Statistical Models of Music-listening Sessions in Social Media

Elena Zheleva
Dept. of Computer Science
Univ. of Maryland, College Park, USA
elena@cs.umd.edu

Eduarda Mendes Rodrigues
Dept. of Informatics Engineering
University of Porto, Portugal
eduardamr@acm.org

John Guiver
Microsoft Research Ltd.
Cambridge, UK
joguiver@microsoft.com

Nataša Milić-Frayling
Microsoft Research Ltd.
Cambridge, UK
natasamf@microsoft.com

ABSTRACT
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. In this paper, we show how to define statistical models of different complexity to describe patterns of song listening in an online music community. First, we adapt the LDA model 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 mood 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.

Categories and Subject Descriptors
H.2.8 [Information Systems]: Data Mining

General Terms
Algorithms, Experimentation

Keywords
social media, sessions, music, taste, mood, graphical models, recommendations, collaborative filtering

1. INTRODUCTION
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 paper, 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 would be 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 incorporate listening moods based on session information. By applying the models to half a million song-listening instances from the Zune Social 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 2 we give an overview of the related work and provide background on graphical models. We then discuss the social media context in Section 3. In Section 4 we describe the data and define the hierarchical graphical models. In Section 5 we present experimental results and then reflect on broader implications of our work in Section 6. We conclude with a summary of our contributions and directions for further work.

2. BACKGROUND

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 similar. Here we provide context for our work by discussing research related to our approach and provide background information with prerequisites for the models we explore.

2.1 Related work

2.1.1 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 would be 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. [4] attempt to detect mood of songs from their acoustical features such as tempo and articulation. Liu et al. [9] use intensity, timbre and rhythm instead. Hu & Downie [8] 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. [4] define four mood labels: happiness, sadness, anger, and fear for training a music classifier. Liu et al. [9] 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 [8] derive a set of five mood clusters from the All Music Guide mood repository to examine the correlation between music genre and mood and artist and mood.

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.

2.1.2 Song recommendations

Ragno et al. [20] 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 [20] 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., [18, 19]). In [18], 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 [19], Platt et al. learn a Gaussian Process kernel to predict user playlists using music metadata such as genre or style as input.

2.1.3 Statistical data modeling

Modeling collections of discrete data has been of growing interest for 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 [3, 7].

Recently, the Latent Dirichlet Model (LDA) [2] 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 4.1.

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. [23] 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 [5].

There are other generative models that combine topic modeling and social network modeling in a single framework [13, 14]. The Author-Role-Topic (ART) model, proposed by McCallum et al. [13], discovers discussion topics in threaded conversations, conditioned on sender-recipient interactions. The Group-Topic (GT) model [14] discovers latent groups in a network and clusters of associated topics based on text. The recent work on recommender systems by Stern et al. [21] 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. [11, 12] also use graphical models for the task of rating prediction. Hoffman et al. [6] propose a probabilistic model which uses audio features to predict song tags.

In our work we use hierarchical graphical models to 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.

2.2 Preliminary concepts

2.2.1 Graphical models and factor graphs

Factor graphs are a useful way of representing probabilistic graphical models. They consist of two types of nodes representing variables and factors, respectively. Figure 3 and Figure 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 explain 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 4 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 [1]) which represent replicated parts of the model, and gates [17] 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.

2.2.2 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 paper we make use of a message-passing algorithm for approximate inference called variational message passing (VMP) [22]. This is one of a class of algorithms that are given a unified treatment in [15].

These algorithms typically make use of a fully factorized approximation of 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.

<table>
<thead>
<tr>
<th>Genre</th>
<th>Sub-genre</th>
</tr>
</thead>
<tbody>
<tr>
<td>Blues/Folk</td>
<td>Baroque</td>
</tr>
<tr>
<td>Christian Gospel</td>
<td>Chamber</td>
</tr>
<tr>
<td>Classical</td>
<td>Choral</td>
</tr>
<tr>
<td>Comedy/Spoken Word</td>
<td>Classical Guitar</td>
</tr>
<tr>
<td>Country</td>
<td>Crossover</td>
</tr>
<tr>
<td>Electronic Dance</td>
<td>Early</td>
</tr>
<tr>
<td>Hip Hop</td>
<td>Opera</td>
</tr>
<tr>
<td>Jazz</td>
<td>Operettes</td>
</tr>
<tr>
<td>Kids</td>
<td>Other Classical</td>
</tr>
<tr>
<td>Latin</td>
<td>Religious</td>
</tr>
<tr>
<td>More</td>
<td>Renaissance</td>
</tr>
<tr>
<td>Pop</td>
<td>Romantic</td>
</tr>
<tr>
<td>R&amp;B</td>
<td></td>
</tr>
<tr>
<td>Reggae/Dancehall</td>
<td>Classic Rock</td>
</tr>
<tr>
<td>Rock</td>
<td>Indie/Modern Rock</td>
</tr>
<tr>
<td>Soundtracks</td>
<td>Metal</td>
</tr>
<tr>
<td>World</td>
<td>New Wave</td>
</tr>
<tr>
<td></td>
<td>Punk Ska</td>
</tr>
<tr>
<td></td>
<td>Rock And Roll</td>
</tr>
</tbody>
</table>

Figure 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.
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 and Figure 4) are two such building blocks, as are the message update equations to deal with plates and gates. Infer.NET [16], 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. SOCIAL MEDIA CONTEXT

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

3.1 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 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 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 instance $(u, m, t)$ 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 $S_u$ of each user includes a sequence of song-listening sessions $S_u = (S_1, ..., S_n) = (m_1, ..., m_N)$. 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 $\psi_i$ over the media items $M$.

Figure 2 shows an example of the data model. The table shows the log of two users $u_1$ and $u_2$ 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 $S3'$.
and switches off the others.

S each media cluster represents a particular taste. It is a direct mixture of clusters, song to as ‘tastes.’ Each taste media cluster is represented as a graphical model that represents consumption of media as the listening mood across song-listening sessions.

4. 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.

4.1 Taste model

Following the LDA model [2], 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 in the user’s playlist \( S_u \) by picking one of the media clusters \( c \), and then picking a song from that media cluster’s mixture \( \psi \). 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 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.

The following process describes the generation of a playlist \( S_u \) for each user \( u \):

1. For each media cluster \( k \)
   
   (a) Choose a distribution over songs, \( \psi_k \sim \text{Dir}(\beta) \)

2. For each user \( u \)
   
   (a) Choose a distribution over media clusters, \( \theta_u \sim \text{Dir}(\alpha) \)

   (b) For each song in the user’s playlist \( S_u \),

   i. Choose a media cluster \( c_{uij} \sim \text{Discrete}(\theta_u) \)

   ii. Observe song \( m_{uij} \sim \text{Discrete}(\psi(c_{uij})) \)

\( \text{Dir}(\alpha) \) is an exchangeable Dirichlet prior, i.e., all pseudo-counts are identical and given by the parameter \( \alpha \). \( \theta(u) \sim \text{Dir}(\alpha) \) is the parameter vector for a user-dependent discrete distribution over media clusters. \( \text{Dir}(\beta) \) is also an exchangeable Dirichlet prior and \( \psi(c) \sim \text{Dir}(\beta) \) is the parameter vector for a 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. [2]. According to this model, the joint probability distribution of the distributions \( \psi \) over songs, the distributions \( \theta \) over clusters, the cluster choice \( c \) for each user and song, and the songs in user \( u \)’s playlist \( S_u = (S_1, ..., S_t(u)) = (m_1, ..., m_N) \), is:

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

We then observe \((m_1, ..., m_N)\) and perform Bayesian inference to recover the posterior marginal distributions of \( \psi \) and \( \theta \).

4.2 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 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 \( \text{Dir}(\beta) \). The following process describes the generation of each user’s playlist \( S_u \):

1. For each media cluster \( k \)
   
   (a) Choose a distribution over songs, \( \psi_k \sim \text{Dir}(\beta) \)

2. For each user \( u \)
   
   (a) Choose a distribution over media clusters, \( \theta_u \sim \text{Dir}(\alpha) \)

   (b) For each session \( S_i \in S_u \)

   i. Choose a media cluster \( c_{ui} \sim \text{Discrete}(\theta_u) \)

   ii. For each song in the session, observe \( m_{uij} \sim \text{Discrete}(\psi(c_{ui})) \)

The joint distribution is:

\[
p(m, c, \psi, \theta | \alpha, \beta) = \prod_{u=1}^n p(\theta_u | \alpha) \prod_{i=1}^t p(c_{ui} | \theta_u) \prod_{j=1}^l p(m_{uij} | \psi(c_{ui})) \prod_{k=1}^K p(\psi_k | \beta).
\]

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 ses-
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.

5. Evaluation

We present results for the problem of playlist generation
and discuss the characteristics of the media clustering ap-
proach by visualizing the genres per cluster, comparing the
discovered latent clusters with the genre taxonomy, investi-
gating 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 listen-
ing 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.

5.1 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.

5.2 Inference

We implemented the statistical models using Infer.NET,
an efficient, general-purpose inference engine for graphical
models [16]. Since exact inference is not possible in the taste
and session models, we used variational message passing [22]
for learning the parameters of each model.

We fixed \( \beta = 0.5 \) and \( \alpha = \frac{1}{K} \). \( \beta \) was set to give the best
performance for the baseline test model (see Section 5.3.1),
and the same value was used for the taste and session mod-
els. The value of \( \alpha \) was set based on limited manual optimi-
ization with respect to the taste model and adopted for
the session model as well.

5.3 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.

5.3.1 Baseline test model

In the unigram model the genres in each music-listening
session are drawn independently from a single discrete dis-
tribution that describes the session. A factor graph of the
model is shown in Figure 5. More specifically, the generative
process is as follows:

1. For each session \( S_i \in S \)
   
   (a) Choose \( \psi_i \sim \text{Dir}(\beta) \)
   
   (b) For each song in the session,
      observe \( m_{ij} \sim \text{Discrete}(\psi_i) \)

Here, \( \text{Dir}(\beta) \) is an exchangeable Dirichlet prior and \( \psi \) is
the parameter vector for a Discrete distribution over songs.
During inference, it learns the distribution over genres based
Figure 7: Comparison of the perplexity of each model for session genres after observing a) 5 seed genres and b) 10 seed genres.

Figure 8: Session model perplexity for session genres after observing 5 or 10 seed genres.

5.3.2 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.

This model assumes that sessions are independent of each other and, unlike the taste and session models, it does not consider latent media clusters.

5.3.3 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 [2]. It captures how well a model trained on observed data would predict 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:

\[
    \text{Perplexity} = \exp \left( \frac{1}{n} \sum_{u=1}^{n} \sum_{S \in S_u} \frac{1}{\text{size}(S)} \ln(p(m_i|\psi(c_{ui}))) \right)
\]

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.

5.3.4 Results

Figure 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...
Figure 9: Resulting media clusters for the session model. Line thickness signifies cluster affiliation strength.

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.

5.4 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 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 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.

5.4.1 Comparing latent clusters with taxonomy

In Section 5.4 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.

5.4.2 Similarity metric

To compare two media clusterings, we employ a similarity metric based on the Mallows distance [10, 24]. 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. [24] 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 $C_2$:

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

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 total count for each Dirichlet and normalize across clusters to get the $\alpha_k$ and $\beta_j$. For the optimization part, we apply linear programming using Microsoft Solver Foundation³.

5.4.3 Cluster comparison results

Figure 10 shows that, as the number of clusters increases, the similarity between the genre clusters derived by the ses-

sion 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 Section 5.3.4.

5.5 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 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 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.

5.6 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 12 shows.

6. 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 friendships ties between users in the social media community.

From the perspective of the service architecture and optimization, clustering media content can contribute to im-
proved 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.

7. CONCLUSION

In our paper we presented a novel and improved statistical model for characterizing user preferences in consuming social media content. By taking into account information about the listening sessions of individual users, 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 essential 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 would require 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.

8. ACKNOWLEDGMENTS

The authors would like to thank Gavin Smyth for sharing his database expertise, Tom Minka and John Winn for discussions on the models, and Jordan Boyd-Graber for providing feedback on an earlier draft of the paper.

9. REFERENCES

Co-evolution of Social and Affiliation Networks

Elena Zheleva  
Dept. of Computer Science  
University of Maryland  
College Park, MD 20742, USA  
elena@cs.umd.edu

Hossam Sharara  
Dept. of Computer Science  
University of Maryland  
College Park, MD 20742, USA  
hossam@cs.umd.edu

Lise Getoor  
Dept. of Computer Science  
University of Maryland  
College Park, MD 20742, USA  
getoor@cs.umd.edu

ABSTRACT

In our work, we address the problem of modeling social network generation which explains both link and group formation. Recent studies on 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 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 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.

Categories and Subject Descriptors
H.2.4 [Mathematics of Computing]: General—Algorithm Analysis  
H.2.8 [Database Management]: Database Applications—Data Mining

General Terms  
Algorithms, Measurement, Experimentation

Keywords

evolution, social network, affiliation network, graph generation, groups

1. INTRODUCTION

In recent years, there has been a proliferation of online social networks. Many of the 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 our work, we 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, LiveJournal and YouTube, collected by Mislove et al. [9].

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.

**Notation.** We study the interactions of two graphs, the social network graph, \( G_s \), and the affiliation graph, \( G_a \). For our purposes, a social network is a graph \( G_s = (V, E_s) \) which has one type of node corresponding to the users that participate in it. Nodes can form links which can be directed or undirected; \( e_s(v_i, v_j, t) \) denotes the link that \( v_i \) and \( v_j \) have formed at time \( t \). A directed link is formed whenever one user links to another. An undirected link requires the approval of both parties in order to be formed.

In an affiliation network \( G_a = (V, H, E_a) \), there are two types of nodes, the social network users \( V \) and groups \( H \) that they have formed. We represent the network as a bipartite graph in which undirected links \( e_a(v_i, h_j, t) \) are formed between user \( v_i \) and group \( h_j \) 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.

2. RELATED WORK

The evolution of social and affiliation networks exhibits a number of properties previously studied in the literature. We describe some of them in more detail in Section 4.2.

2.1 Evolution of 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 [10] and the power-law degree distributions [2, 4]. Time-evolving graphs have also attracted attention recently, where interesting properties have been discovered, such as shrinking diameters, and densification power law [7].

There have been a number of models proposed to capture these properties. For a survey, one can consult the work by Chakrabarti and Faloutsos [3]. For example, unlike the random graph model, the preferential attachment model proposed by Barabasi et al. [2] captures power-law degree distributions. The forest fire model [7] also captures the power-law degree distribution together with densification and shrinking diameters over time. A more recently proposed, microscopic evolution model [6] 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 [8], concentrates on capturing the evolution of connected components in a graph. In our work, we extend the microscopic evolution model by including processes of forming and joining groups of interest.

2.2 Evolution of affiliation networks

To the best of our knowledge, there is no model that captures the evolution of affiliation networks in online communities. However, there are studies that describe the relationship between friendship links and group formation properties [1, 9]. They show that the probability of a user joining a group increases with the number of friends already in the group [1], and that higher degree nodes tend to belong to a higher number of groups [9].

Group detection is a related problem (for a survey, see [5]). 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 communities were formed.

3. 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. [9]. Now, we describe the observations that we discovered by analyzing the datasets, and we relate them to previously observed properties.

3.1 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 also been observed by Mislove et al. [9]. The results are illustrated in Figure 1.

3.2 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 relationship starts declining after a certain point, yielding lower number of group memberships for very high degree nodes. The relationship is illustrated in Figure 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.

3.3 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 3 reports on \(k = 50\) for LiveJournal and Flickr, and on \(k = 25\) for YouTube but this was true for other degrees as well.

3.4 Properties of group members

According to Backstrom et al. [1], 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), meaning that they did not join the group because of a friend.\(^1\)

\(^1\)At \url{http://www.mathwave.com}
4. 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 [6] to dynamic social and affiliation networks.

4.1 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 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 [7].
- **shrinking diameter** - the effective diameter of the network decreases as more nodes join the network [7].

Properties of the affiliation network. We would also like 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 3.2.
• exponential distribution of the number of group affiliations for a particular node degree - see Section 3.3.

4.3 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 [6] 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, 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, \( \lambda \) is the parameter of the exponential distribution of the lifetime, and \( \alpha, \beta \) 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. [6]. We utilize these parts:

Node arrival. New nodes \( V_{t+1} \) arrive at time \( t \) according to a pre-defined arrival process \( N(.) \).

Lifetime sampling. At arrival time \( t \), \( v \) samples lifetime \( a \) from \( \lambda e^{-\lambda a} \); \( v \) becomes inactive after time \( t_{end}(v) = t + a \).

Algorithm 1 Naïve model

1: Set of nodes \( V = \emptyset \)
2: for each time period \( t \in T \) do
3: Set of active nodes at time \( t \), \( V_t = \emptyset \)
4: end for
5: for each time period \( t \in T \) do
6: Node arrival. \( V = V \cup V_{t+1} \)
7: for each new node \( v \in V_{t+1} \) do
8: Lifetime sampling
9: First social linking
10: end for
11: for each node \( v \in V_t \) do
12: Social linking
13: end for
14: for each node \( v \in V_t \cup V_{t+1} \) do
15: Sleep time sampling
16: end for
17: end for
18: Set of groups \( H = \emptyset \).
19: for \( i = 1: \text{number of groups} \) do
20: Group creation. New group \( h_i \) is created and its size \( s \) is sampled from \( s^{-1} \). \( H = H \cup \{h_i\} \).
21: for \( j = 1:s \) do
22: Group joining. Pick a random node \( v \in V \) and form an affiliation link to it \( e_s(v, h_i, \text{null}) \).
23: end for
24: end for

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

Sleep time sampling. \( v \) decides on a discrete sleep time \( \delta \) by sampling from \( \frac{1}{\delta} (\delta^{-\alpha} e^{-\beta \text{degree}(v)} \delta) \). 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 \( V_{t+1} \) that will wake up at time \( t + \delta \).

Social linking. At wake up time \( t \), \( v \) creates an edge \( e_s(v, w, t) \) 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. [6]. Then, it generates and populates groups in such a way that their sizes follow a power-law distribution with an exponent \( k \). Algorithm 1 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 \( \tau \), 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 2 presents the co-evolution model in detail.

Algorithm 2 Co-evolution model

1: Set of nodes \( V = \emptyset \)
2: Set of groups \( H = \emptyset \)
3: for each time period \( t \in T \) do
4: Set of active nodes at time \( t \), \( V_t = \emptyset \)
5: end for
6: for each time period \( t \in T \) do
7: Node arrival. \( V = V \cup V_{t+1} \)
8: for each new node \( v \in V_{t+1} \) do
9: Lifetime sampling
10: First social linking
11: end for
12: for each node \( v \in V_t \) do
13: Social linking
14: Affiliation linking. \( v \) determines \( n_h \), the number of groups to join, sampled from an exponential distribution \( X e^{-X n_h} \) with a mean \( \mu' = \frac{1}{\lambda'} = \rho \text{degree}(v) \).
15: for \( i = 1 : n_h \) do
16: if \( \text{rand}() < \tau \) then
17: Group creation. \( v \) creates group \( h \), and forms edge \( e_s(v, h, t) \). \( H = H \cup \{h_i\} \).
18: else
19: Group joining. \( v \) forms edge \( e_a(v, h, t) \). The 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 probability proportional to the size of \( h \).
20: end if
21: end if
22: end for
23: for each node \( v \in V_t \cup V_{t+1} \) do
24: Sleep time sampling
25: end for
26: end for
Here, we present the parameters of the affiliation network evolution part in more detail. The first parameter, \( \rho \), represents a tuning parameter that controls the density of the affiliation links in the network. The second parameter, \( \gamma \), is the exponent of the power-law that relates node degree with number of group affiliations. The last parameter to our model, \( \tau \), 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 \( \rho \) 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, the relationship between node degree and average number of affiliations is a power-law relation. Even though one can vary the exponent \( \gamma \) 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 evolution 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). 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:

\[
p_v = \begin{cases} 
\eta \times \text{degree}(v) & \text{if } \eta \times \text{degree}(v) < 1 \\
1 & \text{otherwise}
\end{cases}
\]

though other functions of the degree may be more appropriate. The parameter \( \eta \) represents 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 \( \eta \). 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 \( \eta \) actually increases its chances of joining this larger size group of its friend, thus leading to the decreasing relationship noted in the observations.

5. 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.

5.1 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.

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. [6] 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, \( \gamma \), to 0.5.

![Figure 5: Degree distribution in a synthetic network](image-url)

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 5 shows. The network densification property also holds, as illustrated in Figure 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 affiliation network. We start with our first parameter \( \rho \), which represents a tun-
ing factor of the affiliation links’ density. The main
properties that are affected by varying the value of $\rho$ are the total
number of affiliations and the distribution between the node
degree and average number of group affiliations. As illustrated in Figure 7, we can note that the general power distribution persists among different values of $\rho$, but the main effect is the scale of the distribution; as increasing the value of $\rho$, 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 $\rho$ varies between 0 and 25. The total number of affiliation links for each value of $\rho$ is reported in Table 1.

<table>
<thead>
<tr>
<th>$\rho$</th>
<th>Affiliation Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>285,536</td>
</tr>
<tr>
<td>10</td>
<td>2,411,710</td>
</tr>
<tr>
<td>20</td>
<td>4,771,072</td>
</tr>
</tbody>
</table>

Table 1: Number of affiliation links with varying $\rho$

Our next parameter, $\tau$, represents the probability with
which a node creates 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 8, we
note that although the power law distribution of the group
sizes holds for various values of $\tau$ (which is one of the desired
properties), the maximum group size decreases significantly
with increasing the value of $\tau$. This decline in the maximum
group size is caused by the fact that for higher values of $\tau$,
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 2, where the resulting
number of groups in the network and the maximum group
size vary significantly with changing the parameter value.

<table>
<thead>
<tr>
<th>$\tau$</th>
<th>Groups Count</th>
<th>Max Group Size</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.1</td>
<td>66,887</td>
<td>39,753</td>
</tr>
<tr>
<td>0.5</td>
<td>245,143</td>
<td>560</td>
</tr>
<tr>
<td>0.9</td>
<td>332,437</td>
<td>32</td>
</tr>
</tbody>
</table>

Table 2: Number of groups with varying $\tau$

Finally, we investigate the parameter on which $p_v$ de-

pends, $\eta$. $\eta$ represents the extent to which friends influence
the decision of a node to join groups. The outcome of in-
creasing 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 9, we can
easily note that the general distribution captures the desired
properties and the observations in real data. The value of $\eta$
is highly dependent on the social network structure proper-
ties, 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.

5.2 Real data

In this set of experiments, we look for the model param-
eters that will produce a network similar to one of the real-

Figure 6: Densification in a synthetic network

Figure 7: Degree vs. average number of group affiliations with varying $\rho$. 

(a) $\rho = 3$

(b) $\rho = 10$

(c) $\rho = 20$
Table 3: Statistics of a real network (Flickr) vs. a synthetic one

| Real Network (Flickr) | Synthetic Network  
|------------------------|----------------------
| Number of users        | 1,846,198            |
| Number of groups       | 103,648              |
| Number of affiliations  | 8,529,435            |
| Average number of group affiliation per user | 4.62                 |
| Number of groups/Number of users | 0.0561               |

Figure 8: Group size distribution with varying $\tau$

Figure 9: Group size vs. member attributes with varying $\eta$ (dashed line: % ratio of singletons to group size, solid line: % ratio of maximum degree to group size).

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 $\tau$ 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 $\rho$. Finally, as observed in Figure 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 $\eta$. 

world datasets we have used in the observations of Section 3. 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. [6]. 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 3. A summary of the affiliation network statistics of Flickr is given in Table 3.

The Flickr dataset is characterized by a relatively small number of groups in comparison to the number of users,
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 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.

5.3 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 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 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.

6. CONCLUSION

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

Acknowledgements

The authors would like to thank the anonymous reviewers for their valuable feedback. This work was supported by NSF under Grant #0746930.

7. REFERENCES

To Join or Not to Join: The Illusion of Privacy in Social Networks with Mixed Public and Private User Profiles

Elena Zheleva
Department of Computer Science
University of Maryland, College Park
elena@cs.umd.edu

Lise Getoor
Department of Computer Science
University of Maryland, College Park
getoor@cs.umd.edu

ABSTRACT
In order to address privacy concerns, many social media websites allow users to hide their personal profiles from the public. In this work, 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 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.

Categories and Subject Descriptors
H.2.8 [Information Systems]: Data Mining

General Terms
Algorithms, Experimentation

Keywords
privacy, social networks, groups, attribute inference

1. INTRODUCTION
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 surprisingly 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 4. 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 illusory 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.

We motivate the problem in the next section. Then, we describe the data model in Section 3. Section 4 presents the privacy attacks, and Section 5 provides experimental results using these attacks. Section 6 presents related work, and Section 7 discusses the broader implications of our results.

2. MOTIVATION
Disclosing private information means violating the rights of people to control who can access their private informa-


**Figure 1:** Toy instance of the data model.

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.

**3. DATA MODEL**

We represent a social network as a graph $G = (V, E, H)$, where $V$ is a set of $n$ nodes of the same type, $E$ is a set of edges (the friendship links), and $H$ is a set of groups that nodes can belong to. $e_{i,j} \in E$ represents a directed link from node $v_i$ to node $v_j$. Our model handles undirected links by representing them as pairs of directed links. We describe a group as a hyper-edge $h \in H$ among all the nodes who belong to that group; $hU$ denotes the set of users who are connected through hyper-edge $h$ and $v.H$ denotes the groups that node $v$ belongs to. Similarly, $v.F$ is the set of nodes that $v$ has connected to: $v_i.F = \{v_j \mid e_{i,j} \in E\}$. A group can also have a set of properties $h.T$.

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 user profile has a unique id with which the user forms links and participates in groups. Each profile is associated with a sensitive attribute, either observed or hidden. 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 $V_o.A$, 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.

As a running example, we consider the social network presented in Figure 1. It describes a collection of individuals (Ana, Bob, Chris, Don, Emma, Fabio, and Gia), along with their friendship links and their groups of interest. 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). 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" group and the "Yucatan" group. 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.

**4. SENSITIVE-ATTRIBUTE INFEERENCE 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 interactions 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:

\[
v_s \hat{a}_M = \arg\max_{a_i} P_M(v_s, a = a_i; G).
\]

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 basic 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.

### 4.1 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:

\[
P_{\text{BASIC}}(v_s, a = a_i; G) = P(v_s, a = a_i|V_o.A) = \frac{|V_o.a_i|}{|V_o|},
\]

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 friendship information for inferring the attribute value.

### 4.2 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 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.

#### 4.2.1 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: \( \text{Bob}.F = \{\text{Ana},\text{Chris},\text{Emma},\text{Fabio}\} \). 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:

\[
P_{\text{AGG}}(v_s, a = a_i; G) = P(v_s, a = a_i|V_o.A, E) = \frac{|V_o.a_i|}{|V_o|},
\]

where \( V_o^\prime = \{v_o \in V_o | \exists (v_s, v_o) \in E\} \) and \( V_o.a_i = \{v_o \in V_o^\prime|v_o.a = a_i\} \).

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 1), Bob would pick 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.

#### 4.2.2 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. [21] 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 [21]. 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.

![Figure 2: Graphical representation of the models.](image)
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.

4.2.3 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.

4.2.4 Blockmodeling attack (BLOCK)

The next category of link-based methods we explored are approaches based on blockmodeling [24, 2]. 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, $\lambda_i$ is the vector of all link probabilities between block $B_i$ and each block $B_1, \ldots, B_m$. Similarly, let the probability of a link between a single user $v$ and a block $B_i$ be $\lambda(v)_i$, 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:

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

where $\text{sim}()$ 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 [17] when the weight of each directed edge is inversely proportional to the size of the class of the receiving node.

4.3 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.

4.3.1 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.

4.3.2 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, we would like to 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 [14]. 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 document 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:

$$\text{Entropy}(h) = -\sum_{a_i} p(a_i) \log_2 p(a_i)$$

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 1 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 attributes are known. Each node \(v\) is represented as a binary vector where each dimension corresponds to a unique group: \(\{\text{groupId} : \text{isMember}\}, v.a\). 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 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 \(H\) and the group attributes \(T\).

Algorithm 1 Group-based classification model

1: Set of relevant groups \(H_{\text{relevant}} = \emptyset\)
2: for each group \(h \in H\) do
3: \(\text{if isRelevant}(h)\) then
4: \(H_{\text{relevant}} = H_{\text{relevant}} \cup \{h\}\)
5: end if
6: end for
7: trainClassifier\((f, V.o, H_{\text{relevant}})\)
8: for each sensitive node \(v \in V.S\) do
9: \(v.a = f(v, H_{\text{relevant}})\)
10: end for

4.4 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.

5. EXPERIMENTS

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

5.1 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 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 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 [10]. 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 removed 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 (288), 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.

5.2 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 block-modeling attack (BLOCK), 6) the groupmate-link attack (CLIQUE), 7) the group-based classification attack (GROUP) and 8) the attack which uses both links and groups (LINK-GROUP). For the GROUP model, we present results on both the simpler version which considers all groups and the method in which relevant groups are selected. 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, "http://www.dogster.com", http://www.facebook.com, http://www.bibsonomy.org/


\(^2\)At http://code.google.com/apis/

\(^3\)At http://www.geonames.org/export/

\(^4\)At http://www.kde.cs.uni-kassel.de/ws/rsdc08/.
5.3 Sensitive-attribute inference results

Table 2: Attack accuracy assuming 50% private profiles. The successful attacks are shown in bold.

<table>
<thead>
<tr>
<th>Attack model</th>
<th>Flickr (gender)</th>
<th>Facebook (gender)</th>
<th>Facebook (polviews)</th>
<th>Dogster</th>
<th>BibSonomy</th>
</tr>
</thead>
<tbody>
<tr>
<td>BASIC</td>
<td>27.7%</td>
<td>50.0%</td>
<td>56.5%</td>
<td>28.6%</td>
<td>92.2%</td>
</tr>
<tr>
<td>Random guess</td>
<td>1.8%</td>
<td>50.0%</td>
<td>16.7%</td>
<td>14.3%</td>
<td>50%</td>
</tr>
<tr>
<td>BLOCK</td>
<td>8.8%</td>
<td>49.1%</td>
<td>6.1%</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>AGG</td>
<td>28.4%</td>
<td>50.2%</td>
<td>57.6%</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>CC</td>
<td>28.6%</td>
<td>50.4%</td>
<td>56.3%</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>LINK</td>
<td>56.5%</td>
<td>68.6%</td>
<td>58.1%</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>CLIQUE-LINK</td>
<td>46.3%</td>
<td>51.8%</td>
<td>57.1%</td>
<td>60.2%</td>
<td>-</td>
</tr>
<tr>
<td>GROUP</td>
<td>63.5%</td>
<td>73.4%</td>
<td>45.2%</td>
<td>65.5%</td>
<td>94.0%</td>
</tr>
<tr>
<td>GROUP (50% node coverage)</td>
<td>83.6%</td>
<td>77.2%</td>
<td>46.6%</td>
<td>82.0%</td>
<td>96.0%</td>
</tr>
<tr>
<td>LINK-GROUP</td>
<td>64.8%</td>
<td>72.5%</td>
<td>57.8%</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

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 [23].

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.

### 5.3.1 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, 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 inlinks 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 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 3(a) shows that naïvely running GROUP on all group
Figure 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.

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 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 insights. The most heterogeneous group that our method found is "worldwidewondering - 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 4 shows that if users with private profiles do not join low-entropy groups, then GROUP is no longer successful.

5.3.2 Facebook

We performed the same experiments for Facebook as for Flickr but we omit the figures due to space constraints. 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 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).

5.3.3 Dogster

**Link-based attacks.** Due to the fact that this was a random rather than a snowball sample, there were only 492 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 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.

5.3.4 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 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 would be caught. Some of the homogeneous tags with many taggers include “mortgage” and “refinance.”

6. RELATED WORK

To position our work, here, we present a brief overview of related work in privacy and learning in network data.

6.1 Privacy

According to Li et. al. [11], there are two types of privacy attacks in data: identity disclosure and attribute disclosure, and identity disclosure often leads to attribute disclosure. Identity disclosure occurs when the adversary is able to determine the mapping from a record to a specific real-world entity (e.g. an individual). Attribute disclosure occurs when an adversary is able to determine the value of a user at-
tribute that the user intended to stay private. We are interested in attribute disclosure in online social networks using the public profiles, friendship links and group memberships.

The privacy literature recognizes two types of privacy mechanisms: interactive and non-interactive [6]. In the interactive mechanism, an adversary poses queries to a data-base 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. Even though our work is closer to the non-interactive setting, the goal of our data provider is not to anonymize a dataset but to ensure that users’ private data remains private and cannot be inferred using links, groups and public profiles.

Until recently, the literature on anonymization considered only single-table data, in which the rows represent i.i.d. records, and the columns represent record attributes [1, 5, 11, 16, 22]. Real-world data is often relational, and records may be related to one another or to records from other tables. Relational data poses new challenges to preserving the privacy of individuals [3, 8, 15, 18, 19, 25]. For example, in graph data, there is a third type of disclosure attack: link re-identification [25]. Link re-identification is the problem of inferring that two entities participate in a particular type of sensitive relationship or communication. If one anonymizes the data naively by removing personal attributes and replacing them with a random identifier, it still is possible to identify individuals based on their subgraph structure [3, 8, 15]. It is also possible to link records in anonymized data to external relational data sources to disclose attribute values [18]. Our work is complementary in that we assume that the identities of people are known but the value of the sensitive attribute of some of them is not directly available.

We propose several simple models for inferring the hidden sensitive attributes using the observed attributes, link and group information in a single data source. It is important to be aware of the different possible privacy attacks in order to guide anonymization techniques.

He et al. [9] study the use of friendship links in predicting private attributes in a LiveJournal sample. They create synthetic attribute values in the sample, assuming autocorrelation, and show how to use a Bayesian network in predicting sensitive attributes. Lindamood et al. [13] provide another study on a large Facebook sample and show how sensitive attributes can be predicted using other user attributes and friendship links. In contrast, we consider a variety of attacks assuming a richer network structure with social groups, and posit that private-profile attributes are not available. We also test the attacks on four networks with real attributes, showing that autocorrelation is not as ubiquitous as expected.

6.2 Learning in network data

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 [7]. Link-based classification breaks the assumption that data comprises of i.i.d. instances and it can take advantage of autocorrelation, the property that makes the classes of linked objects 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 review of collective classification can be found in the work by Sen et al. [21].

The goal of unsupervised learning or clustering is to group objects together based on their similarity. In social networks, clusters can be found based on attribute and/or structural information. For example, Neville and Jensen [20] describe how autocorrelation in relational data is sometimes caused by the presence of such hidden clusters or groups in the data which influence the attributes of the group members. They use a spectral clustering method based on node links in the data to discover groups, and then use the groups to classify the nodes. Airoldi et al. [2] study mixed-membership clustering of relational data to predict protein function. It is assumed that the cluster assignment is related to the node attribute value in question.

In contrast to these approaches, we are interested in classifying nodes when group membership is explicitly given and only a subset of the groups is related to the node attribute in question. This is different from the case where groups need to be detected because explicit groups can represent a latent common interest that neither attribute nor structural information contains. We propose a relational classification method that makes use of groups with member-set overlaps, and it distinguishes groups that are relevant to classification based on group features such as size and homogeneity.

7. 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. [12] 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. [4] 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?

8. 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 this 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 network providers and help their users understand the potential for leaking information.

9. ACKNOWLEDGMENTS

The authors would like to thank Alan Mislove for providing the code for extracting data from Flickr, Jen Neville for a useful discussion on autocorrelation, Michael Hay for valuable feedback on an initial draft of this paper, and Galileo Namata for help with collective classification. This work was in part supported by NSF under Grant No.0746930.

10. REFERENCES