You Are What You Say: Privacy Risks of Public Mentions

Dan Frankowski, Dan Cosley, Shilad Sen, Loren Terveen, John Riedl
University of Minnesota, Department of Computer Science and Engineering
4-192 EE/CS Building, 200 Union Street SE, Minneapolis, MN 55455, USA
{dfrankow|cosley|ssen|terveen|riedl}@cs.umn.edu

ABSTRACT

In today's data-rich networked world, people express many aspects of their lives online. It is common to segregate different aspects in different places: you might write opinionated rants about movies in your blog under a pseudonym while participating in a forum or web site for scholarly discussion of medical ethics under your real name. However, it may be possible to link these separate identities, because the movies, journal articles, or authors you mention are from a *sparse relation space* whose properties (e.g., many items related to by only a few users) allow *re-identification*. This re-identification violates people's intentions to separate aspects of their life and can have negative consequences; it also may allow other privacy violations, such as obtaining a stronger identifier like name and address.

This paper examines this general problem in a specific setting: re-identification of users from a public web movie forum in a private movie ratings dataset. We present three major results. First, we develop algorithms that can re-identify a large proportion of public users in a sparse relation space. Second, we evaluate whether private dataset owners can protect user privacy by hiding data; we show that this requires extensive and undesirable changes to the dataset, making it impractical. Third, we evaluate two methods for users in a public forum to protect their own privacy, suppression and misdirection. Suppression doesn't work here either. However, we show that a simple misdirection strategy works well: mention a few popular items that you haven't rated.

Categories and Subject Descriptors

K.4.1 [**Computers and Society**]: Public Policy Issues---*privacy*. H.2.7 [**Database Management**]: Database Administration---*Security, integrity, and protection*.

General Terms

Algorithms, Experimentation, Security, Human Factors.

Keywords

k-anonymity, sparse relation space, re-identification, privacy, mentions, datasets, *k*-identification

1. INTRODUCTION

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.
SIGIR'06, August 6–11, 2006, Seattle, Washington, USA.
Copyright 2006 ACM 1-59593-369-7/06/0008...$5.00.

People are judged by their preferences, even for apparently trivial things. During Clarence Thomas' confirmation hearing for the U.S. Supreme Court, he was asked "Did you ever use the term Long Dong Silver in conversation with Professor Hill?", referring to allegations that he harassed Hill by referring to an actor in pornographic videos. If someone had shown that Thomas rented pornography, this could have derailed his nomination. However, Thomas' video rental history was legally private due to the misfortune of his predecessor Robert Bork. During his confirmation hearings in 1987, Bork's movie rental history was leaked. In response, lawmakers passed the 1988 Video Privacy Protection Act making it illegal for video tape service providers to disclose rental or sales information. People are judged by their movie preferences, and there is a belief (in the U.S., at least) that they should be private.

Yet many people reveal their preferences in public, talking about books, movies, songs, and other items on forums and in blogs. They may be revealing more than they think. Tom Owad downloaded 260,000 Amazon wish lists, chose several "dangerous" books and as a proof of concept found the complete address of one of four wish list owners by passing the wish list name, city, and state to Yahoo! PeopleSearch. Owad says, "There are many websites and databases that could be used for this project, but few things tell you as much about a person as the books he chooses to read[1]".

Many organizations keep people's preference, purchase, or usage data. These datasets usually are private to the organization that collects them. People may be comfortable if organizations they feel they have a relationship with collect their data. However, organizations have many reasons to share data:

- Research groups are encouraged to make datasets public (after suitably anonymizing them) to enable further research.
- Consortia may wish to pool or trade data for mutual benefit.
- Government agencies may be permitted or required to make data public (e.g., the census).
- Businesses may choose to sell their data to other businesses.
- Bankrupt businesses may be forced to sell data. For example, Toysmart proposed selling its customer profiles to the highest bidder, in violation of its stated privacy policy.

If these datasets include identifiable information, people's privacy may be violated. Even if obvious identifiers have been removed, they might accidentally contain a uniquely identifying *quasi-identifier*. For example, 87% of the 248 million people in the 1990 U.S. census are likely to be uniquely identified based only on their 5-digit ZIP, gender, and birth date [17].

When quasi-identifiers can be linked to other databases, private information may be disclosed. Sweeney found the medical records

[1] http://www.applefritter.com/bannedbooks

of a former governor of Massachusetts by linking public voter registration data (available for $20) to a database of supposedly anonymized medical records sold to industry [17].

1.1 Linking People in Sparse Relation Spaces

In this paper, we consider a related risk to privacy: re-identifying people by matching data in *sparse relation spaces*. A sparse relation space is a dataset that (a) *relates* people to items; (b) is *sparse*, having relatively few relationships recorded per person; and (c) involves a relatively large *space* of items. Examples of sparse relation spaces include customer purchase data from Target stores, music played on an online music player like iTunes, articles edited in Wikipedia, and books mentioned by bloggers on their public blogs. Sparse relation spaces differ from the databases studied in [17], which have a fixed number of columns (like zip code) and values present for most users.

The problem of re-identification is important for the information retrieval research community for two reasons. First, an increasing amount of data available electronically means re-identification is an increasingly likely application, to which IR techniques can be applied. Re-identification may prove valuable in identifying shills [9], and even fighting terrorism. Second, and less benign, re-identification is an application of IR technology that creates serious privacy risks for users. The IR community should lead the discussion about how user privacy may be preserved.

Privacy loss may occur whenever an agent has access to two sparse relation spaces with overlapping users and items. If there is no overlap, there is no risk. However, overlap is a real possibility as people's relations to items are increasingly available, whether explicitly revealed (forums, blogs, ratings) or implicitly collected (web logs, purchase history). Risks are most severe when one of the datasets is *identified*, i.e., has personally identifying data such as a social security number or a name and address. *Non-identified* datasets lack such data but can be used with an identified dataset to leak sensitive information. For example, in the case of the Massachusetts governor voter registration records are identified, while the medical records were not. When both datasets are non-identified, privacy risks are lower though privacy loss is still possible. For example, two pseudonyms belonging to a single person might be linked, or an e-mail address might be attached to preference information.

Privacy loss requires at least one of the datasets to be *accessible* outside the organization that collects them. *Inaccessible* datasets are held by a single owner, who can cause privacy loss by combining them with an accessible dataset. For example, Amazon might re-identify customers on competitors' websites by comparing their purchase history against reviews written on those sites, and decide to market (or withhold) special offers from them.

Our analysis of privacy loss is grounded in the concept of *k-anonymity* [17]. Sweeney says "A [dataset] release provides *k*-anonymity protection if the information for each person contained in the release cannot be distinguished from at least *k-1* individuals whose information also appears in the release."

Sweeney's definition of *k*-anonymity can be computed on a single dataset. Because we are interested in the problem of re-identifying users between two datasets, we define a related concept, *k*-identification. *k*-identification is a measure of how well an algorithm can narrow each user in a dataset to one of *k* users in another dataset. If *k* is large, or if *k* is small and the *k*-identification rate is low, users can plausibly deny being identified [8]. We will define *k*-identification more precisely later.

1.2 Research Questions

As described earlier, organizations often will wish to release datasets. We became interested in this problem when we wished to release a sparse relation space dataset of movie ratings and wondered about potential risks to users' privacy. We address three questions related to releasing data:

1. **RISKS OF DATASET RELEASE**: What are the risks to user privacy when releasing a dataset?
2. **ALTERING THE DATASET**: How can dataset owners alter the dataset they release to preserve user privacy?
3. **SELF DEFENSE**: How can users protect their own privacy?

After reviewing related work, we will describe our datasets and approach, then explore each question in turn.

2. RELATED WORK

There is a vast literature on privacy from many perspectives. Empirical studies (e.g., [1][18]) have shown that a large majority of internet users are concerned about their privacy. For our purposes, most relevant is work on algorithms to preserve user privacy in datasets that are to be made public.

Verykios et al. [20] survey recent privacy research in the data mining community and identified a number of ways to modify data to preserve privacy. Agrawal et al. [2] investigated one of these methods, *perturbing* attribute values by adding random noise while retaining statistical properties of the original data. Several researchers give methods for constructing association mining rules that minimize privacy disclosures [7][14]. Our work differs because we present techniques and results tailored to *sparse relation spaces* where the mere presence of an attribute can result in a privacy disclosure.

As discussed, Sweeney [17] introduced the concept of *k*-anonymity and also presented two techniques for preserving *k*-anonymity: *suppression* (hiding data), and *generalization* (reducing the fidelity of attribute values). Sweeney gave an algorithm for altering datasets to preserve anonymity while retaining a maximum amount of the original information [16]. Unfortunately the algorithm's computational requirements make it unusable for sparse relation spaces. We present techniques that can be used in real-world systems.

Recommender systems are a common source of sparse relation spaces in which researchers have explored privacy concerns. Polat et al. [12] and Berkovsky et al. [3] showed that perturbing ratings by adding random noise to rating values has only a small effect on recommender accuracy. Ramakrishnan et al. [13] used a graph-theoretic framework to show that *straddlers*, users with eclectic tastes, are more likely to be compromised. Canny described [4] an algorithm for securely building a factor-analytic recommendation model from ratings that users encrypted, a strong kind of generalization. We focus on suppression to protect privacy.

Finally, there is an abundance of research related to text mining of user comments. Drenner et al. described the system infrastructure and text-mining algorithms used in MovieLens to link forum posts to mentioned items [6]. Terveen et al. mined Usenet posts to find

recommended web pages [19]. Dave et al. [5] and Pang et al. [11] gave algorithms for mining users' opinions of an item from their reviews of the items. Both opinion mining authors extract a user's opinion from a textual review with 80% accuracy. Novak et al. [10] investigated re-identifying multiple aliases of a user in a forum based on general properties of their post text. Simulations we present later suggest that marrying our algorithms to opinion mining methods will improve their ability to re-identify people.

3. EXPERIMENTAL SETUP

We conduct several offline experiments using two sparse relation spaces gathered from the MovieLens movie recommender: a set of movie ratings and a set of movie mentions. These datasets were generated from a snapshot of the system on January 17, 2006.

The movie rating dataset contains 12,565,530 ratings (on a scale from 0.5 to 5 stars), 140,132 users, and 8,957 items. The ratings data roughly follows a power law. The most-rated item has 48,730 ratings; the mean number of ratings is 1,403, and the median is 207. This is a typical and important feature of real world sparse relation spaces. The distribution between users and ratings also follows a power law: the user with the most ratings has 6,280, the mean number of ratings is 90, and the median is 33.

The movie mention dataset is drawn from posts in the MovieLens forums. Users can make *movie references* while posting, which are used to integrate MovieLens features into the forums (see Figure 1). Users can explicitly insert references when they mention a movie, or they can be automatically inserted by a movie finding algorithm presented in [6]. The algorithm has few false positives (precision of 0.93) and links most references (recall of 0.78). The dataset contains 3,828 movie references, made by 133 forum posters about 1,685 different movies.

Figure 1: Movie references in a forum post. There are two references: The Life Aquatic, and Finding Neverland.

Like ratings, mentions follow a power law. Table 1 shows the number of users with a given number of mentions, where the mentions are binned exponentially. The bins contain similar numbers of users and have intuitive meaning. Further, we hypothesize that identifying a user depends on the number of mentions: users with more mentions disclose more information. We therefore use this binning strategy in our analyses below.

Table 1. Counts of users with a given number of mentions.

Mentions	1	2-3	4-7	8-15	16-31	32-63	64+
Users	25	21	23	22	18	13	11

4. RQ1: RISKS OF DATASET RELEASE

We now turn to our research questions. Suppose we are a dataset owner and we wish to anonymize our private dataset and release it publicly. *What are the risks to user privacy when releasing a dataset?* Releasing the dataset poses a risk of privacy violation to our users if they can be re-identified. In this section, we explore algorithms that re-identify users from one sparse relation space, movie mentions, in another sparse relation space, movie ratings.

We start by describing how we evaluate the algorithms. Given the movie mentions of a target user t in the forums, we try to re-identify t as a ratings user u from the ratings dataset. A re-identification algorithm is given the movies mentioned by t and returns a *likely list* of ratings users in order of their likelihood of being t. If t is not in the list, then t is not k-identified for any k. Otherwise, let j be t's rank in the list. In the case of ties involving t, let j be the highest rank among the tied users. We say that t is *k-identified* for $k>=j$. For example, if t is third in the list, then t is 3-identified, 4-identified, etc. The *k-identification rate* of the algorithm is the fraction of k-identified users.

For each algorithm, we choose all 133 users with at least one mention as a target user and count how many are k-identified at our chosen evaluation levels of k =1, 5, 10, and 100. These levels allow us to explore performance across a wide range of privacy protection. 1-identification seems most interesting, but for purposes like marketing 10-identification would be useful.

In joining ratings with mentions, we use a key insight: people tend to talk about things they know about—in this case, movies they've rated. In our data, users have on average rated 82% of the movies they mention. Thus, a user u who has rated many of the movies mentioned by t is more likely to be t than one who has not.

In the following subsections we describe successively more refined algorithms that exploit this property: Set Intersection (4.1), TF-IDF (4.2), and Scoring (4.3). We then hypothetically investigate performance if we knew the rating associated with a mention (4.4). We conclude with a discussion of our results (4.5).

4.1 Set Intersection Algorithm

We first examine a naïve re-identification algorithm: *Set Intersection* (Figure 2). We find users in the ratings dataset who are related to every item mentioned by the target user. The likely list contains only users in the ratings database who have rated every movie mention of the given target user t, and considers them equally likely to be t. We ignore the rating value entirely.

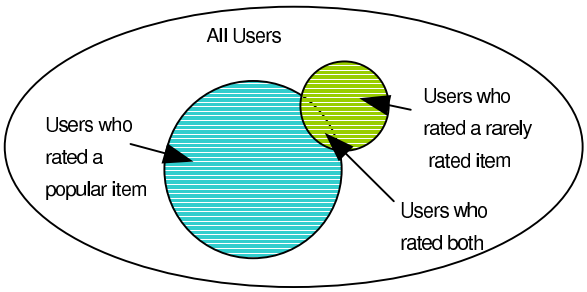

Figure 2: The Set Intersection algorithm finds users who rated all items mentioned. Mentioning a rarely-rated item is more identifying (the circle is smaller).

Set Intersection had k-identification rates of 7%, 12%, 14%, and 23% for k=1, 5, 10, and 100, respectively. In order to understand the behavior of the algorithm, we investigated when it failed, discovering three reasons. "Not narrow" means there were more than k users possible (if we were trying to k-identify). "No one possible" means there were no users who had rated every mention. "Misdirected" means the algorithm found users, but the target user was not among them.

Figure 3 is a stacked graph comparing reasons users are not being 1-identified, by number of mentions. The black region at the bottom ("1-identified") means the user was successfully re-identified. It shows a sweet spot: users are re-identified if they have neither too few nor too many mentions (other values of k show similar sweet spots). Other regions indicate users that were not successfully re-identified. Since only a few mentions may not be very distinguishing, "not narrow" is most probable for a small number of mentions. Since the probability of mentioning an unrated item goes up with many mentions, "no one possible" is the most probable for a large number of mentions. Overall, 20% of users are misdirected, while 31% are not possible.

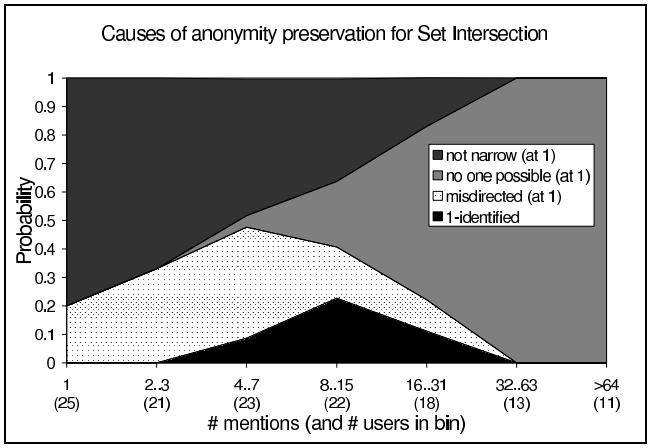

Figure 3: Set Intersection's behavior at 1-identification. "Not narrow" is the main cause of failure for users with few mentions, "No one possible" for those with many mentions.

The set intersection algorithm performs poorly because people sometimes mention movies they haven't rated, and mentioning at least one unrated movie becomes more likely as they mention more movies. Reasons for this include:

- Movies users have not seen yet but want to talk about.
- Mislinking by our movie finding algorithm.
- Movies users have seen but have not rated.

While the fundamental idea seems sound, we need to soften the requirement that users rate every movie they mention.

4.2 TF-IDF Algorithm

To find users who have mentioned items they haven't rated, we consider a slightly more complicated algorithm. Given a target user t in the public mentions dataset, we compute a score for every ratings user u in the private dataset. Users with a higher score are more likely to be t.

We want our scoring function to have the following properties:

- Users who have rated more mentions score higher.
- Users who rate rare movies that are mentioned score higher than users who rate common mentioned movies.

The standard TF-IDF (term frequency-inverse document frequency) algorithm has these properties. TF-IDF calculates a score indicating the relevance of a word in a document based on how often it appears in the document and in the whole corpus. In our case, we map items to words and users to documents. Note that we are not using TF-IDF for text mining. It is simply a way to present queries in a sparse vector space to documents also in a sparse vector space.

For user u and item m, we compute weight w_{um} as follows:

$$w_{um} = tf_{um} \log_2 \frac{|U|}{|u \in U \text{ who rated } m|}$$

where U is the set of all users and tf_{um} denotes the term frequency for the (user, movie) pair. For forum users, tf_{um} is 1 if the user mentioned m and 0 otherwise. For ratings users, tf_{um} is 1 if the user rated m and 0 otherwise. For each user, we create a $\mathbf{w_u}$ vector composed of the weights for all movies. Finally, we measure the similarity between a target forum user t and rating user u using cosine similarity:

$$sim(t,u) = \frac{\mathbf{W}_t \cdot \mathbf{W}_u}{\|\mathbf{W}_t\| \|\mathbf{W}_u\|}$$

This algorithm had k-identification rates of 20%, 32%, 35%, and 50% for k=1, 5, 10, and 100, respectively. This is much better than Set Intersection (7%, 12%, 14%, 23%, respectively), but still had room for improvement. It over-weighted any mention for a ratings user who had rated few movies; this user would then have high scores, perhaps too high.

4.3 Scoring Algorithm

Our next algorithm, Scoring, emphasizes mentions of rarely-rated movies strongly and de-emphasizes the number of ratings a user has. For simplicity, we assume a score is separable. We compute a *sub-score* for every item mentioned by the target user, then multiply the sub-scores to create an overall score for each user.

Equation 1 shows the sub-score computation, given a ratings user u and target user t, supposing that t has mentioned a movie m, and U is the set of all users.

Equation 1: Sub-score for user u, movie mention m.

$$ss(u,m) = \begin{cases} 1 - \dfrac{|\{u' \in U \text{ who rated } m\}| - 1}{|U|} & \text{if } u \text{ rated } m \\ 0.05 & \text{otherwise} \end{cases}$$

If u has rated mention m, then the sub-score is the fraction of users who have not rated m. We do not attach a strong interpretation to this value; it simply gives more weight to rarely rated movies. Subtracting one from the numerator keeps the sub-score greater than zero even if all users have rated m.

If u has not rated m, then the sub-score is a low value in order to penalize u. We arbitrarily chose 0.05, which is lower than any value for a rated mention because no item has been rated by 95% of all users. In essence, a user who has not rated a given mention is considered 10-20 times less likely to be t than another user who has, depending on the popularity of the mentioned movie.

We combine our sub-scores into a single score by multiplication across T, the set of all item mentions of the target user.

Equation 2: Scoring score for user u and target user t.

$$s(u,t) = \prod_{m_i \in T} ss(u, m_i)$$

The ratings user u with the highest score is the most likely to be the target user t.

For example, suppose a target user *t* has mentioned movies A, B, and C, there are 10,000 users in our dataset, and A, B, and C have been rated 20, 500, and 1000 times, respectively. Suppose we are scoring a user u_1 who has rated item A, and user u_2 who has rated B and C. User u_1's score is 0.9981 * 0.05 * 0.05=0.0025, and user u_2's score is 0.05 * 0.9501 * 0.9001=0.043. In this case, we judge u_2 more likely to be *t*. We can see that rating a mention is a strong positive indication, and rating a rare mention even more so.

As a final adjustment, we do not consider users who have rated more than one third (2953) of the movies because users who have rated almost every item will look similar to any target user. This eliminated 12 users, 0.01% of the users in the ratings dataset.

The Scoring algorithm had *k*-identification rates of 31%, 44%, 52%, and 57% for *k*=1, 5, 10, and 100, outperforming TF-IDF (20%, 32%, 35%, and 50%, respectively). Including the heaviest-rating users did reduce 1-identification performance (to 22%), but had little impact at higher *k*. We also tried the algorithm without weighting rated mentions by popularity, instead assigning a flat sub-score of 1 for rated mentions. Again, this did worse at *k*=1 (20%) but had little effect at higher *k*.

4.4 Scoring Algorithm with Ratings

Our previous algorithms ignored the rating values of users in the private dataset. Suppose we could extract not only mentions from users' posts, but also rating values. For example a magic text analyzer might read the text surrounding a movie mention in our forums and guess what the person would have rated the movie.

Happily, we can see how much such an analyzer might help before building it. Since we know the ratings of the target user *t*, we simply use those ratings in place of the hypothetical analyzer. We tried two variations. In the ExactRating version, we assume our analyzer perfectly determines each mention's exact rating value. In the FuzzyRating version, we assume our analyzer can guess a user's rating value to within ±1 star.

We use the mined rating value to restrict the Scoring algorithm. That is, if a user has rated an item, but not sufficiently closely in rating value, we treat that as a non-rating event. This is shown in Equation 3. The function *r(u,i)* is the rating value of user *u* for item *i*, and *r(t,i)* is the rating value of the target user *t* for item *i*.

Equation 3: Modified sub-scoring function for ExactRating (*δ=0*) and FuzzyRating (δ=*1*).

$$ss(u,m)=\begin{cases}1-\dfrac{|\{u'\in U \text{ who rated } m\}|-1}{|U|} & \text{if } u \text{ rated } m \text{ and } |r(u,m)-r(t,m)|\le\delta \\ 0.05 & \text{otherwise}\end{cases}$$

The FuzzyRating algorithm had *k*-identification rates of 47%, 50%, 53%, and 68%, and the ExactRating algorithm had *k*-identification rates of 59%, 68%, 72%, and 81% for *k*=1, 5, 10, and 100, respectively. This compares to Scoring's 31%, 44%, 52%, and 57%. Knowing the actual rating helps, even if we are off by one star, 20% of our ratings scale.

4.5 Discussion

Figure 4 summarizes how the algorithms performed across various levels of *k*-identification. We found that Set Identification performed poorly because it could not handle the case where people mentioned movies they had not rated. This led us to develop two algorithms that handle unrated mentions: TF-IDF and Scoring. Both algorithms give credit to users who have rated a mentioned item. Scoring directly penalizes people who did not rate a mentioned item, while TF-IDF directly penalizes people who rate unmentioned items. In our domain, Scoring outperforms the others at all levels of *k*, 1-identifying almost a third of all users who mention one or more movies and, as Figure 5 shows, 1-identifies 60% of users who mention at least eight movies. We use Scoring as our *k*-identification algorithm for the rest of the paper.

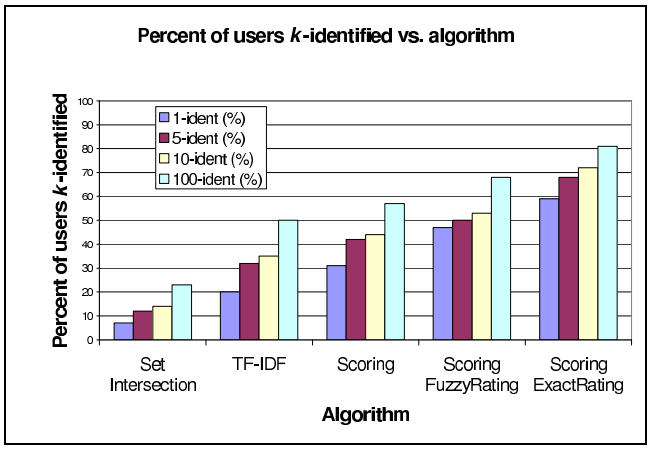

Figure 4: Percent of users *k*-identified by different algorithms for *k*=1, 5, 10, and 100.

Figure 4 also shows that knowing how well someone likes a mentioned item can help. Knowing the rating associated with a mention to within one star improves 1-identification performance by 50% while knowing the exact rating would double it. If one dataset has ratings data, implementing text mining techniques (e.g., [5][11]) to augment the Scoring algorithm would increase its power. Still, Scoring by itself is often effective.

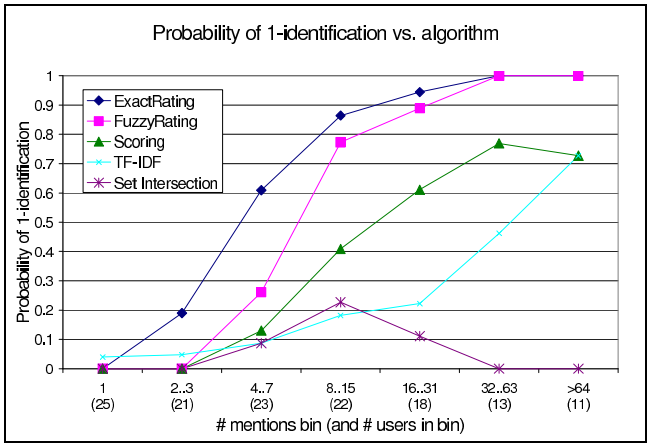

Figure 5: 1-identification rates of Set Intersection, TF-IDF, Scoring, Scoring FuzzyRating, and Scoring ExactRating, binned by number of mentions.

We believe the Scoring algorithm can be used in many contexts. It assumes that if a user is present in two sparse relation spaces, they will tend to have relations to the same items in both spaces, e.g., users mention movies they rate. This seems plausible in many domains. Further, it assumes that rarely-rated items are more revealing. This was true in our data and is likely to be true in most contexts where the number of relations for a given item follows a

power law. One assumption we make that is not as nice is that we know the target user is in the ratings dataset, which may not be true in general (for example, not everyone who has an Amazon wish list is in Yahoo! People Search). Determining whether a user is in both datasets is a hard problem that we leave for future work.

Often mentions that are somewhat identifying are also highly identifying. For example, when a user is 100-identified by Scoring, then 91% of the time that user is also 10-identified, and 54% of the time also 1-identified.

Finally, Figure 5 shows how each algorithm did at 1-identifying people based on number of mentions. In general, more mentions reveal more information about a user. Figure 4 showed that if an algorithm beats another, it does so at all levels of *k*; Figure 5 shows this is also true for the amount of information given to the algorithm. However, as noted before, Set Identification doesn't work if users mention movies they have not rated. In Section 6.2, we explore how people can use this behavior to protect their privacy against the better-performing Scoring algorithm.

5. RQ2: ALTERING THE DATASET

We've shown that there is a risk of privacy violation through re-identification. Therefore, we ask: *How can dataset owners alter the dataset they release to preserve user privacy?*

We considered perturbation, generalization, and suppression as strategies to preserve anonymity. Because the Scoring algorithm ignores ratings, perturbing a rating and weak generalization methods such as mapping ratings to binary or unary scales will have limited effectiveness. Stronger generalization, such as revealing only underlying factors obtained through factor analysis, forces dataset consumers to use the same model as the dataset provider and is not appropriate for applications such as testing recommendation algorithms that require using the relations themselves. Therefore, we focused on suppressing data.

5.1 Data Suppression

Since our successful algorithms rely on rarely-rated movies, we chose to drop these movies. Since we don't know which movies users might mention, we experimented with dropping all ratings of items that have fewer ratings than a specified threshold. As the proportion of *k*-identified users decreases, so do privacy risks.

Figure 6 shows the *k*-identification rate for Scoring after dropping all items below a specified ratings threshold. The X axis is the fraction of items that have been dropped. The Y axis is the percent of users who are *k*-identified for k=1, 5, 10, 100. Figure 6 shows that you have to throw away a lot of items to protect people (88% for k=1), because there are many rarely-rated items. However, throwing away items does not throw away proportionally many ratings. Figure 7 shows the fraction of *ratings* omitted. Throwing away 28% of ratings reduces the 1-identification rate to zero.

5.2 Discussion

It appears dataset owners have to suppress many items to protect users from 1-identification. For many purposes, such as research, throwing away a large fraction of items may be harmful. For example, we released two datasets based on MovieLens ratings that have been used for a wide variety of algorithmic research. We could not have predicted all types of algorithms that have been tested against the datasets. Any way of suppressing ratings would likely affect some algorithms differently from others.

For other purposes such as marketing, suppression may not be so bad. Companies that are primarily interested in selling or analyzing popular items will not need the information provided by rarely-rated items. On the other hand, a rare book seller might find a dataset stripped of unpopular items to be of little value. Datasets released at regular intervals for the purpose of trend analysis would also be adversely affected by suppressing rarely rated items, making it hard to detect emerging fads.

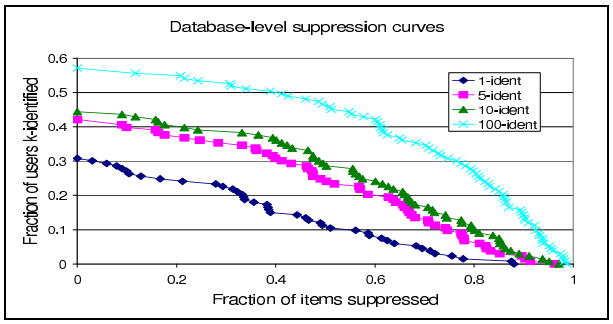

Figure 6: Database-level suppression results. To protect current forum users who mention items from being 1-identified, you have to suppress 88% of all items.

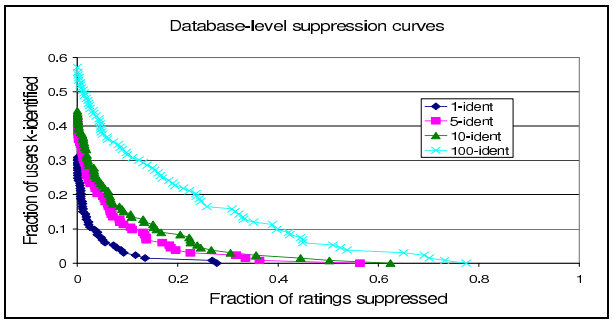

Figure 7: Database-level suppression results. X axis shows fraction of ratings suppressed. To protect current forum users from being 1-identified, you have to suppress 28% of ratings.

6. RQ3: SELF DEFENSE

It appears that suppression is not a very satisfactory strategy for dataset owners who wish to protect user privacy. However, users often have some control over their own information, especially over whether they express it in public. Therefore, we ask: *How can users protect their own privacy?*

6.1 Suppression

Since the Scoring algorithm uses rare movies to identify users, we first chose to try suppressing rarely-rated movies.

Note that the results from dataset-level suppression (section 5.1) hold here. That is, if users choose not to mention movies rated fewer than some threshold number of times, they would be severely restricted in the movies they could mention. Figure 6 says that they would not want to mention at least 88% of the items in order to guarantee no one could be 1-identified.

However, maybe each user would only have to drop a few mentions. Figure 8 shows the *k*-identification rate of the Scoring algorithm after a user suppresses a certain fraction of their mentions. We suppose that they suppress cleverly by lining up the items they have both mentioned and rated in order of how many

times the item has been rated (rarest first), and suppress only the top portion of the list.

The X axis of Figure 8 is the fraction of items (per user) that have been suppressed. We believe it is unreasonable to ask a user to suppress many mentions, so we show only fractions from 0 to 0.5. The Y axis is the *k*-identification rate for k=1, 5, 10, 100. Figure 8 shows that if users suppress some small fraction of their mentions (say, 10 or 20 percent), it often does not help. This is not too surprising, given our previous results at dataset-level suppression.

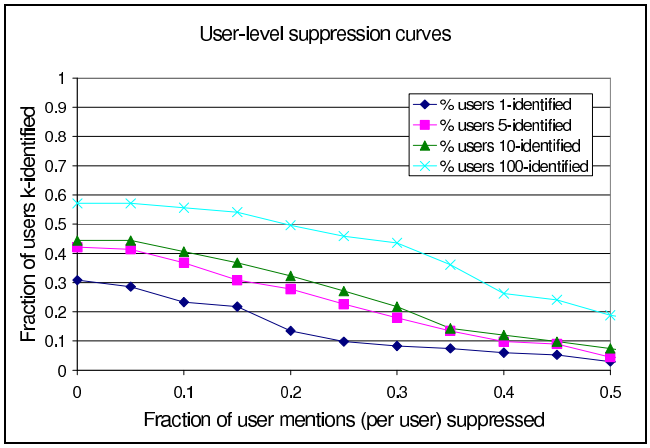

Figure 8: Percent of users *k*-identified by Scoring if each user hides their least popular mentions.

6.2 Misdirection

Since suppressing mentions of identifying items didn't work, we considered the opposite: what if users mention items they have not rated? This might *misdirect* the algorithm to identify one or more other users first. (We discuss the ethics of this below.) While misdirection is related to dataset perturbation [3][12], to the best of our knowledge our approach is novel. Previous work modified the dataset as a whole, while we propose strategies for individuals to change their public behavior.

We measure the effects of misdirection as follows. We create a sorted *misdirecting item list* (see below). Each user takes the first item from the list that they have not rated or mentioned and mentions it. We then measure the new *k*-identification levels. The user then adds another misdirecting mention, we re-compute *k*-identification, and so on.

We use two strategies to create the misdirecting item list. In one, we choose items that have been rated more times than some threshold, in order of increasing popularity. We vary the threshold from 1 to 8192 in powers of 2. In the other strategy, we select from all items in decreasing order of popularity.

Figure 9 shows the results (some thresholds omitted for visual clarity). Mentioning zero misdirecting items gives us 31%, the Scoring value from Figure 4. Mentioning the rarest items with just 1 rating has no effect. Mentioning items with at least 8,192 ratings, which are fairly popular items in this dataset, is effective. Mentioning popular items is most effective, dropping the 1-identification rate from 31% to 13% after 5 mentions. However, even mentioning 20 unrated items leaves some people vulnerable.

For a user, the goal of misdirecting is to make other users in the database more likely to match your mentions. This explains why mentioning popular items is more effective. Using our data and the Scoring algorithm, users in the ratings database are over 10 times more likely to be the target user if they rate even a very popular mentioned item, compared to users who do not mention the item. By mentioning a popular item, thousands of users increase their score. Mentioning a rare item, on the other hand, increases the score of only a few other users.

Therefore, while rarely-rated items are identifying, popular items are misdirecting.

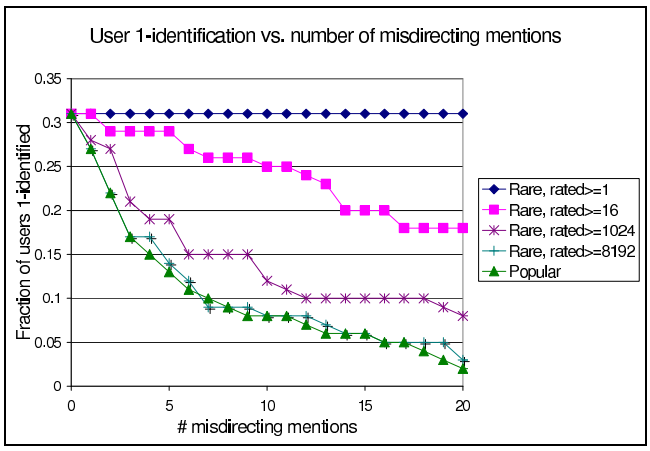

Figure 9: 1-identification after adding misdirecting mentions. More popular items are better for preserving privacy.

6.3 Discussion

We tried suppressing mentions at a user level. Suppressing a small fraction of mentions did not help much. Suppressing a large fraction seems unlikely in many situations (e.g., forums or blogs).

We tried mentioning unrated items in order to misdirect. Popular items helped to misdirect, although 1-identification rates did not drop to zero. Mentioning popular items a user has not rated is a viable strategy for that user to regain some anonymity in a sparse relation space. It is not hard to say something about a popular item or one that you have seen but not rated. A user might even un-rate popular movies after mentioning them. This seems more artificial, however; for example, MovieLens might recommend a movie that had been un-rated.

The fact that the best misdirecting items are popular increases the viability of the misdirection, since one is more likely to be able to think of something to say about a popular item, even if unseen. Knowing this, one might design a re-identification algorithm that ignores the most popular mentions entirely, though this might be challenging if mentions are precious information as they are in our dataset.

Intelligent interfaces might help people manage their privacy. For example, consider a posting privacy advisor in forum software that knew your ratings. We surmise that users would not be willing to be interrupted or advised very often. If true, being advised to suppress more than a few percent of mentions is not good. Even if such an advisor could improve anonymity at a fairly low number of mentions, would people be willing to mention things suggested by a computer? Would they find such a user interface scary, distracting, confusing, or ultimately ignorable (as "Click I Agree to continue" is for many of us)?

Misdirection raises further questions. Convincing users to misdirect seems sneaky if it points to just a few other users (imagine misdirecting pornographic movie ratings to someone else). However, if it were well-known that people engage in misdirection, it might preserve plausible deniability for all. Finally, suppose users were successfully convinced to misdirect, either by directions or a posting advisor. How would that change the nature of public discourse in our sparse relation spaces?

7. CONCLUSION

Being re-identified using items in a sparse relation space can violate privacy: the items themselves might leave one vulnerable to judgment, or they might be used to get at an identifier or quasi-identifier to get information one wishes to keep private.

We found a substantial re-identification risk to users who mention items in a sparse relation space, and we demonstrated successful algorithms to re-identify. In other words, *relationships to items in a sparse relation space can be a quasi-identifier*. This is a serious problem, as there are many such spaces, from purchase relations to books on Amazon to friendship relations to people on MySpace. We explored whether dataset owners can protect user privacy by suppressing ratings and found it required impractically extensive changes to the dataset. We also investigated ways users can protect their own privacy. As with dataset owners, suppression seems impractical--it is hard for users to mitigate risk by not mentioning things.

User-level misdirection (mentioning items not rated) provides some anonymity at a fairly low cost. We also found that although rare items are identifying, popular items are more useful for misdirection. This leaves misdirection as a possible privacy protection strategy, possibly with an advisor. However, there are questions about whether users would accept such a strategy or such an advisor.

We see many fruitful lines for further inquiry:

- Estimate the potential for user re-identification on the web, for example by text mining blogs or forums.
- Build recommender systems based only on public mentions.
- Create new re-identification algorithms, for example using temporal information from ratings, more information from posts, and more theoretically motivated algorithms
- Invent mathematical models and theoretical predictions for *k*-identification in a power-law sparse relation space.
- Investigate more sophisticated suppression algorithms
- Investigate other sparse relation spaces. For example, advise paper reviewers when the citations they have chosen would enable the author to identify them.

The above ideas would further explore the privacy risks we all face as data becomes easier to find, to get, and to combine. When people talk about items in public, they probably reveal more than they think. We have only begun to understand both how much they reveal, and how they can cloak themselves should they wish.

8. ACKNOWLEDGEMENTS

This work was supported by grants IIS 03-24851 and IIS 05-34420 from the National Science Foundation. Thanks also to Sean McNee, Tony Lam, and the rest of GroupLens.

9. REFERENCES

[1] Ackerman, M. S., Cranor, L. F., and Reagle, J. 1999. Privacy in e-commerce: examining user scenarios and privacy preferences. In *Proc. EC99*, pp. 1-8.

[2] Agrawal, R. and Srikant, R. 2000. Privacy-preserving data mining. In *Proc. SIGMOD00*, pp. 439-450.

[3] Berkovsky, S., Eytani, Y., Kuflik, T., and Ricci, R. 2005. Privacy-Enhanced Collaborative Filtering. In *Proc. User Modeling Workshop on Privacy-Enhanced Personalization.*

[4] Canny, J. 2002. Collaborative filtering with privacy via factor analysis. In *Proc. SIGIR02*, pp. 238-245.

[5] Dave, K., Lawrence, S., and Pennock, D. M. 2003. Mining the peanut gallery: opinion extraction and semantic classification of product reviews. In *Proc. WWW03*, pp. 519-528.

[6] Drenner, S., Harper, M., Frankowski, D., Terveen, L., and Riedl, J. 2006. Insert Movie Reference Here: A System to Bridge Conversation and Item-Oriented Web Sites. Accepted for *Proc. CHI06*.

[7] Evfimievski, A., Srikant, R., Agrawal, R., and Gehrke, J. 2002. Privacy preserving mining of association rules. In *Proc. KDD02*, pp. 217-228.

[8] Hong, J.I. and J.A. Landay. An Architecture for Privacy-Sensitive Ubiquitous Computing. In *Mobisys04* pp. 177-189.

[9] Lam, S.K. and Riedl, J. 2004. Shilling recommender systems for fun and profit. In *Proc. WWW04*, pp. 393-402.

[10] Novak, J., Raghavan, P., and Tomkins, A. 2004. Anti-Aliasing on the Web. In *Proc. WWW04*, pp. 30-39.

[11] Pang, B., Lee, L., and Vaithyanathan, S. 2002. Thumbs up? Sentiment classification using machine learning techniques. In *Proc. Empirical Methods in NLP*, pp. 79-86.

[12] Polat, H., Du, W. 2003. Privacy-Preserving Collaborative Filtering Using Randomized Perturbation Techniques. In *Proc. ICDM03*, p. 625.

[13] Ramakrishnan, N., Keller, B. J., Mirza, B. J., Grama, A. Y., and Karypis, G. 2001. Privacy Risks in Recommender Systems. *IEEE Internet Computing* 5(6):54-62.

[14] Rizvi, S., and Haritsa, J. 2002. Maintaining Privacy in Association Rule Mining. In *Proc. VLDB02*, pp. 682-693.

[15] Sarwar, B. M., Karypis, G., Konstan, J.A., and Riedl, J. 2001. Item-based collaborative filtering recommendation algorithms. In *Proc. WWW01*.

[16] Sweeney, L. 2002. Achieving *k*-anonymity privacy protection using generalization and suppression. *Int. J. Uncertain. Fuzziness Knowl.-Based Syst.* 10(5):571-588.

[17] Sweeney, L. 2002. *k*-anonymity: a model for protecting privacy. *Int. J. Uncertain. Fuzziness Knowl.-Based Syst.* 10(5): 557-570.

[18] Taylor, H. 2003. Most People Are "Privacy Pragmatists." The Harris Poll #17. *Harris Interactive (March 19, 2003).*

[19] Terveen, L., et al. 1997. PHOAKS: a system for sharing recommendations. *CACM* 40(3):59-62.

[20] Verykios, V. S., et al. 2004. State-of-the-art in privacy preserving data mining. *SIGMOD Rec.* 33(1):50-57.