each other [62] or authors who are likely to collaborate in the future [80, 125].

Our contributions include the following:

- We propose a general framework for combining social and affiliation networks.
- We show how to instantiate it for overlaying friendship and family networks.
- We show how features of the overlaid networks can be used to accurately predict friendship relationships.
- We validate our results on three social media websites.

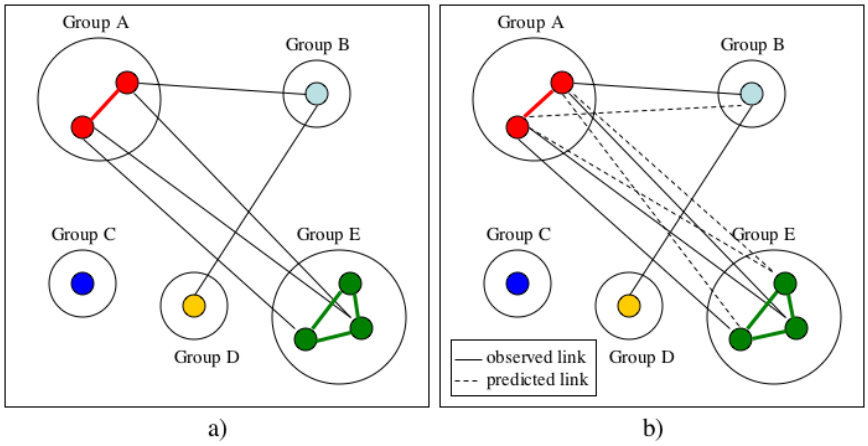


Figure 5.1: Actors in the same tightly-knit group often exhibit structural equivalence, i.e., they have the same connections to all other nodes. Using the original network (a), and a structural equivalence assumption, one can construct a network with new predicted links (b).

In Section 5.1, we describe the link prediction problem that we focus on in this chapter. Section 5.2 addresses the taxonomy of the descriptive, structural, and group features that we used for link prediction in our overlaid networks. We then propose a comparison of our network overlay method with two alternatives in Section 5.3. We describe experimental results in Section 5.4, the generality of our approach in Section 7.5, and discuss conclusions and future work in Section 9.

5.1 Link prediction problem

In this thesis chapter we study the problem of predicting friendship links in multi-relational social networks. This problem is closely related to problems of link prediction [52, 62, 80, 125], link completion [49], and anomalous link discovery [62, 127] which we covered in more depth in the beginning of Part II.

Link prediction in social networks is useful for a variety of tasks. The most straightforward use is for making data entry easier – a link-prediction system can propose links, and users can select the friendship links that they would like to include, rather than users having to enter the friendship links manually. Link prediction is also a core component of any system for dynamic network modeling—the dynamic model can predict which actors are likely to gain popularity, and which are likely to become central according to various social network metrics.

Link prediction is challenging for a number of reasons. When it is posed as a pair-wise classification problem, one of the fundamental challenges is dealing with the large outcome space; if there are n actors, there are n^2 possible relations. In addition, because most social networks are sparsely connected, the prior probability of any link is extremely small, thus we have to contend with a large class skew problem. Furthermore, because the number of links is potentially so large, the number of the negative instances will be huge, so constructing a representative training set is challenging.

In our approach to link prediction in multi-relational social networks, we explore the use of both attribute and structural features, and, in particular, we study how affiliations with closely-knit groups (in our case, family groups) can significantly aid in accurate link (here, friendship) prediction.

5.2 A feature taxonomy for multimodal networks

We identified three classes of features that describe characteristics of potential links in a multimodal network:

- *Descriptive attributes* are attributes inherent to the nodes, and they do not consider the structure of the network.
- *Structural attributes* include characteristics of the networks based on the friendship relationships such as node degree.
- *Group attributes* are based on structural properties of the network when both types of relationships, friendship and family, are considered. The groups in this case are the cliques of family members.

Each feature within a class can be assigned to an actor or to a pair of actors (corresponding to a potential edge). The following sections describe our taxonomy of the features in more detail. For simplicity, we introduce the notation $v_i.F$ to denote the set of friends of actor v_i, and $v_i.M$ to denote the set of family members of the same actor.

5.2.1 Descriptive attributes

The descriptive attributes are attributes of nodes in the social network that do not consider the link structure of the network. These features vary across domains. They provide semantic insight into the inherent properties of each node

in a social network, or compare the values of the same inherent attributes for a pair of nodes.

We define two classes of descriptive attributes for multi-relational social networks:

1. *Actor features.* These are inherent characteristics of an actor, $v.A$.

2. *Actor-pair features.* The actor-pair features compare the values of the same node attribute for a pair of nodes v_i and v_j.

5.2.2 Structural features

The next set of features that we introduce describe features of network structure. The first is a structural features for a single node, v_i, while the remaining describe structural attributes of pairs of nodes, v_i and v_j.

1. *Actor features.* These features describe the link structure around a node.

   Number of friends. The degree, or number of friends, of an actor v_i: $|v_i.F|$.

2. *Actor-pair features.* These features describe how interconnected two nodes are. They measure the sets of friends that two actors have $v_i.F$ and $v_j.F$.

   Number of common friends. The number of friends that the pair of nodes have in common in the network: $|v_i.F \cap v_j.F|$.

   Jaccard coefficient of the friend sets. The Jaccard coefficient over the friend sets of two actors describes the ratio of the number of their common

friends to their total number of friends:

$$Jaccard(v_i, v_j) = \frac{|v_i.F \cap v_j.F|}{|v_i.F \cup v_j.F|}.$$

The *Jaccard coefficient* is a standard metric for measuring the similarity of two sets. Unlike the feature *number of common friends*, it considers the size of the friendship circle of each actor.

Density of common friends. For the set of common friends, the density is the number of friendship links between the common friends over the number of all possible friendship links in the set. The density of common friends of two nodes describes the strength in the community of common friends. Density is also known as *clustering coefficient*.

5.2.3 Group features

The third category of features that we consider are based on group membership; in the networks we studied, the groups are families. These are the features that overlay friendship and affiliation networks.

1. *Actor features.* These are features that describe the groups to which an actor belongs.

   Family Size. This is the simplest attribute and describes the size of an actor's family: $|v_i.M|$.

2. *Actor-pair features.* There are two types of features for modeling these inter-family relations based on the overlapping friend and family sets of two actors $v_i.F$ and $v_j.M$:

   Number of friends in the family. The first feature describes the number of friends v_i has in the family of v_j: $|v_i.F \cap v_j.M|$. This feature allows one to reason about the relationship between an actor and a group of other actors, where the latter is semantically defined over the network through the family relations.

   Portion of friends in the family. The second feature on inter-family relations describes the ratio between the number of friends that v_i has in v_j's family (the same as the above feature) and the size of v_j's family. The rationale behind this feature is that the higher this ratio is, the more likely it is that v_j is close to v_i in the network since more of its family members are friends with v_i.

The idea behind the group features is based on the notion of *structural equivalence* of nodes within a group. Two nodes are structurally equivalent if they have the same links to all other actors. If we can detect tightly-knit groups in a social network and we assume that the nodes in each group are likely to behave similarly, then new links can be predicted by projecting links such that the nodes in the group become structurally equivalent. In our networks, such groups are the family cliques. In a weighted graph, a tight group could map to a clique of nodes with highly-weighted edges.

Figure 5.1 shows an example of how a structural equivalence assumption can help in predicting new links. For example, if one of the actors from Group A is friends with an actor from Group B, as shown on the original network (a), then it may be more likely that there is a link between the other actor from Group A and the actor from Group B, shown as a dashed line in (b).

5.3 Alternative network representations

The traditional approach to studying networks is to treat all relationships as equal. In the previous section, we described overlaying networks with different link types in a way that distinguishes between these types, and uses information about affiliation groups. In other words, our link-prediction approach uses information about the actors V, the family groups H, the friendship relationships E_v, and the family relationships E_m where $E_m = \{(v_i, v_j) | \exists e_h(v_i, h_x), \exists e_h(v_j, h_x)\}$. We call our representation *different-link and affiliation overlay*. Therefore, a logical question one may ask is what the benefits of treating links as different are, and whether affiliation groups really make a difference in link prediction. Our claim is that affiliations are important and that they can have a predictive value. To illustrate the benefit of our approach as compared to the traditional one, we compare it to two alternative representations of the network.

In the first alternative representation, which we call *same-link and no affiliation overlay*, the family and friendship links are treated the same, and affiliation groups *are not* given. More formally, in this representation, the graph consists

of these components: actors V, and a set of edges to which we refer as *implied friendships* $E_{implied} = E_v \cup E_m$. We can compute the descriptive and structural features in this alternative overlay, and use them for link prediction. In our experiments, we investigate whether this alternative overlay can offer the same or better link-prediction accuracy as the different-link and affiliation overlay.

Even if the first alternative overlay does not offer better accuracy, we still need to check whether the predictive value of the different-link and affiliation overlay comes from treating the links as different or from the fact that we are given the affiliation groups. To investigate that, we look at a second alternative overlay, the *same-link and affiliation overlay*, in which the family and friendship links are treated the same, and affiliation groups are given. In this overlay, the graph consists of these components: actors V, groups H, and implied friendships $F_{implied}$. We can compute all classes of features in this alternative overlay, and use them for link prediction.

5.4 Experimental evaluation

5.4.1 Social media data sets

This research is based upon using networks that have two sets of connections: friendship links and family ties. We performed our experiments on three novel datasets describing *petworks*: Dogster, Catster, and Hamsterster [1]. On these

[1] At `http://www.dogster.com`, `http://www.catster.com`, and `http://www.hamsterster.com`.

Figure 5.2: Sample profile on Dogster which includes family and friends.

sites, profiles include photos, personal information, characteristics, as well as membership in community groups. Members also maintain links to friends and family members. As of February 2007, Dogster has approximately 375,000 members. Catster is based on the same platform as Dogster and contains about 150,000 members. Hamsterster has a different platform, but it contains similar information about its members. It is much smaller than Dogster and Catster - about 2,000 members.

These sites are the only three of the hundreds we visited that publicly share both family and friendship connections[2]. However, these are networks where both types of connections are realistic and representative of what we expect to see in other social networks if they collected this data. The family connections

[2]For a full list, see `http://trust.mindswap.org/SocialNetworks`

are representative of real life, since family links are only made between profiles of pets created by the same owner. The friendship linking behavior is in line with patterns seen in other social networks [48].

1. *Actor features:*

   Breed. This is the pet breed such as *golden retriever* or *chihuahua*. A pet can have more than one breed value.

   Breed category. Each breed belongs to a broader category set. For example in Dogster, the major breed categories we identified are *working*, *herding*, *terrier*, *toy*, *sporting*, *non-sporting*, *hound*, and *other*, a catchall for the other breeds that appear in a the site, but not as frequently as the previous ones. When a dog has multiple breeds, its breed category is *mixed*.

   Single Breed. This boolean feature describes whether a pet has a single breed or whether it has multiple breed characteristics.

   Purebred. This is a boolean feature which specifies whether a dog owner considers its pet to be purebred or not.

2. *Actor-pair features.* All of the above features describe characteristics of a single user in the network.

   Same breed. This boolean feature is true if two profiles have at least one common breed.

5.4.2 Data description

We have obtained a random sample of $10,000$ profiles each from Dogster and Catster, and all $2,059$ profiles registered with Hamsterster. Each instance in the test data contained the features for a pair of profiles where some of the features were individual node features. To construct the test data, we chose the pairs of nodes for which there was an existing friendship link, and we sampled from the space of node pairs which did not have a link. We computed the descriptive, structural and group features for each of the profiles.

For each pair of profiles in the test data, we computed the features from the three classes described in Section 5.2. A test instance for a pair of profiles v_i and v_j includes both the individual actor features and the actor-pair features. It has the form

$< v_i$ *features*, v_j *features*, (v_i, v_j)*-pair features*, *class* $>$

where $class$ is the binary class which denotes whether a friendship link exists between the actors.

For Dogster, the sample of 10,000 dogs had around 17,000 links among themselves, and we sample from the non-existing links at a 10:1 ratio (i.e., the non-existing links are 10 times more than the existing links). For Catster, the 10,000 cats had 43,000 links, and for the whole Hamsterster dataset, the number of links was around 22,000. We sampled from the non-existing links in these datasets at the same 10:1 ratio.

Table 5.1: Comparison of F1 values in the three datasets, with the feature types from our taxonomy.

FEATURE TYPE	DOGSTER	CATSTER	HAMSTERSTER
Descriptive	37.6%	0.4%	19.8%
Structural	76.1%	83.1%	59.9%
Group	90.8%	95.2%	89.2%
Descriptive and structural	78.6%	83.0%	60.3%
Descriptive, structural, and group	94.8%	97.9%	90.5%

5.4.3 Experimental setup

We used three well-known classifiers, namely Naïve Bayes, logistic regression and decision trees for our experiments. The goal was to perform binary classification on the test instances and predict friendship links. The implementations of these classifiers were from the latest version of Weka (v3.4.12) from `http://www.cs.waikato.ac.nz/ml/weka/`. We allocated a maximum of 2GB of memory for each classifier we ran. We measured prediction accuracy by computing precision, recall, and their harmonic mean, F1 score, using 10-fold cross-validation.

5.4.4 Link-prediction results

We report only on the results from decision-tree classification because it consistently had the highest accuracy among the three classifiers. Table 5.1 summarizes our results. Adding group features to the descriptive and structural features increased accuracy by 15% to 30%. We discuss the results in more detail in the subsequent subsections.

Descriptive attributes can be useful in combination with structural attributes

In these experiments, we have investigated the predictive power of the simplest features, i.e., the descriptive attributes versus the impact of the structural attributes. Figure 5.3 shows the accuracy results from the decision-tree classifier. When we use only descriptive attributes, the link-prediction accuracy varies across datasets. In Dogster, there is some advantage to using descriptive attributes, yet the accuracy (F1 score) is relatively low (37.6%). In Catster and Hamsterster, building the complete decision trees led to 0.4% and 19.8% accuracy, respectively (using Weka's default pruning parameter, the trees were empty, and the accuracies were 0%). This confirms that, in general, link prediction is a challenging prediction task.

When we used the structural features (such as number of friends that two profiles share), the link-prediction accuracy increased to 76.1% in in Dogster. This suggests that the structural features are much more predictive than simple descriptive attributes. This effect was even more pronounced for Catster and Hamsterster.

In Dogster, combining the node attributes and the structural features leads to futher improvement. Using descriptive attributes together with structural attributes leads to a better F1 score (78.6%) as compared to using either category alone (37.6% and 76.1%, respectively) in Dogster, as shown in Figure 5.3. For Catster and Hamsterster, the difference was less than 0.4%.

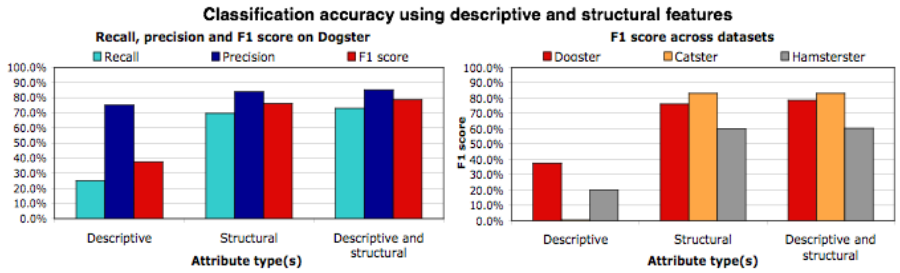


Figure 5.3: a) Recall, precision, and F1 score for Dogster using descriptive and structural attributes; b) F1 score across datasets. Using descriptive attributes together with structural attributes leads to a better F1 score in Dogster but not in Catster and Hamsterster.

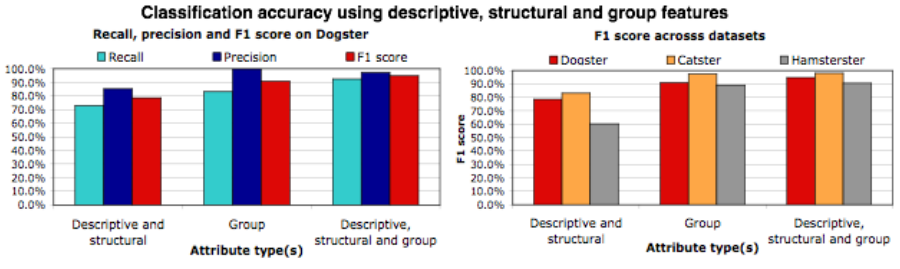


Figure 5.4: Link-prediction accuracy using all feature classes: descriptive, structural and group features. a) Recall, precision, and F1 score for Dogster; b) F1 score across datasets. Group features are highly predictive, yet adding the other features provided benefit too.

Family group features are highly predictive

As the previous experiments showed, structural attributes are stronger predictors than the descriptive attributes alone. Next, we investigate the predictive power of the group features in our taxonomy. In Dogster, Catster and Hamsterster, the group features involve the families and friends of the users. Figure 5.4 shows our comparisons. Our results suggested that family groups are strong predictors for friendship links ($F1 = 90.8\%$ for Dogster). We also ran experiments where we used not only family cliques, but also the structural and descriptive features. In these experiments, the results show that the accuracy (F1) improves by 4% in Dogster, 0.6% in Catster and 1.3% Hamsterster.

Computing more expensive structural attributes is not highly beneficial

Some structural features in our taxonomy were more computationally expensive to construct than others. For example, the feature that described the number of friends is easy to compute, whereas the feature that described the density of common friends for each pair of profiles is the hardest. Using a database, computing density of common friends for all pairs of profiles requires several joins of large tables. In order to investigate the trade-off between computing expensive features and their predictive impact on our results, we have performed the following experiments.

We have designed experiments in which we add more expensive structural features one by one, and assess the link-prediction accuracy at each step. We

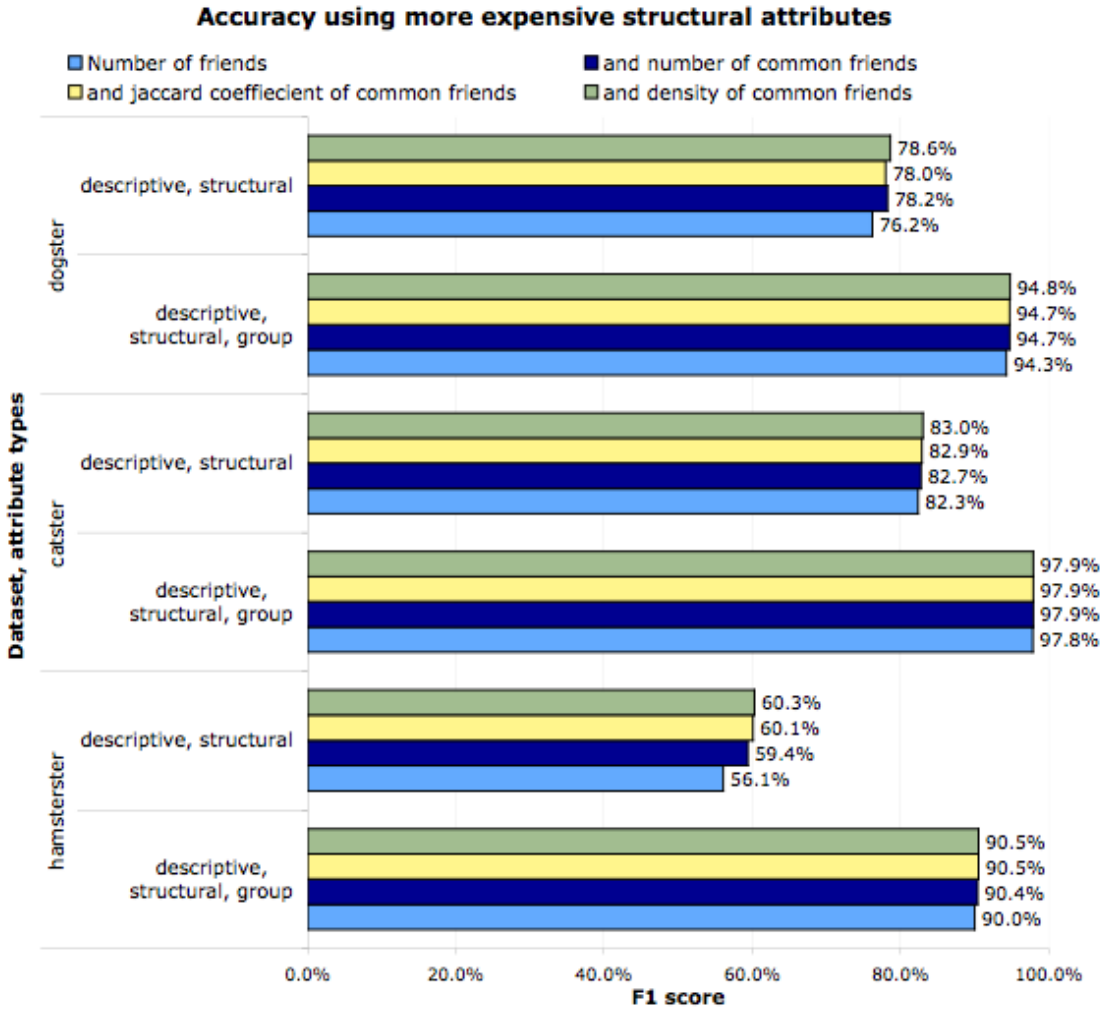


Figure 5.5: Link-prediction accuracy using structural features of increasing computational cost (number of friends, number of common friends, jaccard coefficient of common friends, density of common friends). Computing more expensive structural attributes is not highly beneficial, especially in the presence of group information.

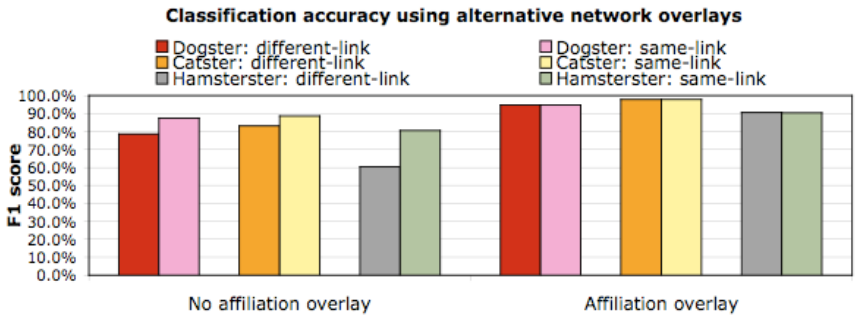


Figure 5.6: Prediction accuracy when links are treated equally, with and without group affiliations. As the results from the affiliation overlays suggest, group features are the main contributor to the high link-prediction accuracy.

used the following combinations of features: (1) using *number of friends* only, (2) using *number of friends* and *number of common friends*, (3) using *number of friends*, *number of common friends* and *jaccard coefficient*, and finally (4) using *number of friends, number of common friends, jaccard coefficient* and *density of common friends*. We are reporting on the results of these four sets of structural features together with the descriptive attributes since we showed in the previous subsection that using descriptive attributes can sometimes be beneficial. We also report on the setting in which group features were used.

Surprisingly, it turned out that computing the more expensive features added very little benefit. Figure 5.5 shows the results of the experiments. For example, in the Dogster case, adding the *number of common friends* of two nodes improved accuracy (F1 score) by 2% over the individual *number of friends*. Computing the most expensive feature *density of common friends* pays off slightly (improves F1 score by 0.4%) only when there are no group attributes. Computing the more expensive *jaccard coefficient* did not pay off over using the simpler feature *number of common friends*. In the Catster and Hamsterster cases, the improvement was less

that 0.5%. Our results also support the claim made in the preferential attachment model [11] that the number of friends of a node (node degree) plays a role in the process of new nodes linking to it. They contradict the link-prediction results in co-authorship networks [80] where *jaccard coefficient* and the *number of common friends* consistently out-performed the metric based on number of friends. This may be inherent to the types of networks discussed.

Alternative network representations

In the next set of experiments, we used the alternative network overlays to test whether there was an advantage to keeping the different types of links and the affiliation groups. We compare our proposed *different-link and affiliation overlay* to the alternative representations *same-link and no affiliation overlay* and *same-link and affiliation overlay* (see Section 5.3). We compute only the descriptive and structural features in the overlay with no affiliation information, and compute all classes of features in the overlays where affiliation information was given.

The results on Figure 5.6 show that when family affiliations were given, it did not matter whether the links were treated as the same type or different types: the link-prediction accuracy was the same. However, in the case when the affiliations were not given, it was better to compute the structural features using both types of relationships but treat them as one type. When family links were treated as friendship links, the accuracy of the predictions made by the structural attributes improved by 6% to 20%. This may be due to the fact that the overlap between friends and family links in the data was very small, and using both

types of links when computing the structural features was beneficial. Using the affiliation information and computing all features on the data led to the best accuracy, and the accuracy was the same both in the different-link and same-link cases. These experiments also confirmed the previous results: group affiliation was the main contributor to the high link-prediction accuracy.

5.5 Discussion

When studying other large social networks, family information is not always relevant or available. However, groups and affiliations are often available, or communities can be discovered.

The networks used here had a binary relationships - friend or family - but a similar effect can be achieved in networks where relationships are weighted. For example, co-authorship networks are widely studied as social networks [11, 52, 80, 117, 118, 125, 127], and edges can be weighted by the number of articles a pair of authors have authored together. In email communication networks - the Enron email corpus [62, 67], for example - the number of messages between two senders can be used as a weight. To mimic the strong family-type relationship we used in this article, a threshold weight can be set. Any edge with a weight over that threshold can be treated as a "strong" relationship (like our family relationship). Clusters of nodes connected with strong ties represent the equivalent of a family unit.

5.6 Conclusions

Link prediction is a notoriously difficult problem. In this research, we found that overlaying friendship and affiliation networks was very effective. For the networks used in our study, we found that family relationships were very useful in predicting friendship links. Our experiments show that we can achieve significantly higher prediction accuracy (between 15% and 30% more accurate) as compared to using more traditional features such as descriptive node attributes and structural features. Family groups helped not only because they represent a clique of actors, but because the family relationship itself was indicative of structural equivalence.

Part III

Privacy in Social and Affiliation Networks

While predictive statistical models allow learning hidden information automatically in social and affiliation networks, they also bring many privacy concerns because of the potentially sensitive nature of personal data. Even though disclosing information on the web is a voluntary activity on the part of the users, users are often unaware of who is able to access their data and how their data can potentially be used.

Data privacy is defined as "freedom from unauthorized intrusion" [140]. However, what constitutes an unauthorized intrusion in social networks is an open question. Because privacy in social networks is a young field, we first identify the space of problems in this emerging area in Chapter 6. When appropriate we present existing work, but many of these problems have not yet been addressed in the research literature. One of the contributions of this chapter is in cataloging the different types of privacy disclosures in social networks. These are studied in the research literature but they are often not explicitly formalized. Chapter 6 allows to present our work in the context of other research on privacy in social networks.

In Chapter 6, we focus on two scenarios for privacy in social networks: privacy breaches and data anonymization. In the first scenario, an adversary is interested in learning the private information of an individual using publicly available social network data, possibly anonymized. In the second scenario, a data provider is interested to release a social network dataset to researchers but preserve the privacy of its users. For this purpose, the data provider needs to provide

a privacy mechanism, so that researchers can access the (possibly perturbed) data in a manner which does not compromise users' privacy. A common assumption in the data anonymization literature is that the data is described by a single table with attribute information for each of the entries. However, social network data can exhibit rich dependencies between entities which can be exploited for learning the private attributes of users, and we explore the consequences of this possibility.

The privacy literature recognizes two types of privacy mechanisms: interactive and non-interactive [34]. In the interactive mechanism, an adversary poses queries to a database, and the database provider gives noisy answers. In the non-interactive setting, a data provider releases an anonymized version of the database to meet privacy concerns.

In addition to defining the space of problems, we study two specific privacy problems in social networks which we present in Chapter 7 and Chapter 8. The first problem is attribute disclosure: inferring the private attributes of social network users using their online social environment [153]. While this work has similarities with both privacy mechanisms, the goal of our data provider is not to anonymize a dataset or provide noisy answers but to ensure that users' private data remains private and cannot be inferred using links, groups and public profiles. The second problem is link re-identification – inferring that two entities participate in a particular type of sensitive relationship or communication [152]. We study this problem in in the non-interactive, anonymization setting. The challenge of anonymizing graph data lies in understanding the complex dependen-

cies and removing sensitive information which can be inferred by direct or indirect means.

In the attribute disclosure work (Chapter 7), we show how an adversary can exploit an online social network with a mixture of public and private user profiles to predict the private attributes of users. We map this problem to a relational classification problem and we propose practical models that use friendship and group membership information (which is often *not* hidden) to infer sensitive attributes. The key novel idea is that in addition to friendship links, groups can be carriers of significant information. We show that on several well-known social media sites, we can easily and accurately recover the information of private-profile users. To the best of our knowledge, this is the first work that uses link-based and group-based classification to study privacy implications in social networks with mixed public and private user profiles.

Traditionally, only two types of privacy attacks have been studied in the privacy literature: attribute disclosure and identity disclosure. We identify a third type of disclosure which can occur in social networks: link disclosure. In our link re-identification work (Chapter 8), we focus on the problem of preserving the privacy of sensitive relationships in the non-interactive, anonymization setting. We propose five different privacy preservation strategies, which vary in terms of the amount of data removed (and hence their utility) and the amount of privacy preserved. We assume the adversary has an accurate predictive model for links, and we show experimentally the success of different link re-identification strategies under varying structural characteristics of the data.

A common assumption in the anonymization literature is that the data is described by a single table with attribute information for each of the entries. However, real-world datasets often exhibit more complexity. Social network data, often represented as a multi-graph, can exhibit rich dependencies between entities. Our link-reidentification work is novel on two fronts: 1) breaking the assumption that the data to be anonymized is a flat-table data and 2) studying a new type of privacy attack which can occur in network data.

Chapter 6

Privacy in Social Networks

Here, we survey the literature on privacy in social networks [154]. We formally define the possible privacy breaches and describe the privacy attacks that have been studied. We present definitions of privacy in the context of anonymization together with existing anonymization techniques. While initial steps have been taken in understanding and overcoming some of the challenges of preserving privacy online, many open problems remain. In particular, some exciting new directions include studying the effect of different types of privacy disclosures on each other, privacy-preserving techniques that prevent sensitive attribute disclosure in networks, a comparison between existing anonymization techniques in terms of utility, and privacy-preserving techniques that meet the individual privacy expectations of online social network users rather than privacy definitions imposed by a data publisher or an online service provider.

In this chapter, we focus on privacy breaches in online social networks, as well as on privacy-preserving techniques for publishing social network data.

In addition, there are other existing surveys on privacy preservation in social networks that focus on different aspects [27, 55, 85, 147, 160]. The surveys on privacy-preserving data publication for networks cover privacy attacks, edge modification, randomization and generalization privacy-preserving strategies for network structure [85, 147, 160] and richer graphs [147]. Clarkson et al. [27] discuss anonymization techniques which aim to prevent identity disclosure. The survey of Hay et al. [55] concentrates on privacy issues with network structure, and it covers attacks and their effectiveness, anonymization strategies, and differential privacy for private query answering.

In Section 6.1, we discuss the different types of privacy breaches: private information that can leak from a social network. We define the types of queries for each type of disclosure, and ways to measure the extent to which a disclosure has occurred in an online or anonymized social network. We are abstracting these definitions from the types of privacy breaches that have been studied in data anonymization. The definitions can be applied both in the anonymization scenario and in the scenario of an intrusion in an online social network. We also provide pointers to work which studies these privacy breaches in the context of anonymization. We present privacy definitions in Section 6.2 and privacy mechanisms for publishing social network data in Section 8.1.

6.1 Privacy breaches in social networks

When studying privacy, it is important to specify what defines a failure to preserve privacy. A *privacy breach* occurs when a piece of sensitive information about an individual is disclosed to an adversary, someone whose goal is to compromise privacy. Traditionally, two types of privacy breaches have been studied: *identity disclosure* and *attribute disclosure*. We discuss these two types in the context of social networks. We also present two more disclosure types, specific to network data: *social link disclosure* and *affiliation link disclosure*.

6.1.1 Identity disclosure

Identity disclosure occurs when an adversary is able to determine the mapping from a profile v in the social network to a specific real-world entity p. Before we are able to provide a formal definition of identity disclosure, let us consider three questions related to the identity of p in which an adversary may be interested.

Definition 1 Mapping query. *In a set of individual profiles V in a social network G, find which profile v maps to a particular individual p. Return v.*

Definition 2 Existence query. *For a particular individual p, find if this individual has a profile v in the network G. Return true or false.*

Definition 3 Co-reference resolution query. *For two individual profiles v_i and v_j, find if they refer to the same individual p. Return true or false.*

A simple way of defining *identity disclosure* is to say that the adversary can answer the *mapping query* correctly and with full certainty. However, unless the adversary knows unique attributes of individual p that can be matched with the observed attributes of profiles in V, this is hard to achieve. One way of formalizing *identity disclosure* for an individual p is to assoicate a random variable $\hat{v}_p$ which ranges over the profiles in the network. We assume that the adversary has a way of computing the probability of each profile v_i belonging to individual p, $Pr(\hat{v}_p = v_i)$. In addition, we introduce a dummy profile v_{dummy} in the network which serves the purpose of absorbing the probability of individual p not having a profile in the network. We assume that p has exactly one profile, and the true profile of p in $V \cup \{v_{dummy}\}$ is v_*. We use the shorthand $Pr_p(v_i) = Pr(\hat{v}_p = v_i)$ to denote the probability that v_i corresponds to p; Pr_p provides a mapping $Pr_p : V \cup \{v_{dummy}\} \rightarrow \mathbb{R}$. We leave it open as to how the adversary constructs Pr_p. Then we can define *identity disclosure* as follows:

Definition 4 Identity disclosure with confidence t. *In a set of individual profiles V in a social network G, identity disclosure occurs with confidence t when $Pr_p(v_*) \geq t$ and $v_* \neq v_{dummy}$.*

An alternative definition of *identity disclosure* considers that the possible values of v_i can be ranked according to their probabilities.

Definition 5 Identity disclosure with *top-k* confidence. *In a set of individual profiles V in a social network G, identity disclosure occurs with* top-k *confidence when v_**

appears in the top k profiles (or top $p\% = k * 100/|V|$), in the list of profiles ranked by Pr_p from high to low.

The majority of research in social network privacy has concentrated on identity disclosure [7, 22, 56, 57, 71, 86, 111, 145, 148, 159, 162]. We discuss it in more detail in Section 8.1.

6.1.2 Attribute disclosure

A common assumption in the privacy literature is that there are three types of possibly overlapping sets of personal attributes:

- Identifying attributes - attributes, such as social security number (SSN), which identify a person uniquely.
- Quasi-identifying attributes - a combination of attributes which can identify a person uniquely, such as name and address.
- Sensitive attributes - attributes that users may like to keep hidden from the public, such as political affiliation and sexual orientation.

Attribute disclosure occurs when an adversary is able to determine the value of a sensitive user attribute, one that the user intended to stay private. This attribute can be an attribute of the node itself, the node's links or the node's affiliations. Without loss of generality, here we discuss the attributes of the node itself. Again, to make this definition more concrete, we assume that each sensitive attribute $v.a_s$ for profile v has an associated random variable $v.\hat{a}_s$ which

ranges over the possible values for $v.a_s$. Let the true value of $v.a_s$ be $v.a_*$. We also assume that the adversary can map the set of possible sensitive attribute values to probabilities, $Pr_a(v.\hat{a}_s = v.a) : v.a \rightarrow \mathbb{R}$, for each possible value $v.a$. Note that this mapping can be different for each node/profile. Now, we can define attribute disclosure as follows:

Definition 6 Attribute disclosure with confidence t. *For a profile v with a hidden attribute value $v.a_s = v.a_*$, attribute disclosure occurs with confidence t when $Pr_a(v.\hat{a}_s = v.a_*) \geq t$.*

Similarly to *identity disclosure*, there is an alternative definition of *attribute disclosure* which considers that the possible values of $v.A_s$ can be ranked according to their probabilities.

Definition 7 Attribute disclosure with *top-k* confidence. *For a profile v with a hidden attribute value $v.a_s = v.a_*$, attribute disclosure occurs with* top-k *confidence when a_* appears in the top k values of the list of possible values ranked by their probabilities Pr_a.*

Clearly, if an adversary can see the identifying attributes in a social network, then answering the identity *mapping query* becomes trivial, and identity disclosure with confidence 1 can occur. For example, if a profile contains a SSN, then identifying the real person behind the profile is trivial since there is a one-to-one mapping between individuals and their social security numbers. Therefore, in order to prevent identity disclosure, the identifying attributes have to be removed from the profiles.

Sometimes, a combination of attributes, known as *quasi-identifying attributes*, can lead to identity disclosure. What constitutes *quasi-identifying attributes* depends on the context. For example, it has been observed that 87% of individuals in the U.S. Census from 1990 can be uniquely identified based on their date of birth, gender and zip code [136]. Another example of quasi-identifiers is a combination of a person's name and address.

Similarly, matching records from different datasets based on quasi-identifying attributes can lead to further privacy breaches. This is known as a *linking attack*. If the identities of users in one dataset are known and the second dataset does not have the identities but it contains sensitive attributes, then the sensitive attributes of the users from the first dataset can be revealed. For example, matching health insurance records, in which the identifying information is removed, with public voter registration records can reveal sensitive health information about voters. Using this attack, Sweeney was able to identify the medical record of the governor of Massachusetts [136].

In the context of social and affiliation networks, there has not been much work on sensitive attribute disclosure. Most studies look at how attributes can be predicted [110, 82, 153], and very few on how they can be protected [22]. We discuss this work in more detail in Section 8.1.

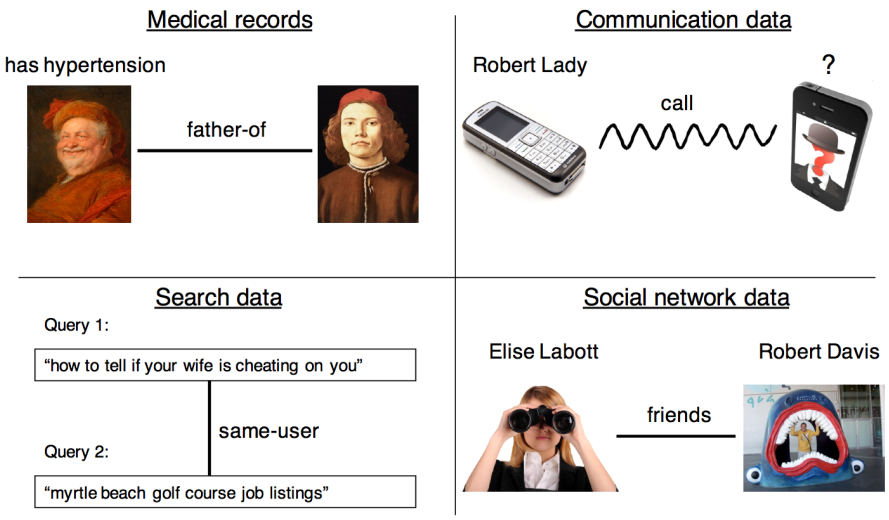


Figure 6.1: Sensitive link examples.

6.1.3 Social link disclosure

Social link disclosure occurs when an adversary is able to find out about the existence of a sensitive relationship between two users, a relationship that these users prefer to remain hidden from the public. Similarly to the previous types of disclosures, we assume that there is a random variable $\hat{e}_{i,j}$ associated with the link existence between two nodes v_i and v_j, and an adversary has a model for assigning a probability to $\hat{e}_{i,j}$, $Pr(\hat{e}_{i,j} = true) : e_{i,j} \rightarrow \mathbb{R}$.

Definition 8 Social link disclosure with confidence t. *For two profiles* v_i *and* v_j, *a social link disclosure occurs with confidence* t *when* $e_v(v_i, v_j) \in E_v$ *and* $Pr(\hat{e}_{i,j} = true) \geq t$.

Note that since the link existence $\hat{e}_{i,j}$ has only two possible values, true and false, the *top-k* definition does not apply to social link disclosure.

Examples of sensitive relationships can be found in social networks, communication data, disease data and others. In social network data, based on the

friendship relationships of a person and the public preferences of the friends such as political affiliation, it may be possible to infer the personal preferences of the person in question as well. In cell phone communication data, finding that an unknown individual has made phone calls to a cell phone number of a known organization can compromise the identity of the unknown individual. In hereditary disease data, knowing the family relationships between individuals who have been diagnosed with hereditary diseases and ones that have not, can help infer the probability of the healthy individuals to develop these diseases. Figure 6.1 presents a summary of these examples.

Researchers have studied attacks that expose sensitive links in social networks [7, 14, 72, 152]. Sensitive edge properties, such as link strength (or weight), have also been the focus of recent work [31, 88].

6.1.4 Affiliation link disclosure

Another type of privacy breach in relational data is *affiliation link disclosure*: whether a person belongs to a particular affiliation group. Whether two users are affiliated with the same group can also be of sensitive nature. Sometimes, affiliation link disclosure can lead to attribute disclosure, social link disclosure, or identity disclosure. Thus, hiding affiliations is a key to preserving the privacy of individuals.

As before, we assume that there is a random variable $\hat{e}_{v,h}$ associated with the existence of an affiliation link between a profile v and a group h, and that an

adversary has a way of computing the probability of $\hat{e}_{v,h}$, $Pr(\hat{e}_{v,h} = true) : e_{v,h} \rightarrow \mathbb{R}$.

Definition 9 Affiliation link disclosure with confidence t. *For a profile v and an affiliation group h, an affiliation link disclosure occurs with confidence t when $e_h(v,h) \in E_h$ and $Pr(\hat{e}_{v,h} = true) \geq t$.*

One type of disclosure can lead to another type. For example, Wondracek et al. [145] show a de-identification attack in which affiliation link disclosure can lead to the identity disclosure of a supposedly anonymous Internet user. An adversary starts the attack by crawling a social networking website and collecting information about the online social group memberships of its users. It is assumed that the identities of the social network users are known. According to the collected data, each user who participates in at least one group has a group signature, which is the set of groups he belongs to. Then, the adversary applies a *history stealing attack* (for more details on the attack, see [145]) which collects the web browsing history of the target Internet user. By finding the group signatures of social network users which match the browsing history of the Internet user, the adversary is able to find a subset of potential social network users who may be the Internet user. In the last step of the attack, the adversary looks for a match between the id's of the potential users and the browsing history of the target individual, which can lead to de-identification of the Internet user.

Another example of affiliation link disclosure leading to identity disclosure is in search data. If we assume that users posing queries to a search engine are

the individuals in the social network, and the search queries they pose are the affiliation groups, then disclosing the links between users and queries can help an adversary identify people in the network. Users interact with search engines in an uninhibited way and reveal a lot of personal information in the text of their queries. There was a scandal in 2006 when AOL, an Internet Service provider, released an "anonymized" sample of over half a million users and their queries posed to the AOL search engine. The release was well-intentioned and meant to boost search ranking research by supplementing it with real-world data. Each user was specified by a unique identifier, and each query contained information about the user identifier, search query, the website the user clicked on, the ranking of that website in the search results, and the timestamp of the query.

Table 6.1: A snapshot of the data released by AOL. Here, we are omitting the timestamps included in the data.

User ID	*Search query*	*Clicked website*	*Ranking*
4417749	clothes for age 60	http://www.news.cornell.edu	10
4417749	dog who urinate on everything	http://www.dogdayusa.com	6
4417749	landscapers in lilburn ga.		
4417749	pine straw in lilburn ga.	http://gwinnett-online.com	9
4417749	gwinnett county yellow pages	http://directory.respond.com	1
4417749	best retirement place in usa	http://www.amazon.com	7
4417749	mini strokes	http://www.ninds.nih.gov	1

One of the problems with the released data was that even though it was in a table format (Table 6.1), its entries were not independent of each other. Shortly after the data release, New York Times reporters linked 454 search queries made by the same individual which gave away enough personal information to identify that individual – Thelma Arnold, a 62-year old widow from Lilburn, Georgia [12]. Her queries included names of people with the same last name as hers,

information about retirement, her location, etc.

Affiliation link disclosure can also lead to attribute disclosure, as illustrated in a *guilt-by-association attack* [29]. This attack assumes that there are groups of users whose sensitive attribute values are the same, thus recovering the sensitive value of one user and the affiliation of another user to the group can help recover the sensitive value of the second user. This attack was used in the BitTorrent file-sharing network to discover the downloading habits of users [26]. Communities were detected based on social links, and monitoring only one user in each community was enough to infer the interests of the other people in the community. In this case the sensitive attribute that users prefer to keep private is whether they violate copyrights. This attack has also been applied to identifying fraudulent callers in a phone network [29]. Cormode et al. [28] study data anonymization to prevent affiliation link disclosure. They refer to affiliation links as associations (see Section 6.3.2).

6.2 Privacy definitions for publishing data

The goal of data mining is discovering new and useful knowledge from data. Sometimes, the data contains sensitive information, and it needs to be sanitized before it is published publicly in order to address privacy concerns. Data sanitization is a complex problem in which hiding private information trades off with utility reduction. The goal of sanitization is to remove or perturb the attributes of the data which help an adversary infer sensitive information. The

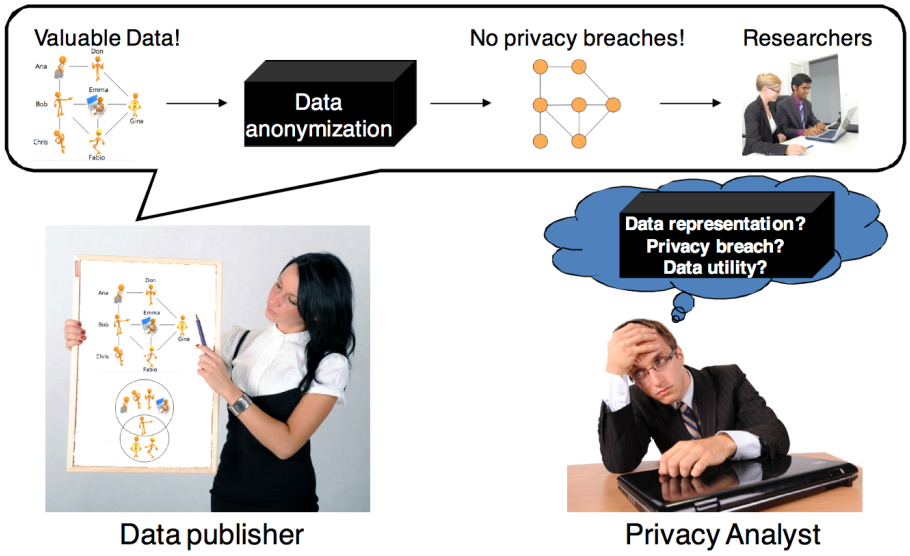


Figure 6.2: Anonymization scenario.

solution depends on the properties of the data and the notions of privacy and utility in the data.

Privacy preservation in the context of social network data is a relatively new research field. Rather than assuming data which is described by a single table of independent records with attribute information for each, it takes into consideration more complex real-world datasets. As discussed earlier, relational data, often represented as a multi-graph, can exhibit rich dependencies between entities. The challenge of sanitizing graph data lies in understanding these dependencies and removing sensitive information which can be inferred by direct or indirect means.

One way in which data providers can sanitize data is by anonymization. Figure 8.1 shows a typical scenario in which a data owner is interested in providing researchers with valuable data and in order to meet privacy concerns, she

consults a privacy analyst before publishing a perturbed version of the data. In the process of anonymizing the data, the identifying information is removed and other attributes are perturbed. Anonymizing techniques have been criticized as often being ad hoc and not providing a principled way of preserving privacy. There are no guarantees that an adversary would not be able to come up with an attack which uses background information and properties of the data, such as node attributes and observed links, to infer the private information of users. Another way of sanitizing data is by providing a private mechanism for accessing the data, such as allowing algorithms which are provably privacy-preserving to run on it. Next, we will discuss privacy preservation definitions. Some of these definitions were not developed specifically for network data but we provide examples from the social network domain.

To formalize privacy preservation, Chawla et al. [24] proposed a framework based on the intuitive definition that "our privacy is protected to the extent we blend in the crowd." Obviously, with the richness of information in online social network profiles, this is hard to achieve and users are easily identifiable. We will look at a simpler case when a data provider is interested in releasing a dataset with online social network profiles. To give a flavor of existing work, we present four existing privacy preservation approaches which make the definition of "blending in the crowd" more concrete.

6.2.1 k-anonymity

k-anonymity protection of data is met if the information for each person contained in the data cannot be distinguished from at least $k-1$ other individuals in the data. k-anonymity can be achieved by suppressing and generalizing the attributes of individuals in the data. Suppressing an attribute value means deleting it from the perturbed data. Generalizing an attribute means replacing it with a less specific but semantically consistent value. One can see that suppression is a special case of generalization, and that suppressing all attributes guarantees k-anonymity. This is why a notion of utility in the data has to be incorporated whenever sanitizing data. The actual objective is to maximize utility by minimizing the amount of generalization and suppression. Achieving k-anonymity by generalization with this objective as a constraint is an NP-hard problem [5]. k-anonymity has been studied mostly for table data, so we begin by presenting its definition using only the nodes V and their attributes $V.A$, i.e., disregarding links and affiliation groups.

Definition 10 **k-anonymity**. *A set of records V satisfies k-anonymity if for every tuple $v \in V$ there exist at least $k-1$ other tuples $v_{i_1}, v_{i_2}, ..., v_{i_{k-1}} \in V$ such that $v_{i_1}.A_q = v_{i_2}.A_q = ... = v_{i_{k-1}}.A_q$ where $A_q \in A$ are the quasi-identifying attributes of the profile.*

Figure 6.3 shows an example of applying 5-anonymity to the data of 10 individuals. The data includes their names, ages, genders and zip codes. The perturbed data meets a 5-anonymity constraint because each individual is indistinguishable from at least 4 other individuals. Here, the assumption is that name is

Identifier	Quasi-identifiers			Sensitive
Name	Age	Sex	Zip	Pol. views
Ana	21	F	20740	liberal
Bob	25	M	83222	liberal
Chris	24	M	20742	liberal
Don	29	M	83209	conservative
Emma	24	F	20640	liberal
Fabio	24	M	20760	liberal
Gina	28	F	83230	liberal
Halle	29	F	83201	conservative
Ian	31	M	83220	conservative
John	24	M	20740	liberal

5-anonymity applied to data →

Equiv. class	Quasi-identifiers			Sensitive
	Age	Sex	Zip	Pol. views
C1	[21,24]	*	20***	liberal
C2	[25,31]	*	832**	liberal
C1	[21,24]	*	20***	liberal
C2	[25,31]	*	832**	conservative
C1	[21,24]	*	20***	liberal
C1	[21,24]	*	20***	liberal
C2	[25,31]	*	832**	liberal
C2	[25,31]	*	832**	conservative
C2	[25,31]	*	832**	conservative
C1	[21,24]	*	20***	liberal

Figure 6.3: 5-anonymity applied to data with 10 records.

an identifying attribute, therefore it has been suppressed. Three of the attributes, *Age*, *Sex* and *Zip code*, are quasi-identifiers, therefore, they have been generalized. The sensitive attributes remain the same.

k-anonymity provides a clustering of the nodes into equivalence classes such that each node is indistinguishable in its quasi-identifying attributes from some minimum number of other nodes. In the previous example, there were two equivalence classes: class $C1$ of individuals whose age is in the range $[21, 24]$ years and have a zip code $20 * **$, and class $C2$ of individuals whose age is in the range $[25, 31]$ years and have a zip code $832 * *$. Note, however, that these equivalent classes are based on node attributes only, and inside each equivalence class, there may be nodes with different identifying structural properties and edges. This makes it hard to define k-anonymity for nodes in social networks. We discuss some approaches later in Section 8.1.

k-anonymity ensures that individuals cannot be uniquely identified by a linking attack. However, it does not necessarily prevent sensitive attribute disclosure. Here, we present two possible attacks on k-anonymized data [93]. The first one can occur when there is little diversity in the sensitive attributes inside an equivalence class. In this case, the sensitive attribute of everyone in the equivalence class becomes known with high certainty. For example, if an adversary wants to figure out Ana's political views knowing that her age is 21 and her zip code is 20740, then he can figure out that her record is in equivalence class $C1$. There is no diversity in the sensitive attribute value of equivalence class $C1$, i.e., everyone in $C1$ has liberal political views, therefore, the adversary is able to infer Ana's political views even though he does not know which row corresponds to her. This is known as the *homogeneity attack* [93].

The second problem with k-anonymity is that in the presence of background knowledge, attribute and identity disclosure can still occur. For example, knowing that someone's friends are liberal, makes it highly likely that this person is liberal as well. In our toy example, the knowledge that Gina's friends, Emma and Fabio, belong to equivalence class $C1$ where everyone is liberal, can help an adversary infer with high certainty that Gina is liberal as well. This is known as the *background attack* [93].

There are a number of definitions derived from k-anonymity tailored to structural properties of network data. Some examples include *k-degree anonymity* [86], *K-Candidate anonymity* [57], *k-automorphism anonymity* [162], *k-neighborhood anonymity* [159, 147], and *(k,l)-grouping* [28]. We introduce the intuition behind

them, together with their definitions in Section 6.3.1 and Section 6.3.2, privacy mechanisms for networks.

6.2.2 l-diversity and t-closeness

A privacy definition which alleviates the problem of sensitive attribute disclosure inherent to k-anonymity is l-diversity [93]. As its name suggests, l-diversity ensures that the sensitive attribute values in each equivalence class are diverse.

Definition 11 **l-diversity**. *A set of records in an equivalence class C is l-diverse if it contains at least l "well-represented" values for each sensitive attribute. A set of nodes V satisfy l-diversity if every equivalence class $C' \subseteq V$ is l-diverse.*

There are a number of ways to define "well-represented." Some examples include using frequency counts and measuring entropy. However, even in the case of l-diverse data, it is possible to infer sensitive attributes when the sensitive distribution in a class is very different from the overall distribution for the same attribute. If the overall distribution is skewed, then the belief of someone's value may change drastically in the anonymized data (*skewness attack*) [79]. For example, only 30% of the records in Figure 6.3 have conservative political views. However, in equivalence class $C2$ this number becomes 60%, thus the belief that a user is conservative increases for users in $C2$. Another possible attack, known as the *similarity attack* [79], works by looking at equivalent classes which contain very similar sensitive attribute values. For example, if Age is a sensitive attribute and an adversary wants to figure out Ana's age knowing that she is in equiva-

lence class $C1$ (based on her *Zip code*), then he would learn that she is between 21 and 24 years old which is a much tighter age range than the range in the whole dataset.

This leads to another privacy definition, t-closeness, which considers the sensitive attribute distribution in each class, and its distance to the overall attribute distribution. The distance can be measured with any similarity score for distributions.

Definition 12 **t-closeness**. *A set of records in an equivalence class C is t-close if the distance between the distribution of a sensitive attribute A_s in C and its distribution in V is no more than a threshold t. A set of nodes V satisfy t-closeness if every equivalence class $C' \subseteq V$ is t-close.*

Just like with k-anonymity, sanitizing data to meet either l-diversity or t-closeness comes with a computational complexity burden. There are other privacy definitions of this flavor but they have all been criticized for being ad hoc. While they guarantee syntactic properties of the released data, they come with no privacy semantics [35].

6.2.3 Differential privacy

The notion of differential privacy was developed as a principled way of defining privacy, so that "the risk to one's privacy [...] should not substantially increase as a result of participating in a database" [34]. This shifts the view on privacy from comparing the prior and posterior beliefs about individuals before and

after publishing a database to evaluating the risk incurred by joining a database. It also imposes a guarantee on the data release mechanism rather than on the data itself. Here, the goal is to provide statistical information about the data while preserving the privacy of users in the data. This privacy definition gives guarantees that are independent of the background information and the computational power of the adversary.

Returning to our running example, if the social network data set is released using a differentially private mechanism, this guarantees that Ana's participation in the social network does not pose a threat to her privacy because the statistics would not look very different without her participation. It *does not* guarantee that one cannot learn sensitive information about Ana using background information but such guarantee is impossible to achieve for any kind of dataset [34].

Definition 13 **ϵ-differential privacy**. *A randomized function K satisfies ϵ-differential privacy if for all data sets D_1 and D_2 differing in at most one element, and any subset S of possible outcomes in $Range(K)$,*

$$P(K(D_1) \in S) \leq exp(\epsilon) \times P(K(D_2) \in S). \tag{6.1}$$

Here, one can think of a profile in the social network as being an element, and V being the data set, thus $D_1 \subseteq V$ and $D_2 \subseteq V$. The randomized function K can be thought of as an algorithm which returns a random variable, possibly with some noise. When developing a differentially private algorithm, one has to keep in mind the utility of the data and incorporate the desired knowledge in

the algorithm. $Range(K)$ is the output range of algorithm K. A common way of achieving ϵ-differential privacy is by adding random noise to the query answer.

One type of algorithm that has been proven to be differentially private is a $count$ query to which one adds Laplacian noise [37]. For example, if the count query is K =*"How many people are younger than 22?"*, then the output range of the query is $Range(K) = \{1, ..., n\}$ where n is the size of the social network. The count query is considered a low-sensitivity query because it has a sensitivity of $\Delta K = 1$ for any D_1 and D_2 differing in one element. Sensitivity is defined as

$$\Delta K = \max_{D_1, D_2} ||K(D_1) - K(D_2)|| \tag{6.2}$$

for any D_1 and D_2 which differ in at most one element. Note that this query has the same sensitivity not only for our specific data but for any data in this format. The Laplacian noise, which is added to the answer, is related to the sensitivity of the query.

A $mean$ query, such as K =*"What is the average age of people in the social network?"*, has an even lower sensitivity for large data sets because removing any profile from the social network changes the output of the query by at most $\Delta K = max(age)/n$. There are also queries, such as $median$ queries, which have high sensitivity and require different techniques for generating noise.

A similar and somewhat weaker definition of differential privacy is the one of (ϵ, δ)-differential privacy which was developed to deal with very unlikely outputs of K [36].

Definition 14 (ϵ,δ)-differential privacy. *A randomized function K satisfies ϵ-differential privacy if for all data sets D_1 and D_2 differing in at most one element, and any subset S of possible outcomes in $Range(K)$,*

$$P(K(D_1) \in S) \leq exp(\epsilon) \times P(K(D_2) \in S) + \delta. \tag{6.3}$$

Generally, ϵ and δ are considered to be very small numbers and are picked according to different considerations, such as the size of the database.

Machanavajjhala et al. provide the first theoretical study of the privacy-utility trade-offs in link recommendation systems for social networks [94]. Recent work has also shown that "differential privacy does not always adequately limit inference about participation in social networks" [65].

6.3 Privacy-preserving mechanisms

So far, we have discussed existing notions of privacy preservation related to the user profiles, mostly ignoring the structural properties of the social network. Next, we discuss how privacy preservation can be achieved considering the network structure: the links between users E_v, and affiliation links E_h to affiliation groups of users H. First, we present existing privacy mechanisms for social networks in Section 6.3.1. Section 6.3.2 includes overview of the mechanisms for affiliation networks. Finally, we describe research which considers both types of networks in Section 6.3.3. Except for the privacy mechanisms based on

differential privacy, each mechanism was developed to counterattack a specific adversarial attack and background knowledge which we also present.

6.3.1 Privacy mechanisms for social networks

The majority of research in this area considers anonymization which strips off all the personal attributes from user profiles but keeps some of the structure coming from the social links between users [7, 56, 57, 86, 159, 162]. We describe this research next. Then, we mention approaches to anonymizing data which consider that there is utility in keeping both user attributes and network structural [22, 152, 159].

Anonymizing network structure

One naíve way of anonymizing a social network is by removing all the attributes of the profiles, and leaving only the social link structure. This creates an anonymized graph which is isomorphic to the original graph. The intuition behind this approach is that if there are no identifying profile attributes, then attribute and identity disclosures cannot occur, and thus the privacy of users is preserved. Contrary to the intuition, this not only removes a lot of important information but it also does not guarantee the privacy of users. Two types of attacks have been proposed to show that identity and social link disclosures occur when it is possible to identify a subgraph in the released graph in which all the node identities are known [7]. The *active attack* assumes that an adversary can insert accounts in the network before the data release, and the *passive attack* as-

sumes that a number of friends can collude and share their linking patterns after the data release.

In the active attack an adversary creates k accounts and links them randomly, then he creates a particular pattern of links to a set of m other users that he is interested to monitor. The goal is to learn whether any two of the monitored nodes have links between them. When the data is released, the adversary can efficiently identify the subgraph of nodes corresponding to his k accounts with provably high probability. Then he can recover the identity of the monitored m nodes and the links between them which leads to social link disclosure for all $\binom{m}{2}$ pairs of nodes. With as few as $k = \Theta(\log n))$ accounts, an adversary can recover the links between as many as $m = \Theta(\log^2 n)$ nodes in an arbitrary graph of size n. The passive attack works in a similar manner. It assumes that the exact time point of the released data snapshot is known and that there are k colluding users who have a record of what their links were at that time point.

Another type of structural background information that has been explored is similar in spirit to the linking attack mentioned in Section 6.1. The existence of an auxiliary social network in which the identity of users is known can help an adversary identify nodes in a target social network [111]. Starting from a set of users which form a clique both in the target and the auxiliary networks, an adversary expands the matching by finding the most likely nodes that correspond to each other in the two networks by using structural information, such as number of user friends (node degree), and number of common neighbors. To validate this attack, it has been shown that the discovered matches sometimes correspond to

matches found using descriptive user attributes such as username and location in the social networks of Twitter and Flickr [111].

Structural privacy. Starting from the idea that certain subgraphs in the social network are unique, researchers have studied the mechanism of protecting individuals from identity disclosure when an adversary has *background information about the graph structure* around a node of interest [56, 57, 86, 148, 159, 162]. Each node has structural properties (subgraph signature) that are the same as the ones of a small set of other nodes in the graph, called a candidate set for this node [57]. Knowing the true structural properties of a node, an adversary may be able to discover the identity of that node in the anonymized network. Structure queries can be posed to the network to discover nodes with specific subgraph signatures.

Looking at the immediate one-hop neighbors, each node has a star-shaped subgraph in which the size of the subgraph is equal to the degree of the node plus one. With the assumption that identity disclosure can occur based on a node's degree, the degree becomes an identifying attribute that a data provider is interested to hide. In our toy network (Figure 1.2), Ana and Don are in each other's candidate sets because they both have degree 2; Emma, Gina and Fabio appear in the same candidate set for either of the three nodes; Bob and Chris are uniquely identifiable because they are the only ones in the network with degrees four and one, respectively. The notion of *k-degree anonymity* [86] was formulated to protect individuals from an adversary who has background information of user's node degrees. It states that each node should have the same degree as at least $k-1$

other nodes in the anonymized network.

Adding the links between the one-hop neighbors of a node, sometimes referred to as the 1.5-hop neighborhood, creates a richer structural signature. Based on this, Ana and Don still have the same subgraph signature, and so do Emma and Fabio. However, Gina has a unique signature and is easily identifiable by an adversary who has knowledge of her true 1.5-hop neighborhood structure. Zhou and Pei [159] formalize the desired property to protect individuals from this type of attack. A graph satisfies *k-neighborhood anonymity* if every node in the network has a 1.5-hop neighborhood graph isomorphic to the 1.5-hop neighborhood graph of at least $k-1$ other nodes. The name of this property was given by Wu et al. [147].

In our example, Ana and Don become uniquely identifiable once we look at their 2-hop neighborhoods. Emma and Fabio have isomorphic signatures regardless of the size of the neighborhood for which the adversary has background information. This leads to the most general privacy preservation definitions of *k-candidate anonymity* [57] and k-automorphism anonymity [162].

Definition 15 **K-Candidate anonymity**. *An anonymized graph satisfies K-Candidate Anonymity with respect to a structural query Q if there is a set of at least K nodes which match Q, and the likelihood of every candidate for a node in this set with respect to Q is less than or equal to $1/k$.*

K-Candidate anonymity [57], considers the structural anonymity of users given a particular structural query, i.e., a subgraph signature. Hay et al. define

three types of structural queries, vertex refinement queries, subgraph queries and hub fingerprint queries [57, 56]. Zou et al. [162] assume a much more powerful adversary who has knowledge of any subgraph signature of a target individual. They propose the notion of *k-automorphism anonymity* to fend off such an adversary.

Definition 16 **k-automorphism anonymity**. *An anonymized graph is k-automorphic if every node in the graph has the same subgraph signature (of arbitrary size) as at least $k-1$ other graph nodes, and the likelihood of every candidate for that node is less than or equal to $1/k$.*

Anonymization. The anonymization strategies for social network structure fall into four main categories:

- **Edge modification**. Since complete removal of the links to keep structural properties private would yield a disconnected graph, edge modification techniques propose edge addition and deletion to meet desired constraints. Liu and Terzi anonymize the network degree sequence to meet k-degree anonymity [86]. This is easy to achieve for low-degree nodes because the degree distribution in social networks often follows a power law. For each distinguishable higher-degree node, where distinguishable is defined as a degree for which there are less than k nodes with that degree, the anonymization algorithm increases its degree artificially so that it becomes indistinguishable from at least $k-1$ other nodes. The objective function of the algorithm is to minimize the number of edge additions and deletions. Zou et

al. [162] propose an edge modification algorithm that achieves k-automorphism anonymity.

- **Randomization**. Anonymization by randomization can be seen as a special case of anonymization by edge modification. It refers to a mechanism which alters the graph structure by removing and adding edges at random, and preserves the total number of edges. Hay et al. [57] show that if this is performed uniformly at random, then it fails to keep important graph metrics of real-world networks. Ying and Wu [148] propose *spectrum-preserving randomization* to address this loss of utility. The graph's spectral properties are the set of eigenvalues of the graph's adjacency matrix to which important graph properties are related. Preserving this spectrum guides the choice of random edges to be added and deleted. However, the impact of this approach on privacy is unclear.

  Two recent studies have presented algorithms for reconstructing randomized networks [141, 146]. Wu et al. [146] take a low rank approximation approach and apply it to a randomized network structure, such that accurate topological features can be recovered. They show that in practice reconstruction may not pose a larger threat to privacy than randomization because the original network is more similar to the randomized network than to the reconstructed network. Vuokko and Terzi [141] consider reconstruction mechanisms for networks where randomization has been applied both to the structure and attributes of the nodes. They identify cases in

which reconstruction can be achieved in polynomial time. The effect of both reconstruction strategies on privacy has not been assessed.

- **Network generalization**. One way to alleviate an attack based on structural background information is by publishing the aggregate information about the structural properties of the nodes [56]. In particular, one can partition the nodes and keep the density information inside and between parts of the partition. Nodes in each partition have the same structural properties, so that an adversary coming with a background knowledge is not able to distinguish between these nodes. In practice, sampling from the anonymized network model creates networks which keep many of the structural properties of the original network, such as degree distribution, path length distribution and transitivity. Network generalization strategies for other network types are discussed in Section 6.3.1 [22, 152] and Section 6.3.3 [14].

- **Differentially private mechanisms**. Differentially private mechanisms refer to algorithms which guarantee that individuals are protected under the definition of differential privacy (see Section 6.2.3). Hay et al. [54] propose an efficient algorithm which allows the public release of one of the most commonly studied network properties, degree distribution, while guaranteeing differential privacy. The algorithm involves a post-processing step on the differentially private output, which ensures a more accurate result. The empirical analysis on real-world and synthetic networks shows that the resulting degree-distribution estimate exhibits low bias and variance, and

can be used for accurate analysis of power-law distributions, commonly occurring in networks.

Anonymizing user attributes and network structure

So far, we have discussed anonymization techniques which perturb the structure of the network but do not consider attributes of the nodes, such as gender, age, nationality, etc. However, providing the (perturbed) structure of social networks is often not sufficient for the purposes of the researchers who study them. In another line of privacy research, the assumption is that anonymized data will have utility only if it contains both structural properties and node attributes.

Anonymization. Zhou and Pei [159] assume that each node has one attribute which they call a label. They show that achieving k-neighborhood anonymity is NP-hard and propose a greedy *edge modification* and *label generalization* algorithm. The algorithm extracts the 1.5-neighborhood signatures for all nodes in the graph and represents them concisely using *DFS trees*. Then it clusters the signatures and anonymizes the ones in each cluster to achieve k-neighborhood anonymity. The objective function of the algorithm is similar to the one of Liu and Terzi [86], the minimization of the number of edge additions.

Zheleva and Getoor [152] study the problem of social link disclosure in graphs with multiplex relations. The assumption is that an adversary has an accurate statistical model for predicting sensitive relationships if given the attributes of nodes and edges in the original data, therefore attributes have to be

perturbed in the released data. They propose anonymization by generalization of the data as a two-step process. In the first step, nodes are treated as a table of records, and their attributes are anonymized to guarantee the privacy of users, for example, to meet one of the privacy definitions described earlier. Using k-anonymity, this creates a partition of the network nodes into equivalence classes. In the second step, the structure of the network is partially preserved by keeping aggregate structural information inside and between the equivalence classes.

Campan and Truta [22] also take a network generalization approach to anonymizing a social network. Their greedy algorithm optimizes a utility function using the attribute and structural information simultaneously rather than as a two-step process. They introduce a structural information loss measure, and adopt an existing measure of attribute information loss. The anonymization algorithm can be adjusted to preserve more of the structural information of the network or the nodes' attribute values.

6.3.2 Privacy mechanisms for affiliation networks

Next, we concentrate on the affiliation network and discuss privacy-preserving techniques developed specifically for this type of network. The affiliation network is represented as a bipartite graph with two types of nodes V and H, and the affiliation links between them E_h. Figure 6.4 shows an illustration of this graph where on the left-hand side there are users, and on the right-hand side there are movies that the users rated. The affiliation links have a weight corre-

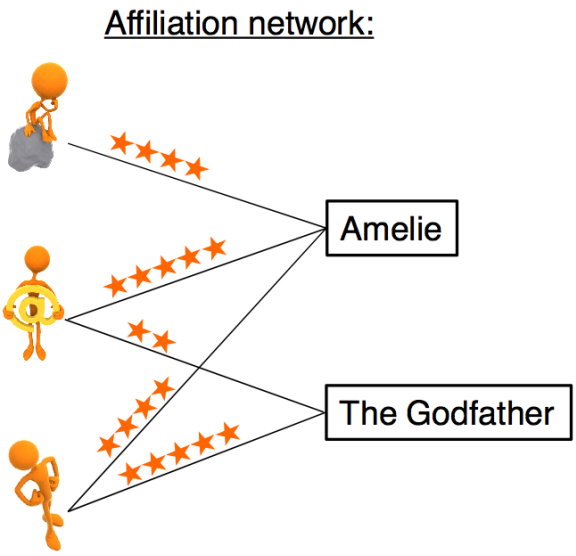


Figure 6.4: An affiliation network as a bipartite graph between three users and two movies. The affiliation links show the ratings that users gave to the movies on a scale from 1 to 5.

sponding to the movie ratings for each user, on a scale from 1 to 5.

Netflix, an online movie rental company, set up a competition aimed at improving their movie recommendation systems. They released a dataset with around 100 million dated ratings from 480 thousand randomly-chosen Netflix customers. To protect customer privacy, each customer id has been replaced with a randomly-assigned id. However, this naive anonymization was found to be vulnerable under a linking attack [110]. Using the dates of user ratings and matching the records released by $Netflix$ and user profiles in $IMDB$, an online movie database, Narayanan and Shmatikov [110] were able to achieve identity and sensitive attribute disclosure for some of the users in the Netflix dataset.

A related problem is the problem of releasing a search query graph in which user information is contained in the affiliation links between search engine queries and clicked website URLs [71]. In particular, there is a bipartite graph of (query,URL)

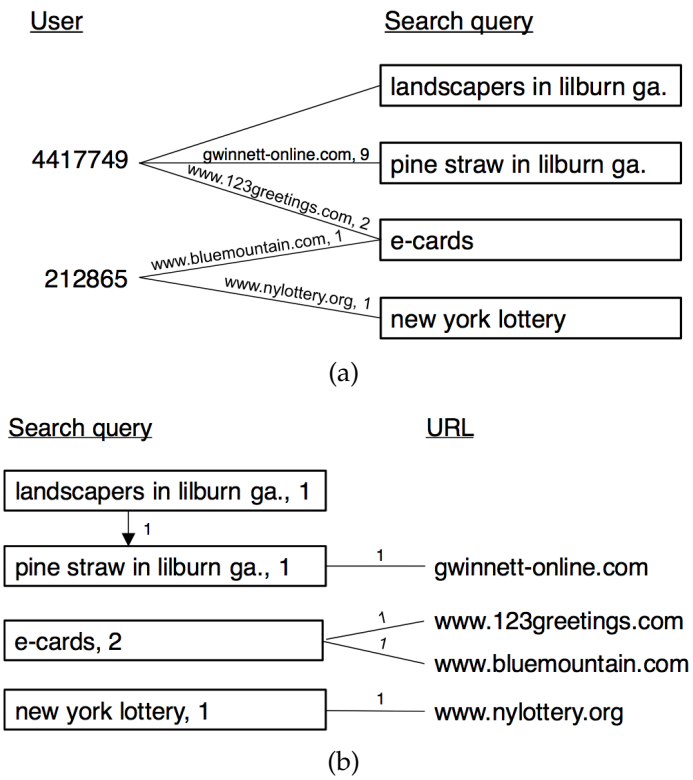


Figure 6.5: a) *User-query graph* representing the users, their queries, the websites they clicked on and the ranking of each website, and b) its reformulation into a *search query graph*.

pairs.Here, the links have a weight corresponding to the number of users who posed a particular query and clicked on the particular URL. In addition, there are links between queries with a weight equal to the number of users who posed the first query and then reformulated it into the second query. Each query also has counts of the number of times the query was posed to the search engine. The utility in such data is in using it for learning better search ranking algorithms. Figure 6.5(a) shows an example a user-query graph. Figure 6.5(b) shows its reformulation into a search query graph where individual users are not represented explicitly but only as aggregate numbers.

Anonymization

Two types of privacy mechanisms for affiliation networks have been studied in the research literature:

- **Network generalization**. Cormode et al. [28] propose a privacy definition for affiliation networks, *(k,l)-grouping*, tailored to prevent sensitive affiliation link disclosure. The authors make the assumption that affiliation links can be predicted based on node attributes and the structure of the network. They show why existing table anonymization techniques fail to preserve the structural properties of the network, and propose a greedy anonymization algorithm which keeps the structure intact but generalizes node attributes. The algorithm requires that each node is indistinguishable from at least $k-1$ other nodes in terms of node properties, and each affiliation group

is indistinguishable from at least $l-1$ other affiliation groups, the basis of (k,l)-grouping. The utility is in being able to answer accurately aggregate queries about users and affiliation groups.

- **Differentially private mechanisms**. A private mechanism for a recommender system has been developed specifically for the movie recommendation setting [104]. The system works by providing differentially private mechanisms for computing counts, rating averages per movie and per user, and the movie-movie covariances in the data. These statistics are sufficient for computing distances based on k-nearest neighbor for predicting the ratings associated with new affiliation links. Using the statistics released by the mechanism, the algorithm performs with an accuracy comparable to the one in the original data.

  Korolova et al. [71] propose an (ϵ,δ)-differentially private algorithm which allows the publication of a search query graph for this purpose. Here the search logs are the database, and pairs of databases D_1 and D_2 are considered to differ in one element when one database excludes the search logs of exactly one user. The algorithm keeps only a limited number of queries and clicks for each user and allows for two types of functions on the graph which are sufficient for evaluating ranking algorithms. The first function gives a search query and its noisy count if it exceeds a pre-specified threshold. The second function publishes the noisy weight of the (query,URL) link for the top URLs for each query which was safe to publish according to the

first function.

6.3.3 Privacy mechanisms for social and affiliation networks

There has not been much research on the privacy implications of the interplay between social and affiliation networks. It is obvious that they inherit all the privacy issues discussed so far for either type of network. What is not so obvious is that the complex dependencies these networks create can allow an adversary to learn private information in intricate ways. In particular, one can use the social environment of users to learn private information about them. One type of attack, which we call an *attribute inference attack*, assumes that an attribute is sensitive only for a subset of the users in the network and that the other users in the network are willing to publish it publicly [153]. The analogy in real-world social networks is the existence of private and public profiles. The attack works by creating a statistical model for predicting the sensitive attribute using the publicly available information and applying that model to predict the users with private profiles. In its basic form, the attack assumes that besides the network structure, the only user attributes that are available are the sensitive attribute value for the public profiles. Naturally, using other profile attributes can create even more powerful statistical models, as Lindamood et al. show [82]. An adversary succeeds when he can recover the sensitive attribute values for a subset of the nodes with high probability.

By taking into account all social and affiliation links, often declared pub-

licly in online social networks, the model can use link-based classification techniques. Link-based classification breaks the assumption that data comprises of independent and identically distributed (iid) nodes and it can take advantage of autocorrelation, the property that attributes of linked objects often correlated with each other. For example, political affiliations of friends tend to be similar, students tend to be friends with other students, etc. A comprehensive survey of models for link-based classification can be found in the work by Sen et al. [131]. The results of Zheleva and Getoor [153] suggest that link-based classification can predict sensitive attributes with high accuracy using information about online social groups, and that social groups have a higher potential for leaking personal information than friendship links.

Anonymization

Bhagat et al. [14] consider attacks for sensitive social link disclosure in social and affiliation networks, to which they refer as *rich interaction graphs*. Two nodes participating in the same group is also considered as a sensitive social link between the two users. Bhagat et al. represent the social and affiliation networks as a bipartite graph, in which one type of nodes are the users and the other type of nodes are affiliation groups. Social links are represented as affiliation groups of size two.

They propose two types of network generalization techniques to prevent social link disclosure. The first technique, a *uniform list approach*, keeps the structure intact, in a manner similar to (k, l)-groupings [28]. It divides nodes into classes

of size m ensuring that each node's interactions fall on nodes of different classes. Each class is split into label lists of size k, thus ensuring that the probability of a link between two users (through a social link or a common affiliation group) is at most $1/k$. If the adversary has a background knowledge of the identities of r of the nodes and k is equal to m, then this probability becomes $1/(k-r)$. The second technique, a *partitioning approach*, also divides the nodes into classes of size m so that each node's interactions fall on nodes of different classes. However, it does not keep the original network structure, and publishes only the number of edges between partitions. The probability of a link between two users is guaranteed to be at most $1/m$ with or without background knowledge. The utility of the anonymized graph is in allowing accurate structural and attribute analysis of the graph.

6.4 Related literature

Research on privacy in online social networks is a very young field which discovers and addresses some of the challenges of preserving the privacy of individuals in an interconnected world [7, 14, 22, 56, 57, 72, 71, 82, 86, 111, 148, 153, 152, 159, 162]. However, privacy research has a longer history in the data mining, database and security communities. For example, privacy-preserving data mining aims at creating data mining algorithms and systems which take into consideration the sensitive content of the data [140, 4]. Chen et al. [25] provide a comprehensive, recent survey of the field of privacy-preserving data publishing. The

database and security communities have studied interactive and non-interactive mechanisms for sharing potentially sensitive data [34]. Most of this research assumes that there are one or more data owners who are interested to provide data access to third parties while meeting privacy constraints. In contrast, access to data in online social networks is often freely available, and users can specify their personal privacy preferences. Addressing the new privacy challenges in this area is an active area of research [66]. The unexpected threats of freely publishing personal data online is exemplified by a number of researchers [1, 82, 111, 153]. boyd points out many privacy concerns and ethical issues, related to the analysis of large online social network data [30]. Measuring the privacy of social network users and enabling them to personalize their online privacy preferences has also been the focus of recent work [87, 41]. Privacy in dynamic social networks has also received recent attention [15, 162].

6.5 Conclusion

Here, we presented the possible privacy breaches in online social networks, together with existing privacy definitions and mechanisms for preserving user privacy. While initial steps have been taken in understanding and overcoming some of the challenges of preserving privacy online, many open problems remain. In particular, some exciting new directions include studying the effect of different types of privacy disclosures on each other, privacy-preserving techniques that prevent sensitive attribute disclosure in networks, a compari-

son between existing anonymization techniques in terms of utility, and privacy-preserving techniques that meet the individual privacy expectations of online social network users rather than privacy definitions imposed by a data publisher or an online service provider.

Chapter 7

Attribute Disclosure

In order to address users' privacy concerns, a number of social media and social network websites, such as Facebook, Orkut and Flickr, allow their participants to set the privacy level of their online profiles and to disclose either some or none of the attributes in their profiles. While some users make use of these features, others are more open to sharing personal information. Some people feel comfortable displaying personal attributes such as age, political affiliation or location, while others do not. In addition, most social-media users utilize the social networking services provided by forming friendship links and affiliating with groups of interest. While a person's profile may remain private, the friendship links and group affiliations are often visible to the public. Unfortunately, these friendships and affiliations leak information; in fact, as we will show, they can leak a surpisingly large amount of information.

The problem we consider is *sensitive attribute inference* in social networks: inferring the private information of users given a social network in which some

profiles and all links and group memberships are public (this is a commonly occurring scenario in existing social media sites). We define the problem formally in Section 7.3. We believe our work is the first one to look at this problem, and to map it to a relational classification problem in network data with groups.

Here, we propose eight privacy attacks for sensitive attribute inference. The attacks use different classifiers and features, and show different ways in which an adversary can utilize links and groups in predicting private information. We evaluate our proposed models using sample datasets from four well-known social media websites: Flickr, Facebook, Dogster and BibSonomy. All of these websites allow their users to form friendships and participate in groups, and our results show that attacks using the group information achieve significantly better accuracy than the models that ignore it. This suggests that group memberships have a strong potential for leaking information, and if they are public, users' privacy in social networks is illusionary at best.

Our contributions include the following:

- We identify a number of novel privacy attacks in social networks with a mixture of public and private profiles.
- We propose that in addition to friendship links, group affiliations can be carriers of significant information.
- We show how to reduce the large number of potential groups in order to improve the attribute accuracy.
- We evaluate our attacks on challenging classification tasks in four social media

datasets.

- We illustrate the privacy implications of publicly affiliating with groups in social networks and discuss how our study affects anonymization of social networks.
- We show how surprisingly easy it is to infer private information from group membership data.

First, we motivate the problem in the next section. Section 7.3 presents the privacy attacks, and Section 8.5 provides experimental results using these attacks. Section 7.5 discusses the broader implications of our results.

7.1 Motivation

Disclosing private information means violating the rights of people to control who can access their private information. In order to prevent private information leakage, it is important to be aware of the ways in which an adversary can attack a social network to learn users' private attributes. Studies on the challenges of preserving the privacy of individuals in social networks have emerged only in the last few years, and they have concentrated on inferring the identity of nodes based on structural properties such as node degree. In contrast, we are interested in inferring sensitive attribute of nodes using approaches developed for relational learning, another active area of research in the last few years.

The novelty of our work is that we study the implications of mixing private and public profiles in a social network. For example, in Facebook many

users choose to set their profiles to private, so that no one but their friends can see their profile details. Yet, fewer people hide their friendship links and even if they do, their friendship links can be found through the backlinks from their public-profile friends. Similarly for group participation information – even if a user makes her profile private, her participation in a public group is shown on the group's membership list. Currently, neither Facebook nor Flickr allow users to hide their group memberships from public groups. Both commercial and governmental entities may employ privacy attacks for targeted marketing, health care screening or political monitoring – just to mention a few. Therefore, social media website providers need to protect their users against undesired eavesdropping and inform them of the possible privacy breaches and providing them with the means to be in full control of their private data.

Our work is also complimentary to work on data anonymization, in which the goal is to perturb data in such a way that the privacy of individuals is preserved. Our goal is not to release anonymized data but to illustrate how social network data can be exploited to predict hidden information: an essential knowledge in the anonymization process.

We identify a new type of privacy breach in relational data, *group membership disclosure*: whether a person belongs to a group relevant to the classification of a sensitive attribute. We conjecture that group membership disclosure can lead to attribute disclosure. Thus, hiding group memberships is a key to preserving the privacy of individuals.

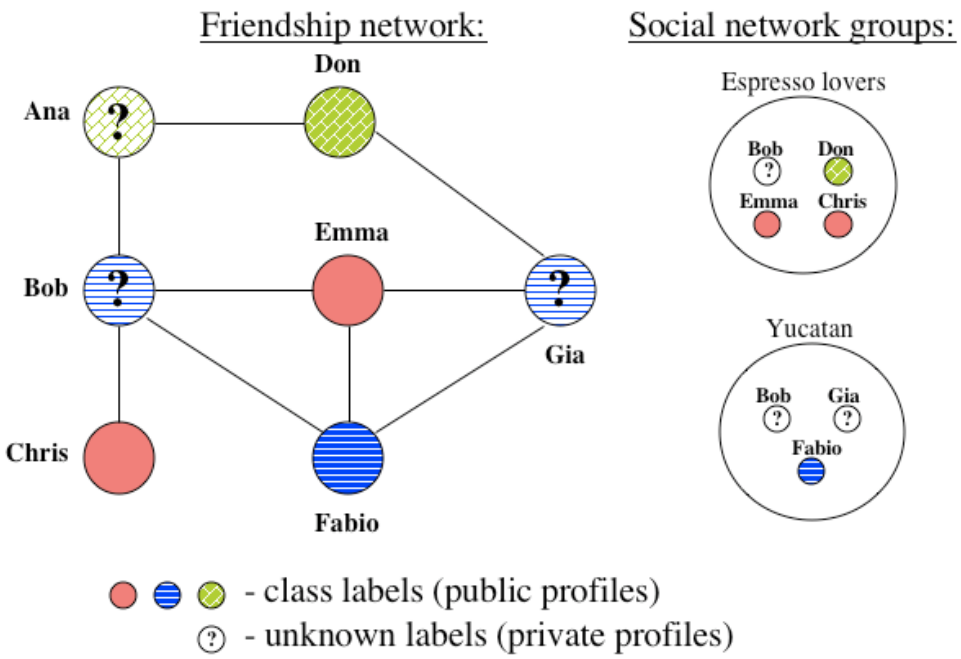


Figure 7.1: Toy instance of the data model.

7.2 Sensitive attributes

The data model is the same as the one presented in Section 1.1 except that it assumes that some of the personal attributes are missing, such as in Figure 7.1. We assume that each node v has a sensitive attribute $v.a$ which is either observed or hidden in the data. A *sensitive attribute* is a personal attribute, such as age, political affiliation or location, which some users in the social network are willing to disclose publicly. A sensitive attribute value can take on one of a set of possible values $\{a_1...a_m\}$. A *private profile* is one for which the sensitive attribute value is unknown, and a *public profile* is the opposite: a profile with an observed sensitive attribute value. We refer to the set of nodes with private profiles as the *sensitive set* of nodes V_s, and to the rest as the *observed set* V_o. The adversary's goal is to predict $V_s.a$, the sensitive attributes of the private profiles.

Here, we study the case where nodes have no other attributes beyond the sensitive attribute. Thus, to make inferences about the sensitive attribute, we need to use some form of relational classifier. While additional attribute information can be helpful and many relational classifiers can make use of it, in our setting this is not possible because all of the private-profile attributes are likely to be hidden.

In our toy example (Figure 7.1), Chris, Don, Emma and Fabio are displaying their attribute values publicly, while Ana, Bob and Gia are keeping theirs private. Emma and Chris have the same sensitive attribute value (marked solid), Bob, Gia and Fabio share the same attribute value (marked with stripes), and Ana and Don have a third value (marked with a brick pattern). While affiliating with some groups may be related to the sensitive attribute, affiliating with others is not. For example, if the sensitive attribute is a person's country of origin, the "Yucatan" group may be relevant. Thus, this group can leak information about sensitive attributes, although the manner in which it is leaked is not necessarily straightforward.

7.3 Sensitive-attribute inference models

The attributes of users who are connected in social networks are often correlated. At the same time, online communities allow very diverse people to connect to each other and form relationships that transcend gender, religion, origin and other boundaries. As this happens, it becomes harder to utilize the complex in-

teractions in online social networks for predicting user attributes.

Attribute disclosure occurs when an adversary is able to infer the sensitive attribute of a real-world entity accurately. The sensitive attribute value of an individual can be modeled as a random variable. This random variable's distribution can depend on the overall network's attribute distribution, the friendship network's attribute distribution and/or the attribute distribution of each group the user joins.

The problem of *sensitive attribute inference* is to infer the hidden sensitive values, $V_s.a$, conditioned on the observed sensitive values, links and group membership in graph G. We assume that the adversary can apply a probabilistic model M for predicting the hidden sensitive attribute values, and he can combine the given graph information in various ways as we discuss next. The prediction of each model is:

$$v_s.\hat{a}_M = \underset{a_i}{\operatorname{argmax}} P_M(v_s.a = a_i; G).$$

where $P_M(v_s.a = a_i; G)$ is the probability that the sensitive attribute value of node $v_s \in V_s$ is a_i according to model M and the observed part of graph G.

We assume that the overall distribution of the sensitive attribute is either known or it can be found using the public profiles. An attack using this distribution is a *baseline attack*. A *successful attack* is one which, given extra knowledge, e.g., friendship links or group affiliations, has a significantly higher accuracy than

the baseline attack. The extra knowledge *compromises* the privacy of users if there is an attack which uses it and is successful.

7.3.1 Attacks without links and groups

In the absence of relationship and group information, the only available information is the overall marginal distribution for the sensitive attribute in the public profiles. So, the simplest model is to use this as the basis for predicting the sensitive attributes of the private profiles. More precisely, according to this model, BASIC, the probability of a sensitive attribute value can be estimated as the fraction of observed users who have that sensitive attribute value:

$$P_{BASIC}(v_s.a = a_i; G) = P(v_s.a = a_i | V_o.a) = \frac{|V_o.a_i|}{|V_o|},$$

where $|V_o.a_i|$ is the number of public profiles with sensitive attribute value a_i and $|V_o|$ is the total number of public profiles. The adversary using model BASIC picks the most probable attribute value which in this case is the overall mode of the multinomial attribute distribution. In our toy example, the most common observed sensitive attribute is the value that Chris and Emma share. Therefore, the adversary would predict that Ana, Bob and Gia have the same attribute value as well. An obvious problem with this approach is that if there is a sensitive attribute value that is predominant in the observed data, it will be predicted for all users with private profiles. Nevertheless, this attack is always at least as good as a random guess, and we use it as a simple baseline. Next, we look at using

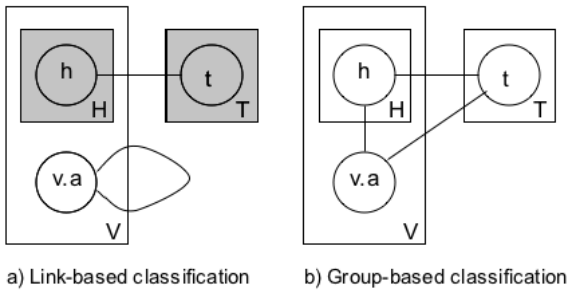


Figure 7.2: Graphical representation of the models. Grayed areas correspond to variables that are ignored in the model.

friendship information for inferring the attribute value.

7.3.2 Privacy attacks using links

Link-based privacy attacks take advantage of $autocorrelation$, the property that the attribute values of linked objects are correlated. An example of autocorrelation is that people who are friends often share common characteristics (as in the proverb "Tell me who your friends are, and I'll tell you who you are"). Figure 7.2(a) shows a graphical representation of the link-based classification model. There is a random variable associated with each sensitive attribute $v.a$, and the sensitive attributes of linked nodes are correlated. The greying of the other two types of random variables means that the group information is not used in this model.

Friend-aggregate model (AGG)

The nodes and their links produce a graph structure in which one can identify circles of close friends. For example, the circle of Bob's friends is the set of users that he has links to: $Bob.F = \{Ana, Chris, Emma, Fabio\}$. The friend-

aggregate model AGG looks at the sensitive attribute distribution amongst the friends of the person under question. According to this model, the probability of the sensitive attribute value can be estimated by:

$$P_{AGG}(v_s.a = a_i; G) = P(v_s.a = a_i | V_o.a, E_v) = \frac{|V'_o.a_i|}{|V'_o|}$$

where $V'_o = \{v_o \in V_o | \exists e_v(v_s, v_o) \in E_v\}$ and $V'_o.a_i = \{v_o \in V'_o | v_o.a = a_i\}$.

Again, the adversary using this model picks the most probable attribute value (i.e., the mode of the friends' attribute distribution). In our toy example (Figure 7.1), Bob will be assigned the same value as Emma and Chris, Ana the same label as Don, and Gia will be undecided between Don's, Emma's and Fabio's label. One problem with this method is the one when person's friends are very diverse, as in Gia's case, it will be difficult to make a prediction.

Collective classification model (CC)

Collective classification also takes advantage of autocorrelation between linked objects. Unlike more traditional methods, in which each instance is classified independently of the rest, collective classification aims at learning and inferring class labels of linked objects together. In our setting, it makes use of not only the public profiles but also the inferred values for connected private profiles. Collective classification has been an active area of research in the last decade (see Sen et al. [131] for a survey). Some of the approximate inference algorithms proposed include iterative classification (ICA), Gibbs sampling, loopy belief propagation

and mean-field relaxation labeling.

For our experiments, we have chosen to use ICA because it is simple, fast and has been shown to perform well on a number of problems [131]. In our setting, ICA first assigns a label to each private profile based on the labels of the friends with public profiles, then it iteratively re-assigns labels considering the labels of both public and private-profile friends. The assignment is based on a local classifier which takes the friends' class labels as features. For example, a simple classifier could assign a label based on the majority of the friends labels. A more sophisticated classifier can be trained using the counts of friends' labels.

Flat-link model (LINK)

Another approach to dealing with links is to "flatten" the data by considering the adjacency matrix of the graph. In this model, each row in the matrix is a user instance. In other words, each user has a list of binary features of the size of the network, and each feature has a value of 1 if the user is friends with the person who corresponds to this feature, and 0 otherwise. The user instance also has a class label which is known if the user's profile is public, and unknown if it is private. The instances with public profiles are the training data which can be fed to any traditional classifier, such as Naïve Bayes, logistic regression or SVM. The learned model can then be applied to predict the private profile labels.

Blockmodeling attack (BLOCK)

The next category of link-based methods we explored are approaches based on blockmodeling [142, 6]. The basic idea behind *stochastic blockmodeling* is that users form natural clusters or blocks, and their interactions can be explained by the blocks they belong to. In particular, the link probability between two users is the same as the link probability between their corresponding blocks. If sensitive attribute values separate users into blocks, then based on the observed interactions of a private-profile user with public-profile users, one can predict the most likely block the user belongs to and thus discover the attribute value. Let block B_i denote the set of public profiles that have attribute value a_i, and $\lambda_{i,j}$ the probability that a link exists between users in block B_i and users in block B_j. Thus, λ_i is the vector of all link probabilities between block B_i and each block $B_1, ..., B_m$. Similarly, let the probability of a link between a single user v and a block B_j be $\lambda(v)_j$ with $\lambda(v)$ being the vector of link probabilities between v and each block. To find the probability that a private-profile user belongs to a particular block, the model looks at the maximum similarity between the interaction patterns (link probability to each block) of the node in question and the overall interactions between blocks. After finding the most likely block, the sensitive attribute value is predicted. The probability of an attribute value using the blockmodeling attack, BLOCK, is estimated by:

$$P_{BLOCK}(v_s.a_i; G) = P(v_s.a_i|V_o.A, E_v, \lambda) = \frac{1}{Z} sim(\lambda_i, \lambda(v))$$

where $sim()$ can be any vector similarity function and Z is a normalization factor. We compute maximum similarity using the minimum L2 norm. This model is similar to the class-distribution relational-neighbour classifier described in [95] when the weight of each directed edge is inversely proportional to the size of the class of the receiving node.

7.3.3 Privacy attacks using groups

In addition to link or friendship information, social networks offer a very rich structure through the group memberships of users. All individuals in a group are bound together by some observed or hidden interest(s) that they share, and individuals often belong to more than one group. Thus, groups offer a broad perspective on a person, and it may be possible to use them for sensitive attribute inference. If a user belongs to only one group (as it is Gia's case in the toy example), then it is straightforward to infer a label using an aggregate, e.g., the mode, of her groupmates' labels, similar to the friend-aggregate model. This problem becomes more complex when there are multiple groups that a user belongs to, and their distributions suggest different values for the sensitive attribute. We propose two models for utilizing the groups in predicting the sensitive attribute – a model which assumes that all groupmates are friends and one which takes

groups as classifier features.

Groupmate-link model (CLIQUE)

One can think of groupmates as friends to whom users are implicitly linked. In this model, we assume that each group is a clique of friends, thus creating a friendship link between users who belong to at least one group together. This data representation allows us to apply any of the link-based models that we have already described. The advantage of this model is that it simplifies the problem to a link-based classification problem, which has been studied more thoroughly. One of the disadvantages is that it doesn't account for the strength of the relationship between two people, e.g. number of common groups.

Group-based classification model (GROUP)

Another approach to dealing with groups is to consider each group as a feature in a classifier. While some groups may be useful in inferring the sensitive attribute, a problem in many of the datasets that we encountered was that users were members of a very large number of groups, so identifying which groups are likely to be predictive is a key. Ideally, the model would discard group memberships irrelevant to the classification task. For example, the group "Yucatan" may be relevant for finding where a person is from, but "Espresso lovers" may not be.

To select the relevant groups, one can apply standard feature selection criteria [84]. If there are N groups, the number of candidate group subsets is 2^N, and finding an optimal feature subset is intractable. Similar to pruning words in doc-

ument classification, one can prune groups based on their properties and evaluate their predictive accuracy. Example group properties include density, size and homogeneity. Smaller groups may be more predictive than large groups, and groups with high homogeneity may be more predictive of the class value. For example, if the classification task is to predict the country that people are from, a cultural group in which 90% of the people are from the same country is more likely to be predictive of the country class label. One way to measure group homogeneity is by computing the entropy of the group: $Entropy(h) = -\sum_{i=1}^{m} p(a_i) \log_2 p(a_i)$ where m is the number of possible node class values and $p(a_i)$ is the fraction of observed members that have class value a_i: $p(a_i) = \frac{|h.V.a_i|}{|h.V|}$.

For example, the group "Yucatan" has an entropy of 0 because only one attribute value is represented there, therefore its homogeneity is very high. We also consider the confidence in the computed group entropy. One way to measure this is through the percent of public profiles in the group.

The group-based classification approach contains three main steps as Algorithm 6 shows. In the first step, the algorithm performs feature selection: it selects the groups that are relevant to the node classification task. This can either be done automatically or by a domain expert. Ideally, when the number of groups is high, the feature selection should be automated. For example, the function $isRelevant(h)$ can return $true$ if the entropy of group h is low. In the second step, the algorithm learns a global function f, e.g., trains a classifier, that takes the relevant groups of a node as features and returns the sensitive attribute value. This step uses only the nodes from the observed set whose sensitive at-

tributes are known. Each node v is represented as a binary vector where each dimension corresponds to a unique group: $\{groupId : isMember\}$, $v.a$. Only memberships to relevant groups are considered and $v.a$ is the class coming from a multinomial distribution which denotes the sensitive-attribute value. In the third step, the classifier returns the predicted sensitive attribute for each private profile. Figure 7.2(b) shows a graphical representation of the group-based classification model. It shows that there is a dependence between the nodes' sensitive attributes $V.a$, the group memberships E_h and the group attributes $H.A$.

Algorithm 6 Group-based classification model

```
1: Set of relevant groups H_relevant = ∅
2: for each group h ∈ H do
3:    if isRelevant(h) then
4:       H_relevant = H_relevant ∪ {h}
5:    end if
6: end for
7: trainClassifier(f, V_o, H_relevant)
8: for each sensitive node v ∈ V_s do
9:    v.â = f(v.H_relevant)
10: end for
```

7.3.4 Privacy attacks using links and groups

It is possible to construct a method which uses both links and groups to predict the sensitive attributes of users. We use a simple method which combines the flat-link and the group-based classification models into one: LINK-GROUP. This model uses all links and groups as features, thus utilizing the full power of available information. Like LINK and GROUP, LINK-GROUP can use any traditional classifier.

Table 7.1: Properties of the four datasets.

PROPERTY	FLICKR	FACEBOOK	DOGSTER	BIBSONOMY
No. of users	9,179	1,598/965	2,632	31,715
No. of links	941,677	86,007/33,597	4,482	N/A
No. of groups	47,754	2,932/2,497	1,042	132,554
Avg. in-sample degree	142	108/70	1	N/A
Avg. no. groups per user	162	24/25	1	98
Avg. group size	31	10/9	3	9
Largest group size	4,527	290/221	118	7,182
% same-label node links	23.5%	49.9%/40.3%	-	N/A
No. of possible labels	55	2/6	7	2
Sensitive attribute	*location*	*gender/polviews*	*breed category*	*spammer*

7.4 Experiments

We evaluated the effectiveness of each of the proposed models for inferring sensitive attributes in social networks.

7.4.1 Data description

For our evaluation, we studied four diverse online communities: the photo-sharing website Flickr, the social network Facebook, Dogster, an online social network for dogs, and the social bookmarking system BibSonomy[1]. Table 7.1 shows properties of the datasets, including the sensitive attributes.

Flickr is a photo-sharing community in which users can display photographs, create directed friendship links and participate in groups of common interest. Users have the choice of providing personal information on their profiles, such as gender, marital status and location. We collected a snowball sample of $14,451$ users from it. To resolve their locations (which users enter manually, as opposed

[1]At `http://www.flickr.com`, `http://www.facebook.com`, `http://www.dogster.com`, `http://www.bibsonomy.org/`

to choosing them from a list), we used a two-step process. First, we used Google Maps API[2] to find the latitude and longitude of each location. Then, we mapped the latitude and longitude back to a country location using the reverse-geocoding capabilities of GeoNames[3]. We discarded the profiles with no resolved country location (34%), and ones that belonged to a country with less than 10 representatives. The resulting sample contained $9,179$ users from 55 countries. There were $47,754$ groups with at least 2 members in the sample.

Facebook is a social network which allows users to communicate with each other, to form undirected friendship links and participate in groups and events. We used a part of the Facebook network, available for research purposes [78]. It contains all $1,598$ profiles of first-year students in a small college. The dataset does not contain group information but it contains the favorite books, music and movies of the users, and we considered them to be the groups that unify people. $1,225$ of the users share at least one group with another person, and $1,576$ users have friendship links. All profiles have gender and 965 have self-declared political views. We use six labels of political views - *very liberal or liberal* (545 profiles), *moderate* (210), *conservative or very conservative* (114), *libertarian* (29), *apathetic* (18), and *other* (49).

Dogster is a website where dog owners can create profiles describing their dogs, as well as participate in group memberships. Members maintain links to friends and family. From a random sample of $10,000$ Dogster profiles, we re-

[2]At `http://code.google.com/apis/`.
[3]At `http://www.geonames.org/export/`.

Table 7.2: Attack accuracy assuming 50% private profiles in Flickr, Facebook (FB), Dogster and BibSonomy (Bib). The successful attacks are shown in bold.

ATTACK MODEL	FLICKR	FB-GENDER	FB-POLVIEWS	DOGSTER	BIB
BASIC	27.7%	50.0%	56.5%	28.6%	92.2%
Random guess	1.8%	50.0%	16.7%	14.3%	50%
BLOCK	8.8%	49.1%	6.1%	-	-
AGG	28.4%	50.2%	57.6%	-	-
CC	28.6%	50.4%	56.3%	-	-
LINK	**56.5%**	**68.6%**	58.1%	-	-
CLIQUE-LINK	**46.3%**	51.8%	57.1%	**60.2%**	-
GROUP	**63.5%**	**73.4%**	45.2%	**65.5%**	**94.0%**
GROUP*	**83.6%**	**77.2%**	46.6%	**82.0%**	**96.0%**
LINK-GROUP	**64.8**%	**72.5**%	57.8%	-	-

moved the ones that do not participate in any groups. The remaining $2,632$ dogs participate in $1,042$ groups with at least two members each. Dogs have breeds, and each breed belongs to a broader type set. In our dataset, there were mostly *toy* dogs (749). The other breed categories were *working* (268), *herding* (202), *terrier* (232), *sporting* (308), *non-sporting* (225), *hound* (152) and *mixed dogs* (506).

The fourth dataset contains publicly available data from the social bookmarking website BibSonomy[4], in which users can tag bookmarks and publications. Although BibSonomy allows users to form friendships and join groups of interest, the dataset did not contain this information. Therefore, we consider each tag placed by a person to be a group to which a user belongs. There are no links between users other than the group affiliations. There are $31,715$ users with at least one tag, 98.7% of which posted the same tag with at least one other user. The sensitive attribute is the binary attribute of whether someone is a spammer or not.

[4]At `http://www.kde.cs.uni-kassel.de/ws/rsdc08/`.

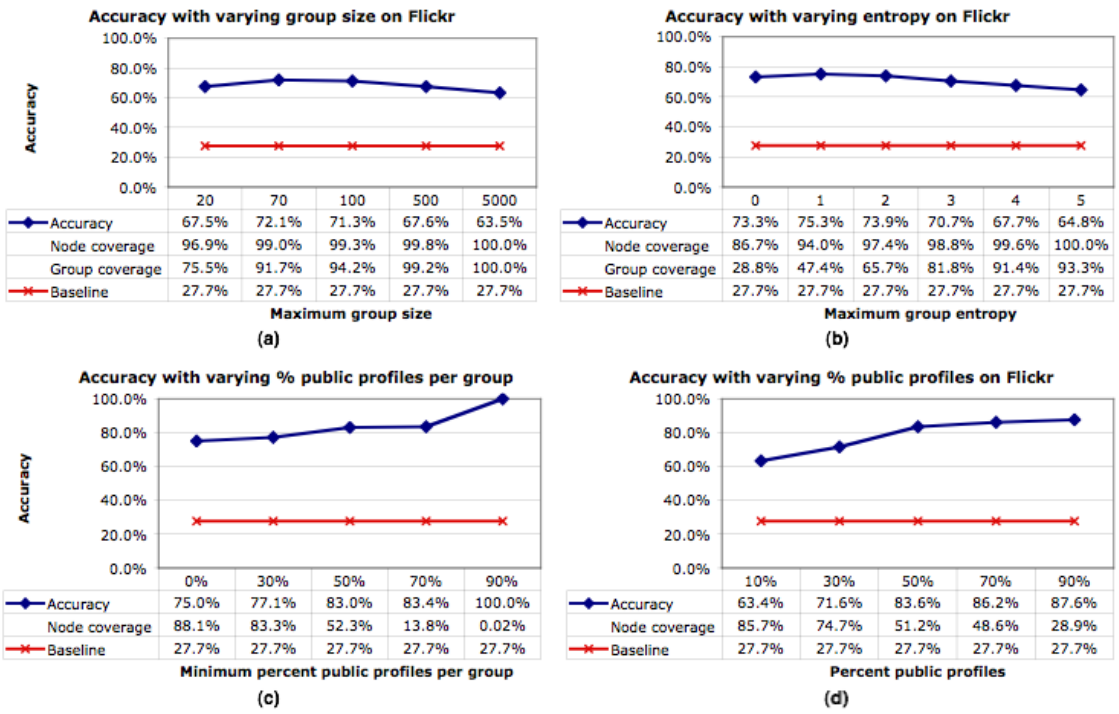


Figure 7.3: GROUP prediction accuracy on Flickr with 50% private profiles and relevant groups chosen based on (a) varying size, (b) varying entropy, and (c) a varying minimum requirement for the number of public profiles per group (maximum entropy cutoff at 0.5). Accuracy for various percent of public profiles in the network (d): the less public profiles, the worse the accuracy and therefore, the better the privacy of users.

7.4.2 Experimental setup

We ran experiments for each of the presented attack models: 1) the baseline model, an attack in the absence of link and group information (BASIC), 2) the friend-aggregate attack (AGG), 3) the collective classification attack (CC), 4) the flat-link attack (LINK) and 5) the blockmodeling attack (BLOCK), 6) the groupmate-link attack (CLIQUE), 7) the group-based classification attack with all groups (GROUP), 8) the group-based classification attack in which relevant groups are selected in a way to have 50% node coverage (GROUP*), and 9) the attack which uses both links and groups (LINK-GROUP). For the BLOCK model,

we present leave-one-out experiments assuming that complete information is given in the network in order to predict the sensitive attribute of a user. For the AGG, CC, LINK, CLIQUE, GROUP and LINK-GROUP models, we split the data into test and training by randomly assigning each profile to be private with a probability $n\%$. For LINK, GROUP and LINK-GROUP, we used an implementation of SVM for multi-value classification [139].

Groups were marked as relevant to the classification task either based on maximum size cutoff, maximum entropy cutoff and/or minimum percent of public profiles in the group. For each experiment, we measure accuracy, node coverage and group coverage. Accuracy is the correct classification rate, node coverage is the portion of private profiles for which we can predict the sensitive attribute, and group coverage is the portion of groups used for classification. The reported results are the averages over 5 trials for each set of parameters. We consider an attack to be successful if its average accuracy minus its standard deviation was larger than the baseline accuracy plus its standard deviation.

7.4.3 Sensitive-attribute inference results

Table 7.2 provides a summary of the results, assuming 50% private profiles. We see a wide variation in the performance of the different methods. GROUP* considers 50% node coverage, i.e. it shows the accuracy for half of the private-profile users who participate in a group with at least one other user. We also present experiments for varying $\%$ of private profiles (Figure 7.3(d) and Fig-

ure 7.5).

Flickr

Link-based attacks. Not surprisingly, in the absence of link and group information, our baseline achieved a relatively low accuracy (27.7%). However, surprisingly, the link-based methods AGG and CC also performed quite badly. AGG's accuracy was 28.4%, predicting that most users were from the United States. The iterative collective classification attack, CC, performed slightly, but not significantly, better (28.6%). Clearly, Flickr users do not form friendships based on their country of origin and country attribute in Flickr is not autocorrelated (only 23% of the links are between users from the same country). Another possible explanation is that the class had a very skewed distribution which persisted in friendship circles. The blockmodeling attack, BLOCK, performed worse, with only 8.8% accuracy, showing that users from a particular country did not form a natural block to explain their linking patterns. The only successful link-based attack was the "flattened" link model, LINK. With simple binary features, it achieved an accuracy of 56.5%. We performed experiments based on both in-links and outlinks, as well as ignoring the direction of the links. The results were slightly better using undirected links, and these are the results we report.

From a privacy perspective, the results from the link-based models are actually positive, showing that in this dataset, exposing the friendship links is not a serious threat to privacy for the studied attribute. The only model which performed well, LINK, shows that if an adversary tries to predict private attributes

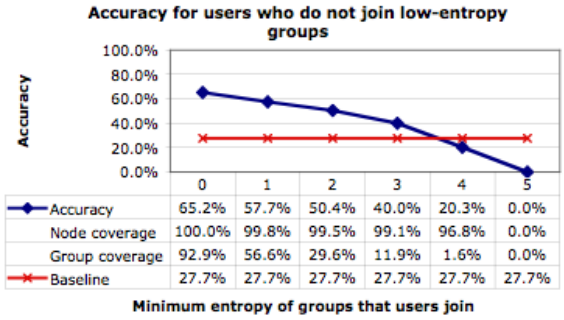


Figure 7.4: Assuming 50% public profiles, the GROUP accuracy drops significantly if Flickr users with private profiles do not join low-entropy groups.

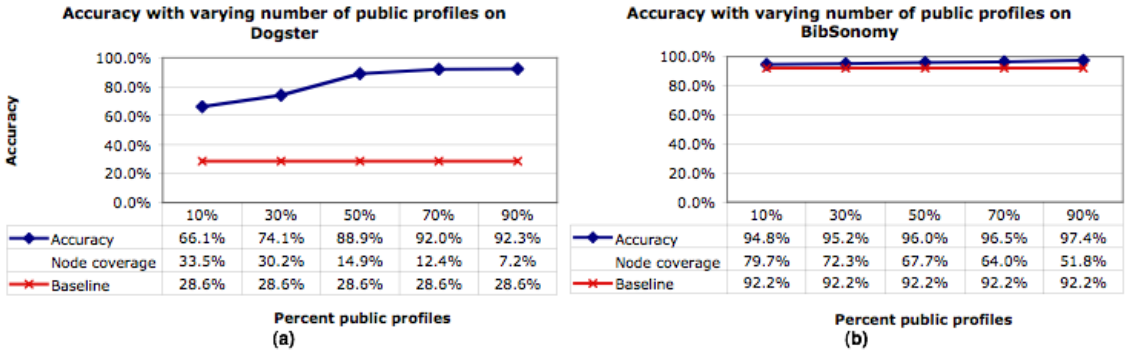


Figure 7.5: GROUP prediction accuracy on (a) Dogster and (b) BibSonomy.

of users using it, then he has almost a 50-50 chance of being wrong.

Group-based attacks. Next, we evaluate the attacks which used groups. For the CLIQUE model, we converted the groupmate relationships into friendship relationships. This led to an extremely high densification of the network. From an average of 142 friends per user, the average node degree became $7,239$ (out of maximum possible $9,178$). Since the CLIQUE model can use any of the link-based models, we chose to use it with the LINK model because it performed best from the link-based models. This CLIQUE-LINK model has an accuracy of 46.3% and due to the lack of sparsity, its training took much longer time than any of the other approaches.

The group-based classification results were more promising. We evaluated

our methods under a wide range of conditions, and we report on the ones that provided more insight in terms of high accuracy and node coverage. Figure 7.3(a) shows that naïvely running GROUP on all group memberships, the prediction accuracy was 63.5%. However, as larger groups are excluded, the accuracy improves even further (72.1%). This shows that medium to small-sized groups are more informative. Choosing the relevant groups based solely on their entropy shows even better results (Figure 7.3(b)). Using the groups with entropy lower than 0.5 resulted in the best accuracy. We also pruned groups based on varying percentages of public profiles per group which raised the accuracy even further (Figure 7.3(c)). Other advantages of choosing relevant groups were that it reduced the group space by 71.2% and that SVM training time was much shorter. The disadvantage is that as we prune groups, some of the users do not belong to any of the chosen groups, thus the node coverage decreases: 51% of the private profile attributes were predicted with 83.6% accuracy.

For privacy purposes, this is a strong result, and it means that groups can help an adversary predict the sensitive attribute for half of the users with private profiles with a high accuracy. Figure 7.3(d) shows that the more the private profiles in the network, the worse the accuracy. However, even in the case of mostly private profiles, the GROUP attack is still successful (63.4%). The reported results are for the case when the minimum portion of public profiles per group is equal to the portion in the overall network and the cutoff for the maximum group entropy is at 0.5.

Looking at the most and least relevant groups also provides interesting in-

sights. The most heterogeneous group that our method found is ”worldwide-wondering - a travel atlas.” As its name suggests, it pertains to users from different countries and using it to predict someone’s country seems useless. Some of the larger homogeneous groups include ”Beautiful NC,” ”Disegni e scritte sui muri” and ”*Nederland belicht*”. Other homogeneous groups were related to country but not in such an obvious manner. For example, one of them has the nondescript name ”::PONX::” which turned out to be the title of a Mexican magazine. For one user we looked at, this group helped us determine that although he claims to be from all over the world, he is most likely from Mexico.

Mixed model. The model which uses both links and groups as features, LINK-GROUP, did not perform statistically different from the GROUP model (64.8%). This showed that adding the links to the GROUP model did not lead to an additional benefit.

Insights on privacy preservation. Since including only low-entropy groups significantly boosts the success of the group-based attack, we conjectured that not participating in low-entropy groups helps people preserve their privacy better. Figure 7.4 shows that if users with private profiles do not join low-entropy groups, then GROUP is no longer successful.

Facebook

We performed the same experiments for Facebook as for Flickr, and we provide a summary of the results here.

Link-based attacks. In predicting gender, we found that while AGG, CC and BLOCK performed similarly to the baseline, LINK's accuracy varied between 65.3% and 73.5%. In predicting the political views, the link-based methods performed similarly to the baseline as Table 7.2 shows. LINK's average accuracy was not significantly different from the rest. We also performed binary classification to predict whether someone is liberal or not and the results were similar. The best-performing method was LINK with 61.8% accuracy. From privacy perspective, this result means that while it is easy to predict gender, it is hard to predict the political views of Facebook users based on their friendships.

Group-based attacks. The GROUP attack was successful in predicting gender (73.4%) when using all groups. Selecting groups that have at least 50% public profiles per group raised the accuracy by 4% but dropped the node coverage by a half. Predicting political views with GROUP was not successful (45.2%); some possible explanations are that the groups we considered are not real social groups and that books, movies and music taste of first-year college students may not be related to their political views. The relatively low number of groups may also have had an effect.

Mixed model. Again, LINK-GROUP did not perform statistically different from the other best-performing models (72.5% for gender, 57.8% for political views).

Dogster

Link-based attacks. Due to the fact that this was a random rather than a snowball sample, there were only 432 nodes with links, and link-based methods are at an unfair disadvantage, so we do not report their results here.

Group-based attacks. The baseline accuracy was 28.6%. CLIQUE-LINK's accuracy was significantly higher (60.2%), as was GROUP's accuracy (65.5%) when there were 50% public profiles. Pruning groups based on entropy led to even higher accuracy (88.9%) but had lower node coverage (14.9%). Figure 7.5(a) shows the accuracy and node coverage for various private profile percentage assumptions. We tried different options for the maximum group entropy required, and here, we report on the results for 0.5. The accuracy increased significantly as the number of public profiles in the network increased with one exception: the accuracies for 70% and 90% public profiles did not have a statistically significant difference. A group named "All Fur Fun" was the least homogeneous of all groups, i.e., had the highest group entropy of 2.7. The online profile of the group shows that this is a group that invites all dogs to party together, so it is not surprising that dogs of many different breeds join.

BibSonomy

Group-based attacks. We used the BibSonomy data to see whether the group-based classification approach can help in predicting whether someone is a spammer or not. There is a large class skew in the data: most of the labeled user profiles

are spammer profiles and the baseline accuracy is 92.2%. Using all groups when 50% of the profiles are public leads to a statistically significant improvement in the accuracy (94%) and has a very good node coverage (98.5%); this covers almost all users with tags that at least one other user uses (98.7%). The accuracy results for BibSonomy are presented in Figure 7.5(b). We explored different options for the minimum entropy required, and we report on the results for it being 0, i.e., only completely homogeneous groups were chosen. As in the other results, the coverage gets lower when the most homogeneous groups are chosen (which in the spam case is actually undesirable). Precision was 99.9-100% in all group-based classification cases, meaning that virtually all predicted spammers were such, whereas in the baseline case, it is 92.2%. The results also suggest that if more profiles were labeled, then more covered spammers can be caught. Some of the homogeneous tags with many taggers include "mortgage" and "refinance."

7.5 Discussion

Privacy. Our work shows that groups can leak a significant amount of information and not joining homogeneous groups preserves privacy better. People who are concerned about their privacy should consider properties of the groups they join, and social network providers should warn their users of the privacy breaches associated with joining groups. Obviously, in dynamically-evolving environments, it is harder to assess whether a group will remain diverse as more people join and leave it. Another privacy aspect is the ability to join public groups

but display group memberships only to friends. Currently, neither Facebook nor Flickr allow group memberships to be private and this is a desirable solution to the problem we have discussed.

Surprisingly, link-based methods did not perform as well as we expected. This suggests that breaking privacy in social networks with mixed private and public profiles is not necessarily straightforward, and using friends in classifying people has to be treated with care. We also conjecture that this depends on the dataset. For example, while link-based methods were not very successful in predicting the location of users in Flickr, they may work well in LiveJournal; for example, a study by Liben-Nowell et al. [81] showed that most of the friendship links in LiveJournal are related to geographical proximity. Another important point to consider is the nature of the sensitive attribute we are trying to predict. For example, predicting someone's political views may be a very hard task in general. Recent research by Baldassarri et. al. [9] shows that most Americans are neither consistently liberal nor conservative, and thus labeling a person as one or the other is inappropriate.

In some cases, the assumption that unpublished private attributes can be predicted from those made public may not hold. This happens when the attribute distribution in private profiles is very different from the one in public profiles. An extreme example is a disease attribute which shows values for common diseases such as Flu, Fever, etc, in public profiles, whereas more sensitive values such as HIV appear only in private profiles. In a similar example, young people tend to make their age public, and older ones tend to keep it secret. We plan to address

this issue in future work.

Data anonymization. The challenge of anonymizing graph data lies in understanding the rich dependencies in the data and removing sensitive information which can be inferred by direct or indirect means. Here, we show attribute-disclosure attacks in data which is meant to be partially private. Our results suggest that a data provider should consider removing groups that are homogeneous in respect to sensitive attributes before releasing an anonymized dataset in the public domain. Our privacy attacks are also meant to show that more sophisticated anonymization techniques are necessary.

Data mining. We show that it is possible to predict the attributes of some users with hidden profiles and create better statistics of the attribute's overall distribution. For example, if a marketing company can predict the gender and location of users with hidden profiles, it can improve its targeted marketing. As groups with higher entropy are added, the uncertainty associated with the attribute prediction increases, and it becomes harder to utilize the existence of diverse groups for sensitive attribute inference.

Remaining research questions. There are a number of interesting questions that remain to be answered: What are the properties that make a social network vulnerable to a group-based attack? Are profiles on social media websites more or less vulnerable than ones on a purely networking website? What are the specific privacy guidelines that a social network website provider should follow to ensure its users are protected against unintended privacy leaks? Do users with private profiles have group-membership patterns that are different and more

privacy-preserving from public-profile members? These are questions of social relevance and we hope that our research will inspire more work in this area.

7.6 Conclusion

While having a private profile is a good idea for the privacy-concerned users, their links to other people and affiliations with public groups pose a threat to their privacy. In our attribute disclosure work, we showed how one can exploit a social network with mixed profiles to predict the sensitive attributes of users. Using group information, we were able to discover the sensitive attribute values of some users with surprisingly high accuracy on four real-world social-media datasets. We hope that these results will raise the privacy awareness of social media users and will motivate social media websites to enable greater control over release of information and to help their users understand the potential for leaking information.

Chapter 8

Link Disclosure

The goal of data mining is discovering new and useful knowledge from data. Sometimes, the data contains sensitive information, and it needs to be sanitized before it is given to data mining researchers and the public in order to address privacy concerns. Data sanitization is a complex problem in which hiding private information trades off with utility reduction. The goal of sanitization is to remove or change the attributes of the data which help an adversary infer sensitive information. The solution depends on the properties of the data and the notions of privacy and utility in the data.

Most of the work in this area makes the assumption that the data is described by a single table with attribute information for each of the entries. However, real-world datasets often exhibit more complexity. Relational data, often represented as a multi-graph, can exhibit rich dependencies between entities. The challenge of anonymizing graph data lies in understanding these dependencies and removing sensitive information which can be inferred by direct or indirect

means.

Very little work has been done in this direction, and there has been a growing interest in it. The existing work looks at the identifying structural properties of the graph nodes [7, 56], or considers relations to be attributes of nodes [114]. Our work assumes that the anonymized data will be useful only if it contains both structural properties and node attributes. We study anonymization techniques to match this assumption.

In this section, we focus on the problem of preserving the privacy of sensitive relationships in graph data. We refer to the problem of inferring sensitive relationships from anonymized graph data as *link re-identification*. We propose five different privacy preservation strategies, which vary in terms of the amount of data removed (and hence their utility) and the amount of privacy preserved. We assume the adversary has an accurate predictive model for links, and we show experimentally the success of different link re-identification strategies under varying structural characteristics of the data.

Unlike existing work on privacy preservation which concentrates on hiding the identity or attributes of entities, we look at the case where relationships between entities are to be kept private. Finding out about the existence of these sensitive relationships leads to a privacy breach. We refer to the problem of inferring sensitive relationships from anonymized graph data as *link re-identification*.

Examples of sensitive relationships can be found in social networks, communication data, search engine data, disease data and others. In social network data, based on the friendship relationships of a person and the public preferences

of the friends such as political affiliation, it may be possible to infer the personal preferences of the person in question as well. In cell phone communication data, finding that an unknown individual has made phone calls to a cell phone number of a known organization can compromise the identity of the unknown individual. Another example is in search data: being able to link search queries made by the same individual can give personal information that helps identify that individual. In hereditary disease data, knowing the family relationships between individuals who have been diagnosed with hereditary diseases and ones that have not, can help infer the probability of the healthy individuals to develop these diseases.

We consider the node data to be anonymized using a known single-table definition such as k-anonymization [130] or the more recently proposed t-closeness [79]. For the edge data, we propose five different anonymization strategies. The most conservative approach is to remove the relationships altogether, thus preserving any privacy that these relationships may compromise. We assume that while all of the sensitive relationships are removed, all or a portion of the relationships of other types are left intact in the anonymized data. We propose a method which allows modeling the influence of data attributes on sensitive relationships, and studying how different anonymization techniques can preserve privacy. The privacy breach is measured by counting the number of sensitive relationships that can be inferred from the anonymized data. The utility of the data is measured by counting how many attributes or observations have to be deleted in the sanitization process.

To formalize privacy preservation, Chawla et al. [24] propose a framework

based on the intuitive definition that "our privacy is protected to the extent we blend in the crowd." What needs to be specified in this general framework is an abstraction of the concept of a database, the adversary information and its functionality, and when an adversary succeeds. Starting from this idea, we define the relational privacy framework for link re-identification. First, we discuss methods for anonymizing graph data and the resulting adversary information in Section 8.1. Section 8.2 covers graph-based privacy attacks, Section 8.3 discusses general link re-identification attacks, and Section 8.4 discusses link re-identification in anonymized data and when an adversary succeeds. Section 8.5 presents the benefits and disadvantages of each anonymization method in an experimental setting.

For the purpose of this work, we consider a multiplex graph $G = (V, E_v)$, composed of a set of nodes V and sets of social links $E_v = \{E^1, \ldots, E^k, E^s\}$ of $k + 1$ different types. The links $E^1, \ldots, E^k$ are the *observed* relationships, and E^s is the sensitive relationship, meaning that it is undesirable to disclose the e^s edges to the adversary. In this representation, all affiliation links E_h are converted into social links where a social link of type E^{h_x} exists between two nodes if both of them have affiliation links to groups h_j, i.e. $\exists e^{h_x}(v_i, v_j) \in E^{h_x} \iff \exists e_h(v_i, h_x) and \exists e_h(v_j, h_x)$. We use the short notation $e^k_{i,j}$ to specify $e^k(v_i, v_j)$.

In the process of anonymizing the data, the sensitive relationships are always removed, i.e., they are not provided in the released data. However, it may be possible to predict some of these relationships using other observed relationships and/or node attributes. For the purposes of this chapter, we focus on pre-

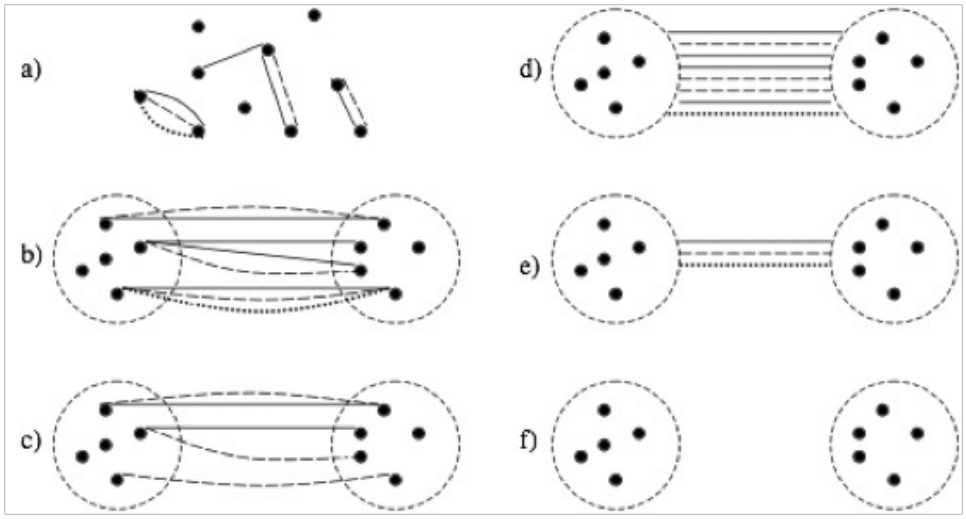


Figure 8.1: The original data graph (a)) and the output from five anonymization approaches to graph data: b) revealing the observations between nodes, c) removing 50% of the observations , d) revealing all the observations between equivalence classes of nodes (cluster-edge anonymization), e) constrained revealing of the observations between equivalence classes of nodes (cluster-edge anonymization with constraints), f) removing all relational observations. There are three different edge types in the original data graph represented by different line styles. Clusters resulting from node anonymization are circled with dotted lines.

dicting sensitive edges based on the observed edges, but it is straightforward to include node and edge attributes and interesting to also consider structural properties. If the sensitive edges can be identified, then we say that there has been a *privacy breach*.

In addition, the data can include certain *constraints* which specify the number of relationships of a particular type or the number of relationships connecting any two nodes. Constraints can also be inequality constraints describing the maximum or minimum number of relationships.

As a motivating example, consider the case where the entities are students, and the relationships between students v_i and v_j include taking a class h_c to-

gether ($e^{h_c}(v_i, v_j)$), belonging to the same research group h_g ($e^{h_g}(v_i, v_j)$), and being friends ($e^s(v_i, v_j)$). We can consider the class and research groups as the types of the edges, so that students can take more than one class together, and they can belong to more than one research group. In this case, we may consider e^s, the friendship between two people, to be the sensitive relationship. We are interested in understanding how difficult it is to determine friendship based on class and research group rosters.

8.1 Graph anonymization

The process of anonymization involves taking the unanonymized graph data, making some modifications, and constructing a new *released graph* which will be made available to the adversary. The modifications include changes to both the nodes and edges of the graph. We discuss several graph anonymization strategies and, for each approach, we discuss the tradeoffs between privacy preservation and the utility of the anonymized data.

We assume that the adversary has the information contained in the released graph data, and the constraints on the data. The *adversary succeeds* when she can figure out whether two nodes exhibit a sensitive relationship, i.e., when she is able to correctly predict a sensitive link between them. For example, if the adversary can figure out which students are likely to be friends given the released graph, then the data discloses private information about the two individuals.

8.1.1 Node anonymization

We assume that the nodes have been anonymized with one of the techniques introduced for single table data. For example, the nodes could be k-anonymized using t-closeness [79]. This anonymization provides a clustering of the nodes into m equivalence classes $(C_1, \ldots, C_m)$ such that each node is indistinguishable in its quasi-identifying attributes from some minimum number of other nodes. We use the following notation $C(v_i) = C_k$ to specify that a node v_i belongs to equivalence class C_k.

The anonymization of nodes creates equivalent classes of nodes. Note, however, that these equivalent classes are based on node attributes only, and inside each equivalence class, there may be nodes with different identifying structural properties and edges.

8.1.2 Edge anonymization

For the relational part of the graph, we describe five possible anonymization approaches. They range from one which removes the least amount of information to a very restrictive one, which removes the greatest amount of relational data. Figure 8.1(a) shows a simple data graph in which there are ten nodes and eight observed edges. There are three edge types, and each one is represented by a different line style. We will illustrate each of our techniques on this graph. For each approach, we discuss the tradeoffs between privacy preservation and the utility of the anonymized data.

Intact edges

The first (trivial) edge anonymization option is to only remove the sensitive edges, leaving all other observational edges intact. Figure 8.1(b) shows an illustration of this technique applied to the original data graph of Figure 8.1(a).

In our running example, we remove the friendship relationships, since they are the sensitive relationships, but we leave intact the information about students taking classes together and being members of the same research group. Since the relational observations remain in the graph, this anonymization technique should have a high utility. But it is likely to have low privacy preservation.

Intact-Edge Anonymization Algorithm

1: Input: $G = (V, E^1, \ldots, E^s)$
2: Output: $G' = (V', E^{1'}, \ldots, E^{k'})$
3: V′=anonymize-nodes(V)
4: **for** t=1 to k **do**
5: $\quad E^{t'} = E^t$
6: **end for**

Figure 8.2: Algorithm for anonymizing graph data by removing only the sensitive edges.

Partial-edge removal

Another anonymization option is to remove some portion of the relational observations. We could either remove a particular type of observation which contributes to the overall likelihood of a sensitive relationship, or remove a certain percentage of observations that meet some pre-specified criteria (e.g., at random, connecting high-degree nodes, etc.). Figure 8.1(c) shows an illustration of this technique when the edges are removed at random.

This partial edge removal process should increase the privacy preservation and reduce the utility of the data as compared to the previous method. Removing observations should reduce the number of node pairs with highly likely sensitive relationships but it does not remove them completely. For those pairs of nodes, private information may be disclosed.

Partial-Edge Anonymization Algorithm

1: Input: $G = (V, E^1, \ldots, E^k, E^s)$, percent-removed
2: Output: $G' = (V', E^{1'}, \ldots, E^{k'}))$
3: V′=anonymize-nodes(V)
4: **for** t=1 to k **do**
5: $E^{t'} = E^t$
6: removed = $\lceil$percent-removed $\times \|E^{t'}\| \rceil$
7: **for** i=1 to removed **do**
8: e_i = random edge from $E^{t'}$
9: $E^{t'} = E^{t'} \setminus \{e_i\}$
10: **end for**
11: **end for**

Figure 8.3: Algorithm for anonymizing graph data by removing randomly a portion of the observed edges.

Cluster-edge anonymization

In the above approaches, while the nodes had been anonymized, the number of nodes in the graph was still the same, and the edges were essentially between copies of the anonymized nodes. Another approach is to collapse the anonymized nodes into a single node for each cluster, and then consider which edges to include in the collapsed graph.

The simplest approach is to leave the sets of edges intact, and maintain the counts of the number of edges between the clusters for each edge type. We refer to this technique as *cluster-edge* anonymization. Figure 8.4 presents the algorithm for

this technique, and Figure 8.1(d) shows an illustration of the result from applying the algorithm.

Cluster-Edge Anonymization Algorithm

Input: $G = (V, E^1, \ldots, E^k, E^s)$,
Output: $G' = (V', E^{1'}, \ldots, E^{k'}))$
$V' = \{C_1, \ldots, C_m\}$
for t=1 to k **do**
    $E^{t'} = \emptyset$
    for all $(v_i, v_j) \in E^t$ **do**
        $C_i = C(v_i)$
        $C_j = C(v_j)$
        $E^{t'} = E^{t'} \cup \{(C_i, C_j)\}$
    end for
end for

Figure 8.4: Algorithm for cluster-edge anonymization technique.

Cluster-edge anonymization with constraints

Next, we consider using a stricter method for sanitizing observed edges than the previous technique. The *cluster-edge anonymization with constraints* technique creates edges between equivalence classes as above, but it requires the equivalence class nodes to have the same constraints as any two nodes in the original data. For example, if there can be at most two edges of a certain type between entities, there can be at most two edges of a certain type between the cluster nodes. This, in effect, removes some of the count information that is revealed in the previous anonymization technique.

In order to determine the number of edges of a particular type connecting two equivalence classes, the anonymization algorithm picks the maximum of the number of edges of that type between any two nodes in the original graph. In

Cluster-Edge Anonymization with Constraints Algorithm

```
1: Input: G = (V, E)
2: Output: G' = (V', E')
3: V' = {C_1, ..., C_m}
4: for t=1 to k do
5:    E^t' = ∅
6:    for all (v_i, v_j) ∈ E^t do
7:       C_i = C(v_i)
8:       C_j = C(v_j)
9:       if (C_i, C_j) ∉ E^t' then
10:         E^t' = E^t' ∪ {(C_i, C_j)}
11:      end if
12:   end for
13: end for
```

Figure 8.5: Algorithm for cluster-edge with constraints anonymization technique.

our earlier example, if the maximum number of common classes that any pair of students from the two equivalence classes takes is one class together, then the equivalence classes are connected by one class edge. Figure 8.1(e) shows an illustration of this technique.

This information will keep some of the utility of the data but it will say nothing of the distribution of observations. The anonymized data hides whether all observations appear on one two-node edge or on all two-node edges, and whether they ever appear in the same two-node edge. This may reduce the privacy breach on each sensitive relationship.

Removed edges

The most conservative anonymization option is to remove all the edges. Depending on the intended uses of an anonymized social network, removing the node and/or edge attributes completely may be undesirable. For example,

if one wants to know whether any first-year students took a particular course together, then all the three types of information, i.e., edges, edge attributes (such as edge type) and node attributes, are necessary. In our toy example, while taking a course together is information contained in a network edge, the name of the course is an edge attribute, and the year of enrollment is a node attribute. In this case, this anonymization technique leads to very low utility, yet high privacy preservation.

No-Edge Anonymization Algorithm

1: Input: G=(V,E)
2: Output: G′=(V′,∅)
3: V′=anonymize-nodes(V)

Figure 8.6: Algorithm for anonymizing graph data by removing the edges

8.2 Graph-based privacy attacks

According to Li et. al. [79], there are two types of privacy attacks in data: *identity disclosure* and *attribute disclosure*. In graph data, there is a third type of attack: *link re-identification*. Identity disclosure occurs when the adversary is able to determine the mapping from an anonymized record to a specific real-world entity (e.g. an individual). Attribute disclosure occurs when the adversary is able to infer the attributes of a real-world entity more accurately than it would be possible before the data release. Identity disclosure often leads to attribute disclosure [79]. Both identity disclosure and attribute disclosure have been studied very widely in the privacy community [5, 7, 13, 24, 56, 79, 93, 112, 113, 114, 130, 136].

Rather than focus on these two kinds of attack, the focus of our chapter is on link re-identification. Link re-identification is the problem of inferring that two entities participate in a particular type of sensitive relationship or communication. *Sensitive conclusions* are more general statements that an adversary can make about the data, and can involve both node, edge and structural information. These conclusions can be the results of aggregate queries. For example, in a database describing medical data informal about company employees, finding that almost all people who work for a particular company have a drinking problem may be undesirable. Depending on the representation of the data, this can be revealed by using both the node attributes and the co-worker relationship.

8.3 Link re-identification attacks

The extent of a privacy breach is often determined by data domain knowledge of the adversary. The domain knowledge can influence accurate inference in subtle ways. The goal of the adversary is to determine whether a sensitive relationship exists. There are different types of information that can be used to infer a sensitive relationship: node attributes, edge existence, and structural properties. Based on the domain knowledge of the adversary, she can construct rules for finding likely sensitive relationships. In this work, we assume that the adversary has an accurate probabilistic model for link prediction, which we will describe below.

In our running example, the sensitive friendship link may be re-identified

based on node attributes, edge existence or structural properties. For example, consider two student nodes containing a boolean attribute "Talkative." Two nodes that both have it set to "true" may be more likely to be friends than two nodes that both have it set to "false." This inference is based on node attributes. An example of re-identification based on edge existence is two students in the same research group who are more likely to be friends compared to if they are in different research groups. A re-identification that is based on a structural property such as node degree says that two students are more likely to be friends if they are likely to correspond to high degree nodes in the graph. A more complex observation is one which uses the result of an inferred relationship. For example, if each of two students is highly likely to be a friend with a third person based on other observations, then the two students are more likely to be friends too.

8.3.1 Link re-identification using observations

We assume that the adversary has a probabilistic model for predicting the existence of a sensitive edge based on a set of observations $\boldsymbol{O}$: $P(e_{ij}^s|\boldsymbol{O})$. In this work, we assume a simple *noisy-or model* [123] for the existence of the sensitive edge. The noisy-or model can capture the fact that each observed edge contributes (in a probabilistic way) to the probability of the sensitive edge existing; it makes the simplifying assumption that each factor is an independent cause for the sensitive edge. Here, we focus on re-identification based on edge existence, so the observations that we consider are sets of edges, e_{ij}^l. For simplicity, we label

these observations $o_1, \ldots, o_n$. For each observed edge, we assume that we have a *noise* parameter, $\lambda_1, \ldots, \lambda_n$, and, in addition, we have a *leak* parameter λ_0 which captures the probability that the sensitive edge is there due to other, unmodeled, reasons. A noise parameter λ_i captures the independent influence of an observed relationship o_i on the existence of a sensitive relationship. Then, according to the noisy-or model, the probability of a sensitive edge is:

$$P(e^s_{ij} = 1) = P(e^s_{ij} = 1|o_1, ..., o_n) \quad = \quad 1 - \prod_{l=0}^{n}(1 - \lambda_l)$$

The above formula applies only when the observations are certain. It is also possible that the observation existence is not known. In that case, there are probabilities $P(o_1), \ldots, P(o_n)$ associated with the existence of each observation, and the probability of a sensitive edge is:

$$P(e^s_{ij} = 1) = \sum_{\{\mathbf{o}\}} P(e^s_{ij} = 1|\mathbf{o}) \prod_{k=1}^{n} P(o_k)$$

where

$$P(e^s_{ij} = 1|\mathbf{o}) \quad = \quad 1 - (1 - \lambda_0) \prod_{l=1}^{n}(1 - \lambda_l)^{o_l}$$

More details about this model can be found in [132].

The noisy-or function is applicable when there are a few observations that can cause an event, and each one can contribute positively to the likelihood of the event, independent of the rest. The function has some nice properties: 1) the result of it is always between 0 and 1 when the input probabilities are in that range; 2) the final result is independent of the order in which the observations are added; 3) it can accommodate different number of observations; 4) adding a new positive observation always increases the overall likelihood. We use this function to measure how likely each sensitive relationship is, and to find whether there are parts of the graph that are vulnerable to an adversary attack. It is also possible to express the dependence between events in an explicit probability model such as a Bayesian or a Markov network, when the dependences between observations are known.

8.3.2 Amount of information disclosed

Based on the noisy-or model for each pair of nodes, it is possible to determine the number of node pairs that are likely to participate in a sensitive relationship. In the anonymized data, it is desirable to have few sensitive relationships which can be inferred with high likelihood. To formalize this desirable property, we can compute the percentage of all possible two-node relationships which have a high likelihood and make sure that it is below some allowed level δ:

$$\frac{|relationships(P(e_{ij}^{s}) > \rho)|}{|V|^2} < \delta \tag{8.1}$$

where ρ is the threshold for predicting that a sensitive relationship exists and $relationships(P(e^s_{ij}) > \rho)$ returns the set of all sensitive relationships which have likelihood above ρ. For example, if it is true for the given data that 15% of the possible pair relationships have a true likelihood of exhibiting a sensitive relationship higher than 0.8, then

$$\frac{|relationships(P(e^s_{ij}) > 0.8)|}{|V|^2} <= 0.15.$$

For each anonymization technique, it is possible to find the highest possible δ that satisfies a particular ρ level. This can be used to compare the privacy preservation for each technique. The higher the δ, the lower the privacy preservation.

8.3.3 Utility

Utility in the data is hard to measure, and we make an assumption that the more observations there are in the anonymized data, the better. To measure utility, we use a very simple approach. We count the number of observations which were removed in the process of anonymization. The lower the number of removed observations, the higher the overall utility. For the intact edge and the cluster-edge anonymization techniques, no relational observations are deleted, therefore, these two techniques have the highest utility. For the partial edge removal technique, the utility depends on the percentage of edges removed. For the cluster-based with constraints technique, it is much lower, since the graph is collapsed, and many edges are removed. The exact number can be computed

using the properties and constraints of the data such as number of nodes, edges of each type, and the size of the equivalence classes. Note that a more sophisticated measure of utility would also consider the loss of structural properties in the anonymized data. In the case when all the edges are removed, the utility is 0.

8.4 Link re-identification in anonymized data

In the first two types of link anonymization (intact and partial), the noisy-or model can be used directly to compute the probability of a sensitive edge. In the other two cases, one has to consider the probability that an observed edge exists between two nodes, and apply the noisy-or.

8.4.1 Link re-identification in cluster-edge anonymization

In the case of keeping edges between equivalence classes, the probability of an observation existing between two nodes is not given and it needs to be estimated. The noisy-or function will need to take into consideration the probability associated with each observation in order to compute the likelihood of a sensitive relationship. When the number of relationships of each type (e.g., course, research group, etc.) between two equivalence classes is given, the distribution is not uniform, and the probability of an observation *P(o)=P(observation(v_i, v_j))* existing between two students can be computed directly from the counts of relationships between their equivalence classes. *P($e_{i,j}^{h_c}$)* expresses the probability that there exists a class edge between any two students v_i and v_j from two equivalence

classes $C(v_i)$ and $C(v_j)$, i.e., the students take a course h_c together. It is equal to the number of possible student pairs from the two equivalence classes who take a course together as a fraction of the number of possible relationships in the graph $|V|^2$.

8.4.2 Link re-identification in cluster-edge anonymization with constraints

In the constrained cluster-edge anonymization approach, the number of relationships between equivalence classes is not given. Therefore, the probability of an observation existing between any two edges has to be taken into account in the noisy-or model. To estimate this probability, an adversary can assume a uniform distribution, meaning that the probability of an observation existing between any two edges is the same for all edges in the graph. This estimate is worse than the cluster-edge anonymization method. Using the constraints on the data, it is possible to get estimates of this probability. For example, if it is known that there are 50 pairs of students who take courses together, and there are 100 possible pairs, then the probability of any two students taking any class h_c together is $P(e_{i,j}^{h_c})=0.5$. If the adversary knows the number of offered courses m, the number of courses per person n, the number of students $s = |V|$, and assumes that all courses have the same number of people $p = \frac{s*n}{m}$, then the number of possible pairs who take courses together can be calculated as $n * (p - 1)$. This number can be used to compute in a manner similar to the cluster-edge anonymization

method $P(e_{i,j}^{h_c})=\frac{n*(p-1)}{|V|^2}$.

One can also use an expected value of any two-node relationship to be sensitive by looking at the likelihood distribution of all relationships. However, we found that this does not measure privacy well because an adversary is more interested in the highly likely relationships.

An observation probability shows the percentage of edges between two nodes from two different equivalence classes that contain the observation. For example, if the two equivalence classes have exactly 10 nodes each, and the observation exists for 30 of the two-node edges, then the edge probability is $P(\texttt{observation}(v_i, v_j))=0.3$ where $\texttt{observation}(v_i, v_j)$ is either $e_{i,j}^{h_c}$, or $e_{i,j}^{h_g}$ for any h_c and h_g. This increases the utility of the data as compared to the case when no probabilities are included, but it can also decrease the privacy preservation. An exception is the case when observations between equivalence classes have exactly the same distribution as the overall uniform distribution.

8.5 Experiments

The effectiveness of the anonymization approaches depends on the structural and statistical characteristics of the underlying graph. In order to study the influence of each anonymization approach on privacy preservation, we apply them to synthetic data generated under varying statistical and structural assumptions and compute the information disclosed. We show how many relationships are revealed at different probability thresholds. First, we describe the data gener-

ator.

8.5.1 Data generator

The data generator creates data according to the data model described in Section 1.1. The input to the data generator includes: the number of nodes, maximum number of nodes which can participate in a relationship (e.g., the maximum number of students taking the same class), the maximum number of relationships that each student can have with any other student (e.g., maximum number of classes that a student can take). For all observation types, the probability of two nodes exhibiting a sensitive relationship given the observation type is given and the leak probability, the probability of two nodes exhibiting a sensitive relationship due to unobserved causes.

For the concrete example, the data generator starts by creating a set of students, a set of classes, and a set of research groups. There are constraints on how many classes each student takes, and on how many research groups each student belongs. There are also constraints on the maximum number of students per class and on the maximum number of students per group. For each student, the generator picks random classes to enroll into up to the maximum number of classes per student possible. Similarly, each student is assigned to a random research group.

The nodes in the data graph represent students. There is a e^{h_c} edge connecting two students for each class they take together, and there is e^{h_g} edge if they

belong to the same research group. These pieces of information represent observations indicating that two students may be friends, i.e., that they may exhibit a sensitive relationship. The ground truth is generated by computing the probability of a friendship between each two students using the noisy-or model, and assigning the friendship a true value with a probability equal to that likelihood.

The parameters given to the data generator can be varied. We explore graphs which vary in their density, therefore we allow the number of classes and research groups to vary while fixing the number of nodes/students to 100. The constraints on the data are that each student takes two classes, and belongs to one research group. Also, a class can have no more than 25 people, and a group can have no more than 15. We picked probabilities which make sense in the domain. The prior probability of two students knowing each other is $P(e^s_{i,j})$=*0.2*. It is relatively high because the students are from the same department. The probability that two students know each other if they are in the same class h_c is $P(e^s_{i,j}|e^{h_c}_{i,j})$=*0.4*. The probability that two students know each other if they are in the same research group is $P(e^s_{i,j}|e^{h_g}_{i,j})$=*0.6*.

8.5.2 Evaluating privacy preservation in anonymized data

We begin by studying the privacy preservation in the data that results from each of the anonymization techniques. In particular, we study the number of correctly identified sensitive relationships for the following anonymization functions: 1) when the anonymization function leaves the edges between nodes in-

tact (8.1.2), 2) when it removes 50% of the observations chosen at random (8.1.2), 3) when it leaves edges between node equivalence classes in the cluster-edge anonymization (8.1.2), and 4) when it leaves edges between node equivalence classes with a constrained number of observations (8.1.2). For the last two, each node is assigned randomly to an equivalence class. We vary k, the number of nodes in each equivalence class, and show the results for $k = 2$ and $k = 6$ because they exhibit the tendencies of varying k well.

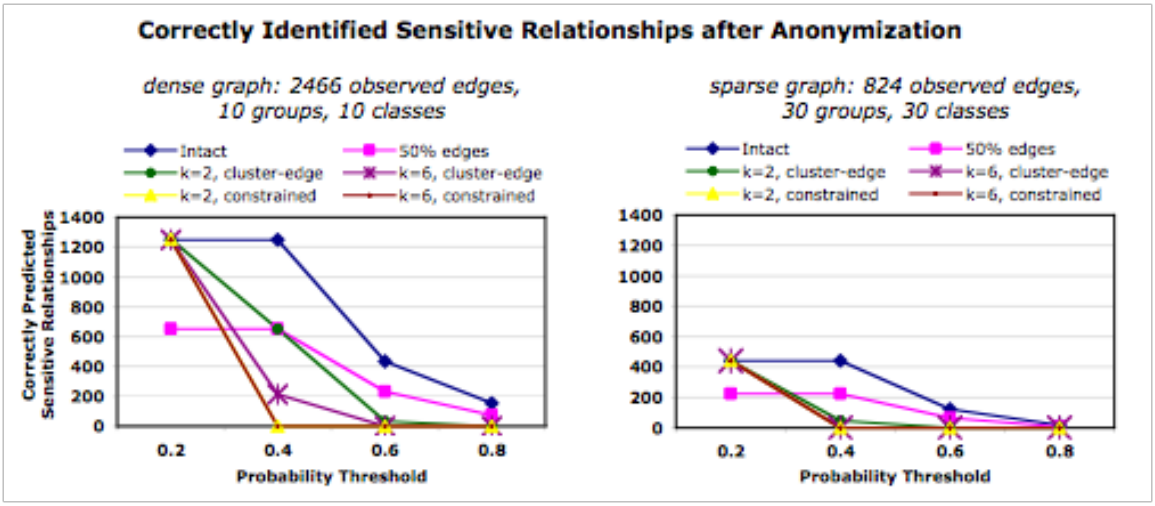


Figure 8.7: Comparison between the number of sensitive relationships found after each of six anonymization techniques has been applied. The number of revealed friendships decreases as the friendship likelihood threshold increases. The two constrained cluster-edge methods (at $k = 2$ and $k = 6$) reveal the same number of relationships in both graphs. In the sparse graph, the cluster-edge method at $k = 6$ (not constrained) also overlaps with the two constrained methods.

The data was generated with the default parameters, varying the number of classes and the number of research groups between 10 and 30. A graph, in which there are 10 research groups and 10 classes, is very dense, and a graph at the other extreme with 30 research groups and 30 classes is very sparse. We show these

"extreme" cases in Figure 8.7 and Figure 8.8. To account for the randomness in the generated graph, we ran the experiments on 100 generated graphs, and present the average performance. Note that when using the default data parameters (at most two classes taken by each student and at most one group of which a student is a member), the maximum possible likelihood for their friendship is 0.89.

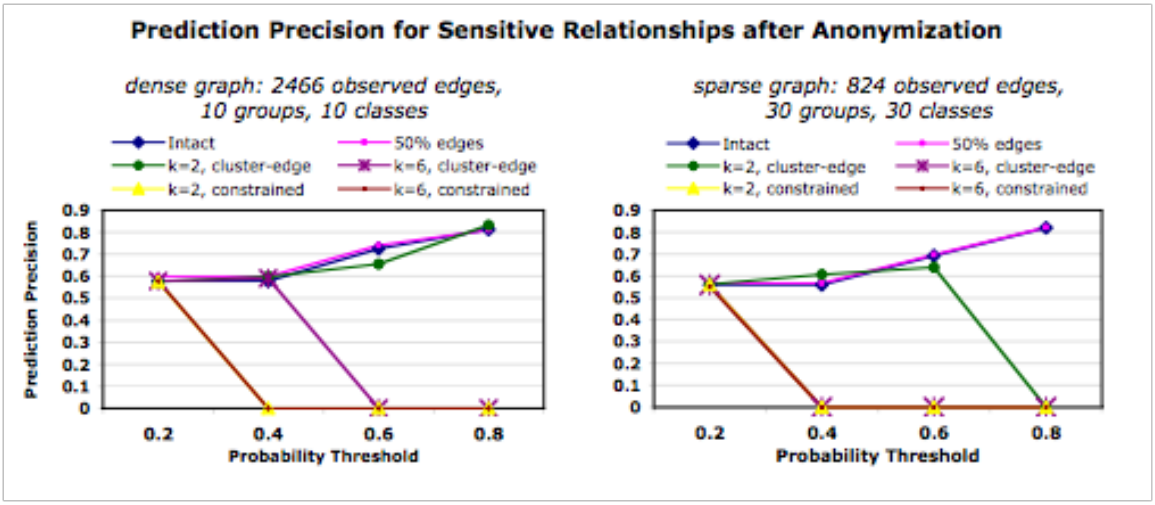


Figure 8.8: Comparison between the precision of predicted friendships found after one of six anonymization techniques has been applied. At low threshold values, the number of revealed friendships is large but the precision is low. The precision of the method that removes 50% of the edges at random overlaps with the precision of the intact-edge method in the sparse graph, and nearly overlaps in the dense graph. The precision of the two constrained cluster-edge methods (at $k = 2$ and $k = 6$) overlap as well.

We measure the precision, recall rate and the number of inferred sensitive relationships in the anonymized graphs. The precision shows how many of the predicted sensitive relationships are true sensitive relationships. The recall rate measures what portion of the true sensitive relationships can be predicted. Translated into the privacy domain, the recall rate measures what portion of the true sensitive relationships have been compromised, and the precision shows what is

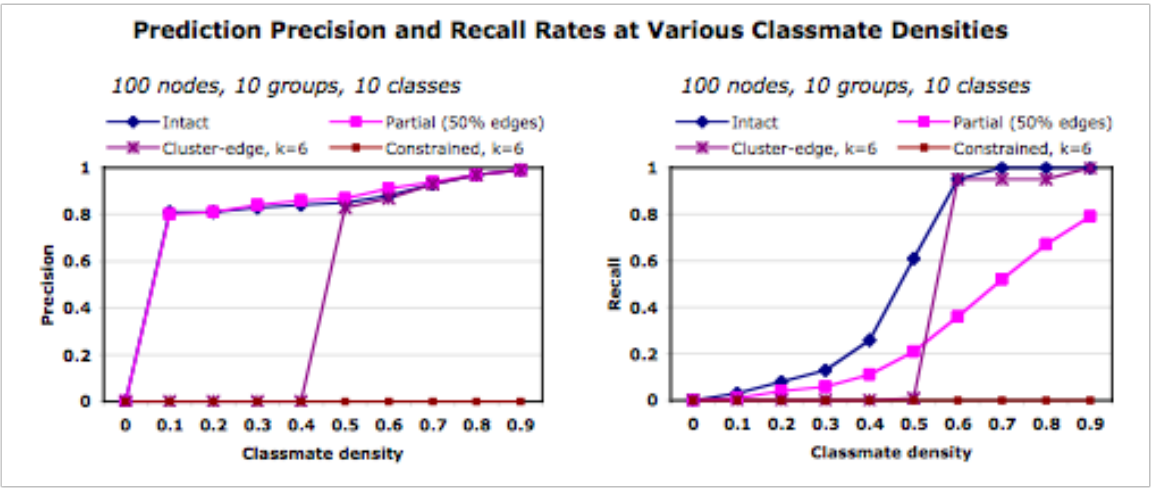


Figure 8.9: Comparison between the precision at different classmate density levels (a)) shows that at high density levels, the cluster-edge anonymization preserve privacy as badly as the anonymization which deletes 50 % of the edges. Moreover, the recall rate at these levels (b)) is much higher for the cluster-edge method. The groupmate density is kept constant at 0.1.

the chance that a predicted relationship is really a sensitive one. For example, if the analysis predicts 10 sensitive relationships and only 5 of them are true, then the precision is 0.5. If there are a total of 100 true sensitive relationships in the network, then the recall rate is 0.05. Ideally, a model for predicting sensitive information should have a high precision and a high recall rate when tested on the original data, and a low precision and a low recall rate when tested on the anonymized data.

A low precision in the anonymized data is more crucial than a low recall rate. A combination of a high precision with a low recall rate in the anonymized data is undesirable because it means that the anonymization can hide most of the sensitive relationships but the ones that can be predicted are highly likely to be true. Results with a low precision and a high recall rate are not as bad. In this case,

even though the anonymization allows many of the true sensitive relationships to be predicted, the true sensitive relationships are indistinguishable from many non-sensitive relationships.

8.5.3 Results

Figure 8.7 shows a comparison between the number of sensitive relationships inferred after each of six anonymization techniques has been applied. It shows that at higher thresholds (0.6 and 0.8), keeping all the edges between node equivalence classes preserves privacy much better than deleting 50% of the two-node edges, while having higher utility as discussed in Section 8.3.3. As expected, for lower k, the privacy preservation is lower: the number of revealed relationships is higher in the data anonymized with the cluster-edge method. In the data anonymized with the cluster-edge method with constraints, varying k yielded to the same results, which is why the graphs of $k = 2$ overlap with the graphs, in which $k = 6$.

We also ran the experiments for other combinations of class and group parameters in the range [10,30]. The experiments confirmed that as the number of observed edges decreases, so does the number of correctly identified sensitive relationships. However, the behavior at different thresholds is proportionately the same for all anonymization methods except the cluster-edge method. In the cluster-edge method, the privacy is preserved better in the sparse graph for both k levels, as seen by comparing the dense and the sparse graph results at threshold

0.4. In the sparse graph, the results when $k = 6$ are the same as the ones of the cluster-edge with constraints.

Figure 8.8 shows that even though lower probability thresholds reveal more sensitive relationships, the precision is low. At higher probability thresholds, the precision is high but on a very small number of predicted relationships.

Experimenting with the number of nodes in the network showed that the precision and sensitivity results were invariant to the network size when the friendship, groupmate and classmate densities were kept constant. The density values were 0.36, 0.1 and 0.2, respectively. The tested networks were of size 100, 200, 300 and 400 nodes. Other constant parameters were the number of groups, 10, the number of classes, 10, and the k-anonymization parameter $k = 6$.

We also varied the multigraph classmate density by varying the number of classes each student joined. Since this parameter was used in the data generator as well, it affected the friendship density of the original graph. The correlation between the two densities was positive. We found that at high classmate density levels the claim that the cluster-edge anonymization preserves privacy better than the anonymization which deletes 50% of the edges no longer held. As Figure 8.9a) shows that as the class density goes above 0.4 (friendship density is 0.63), the precision of predicted sensitive links is almost the same for the two methods. Moreover, as Figure 8.9b) at levels above 0.5 (friendship density is 0.76), the data anonymized with the cluster-edge method has much higher recall rate. Again, the number of nodes was 100, the number of groups was 10, the number of classes was 10, and the k-anonymization parameter k was 6.

8.6 Conclusion

Here, we presented the problem of link re-identification. We have proposed several approaches for anonymizing graph data and done an initial empirical evaluation of the effectiveness of the different strategies. Understanding and appreciating the subtleties in the effectiveness of techniques is an important and timely topic for data providers interested in releasing social network datasets.

Chapter 9

Conclusion

Real-world social network data exhibits complex interactions and dependencies. Understanding the processes that govern the generation of user content and network growth in social media is not trivial, and the predictive and descriptive algorithms in this domain need to reflect the inherent structure of the data.

In this thesis, we have taken an initial step towards understanding the user behavior in social and affiliation networks. We have shown the importance of studying both social and affiliation networks in a variety of settings. In particular, we addressed the following three related subject areas: 1) prediction of user attributes and latent user preferences, 2) network evolution and link prediction, and 3) privacy in social networks. Our work suggests that affiliation networks are just as important if not more important when studying online social networks. The information in them complements information found on the social network built around pairwise, or user-user links. Social and affiliation networks allow us to study the macro behavior and micro incentives of users and to build better

user behavior models. Our work also shows that affiliation network information is very important to take into account when studying privacy issues in social networks.

With this thesis, we hope to motivate further research in social and affiliation networks which goes beyond pairwise interactions and studies models for complex group behavior using observed and unobserved user characteristics, roles and interactions. Exciting new directions include exploring principled approaches to modeling collective group behavior in complex networks to gain insight into people's motivations, preferences and interests, and developing dynamic network evolution models which consider not only structural properties but also attribute correlations. We envision a number of applications for these types of models, such as building better personalized services which focus on both the predictive and privacy aspect of the algorithms, studying human emotional and physical health based on people's support networks, as well as designing more successful marketing or political campaigns.

Bibliography

[1] A. Acquisti and R. Gross. Predicting social security numbers from public data. In *Proceedings of the National Academy of Sciences (PNAS)*, 2009.

[2] L. Adamic and E. Adar. Friends and neighbors on the web. *Social Networks*, 25(3):211–230, 2003.

[3] C. Aggarwal, editor. *Social Network Data Analytics*. Springer, 2011.

[4] C. C. Aggarwal and P. S. Yu. *Privacy-Preserving Data Mining: Models and Algorithms*. Springer, 2008.

[5] G. Aggarwal, T. Feder, K. Kenthapadi, R. Motwani, R. Panigrahy, D. Thomas, and A. Zhu. Approximation algorithms for k-anonimity. *Journal of Privacy Technology*, Nov. 2005.

[6] E. Airoldi, D. Blei, S. Fienberg, and E. Xing. Mixed-membership stochastic blockmodels. *Journal of Machine Learning Research (JMLR)*, 9:1981–2014, 2008.

[7] L. Backstrom, C. Dwork, and J. Kleinberg. Wherefore art thou r3579x: anonymized social networks, hidden patterns, and structural steganography. In *International World Wide Web Conference (WWW)*, 2007.

[8] L. Backstrom, D. Huttenlocher, J. Kleinberg, and X. Lan. Group formation in large social networks: Membership, growth, and evolution. *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, 2006.

[9] D. Baldassarri and A. Gelman. Partisans without constraint: Political polarization and trends in american public opinion. *American Journal of Sociology*, 114(2):408–446, September 2008.

[10] A.-L. Barabási and R. Albert. Emergence of scaling in random networks. *Science*, 286(5439):509–512, October 1999.

[11] A.-L. Barabási, H. Jeong, Z. Neda, E. Ravasz, A. Schubert, and T. Vicsek. Evolution of the social network of scientific collaborations. *PHYSICA A*, 311:3, 2002.

[12] M. Barbaro and T. Zeller. A face is exposed for aol searcher no. 4417749. *New York Times*, August 2006.

[13] R. Bayardo and R. Agrawal. Data privacy through optimal k-anonymization. In *IEEE International Conference on Data Engineering (ICDE)*, 2005.

[14] S. Bhagat, G. Cormode, B. Krishnamurthy, and D. Srivastava. Class-based graph anonymization for social network data. In *International Conference on Very Large Databases (VLDB)*, 2009.

[15] S. Bhagat, G. Cormode, B. Krishnamurthy, and D. Srivastava. Privacy in dynamic social networks. In *International World Wide Web Conference (WWW) Poster*, 2010.

[16] S. Bhagat, G. Cormode, and S. Muthukrishnan. Node classification in social networks. In C. Aggarwal, editor, *Social Network Data Analytics*, pages 115–148. Springer, 1 edition, 2011.

[17] I. Bhattacharya and L. Getoor. A latent dirichlet model for unsupervised entity resolution. In *SIAM International Conference on Data Mining (SDM)*, 2006.

[18] C. M. Bishop. *Pattern Recognition and Machine Learning*. Springer, 2006.

[19] D. Blei and J. Lafferty. Topic models. *Text Mining: Classification, Clustering, and Applications*, 2009.

[20] D. Blei, A. Ng, and M. Jordan. Latent dirichlet allocation. *Journal of Machine Learning Research (JMLR)*, 3:993–1022, January 2003.

[21] Y. Boykov, O. Veksler, and R. Zabih. Fast approximate energy minimization via graph cuts. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 23:1222–1239, November 2001.

[22] A. Campan and T. M. Truta. A clustering approach for data and structural anonymity in social networks. *KDD Workshop on Privacy, Security, and Trust in KDD (PinKDD)*, 2008.

[23] D. Chakrabarti and C. Faloutsos. Graph mining: Laws, generators, and algorithms. *ACM Computing Surveys*, 38(1), 2006.

[24] S. Chawla, C. Dwork, F. Mcsherry, A. Smith, and H. Wee. Toward privacy in public databases. In *Theory of Cryptography Conference (TCC)*, 2005.

[25] B.-C. Chen, D. Kifer, K. LeFevre, and A. Machanavajjhala. Privacy-preserving data publishing. *Foundations and trends in databases*, 2(1–2):1–167, 2009.

[26] D. R. Choffnes, J. Duch, D. Malmgren, R. Guimera, F. E. Bustamante, and L. Amaral. Swarmscreen: Privacy through plausible deniability in p2p systems tech. Technical Report NWU-EECS-09-04, Department of EECS, Northwestern University, June 2009.

[27] K. Clarkson, K. Liu, and E. Terzi. Towards identity anonymization in social networks. In P.Yu, J. Han, and C. Faloutsos, editors, *Link Mining: Models Algorithms and Applications*. Springer, 2010.

[28] G. Cormode, D. Srivastava, T. Yu, and Q. Zhang. Anonymizing bipartite graph data using safe groupings. In *International Conference on Very Large Databases (VLDB)*, 2008.

[29] C. Cortes, D. Pregibon, and C. Volinsky. Communities of interest. In *Advances in Intelligent Data Analysis*, 2001.

[30] danah boyd. Privacy and publicity in the context of big data. In *WWW Invited Talk*, 2010. Available at `http://www.danah.org/papers/talks/2010/WWW2010.html`.

[31] S. Das, mer Egecioglu, and A. E. Abbadi. Anonymizing weighted social network graphs. In *IEEE International Conference on Data Engineering (ICDE)*, 2010.

[32] S. C. Deerwester, S. T. Dumais, T. K. Landauer, G. W. Furnas, and R. A. Harshman. Indexing by latent semantic analysis. *Journal of the American Society of Information Science*, 41(6), 1990.

[33] L. Dietz, S. Bickel, and S. Tobias. Unsupervised prediction of citation influences. In *International Conference on Machine Learning (ICML)*, 2007.

[34] C. Dwork. Differential privacy. In *International Colloquium on Automata, Languages and Programming (ICALP)*, 2006.

[35] C. Dwork. An ad omnia approach to defining and achieving private data analysis. *KDD Workshop on Privacy, Security, and Trust in KDD (PinKDD) 2007*, 4890:1–13, 2008.

[36] C. Dwork, K. Kenthapadi, F. McSherry, I. Mironov, and M. Naor. Our data, ourselves: privacy via distributed noise generation. In *International Conference on the Theory and Applications of Cryptographic Techniques (EUROCRYPT)*, 2006.

[37] C. Dwork, F. McSherry, K. Nissim, and A. Smith. Calibrating noise to sensitivity in private data analysis. In *Theory of Cryptography Conference (TCC)*, 2005.

[38] D. Easley and J. Kleinberg, editors. *Networks, Crowds, and Markets*. Cambridge University Press, 2010.

[39] E. Erosheva, S. Fienberg, and J. Lafferty. Mixed-membership models of scientific publications. In *Proceedings of the National Academy of Sciences (PNAS)*, 2004.

[40] M. Faloutsos, P. Faloutsos, and C. Faloutsos. On power-law relationships of the internet topology. *ACM SIGCOMM*, pages 251–262, September 1999.

[41] L. Fang and K. LeFevre. Privacy wizards for social networking sites. In *International World Wide Web Conference (WWW)*, 2010.

[42] K. Faust and S. Wasserman. Blockmodels: Interpretation and evaluation. *Social Networks*, 14:5–61, 1992.

[43] Y. Feng, Y. Zhuang, and Y. Pan. Music information retrieval by detecting mood via computational media aesthetics. In *IEEE/WIC/ACM International Conference on Web intelligence (WI)*, 2003.

[44] L. Freeman, editor. *The Development of Social Network Analysis*. Vancouver: Empirical Press, 2006.

[45] J. Friedland and D. Jensen. Finding Tribes: Identifying Close-Knit Individuals from Employment Patterns. In *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, 2007.

[46] L. Getoor and C. P. Diehl. Link mining: a survey. *SIGKDD Explorations Newsletter*, 7(2):3–12, December 2005.

[47] L. Getoor and B. Taskar, editors. *Introduction to statistical relational learning*. MIT Press, 2007.

[48] J. Golbeck. The dynamics of web-based social networks: Membership, relationships, and change. *First Monday*, 12(11), 2007.

[49] A. Goldenberg, J. Kubica, P. Komarek, A. Moore, and J. Schneider. A comparison of statistical and machine learning algorithms on the task of link completion. In *KDD Workshop on Link Analysis for Detecting Complex Behavior*, 2003.

[50] A. Goldenberg, A. Zheng, S. Fienberg, and A. E.M. A survey of statistical network models. In *Foundations and Trends in Machine Learning*, volume 2, pages 129–233. Now Publishers, 2009.

[51] D. Hand, H. Mannila, and P. Smyth. *Principles of Data Mining*. The MIT Press, 2001.

[52] M. Hasan, V. Chaoji, S. Salem, and M. Zaki. Link Prediction using Supervised Learning. In *SDM Workshop on Link Analysis, Counter-terrorism and Security*, 2006.

[53] M. A. Hasan and M. J. Zaki. A survey of link prediction in social networks. In C. Aggarwal, editor, *Social Network Data Analytics*, pages 243–276. Springer, 1 edition, 2011.

[54] M. Hay, C. Li, G. Miklau, and D. Jensen. Accurate estimation of the degree distribution of private networks. In *IEEE International Conference on Data Mining (ICDM)*, 2009.

[55] M. Hay, G. Miklau, and D. Jensen. Enabling accurate analysis of private network data. In F. Bonchi and E. Ferrari, editors, *Privacy-Aware Knowledge Discovery: Novel Applications and New Techniques*. Chapman & Hall/CRC Press, 2010.

[56] M. Hay, G. Miklau, D. Jensen, and D. Towsley. Resisting structural identification in anonymized social networks. In *International Conference on Very Large Databases (VLDB)*, August 2008.

[57] M. Hay, G. Miklau, D. Jensen, P. Weis, and S. Srivastava. Anonymizing social networks. Technical report, University of Massachusetts, Amherst, March 2007.

[58] K. Henderson and T. Eliassi-Rad. Applying latent dirichlet allocation to group discovery in large graphs. In *ACM SIGAPP Symposium On Applied Computing (SAC)*, pages 1456–1461, 2009.

[59] M. Hoffman, D. Blei, and P. Cook. Easy as cba: a simple probabilistic model for tagging music. In *International Symposium on Music Information Retrieval (ISMIR)*, 2009.

[60] T. Hofmann. Probabilistic latent semantic analysis. In *Conference on Uncertainty in Artificial Intelligence (UAI)*, 1999.

[61] X. Hu and J. S. Downie. Exploring mood metadata: Relationships with genre, artist and usage. In *International Symposium on Music Information Retrieval (ISMIR)*, pages 67–72, 2007.

[62] Z. Huang and D. Zeng. A Link Prediction Approach to Anomalous Email Detection. In *IEEE International Conference on Systems, Man, and Cybernetics*, 2006.

[63] M. I. Jordan and Y. Weiss. Graphical models: Probabilistic inference. In M. Arbib, editor, *The Handbook of Brain Theory and Neural Networks*, pages 490–496. MIT Press, 2 edition, 2002.

[64] C. Kemp, J. B. Tenenbaum, T. L. Griffiths, T. Yamada, and N. Ueda. Learning systems of concepts with an infinite relational model. In *AAAI Conference on Artificial Intelligence (AAAI)*, 2006.

[65] D. Kifer and A. Machanavajjhala. No free lunch in data privacy. In *ACM SIGMOD International Conference on Management of Data (SIGMOD)*, 2011.

[66] J. M. Kleinberg. Challenges in mining social network data: processes, privacy, and paradoxes. In *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, pages 4–5, 2007.

[67] B. Klimt and Y. Yang. The enron corpus: A new dataset for email classification research. In J.-F. Boulicaut, F. Esposito, F. Giannotti, and D. Pedreschi, editors, *European Conference on Machine Learning (ECML)*, volume 3201 of *Lecture Notes in Computer Science*, pages 217–226. Springer, 2004.

[68] P. Kohli, L. Ladicky, and P. Torr. Robust higher order potentials for enforcing label consistency. *International Journal of Computer Vision (IJCV)*, 82:302–324, May 2009.

[69] D. Koller and N. Friedman. *Probabilistic Graphical Models: Principles and Techniques*. The MIT Press, 2009.

[70] V. Kolmogorov. Convergent tree-reweighted message passing for energy minimization. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 28(10):1568–1583, October 2006.

[71] A. Korolova, K. Kenthapadi, N. Mishra, and A. Ntoulas. Releasing search queries and clicks privately. In *International World Wide Web Conference (WWW)*, 2009.

[72] A. Korolova, R. Motwani, S. U. Nabar, and Y. Xu. Link privacy in social networks. In *ACM Conference on Information and Knowledge Management (CIKM)*, 2008.

[73] J. Kubica, A. Moore, J. Schneider, and Y. Yang. Stochastic link and group detection. In *AAAI Conference on Artificial Intelligence (AAAI)*, 2002.

[74] J. M. Kubica, A. Moore, D. Cohn, and J. Schneider. cGraph: A Fast Graph-Based Method for Link Analysis and Queries. In *IJCAI Workshop on Text-Mining & Link-Analysis*, 2003.

[75] S. Lattanzi and D. Sivakumar. Affiliation networks. *ACM Symposium on Theory of Computing (STOC)*, June 2009.

[76] J. Leskovec, L. Backstrom, R. Kumar, and A. Tomkins. Microscopic evolution of social networks. *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, August 2008.

[77] J. Leskovec, J. Kleinberg, and C. Faloutsos. Graphs over time: densification laws, shrinking diameters and possible explanations. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, pages 177–187, 2005.

[78] K. Lewis, J. Kaufman, M. Gonzalez, A. Wimmer, and N. Christakis. Tastes, ties, and time: a new social network dataset using facebook.com. *Social Networks*, 30:330–342, 2008.

[79] N. Li, T. Li, and S. Venkatasubramanian. t-closeness: Privacy beyond k-anon. and l-diversity. In *IEEE International Conference on Data Engineering (ICDE)*, 2007.

[80] D. Liben-Nowell and J. Kleinberg. The Link Prediction Problem for Social Networks. In *Proceedings of the 12th International Conference on Information and Knowledge Management (CIKM)*, 2003.

[81] D. Liben-Nowell, J. Novak, R. Kumar, P. Raghavan, and A. Tomkins. Geographic routing in social networks. *Proceedings of the National Academy of Sciences (PNAS)*, 102(33):11623–11628, August 2005.

[82] J. Lindamood, R. Heatherly, M. Kantarcioglu, and B. Thuraisingham. Inferring private information using social network data. In *International World Wide Web Conference (WWW) Poster*, 2009.

[83] D. Liu, L. Lu, and H.-J. Zhang. Automatic mood detection from acoustic music data. In *International Symposium on Music Information Retrieval (ISMIR)*, 2003.

[84] H. Liu and L. Yu. Toward integrating feature selection algorithms for classification and clustering. *IEEE Transactions on Knowledge and Data Engineering (TKDE)*, 17(4):491–502, April 2005.

[85] 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 of Data Mining*, chapter 21, pages 419–437. Chapman & Hall/CRC, 2008.

[86] K. Liu and E. Terzi. Towards identity anonymization on graphs. In *ACM SIGMOD International Conference on Management of Data (SIGMOD)*, 2008.

[87] K. Liu and E. Terzi. A framework for computing the privacy scores of users in online social networks. In *IEEE International Conference on Data Mining (ICDM)*, 2009.

[88] L. Liu, J. Wang, J. Liu, and J. Zhang. Privacy preservation in social networks with sensitive edge weights. In *SIAM International Conference on Data Mining (SDM)*, 2009.

[89] Y. Liu, A. Niculescu-Mizil, and W. Gryc. Topic-link lda: joint models of topic and author community. In *International Conference on Machine Learning (ICML)*, 2009.

[90] H.-A. Loeliger. An introduction to factor graphs. In *IEEE Signal Processing Magazine*, pages 28–41, January 2004.

[91] L. Lü and T. Zhou. Link prediction in complex networks: A survey. *Physica A: Statistical Mechanics and its Applications*, 390(6):1150 – 1170, 2011.

[92] Q. Lu and L. Getoor. Link based classification. In *International Conference on Machine Learning (ICML)*, 2003.

[93] A. Machanavajjhala, J. Gehrke, D. Kifer, and M. Venkitasubramaniam. l-diversity: Privacy beyond k-anonymity. In *IEEE International Conference on Data Engineering (ICDE)*, 2006.

[94] A. Machanavajjhala, A. Korolova, and A. D. Sarma. Personalized social recommendations - accurate or private? In *International Conference on Very Large Databases (VLDB)*, 2011.

[95] S. Macskassy and F. Provost. Classification in networked data: A toolkit and a univariate case study. *Journal of Machine Learning Research (JMLR)*, 8:935–983, May 2007.

[96] C. L. Mallows. A note on asymptotic joint normality. *The Annals of Mathematical Statistics*, 43(2):508–515, 1972.

[97] B. Marlin. Modeling user rating profiles for collaborative filtering. In *Neural Information Processing Systems Conference (NIPS)*, 2003.

[98] B. Marlin and R. Zemel. Collaborative prediction and ranking with non-random missing data. In *ACM Conference on Recommender Systems (RecSys)*, 2009.

[99] A. McCallum, X. Wang, and A. Corrada-Emmanuel. Topic and role discovery in social networks with experiments on enron and academic email. *Journal of Artificial Intelligence Research (JAIR)*, 2007.

[100] A. McCallum, X. Wang, and N. Mohanty. Joint group and topic discovery from relations and text. In *Statistical Network Analysis: Models, Issues and New Directions*, volume 4503, pages 28–44. Lecture Notes in Computer Science, 2007.

[101] L. McDowell, K. Gupta, and D. Aha. Cautious inference in collective classification. In *AAAI Conference on Artificial Intelligence (AAAI)*, 2007.

[102] M. McGlohon, L. Akoglu, and C. Faloutsos. Weighted graphs and disconnected components. *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, August 2008.

[103] M. McGlohon, L. Akoglu, and C. Faloutsos. Statistical properties of social networks. In C. Aggarwal, editor, *Social Network Data Analytics*, pages 243–276. Springer, 1 edition, 2011.

[104] F. McSherry and I. Mironov. Differentially private recommender systems: building privacy into the netflix prize contenders. In *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, 2009.

[105] T. Minka. Divergence measures and message passing. Technical report, Microsoft Research, 2005.

[106] T. Minka, J. Winn, J. Guiver, and A. Kannan. Infer.NET 2.3, 2009. Microsoft Research Cambridge. http://research.microsoft.com/infernet.

[107] T. Minka and J. M. Winn. Gates. In *Neural Information Processing Systems Conference (NIPS)*, 2008.

[108] A. Mislove, M. Marcon, K. P. Gummadi, P. Druschel, and B. Bhattacharjee. Measurement and analysis of online social networks. *Internet Measurement Conference (IMC)*, October 2007.

[109] T. Mitchell. *Machine Learning*. McGraw-Hill, 1997.

[110] A. Narayanan and V. Shmatikov. Robust de-anonymization of large sparse datasets. *IEEE Symposium on Security and Privacy*, 2008.

[111] A. Narayanan and V. Shmatikov. De-anonymizing social networks. In *IEEE Symposium on Security and Privacy*, 2009.

[112] M. E. Nergiz, M. Atzori, and C. Clifton. Hiding the presence of individuals from shared databases. In *ACM SIGMOD International Conference on Management of Data (SIGMOD)*, 2007.

[113] M. E. Nergiz and C. Clifton. Thoughts on k-anonymization. In *ICDE Workshop on Privacy Data Management*, 2006.

[114] M. E. Nergiz and C. Clifton. Multirelational k-anonymity. In *IEEE International Conference on Data Engineering (ICDE)*, 2007.

[115] J. Neville and D. Jensen. Iterative classification in relational data. In *AAAI Workshop on Statistical Relational Learning*, 2000.

[116] J. Neville and D. Jensen. Leveraging relational autocorrelation with latent group models. In *IEEE International Conference on Data Mining (ICDM)*, 2005.

[117] M. Newman. Who is the best connected scientist? a study of scientific coauthorship networks. Working Papers 00-12-064, Santa Fe Institute, Dec. 2000. available at http://ideas.repec.org/p/wop/safiwp/00-12-064.html.

[118] M. Newman. Coauthorship networks and patterns of scientific collaboration. *Proceedings of the National Academy of Sciences (PNAS)*, 101:5200–5205, 2004.

[119] M. Newman, A.-L. Barabási, and D. J. Watts, editors. *The Structure and Dynamics of Networks*. Princeton University Press, 2006.

[120] M. E. J. Newman. The structure and function of complex networks. In *SIAM Review*, volume 45, pages 167–256. 2003.

[121] P. E. os and A. Rényi. On random graphs. *Publicationes Mathematicae*, 6:290–297, 1959.

[122] E. Pampalk, T. Pohle, and G. Widmer. Dynamic playlist generation based on skipping behavior. In *International Symposium on Music Information Retrieval (ISMIR)*, 2005.

[123] J. Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann Publishers, Inc., San Mateo, California, 1988.

[124] J. C. Platt, C. J. C. Burges, S. Swenson, C. Weare, and A. Zheng. Learning a gaussian process prior for automatically generating music playlists. In *Neural Information Processing Systems Conference (NIPS)*, 2001.

[125] A. Popescul and L. H. Ungar. Statistical relational learning for link prediction. In *IJCAI Workshop on Learning Statistical Models from Relational Data*, 2003.

[126] R. Ragno, C. J. C. Burges, and C. Herley. Inferring similarity between music objects with application to playlist generation. In *ACM SIGMM Workshop on Multimedia Information Retrieval (MIR)*, 2005.

[127] M. J. Rattigan and D. Jensen. The case for anomalous link discovery. *SIGKDD Explorations Newsletter*, 7(2):41–47, 2005.

[128] A. Réka and A.-L. Barabási. Statistical mechanics of complex networks. In *Reviews of Modern Physics*, volume 74, pages 47–97. January 2002.

[129] M. Rosen-Zvi, T. Griffiths, M. Steyvers, and P. Smyth. The author-topic model for authors and documents. In *Conference on Uncertainty in Artificial Intelligence (UAI)*, 2004.

[130] P. Samarati. Protecting respondents' identities in microdata release. *IEEE Transactions on Knowledge and Data Engineering (TKDE)*, 13(6):1010–1027, 2001.

[131] P. Sen, G. M. Namata, M. Bilgic, L. Getoor, B. Gallagher, and T. Eliassi-Rad. Collective classification in network data. *AI Magazine*, 29(3):93–106, 2008.

[132] T. Singliar and M. Hauskrecht. Noisy-or component analysis and its application to link analysis. *Journal of Machine Learning Research (JMLR)*, 7:2189–2213, 2006.

[133] M. Spiliopoulou. Evolution in social networks: A survey. In C. Aggarwal, editor, *Social Network Data Analytics*, pages 149–176. Springer, 1 edition, 2011.

[134] D. H. Stern, R. Herbrich, and T. Graepel. Matchbox: large scale online bayesian recommendations. In *International World Wide Web Conference (WWW)*, 2009.

[135] M. Steyvers and T. Griffiths. Probabilistic topic models. *Handbook of Latent Semantic Analysis*, 2007.

[136] L. Sweeney. Achieving k-anonymity privacy protection using generalization and suppression. *International Journal of Uncertainty*, 10(5):571–588, 2002.

[137] B. Taskar, P. Abbeel, and D. Koller. Discriminative probabilistic models for relational data. In *Conference on Uncertainty in Artificial Intelligence (UAI)*, 2002.

[138] B. Taskar, M.-F. Wong, P. Abbeel, and D. Koller. Link Prediction in Relational Data. In *Neural Information Processing Systems Conference (NIPS)*, 2003.

[139] I. Tsochantaridis, T. Hofmann, T. Joachims, and Y. Altun. Support vector learning for interdependent and structured output spaces. *International Conference on Machine Learning (ICML)*, 2004.

[140] J. Vaidya, C. Clifton, and Y. Zhu. *Privacy Preserving Data Mining*. Springer, 2006.

[141] N. Vuokko and E. Terzi. Reconstructing randomized social networks. In *SIAM International Conference on Data Mining (SDM)*, 2010.

[142] Y. Wang and G. Wong. Stochastic blockmodels for directed graphs. *Journal of the American Statistical Association (JASA)*, 1987.

[143] D. J. Watts and S. H. Strogatz. Collective dynamics of 'small-world' networks. *Nature*, 393:440–442, 1998.

[144] J. Winn and C. Bishop. Variational message passing. *Journal of Machine Learning Research (JMLR)*, 6:661–694, 2005.

[145] G. Wondracek, T. Holz, E. Kirda, and C. Kruegel. A practical attack to deanonymize social network users. In *IEEE Symposium on Security and Privacy*, 2010.

[146] L. Wu, X. Ying, and X. Wu. Reconstruction of randomized graph via low rank approximation. In *SIAM International Conference on Data Mining (SDM)*, 2010.

[147] X. Wu, X. Ying, K. Liu, and L. Chen. A survey of algorithms for privacy-preserving social network analysis. In C. Aggarwal and H. Wang, editors, *Managing and Mining Graph Data*. Kluwer Academic Publishers, 2009.

[148] X. Ying and X. Wu. Randomizing social networks: a spectrum preserving approach. In *SIAM International Conference on Data Mining (SDM)*, 2008.

[149] P. S. Yu, C. Faloutsos, and J. Han, editors. *Link Mining: Models, Algorithms and Applications*. Springer-Verlag, 2010.

[150] G.-X. Yuan, K.-W. Chang, C.-J. Hsieh, and C.-J. Lin. A comparison of optimization methods and software for large-scale l1-regularized linear classification. *Journal of Machine Learning Research (JMLR)*, 11:3183–3234, 2010.

[151] H. Zhang, B. Qiu, C. L. Giles, H. C. Foley, and J. Yen. An lda-based community structure discovery approach for large-scale social networks. In *IEEE International Conference on Intelligence and Security Informatics (ISI)*, 2007.

[152] E. Zheleva and L. Getoor. Preserving the privacy of sensitive relationships in graph data. *KDD Workshop on Privacy, Security, and Trust in KDD (PinKDD) 2007*, 4890:153–171, 2008.

[153] E. Zheleva and L. Getoor. To join or not to join: the illusion of privacy in social networks with mixed public and private user profiles. In *International World Wide Web Conference (WWW)*, 2009.

[154] E. Zheleva and L. Getoor. Privacy in social networks: A survey. In C. Aggarwal, editor, *Social Network Data Analytics*, pages 277–306. Springer, 1 edition, 2011.

[155] E. Zheleva, L. Getoor, J. Golbeck, and U. Kuter. Using friendship ties and family circles for link prediction. In *KDD Workshop on Social Network Mining and Analysis (SNA-KDD)*, 2008.

[156] E. Zheleva, L. Getoor, and S. Sarawagi. Higher-order graphical models for classification in social and affiliation networks. In *NIPS Workshop on Networks Across Disciplines: Theory and Applications*, 2010.

[157] E. Zheleva, J. Guiver, E. M. Rodrigues, and N. Milic-Frayling. Statistical models of music-listening sessions in social media. In *International World Wide Web Conference (WWW)*, 2010.

[158] E. Zheleva, H. Sharara, and L. Getoor. Co-evolution of social and affiliation networks. In *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, 2009.

[159] B. Zhou and J. Pei. Preserving privacy in social networks against neighborhood attacks. In *IEEE International Conference on Data Engineering (ICDE)*, 2008.

[160] B. Zhou, J. Pei, and W.-S. Luk. A brief survey on anonymization techniques for privacy preserving publishing of social network data. *SIGKDD Explorations Newsletter*, 10(2), 2009.

[161] D. Zhou, J. Li, and H. Zha. A new mallows distance based metric for comparing clusterings. In *International Conference on Machine Learning (ICML)*, 2005.

[162] L. Zou, L. Chen, and M. T. zsu. K-automorphism: A general framework for privacy preserving network publication. In *International Conference on Very Large Databases (VLDB)*, 2008.