Robust De-anonymization of Large Sparse Datasets

Arvind Narayanan and Vitaly Shmatikov
The University of Texas at Austin

Abstract

We present a new class of statistical de-anonymization attacks against high-dimensional micro-data, such as individual preferences, recommendations, transaction records and so on. Our techniques are robust to perturbation in the data and tolerate some mistakes in the adversary's background knowledge.

We apply our de-anonymization methodology to the Netflix Prize dataset, which contains anonymous movie ratings of 500,000 subscribers of Netflix, the world's largest online movie rental service. We demonstrate that an adversary who knows only a little bit about an individual subscriber can easily identify this subscriber's record in the dataset. Using the Internet Movie Database as the source of background knowledge, we successfully identified the Netflix records of known users, uncovering their apparent political preferences and other potentially sensitive information.

1 Introduction

Datasets containing *micro-data*, that is, information about specific individuals, are increasingly becoming public in response to "open government" laws and to support data mining research. Some datasets include legally protected information such as health histories; others contain individual preferences and transactions, which many people may view as private or sensitive.

Privacy risks of publishing micro-data are well-known. Even if identifiers such as names and Social Security numbers have been removed, the adversary can use background knowledge and cross-correlation with other databases to re-identify individual data records. Famous attacks include de-anonymization of a Massachusetts hospital discharge database by joining it with a public voter database [25] and privacy breaches caused by (ostensibly anonymized) AOL search data [16].

Micro-data are characterized by high dimensionality and sparsity. Each record contains many attributes (*i.e.*, columns in a database schema), which can be viewed as dimensions. Sparsity means that for the average record, there are no "similar" records in the multi-dimensional space defined by the attributes. This sparsity is empirically well-established [7, 4, 19] and related to the "fat tail" phenomenon: individual transaction and preference records tend to include statistically rare attributes.

Our contributions. Our first contribution is a formal model for privacy breaches in anonymized micro-data (section 3). We present two definitions, one based on the probability of successful de-anonymization, the other on the amount of information recovered about the target. Unlike previous work [25], we do not assume *a priori* that the adversary's knowledge is limited to a fixed set of "quasi-identifier" attributes. Our model thus encompasses a much broader class of de-anonymization attacks than simple cross-database correlation.

Our second contribution is a very general class of de-anonymization algorithms, demonstrating the fundamental limits of privacy in public micro-data (section 4). Under very mild assumptions about the distribution from which the records are drawn, the adversary with a small amount of background knowledge about an individual can use it to identify, with high probability, this individual's record in the anonymized dataset and to learn all anonymously released information about him or her, including sensitive attributes. For *sparse* datasets, such as most real-world datasets of individual transactions, preferences, and recommendations, very little background knowledge is needed (as few as 5-10 attributes in our case study). Our de-anonymization algorithm is *robust* to the imprecision of the adversary's background knowledge and to perturbation that may have been applied to the data prior to release. It works even if only a *subset* of the original dataset has been published.

Our third contribution is a practical analysis of the Netflix Prize dataset, containing anonymized movie ratings of 500,000 Netflix subscribers (section 5). Netflix—the world's largest online DVD rental service—published this dataset to support the Netflix Prize data mining contest. We demonstrate that an adversary who knows a little bit about some subscriber can easily identify her record if it is present in the dataset, or, at the very least, identify a small set of records which include the subscriber's record. The adversary's background knowledge need not be precise, *e.g.*, the dates may only be known to the adversary with a 14-day error, the ratings may be known only approximately, and some of the ratings and dates may even be completely wrong. Because our algorithm is robust, if it uniquely identifies a record in the published dataset, with high probability this identification is not a false positive.

2 Related work

Unlike statistical databases [1, 3, 5], micro-data include actual records of individuals even after anonymization. A popular approach to micro-data privacy is k-anonymity [27, 9]. The data publisher decides in advance which of the attributes may be available to the adversary (these are called "quasi-identifiers"), and which are the sensitive attributes to be protected. k-anonymization ensures that each quasi-identifier tuple occurs in at least k records in the anonymized database. This does not guarantee any privacy, because the values of sensitive attributes associated with a given quasi-identifier may not be sufficiently diverse [20, 21] or the adversary may know more than just the quasi-identifiers [20]. Furthermore, k-anonymization completely fails on high-dimensional datasets [2], such as the Netflix Prize dataset and most real-world datasets of individual recommendations and purchases.

The de-anonymization algorithm presented in this paper does not assume that the attributes are divided *a priori* into quasi-identifiers and sensitive attributes. Examples include anonymized transaction records (if the adversary knows a few of the individual's purchases, can he learn *all* of her purchases?), recommendations and ratings (if the adversary knows a few movies that the individual watched, can he learn *all* movies she watched?), Web browsing and search histories, and so on. In such datasets, it is hard to tell in advance which attributes might be available to the adversary; the adversary's background knowledge may even vary from individual to individual. Unlike [25, 22, 14], our algorithm is *robust*. It works even if the published records have been perturbed, if only a subset of the original dataset has been published, and if there are mistakes in the adversary's background knowledge.

Our definition of privacy breach is somewhat similar to that of Chawla *et al.* [8]. We discuss the differences in section 3. There is theoretical evidence that for any (sanitized) database with meaningful utility, there is *always* some auxiliary or background information that results in a privacy breach [11]. In this paper, we aim to quantify the amount of auxiliary information required and its relationship to the percentage of records which would experience a significant privacy loss.

We are aware of only one previous paper that considered privacy of movie ratings. In collaboration with the MovieLens recommendation service, Frankowski *et al.* correlated public mentions of movies in the MovieLens discussion forum with the users' movie rating histories in the internal MovieLens dataset [14]. The algorithm uses the entire public record as the background knowledge (29 ratings per user, on average), and is not robust if this knowledge is imprecise, *e.g.*, if the user publicly mentioned movies which he did not rate.

While our algorithm follows the same basic scoring paradigm as [14], our scoring function is more complex and our selection criterion is nontrivial and an important innovation in its own right. Furthermore, our case study is based solely on public data and does *not* involve cross-correlating internal Netflix datasets (to which we do not have access) with public forums. It requires much less background knowledge (2-8 ratings per user), which need not be precise. Furthermore, our analysis has privacy implications for 500,000 Netflix subscribers whose records have been published; by contrast, the largest public MovieLens datasets contains only 6,000 records.

3 Model

Database. Define database $\mathcal{D}$ to be an $N \times M$ matrix where each row is a record associated with some individual, and the columns are attributes. We are interested in databases containing individual preferences or transactions. The number of columns thus reflects the total number of items in the space we are considering, ranging from a few thousand for movies to millions for (say) the `amazon.com` catalog.

Each attribute (column) can be thought of as a dimension, and each individual record as a point in the multidimensional attribute space. To keep our analysis general, we will not fix the space X from which attributes are drawn. They may be boolean (*e.g.*, has this book been rated?), integer (*e.g.*, the book's rating on a 1-10 scale), date, or a tuple such as a (rating, date) pair.

A typical reason to publish anonymized micro-data is "collaborative filtering," *i.e.*, predicting a consumer's future choices from his past behavior using the knowledge of what similar consumers did. Technically, the goal is to predict the value of some attributes using a combination of other attributes. This is used in shopping recommender systems, aggressive caching in Web browsers, and other applications [28].

Sparsity and similarity. Preference databases with thousands of attributes are necessarily *sparse*, *i.e.*, each individual record contains values only for a small fraction of attributes. For example, the shopping history of even the most profligate Amazon shopper contains only a tiny fraction of all available items. We call these attributes *non-null*; the set of non-null attributes is the *support* of a record (denoted $\mathsf{supp}(r)$). Null attributes are denoted $\perp$. The support of a column is defined analogously. Even though points corresponding to database records are very sparse in the attribute space, each record may have dozens or hundreds of non-null attributes, making the database truly high-dimensional.

The distribution of per-attribute support sizes is typically heavy- or *long-tailed*, roughly following the power law [7, 4]. This means that although the supports of the columns corresponding to "unpopular" items are small, these items are so numerous that they make up the bulk of the non-null entries in the database. Thus, any attempt to approximate the database by projecting it down to the most common columns is bound to failure.[1]

Unlike "quasi-identifiers" [27, 9], there are no attributes that can be used directly for de-anonymization. In a large database, for any except the rarest attributes, there are hundreds of records with the same value of this attribute. Therefore, it is *not* a quasi-identifier. At the same time, knowledge that a particular individual has a certain attribute value does reveal *some* information, since attribute values and even the mere fact that a given attribute is non-null vary from record to record.

The similarity measure Sim is a function that maps a pair of attributes (or more generally, a pair of records) to the interval $[0, 1]$. It captures the intuitive notion of two values being "similar." Typically, Sim on attributes will behave like an indicator function. For example, in our analysis of the Netflix Prize dataset, Sim outputs 1 on a pair of movies rated by different subscribers if and only if both the ratings and the dates are within a certain threshold of each other; it outputs 0 otherwise.

To define Sim over two records r_1, r_2, we "generalize" the cosine similarity measure:

$$\mathsf{Sim}(r_1, r_2) = \frac{\sum \mathsf{Sim}(r_{1i}, r_{2i})}{|\mathsf{supp}(r_1) \cup \mathsf{supp}(r_2)|}$$

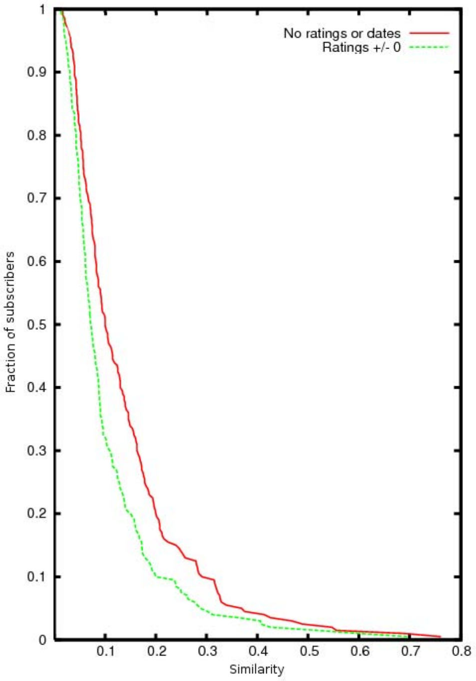

Figure 1. X-axis (x) is the similarity to the "neighbor" with the highest similarity score; Y-axis is the fraction of subscribers whose nearest-neighbor similarity is at least x.

Definition 1 (Sparsity) *A database D is (ϵ, δ)-sparse w.r.t. the similarity measure Sim if*

$$\Pr_r[\mathsf{Sim}(r, r') > \epsilon\ \forall r' \neq r] \leq \delta$$

As a real-world example, in fig. 1 we show that the Netflix Prize dataset is overwhelmingly sparse. For the vast majority of records, there isn't a *single* record with similarity score over 0.5 in the entire 500,000-record dataset, even if we consider only the sets of movies rated without taking into account numerical ratings or dates.

Sanitization and sampling. Database sanitization methods include generalization and suppression [26, 9], as well as perturbation. The data publisher may only release a (possibly non-uniform) sample of the database. Our algorithm is designed to work against data that have been both anonymized and sanitized.

If the database is published for collaborative filtering or similar data mining purposes (as in the case of the Netflix Prize dataset), the "error" introduced by sanitization *cannot* be large, otherwise data utility will be

[1] The same effect causes k-anonymization to fail on high-dimensional databases [2].

lost. We make this precise in our analysis. Our definition of privacy breach allows the adversary to identify not just his target record, but *any* record as long as it is sufficiently similar (via Sim) to the target and can thus be used to determine its attributes with high probability.

From the viewpoint of our de-anonymization algorithm, there is no difference between the perturbation of the published records and the imprecision of the adversary's knowledge about his target. In either case, there is a small discrepancy between the attribute value(s) in the anonymous record and the same value(s) as known to the adversary. In the rest of the paper, we treat perturbation simply as imprecision of the adversary's knowledge. The algorithm is designed to be robust to the latter.

Adversary model. We sample record r randomly from database D and give *auxiliary information* or background knowledge related to r to the adversary. It is restricted to a subset of (possibly imprecise, perturbed, or simply incorrect) values of r's attributes, modeled as an arbitrary probabilistic function $\mathsf{Aux}\colon X^M \to X^M$. The attributes given to the adversary may be chosen uniformly from the support of r, or according to some other rule.[2] Given this auxiliary information and an anonymized sample $\hat{D}$ of D, the adversary's goal is to reconstruct attribute values of the entire record r. Note that there is no artificial distinction between quasi-identifiers and sensitive attributes.

If the published records are sanitized by adding random noise Z_S, and the noise used in generating Aux is Z_A, then the adversary's task is equivalent to the scenario where the data are not perturbed but noise Z_S+Z_A is used in generating Aux. This makes perturbation equivalent to imprecision of Aux.

Privacy breach: formal definitions. What does it mean to de-anonymize a record r? The naive answer is to find the "right" anonymized record in the public sample $\hat{D}$. This is hard to capture formally, however, because it requires assumptions about the data publishing process (*e.g.*, what if $\hat{D}$ contains two copies of every original record?). Fundamentally, the adversary's objective is is to learn as much as he can about r's attributes that he doesn't already know. We give two different (but related) formal definitions, because there are two distinct scenarios for privacy breaches in large databases.

The first scenario is automated large-scale de-anonymization. For every record r about which he has some information, the adversary must produce a single "prediction" for all attributes of r. An example is the attack that inspired k-anonymity [25]: taking the demographic data from a voter database as auxiliary information, the adversary joins it with the anonymized hospital discharge database and uses the resulting combination to determine the values of medical attributes for each person who appears in both databases.

Definition 2 *A database D can be (θ, ω)-deanonymized w.r.t. auxiliary information Aux if there exists an algorithm A which, on inputs D and $\mathsf{Aux}(r)$ where $r \leftarrow D$ outputs r' such that*

$$\Pr[\mathsf{Sim}(r, r') \geq \theta] \geq \omega$$

Definition 2 can be interpreted as an *amplification of background knowledge*: the adversary starts with $\mathsf{aux} = \mathsf{Aux}(r)$ which is close to r on a small subset of attributes, and uses this to compute r' which is close to r on the entire set of attributes. This captures the **adversary's ability to gain information about his target record**. As long he finds *some* record which is guaranteed to be very similar to the target record, *i.e.*, contains the same or similar attribute values, privacy breach has occurred.

If operating on a sample $\hat{D}$, the de-anonymization algorithm must also detect whether the target record is part of the sample, or has not been released at all. In the following, the probability is taken over the randomness of the sampling of r from $\hat{D}$, Aux and A itself.

Definition 3 (De-anonymization) *An arbitrary subset $\hat{D}$ of a database D can be (θ, ω)-deanonymized w.r.t. auxiliary information Aux if there exists an algorithm A which, on inputs $\hat{D}$ and $\mathsf{Aux}(r)$ where $r \leftarrow D$*

- *If $r \in \hat{D}$, outputs r' s.t. $\Pr[\mathsf{Sim}(r, r') \geq \theta] \geq \omega$*
- *if $r \notin \hat{D}$, outputs $\perp$ with probability at least ω*

The same error threshold $(1 - \omega)$ is used for both false positives and false negatives because the parameters of the algorithm can be adjusted so that both rates are equal; this is the "equal error rate."

In the second privacy breach scenario, the adversary produces a set or "lineup" of candidate records that include his target record r, either because there is not enough auxiliary information to identify r in the lineup or because he expects to perform additional analysis to complete de-anonymization. This is similar to communication anonymity in mix networks [24].

The *number* of candidate records is not a good metric, because some of the records may be much likelier candidates than others. Instead, we consider the probability distribution over the candidate records, and use

[2] For example, in the Netflix Prize case study we also pick uniformly from among the attributes whose supports are below a certain threshold, *e.g.*, movies that are outside the most popular 100 or 500 movies.

as the metric the conditional *entropy* of r given aux. In the absence of an "oracle" to identify the target record r in the lineup, the entropy of the distribution itself can be used as a metric [24, 10]. If the adversary has such an "oracle" (this is a technical device used to measure the adversary's success; in the real world, the adversary may not have an oracle telling him whether de-anonymization succeeded), then privacy breach can be quantified as follows: *how many bits of additional information does the adversary need in order to output a record which is similar to his target record?*

Thus, suppose that after executing the de-anonymization algorithm, the adversary outputs records $r'_1, \ldots r'_k$ and the corresponding probabilities $p_1, \ldots p_k$. The latter can be viewed as an *entropy encoding* of the candidate records. According to Shannon's source coding theorem, the optimal code length for record r'_i is $(-\log p_i)$. We denote by $H_S(\Pi, x)$ this Shannon entropy of a record x w.r.t. a probability distribution Π. In the following, the expectation is taken over the coin tosses of A, the sampling of r and Aux.

Definition 4 (Entropic de-anonymization) *A database D can be (θ, H)-deanonymized w.r.t. auxiliary information Aux if there exists an algorithm A which, on inputs D and $\mathsf{Aux}(r)$ where $r \leftarrow D$ outputs a set of candidate records D' and probability distribution Π such that*

$$E[\mathit{min}_{r' \in D', \mathsf{Sim}(r,r') \geq \theta} H_S(\Pi, r')] \leq H$$

This definition measures the minimum Shannon entropy of the candidate set of records which are similar to the target record. As we will show, in sparse databases this set is likely to contain a single record, thus taking the minimum is but a syntactic requirement.

When the minimum is taken over an empty set, we define it to be $H_0 = \log_2 N$, the *a priori* entropy of the target record. This models outputting a random record from the entire database when the adversary cannot compute a lineup of plausible candidates. Formally, the adversary's algorithm A can be converted into an algorithm A', which outputs the mean of two distributions: one is the output of A, the other is the uniform distribution over D. Observe that for A', the minimum is always taken over a non-empty set, and the expectation for A' differs from that for A by at most 1 bit.

Chawla *et al.* [8] give a definition of privacy breach via *isolation* which is similar to ours, but requires a metric on attributes, whereas our general similarity measure does not naturally lead to a metric (there is no feasible way to derive a distance function from it that satisfies the triangle inequality). This appears to be essential for achieving robustness to completely erroneous attributes in the adversary's auxiliary information.

4 De-anonymization algorithm

We start by describing an algorithm template or meta-algorithm. The inputs are a sample $\hat{D}$ of database D and auxiliary information $\mathsf{aux} = Aux(r), r \leftarrow D$. The output is either a record $r' \in \hat{D}$, or a set of candidate records and a probability distribution over those records (following Definitions 3 and 4, respectively).

The three main components of the algorithm are the scoring function, matching criterion, and record selection. The **scoring function** Score assigns a numerical score to each record in $\hat{D}$ based on how well it matches the adversary's auxiliary information Aux. The **matching criterion** is the algorithm applied by the adversary to the set of scores to determine if there is a match. Finally, **record selection** selects one "best-guess" record or a probability distribution, if needed.

1. Compute $\mathsf{Score}(\mathsf{aux}, r')$ for each $r' \in \hat{D}$.

2. Apply the matching criterion to the resulting set of scores and compute the matching set; if the matching set is empty, output $\perp$ and exit.

3. If a "best guess" is required (de-anonymization according to Defs. 2 and 3), output $r' \in \hat{D}$ with the highest score. If a probability distribution over candidate records is required (de-anonymization according to Def. 4), compute and output some non-decreasing distribution based on the scores.

Algorithm Scoreboard. The following simple instantiation of the above template is sufficiently tractable to be formally analyzed in the rest of this section.

- $\mathsf{Score}(\mathsf{aux}, r') = \mathsf{min}_{i \in \mathsf{supp(aux)}} \mathsf{Sim}(\mathsf{aux}_i, r'_i)$, *i.e.*, the score of a candidate record is determined by the least similar attribute between it and the adversary's auxiliary information.

- The matching set $D' = \{r' \in \hat{D} : \mathsf{Score}(\mathsf{aux}, r') > \alpha\}$ for some fixed constant α. The matching criterion is that D' be nonempty.

- Probability distribution is uniform on D'.

Algorithm Scoreboard-RH. Algorithm Scoreboard is not sufficiently robust for some applications; in particular, it fails if any of the attributes in the adversary's auxiliary information are completely incorrect.

The following algorithm incorporates several heuristics which have proved useful in practical analysis (see section 5). First, the scoring function gives higher weight to statistically rare attributes. Intuitively, if the auxiliary information tells the adversary that his target has a certain rare attribute, this helps de-anonymization much more than the knowledge of a common attribute (*e.g.*, it is more useful to know that the target has purchased "The Dedalus Book of French Horror" than the fact that she purchased a Harry Potter book).

Second, to improve robustness, the matching criterion requires that the top score be significantly above the second-best score. This measures how much the identified record "stands out" from other candidate records.

- $\mathsf{Score}(\mathsf{aux}, r') = \sum_{i \in \mathsf{supp}(\mathsf{aux})} \mathsf{wt}(i) \mathsf{Sim}(\mathsf{aux}_i, r'_i)$ where $\mathsf{wt}(i) = \frac{1}{\log |\mathsf{supp}(i)|}$.[3]
- If a "best guess" is required, compute $\mathsf{max} = \mathsf{max}(S), \mathsf{max}_2 = \mathsf{max}_2(S)$ and $\sigma = \sigma(S)$ where $S = \{\mathsf{Score}(\mathsf{aux}, r') : r' \in \hat{D}\}$, *i.e.*, the highest and second-highest scores and the standard deviation of the scores. If $\frac{\mathsf{max} - \mathsf{max}_2}{\sigma} < \phi$, where ϕ is a fixed parameter called the *eccentricity*, then there is no match; otherwise, the matching set consists of the record with the highest score.[4]
- If entropic de-anonymization is required, output distribution $\Pi(r') = c \cdot e^{\frac{\mathsf{Score}(\mathsf{aux}, r')}{\sigma}}$ for each r', where c is a constant that makes the distribution sum up to 1. This weighs each matching record in inverse proportion to the likelihood that the match in question is a statistical fluke.

Note that there are two ways in which this algorithm can fail to find the correct record. First, an incorrect record may be assigned the highest score. Second, the correct record may not have a score which is significantly higher than the second-highest score.

4.1 Analysis: general case

We now quantify the amount of auxiliary information needed to de-anonymize an arbitrary dataset using Algorithm $\mathsf{Scoreboard}$. The smaller the required information (*i.e.*, the fewer attribute values the adversary needs to know about his target), the easier the attack.

We start with the worst-case analysis and calculate how much auxiliary information is needed without any assumptions about the distribution from which the data are drawn. In section 4.2, we will show that much less auxiliary information is needed to de-anonymize records drawn from *sparse* distributions (real-world transaction and recommendation datasets are all sparse).

Let aux be the auxiliary information about some record r; aux consists of m (non-null) attribute values, which are close to the corresponding values of attributes in r, that is, $|\mathsf{aux}| = m$ and $\mathsf{Sim}(\mathsf{aux}_i, r_i) \geq 1 - \epsilon \; \forall i \in \mathsf{supp}(\mathsf{aux})$, where aux_i (respectively, r_i) is the ith attribute of aux (respectively, r).

Theorem 1 *Let $0 < \epsilon, \delta < 1$ and let D be the database. Let Aux be such that $\mathsf{aux} = \mathsf{Aux}(r)$ consists of at least $m \geq \frac{\log N - \log \epsilon}{-\log(1-\delta)}$ randomly selected attribute values of the target record r, where $\forall i \in \mathsf{supp}(\mathsf{aux})$, $\mathsf{Sim}(\mathsf{aux}_i, r_i) \geq 1 - \epsilon$. Then D can be $(1 - \epsilon - \delta, 1 - \epsilon)$-deanonymized w.r.t. Aux.*

Proof. Use Algorithm $\mathsf{Scoreboard}$ with $\alpha = 1 - \epsilon$ to compute the set of all records in $\hat{D}$ that match aux, then output a record r' at random from the matching set. It is sufficient to prove that this randomly chosen r' must be very similar to the target record r. (This satisfies our definition of a privacy breach because it gives the adversary almost everything he may want to learn about r.)

Record r' is a *false match* if $\mathsf{Sim}(r, r') \leq 1 - \epsilon - \delta$ (*i.e.*, the likelihood that r' is similar to the target r is below the threshold). We first show that, with high probability, there are no false matches in the matching set.

Lemma 1 *If r' is a false match, then* $\Pr_{i \in \mathsf{supp}(r)}[\mathsf{Sim}(r_i, r'_i) \geq 1 - \epsilon] < 1 - \delta$

Lemma 1 holds, because the contrary implies $\mathsf{Sim}(r, r') \geq (1 - \epsilon)(1 - \delta) \geq (1 - \epsilon - \delta)$, contradicting the assumption that r' is a false match. Therefore, the probability that the false match r' belongs to the matching set is at most $(1 - \delta)^m$. By a union bound, the probability that the matching set contains even a single false match is at most $N(1 - \delta)^m$. If $m = \frac{\log \frac{N}{\epsilon}}{\log \frac{1}{1-\delta}}$, then the probability that the matching set contains any false matches is no more than ϵ.

Therefore, with probability $1 - \epsilon$, there are no false matches. Thus for every record r' in the matching set, $\mathsf{Sim}(r, r') \geq 1 - \epsilon - \delta$, *i.e.*, any r' must be similar to the true record r. To complete the proof, observe that the matching set contains at least one record, r itself.

When δ is small, $m = \frac{\log N - \log \epsilon}{\delta}$. This depends logarithmically on ϵ and linearly on δ: the chance that the algorithm fails completely is very small even if attribute-wise accuracy is not very high. Also note that the matching set need not be small. Even if the algorithm returns

[3] Without loss of generality, we assume $\forall i \; |\mathsf{supp}(i)| > 0$.

[4] Increasing ϕ increases the false negative rate, *i.e.*, the chance of erroneously dismissing a correct match, and decreases the false positive rate; ϕ may be chosen so that the two rates are equal.

many records, with high probability they are *all* similar to the target record r, and thus any one of them can be used to learn the unknown attributes of r.

4.2 Analysis: sparse datasets

Most real-world datasets containing individual transactions, preferences, and so on are *sparse*. Sparsity increases the probability that de-anonymization succeeds, decreases the amount of auxiliary information needed, and improves robustness to both perturbation in the data and mistakes in the auxiliary information.

Our assumptions about data sparsity are very mild. We only assume $(1-\epsilon-\delta, \ldots)$ sparsity, *i.e.*, we assume that the average record does not have *extremely* similar peers in the dataset (real-world records tend not to have even *approximately* similar peers—see fig. 1).

Theorem 2 *Let ϵ, δ, and* aux *be as in Theorem 1. If the database D is $(1-\epsilon-\delta, \epsilon)$-sparse, then D can be $(1, 1-\epsilon)$-deanonymized.* □

The proof is essentially the same as for Theorem 1, but in this case *any* $r' \neq r$ from the matching set must be a false match. Because with probability $1-\epsilon$, Scoreboard outputs no false matches, the matching set consists of exactly one record: the true target record r.

De-anonymization in the sense of Definition 4 requires even less auxiliary information. Recall that in this kind of privacy breach, the adversary outputs a "lineup" of k suspect records, one of which is the true record. This k-deanonymization is equivalent to $(1, \frac{1}{k})$-deanonymization in our framework.

Theorem 3 *Let D be $(1-\epsilon-\delta, \epsilon)$-sparse and* aux *be as in Theorem 1 with $m = \frac{\log \frac{N}{k-1}}{\log \frac{1}{1-\delta}}$. Then*

- *D can be $(1, \frac{1}{k})$-deanonymized.*
- *D can be $(1, \log k)$-deanonymized (entropically).*

By the same argument as in the proof of Theorem 1, if the adversary knows $m = \frac{\log \frac{N}{k-1}}{\log \frac{1}{1-\delta}}$ attributes, then the expected number of false matches in the matching set is at most $k-1$. Let X be the random variable representing this number. A random record from the matching set is a false match with probability of at least $\frac{1}{X}$. Since $\frac{1}{x}$ is a convex function, apply Jensen's inequality [18] to obtain $E[\frac{1}{X}] \geq \frac{1}{E(X)} \geq \frac{1}{k}$.

Similarly, if the adversary outputs the uniform distribution over the matching set, its entropy is $\log X$. Since $\log x$ is a concave function, by Jensen's inequality $E[\log X] \leq \log E(X) \leq \log k$.

Neither claim follows directly from the other. □

4.3 De-anonymization from a sample

We now consider the scenario in which the released database $\hat{D} \subsetneq D$ is a sample of the original database D, *i.e.*, only some of the anonymized records are available to the adversary. This is the case, for example, for the Netflix Prize dataset (the subject of our case study in section 5), where the publicly available anonymized sample contains less than $\frac{1}{10}$ of the original data.

In this scenario, even though the original database D contains the adversary's target record r, this record may not appear in $\hat{D}$ even in anonymized form. The adversary can still apply Scoreboard, but the matching set may be empty, in which case the adversary outputs $\perp$ (indicating that de-anonymization fails). If the matching set is not empty, he proceeds as before: picks a random record r' and learn the attributes of r on the basis of r'. We now demonstrate the equivalent of Theorem 1: de-anonymization succeeds as long as r is in the public sample; otherwise, the adversary can detect, with high probability, that r is not in the public sample.

Theorem 4 *Let ϵ, δ, D, and* aux *be as in Theorem 1, and $\hat{D} \subset D$. Then $\hat{D}$ can be $(1-\epsilon-\delta, 1-\epsilon)$-deanonymized w.r.t.* aux*.* □

The bound on the probability of a false match given in the proof of Theorem 1 still holds, and the adversary is guaranteed at least one match as long as his target record r is in $\hat{D}$. Therefore, if $r \notin \hat{D}$, the adversary outputs $\perp$ with probability at least $1-\epsilon$. If $r \in \hat{D}$, then again the adversary succeeds with probability at least $1-\epsilon$.

Theorems 2 and 3 do not translate directly. For each record in the public sample $\hat{D}$, there could be any number of similar records in $D \setminus \hat{D}$, the part of the database that is not available to the adversary.

Fortunately, if D is sparse, then theorems 2 and 3 still hold, and de-anonymization succeeds with a very small amount of auxiliary information. We now show that if the random sample $\hat{D}$ is sparse, then the entire database D must also be sparse. Therefore, the adversary can simply apply the de-anonymization algorithm to the sample. If he finds the target record r, then with high probability this is not a false positive.

Theorem 5 *If database D is not (ϵ, δ)-sparse, then a random $\frac{1}{\lambda}$-subset $\hat{D}$ is not $(\epsilon, \frac{\delta\gamma}{\lambda})$-sparse with probability at least $1-\gamma$.* □

For each $r \in \hat{D}$, the "nearest neighbor" r' of r in D has a probability $\frac{1}{\lambda}$ of being included in $\hat{D}$. Therefore, the expected probability that the similarity with the nearest neighbor is at least $1-\epsilon$ is at least $\frac{\delta}{\lambda}$. (Here the expectation is over the set of all possible samples and the probability is over the choice of the record in $\hat{D}$.) Applying Markov's inequality, the probability, taken over the choice $\hat{D}$, that $\hat{D}$ is sparse, *i.e.*, that the similarity with the nearest neighbor is $\frac{\delta\gamma}{\lambda}$, is no more than γ. □

The above bound is quite pessimistic. Intuitively, for any "reasonable" dataset, the sparsity of a random sample will be about the same as that of the original dataset.

Theorem 5 can be interpreted as follows. Consider the adversary who has access to a sparse sample $\hat{D}$, but not the entire database D. Theorem 5 says that either a very-low-probability event has occurred, or D itself is sparse. Note that it is meaningless to try to bound the probability that D is sparse because we do not have a probability distribution on how D itself is created.

Intuitively, this says that unless the sample is specially tailored, sparsity of the sample implies sparsity of the entire database. The alternative is that the similarity between a random record in the sample and its nearest neighbor is very different from the corresponding distribution in the full database. In practice, most, if not all anonymized datasets are published to support research on data mining and collaborative filtering. Tailoring the published sample in such a way that its nearest-neighbor similarity is radically different from that of the original data would completely destroy utility of the sample for learning new collaborative filters, which are often based on the set of nearest neighbors. Therefore, in real-world anonymous data publishing scenarios—including, for example, the Netflix Prize dataset—sparsity of the sample should imply sparsity of the original dataset.

5 Case study: Netflix Prize dataset

On October 2, 2006, Netflix, the world's largest online DVD rental service, announced the $1-million Netflix Prize for improving their movie recommendation service [15]. To aid contestants, Netflix publicly released a dataset containing $100,480,507$ movie ratings, created by $480,189$ Netflix subscribers between December 1999 and December 2005.

Among the Frequently Asked Questions about the Netflix Prize [23], there is the following question: "Is there any customer information in the dataset that should be kept private?" The answer is as follows:

> "No, all customer identifying information has been removed; all that remains are ratings and dates. This follows our privacy policy [...] Even if, for example, you knew all your own ratings and their dates you probably couldn't identify them reliably in the data because only a small sample was included (less than one-tenth of our complete dataset) and that data was subject to perturbation. Of course, since you know all your own ratings that really isn't a privacy problem is it?"

Removing identifying information is not sufficient for anonymity. An adversary may have auxiliary information about a subscriber's movie preferences: the titles of a few of the movies that this subscriber watched, whether she liked them or not, maybe even approximate dates when she watched them. We emphasize that even if it is hard to collect such information for a large number of subscribers, targeted de-anonymization—for example, a boss using the Netflix Prize dataset to find an employee's entire movie viewing history after a casual conversation—still presents a serious threat to privacy.

We investigate the following question: *How much does the adversary need to know about a Netflix subscriber in order to identify her record if it is present in the dataset, and thus learn her complete movie viewing history?* Formally, we study the relationship between the size of aux and $(1,\omega)$- and $(1,H)$-deanonymization.

Does privacy of Netflix ratings matter? The issue is *not* "Does the average Netflix subscriber care about the privacy of his movie viewing history?," but "Are there *any* Netflix subscribers whose privacy can be compromised by analyzing the Netflix Prize dataset?" As shown by our experiments below, it is possible to learn sensitive *non-public* information about a person from his or her movie viewing history. We assert that even if the vast majority of Netflix subscribers did not care about the privacy of their movie ratings (which is not obvious by any means), our analysis would still indicate serious privacy issues with the Netflix Prize dataset.

Moreover, the linkage between an individual and her movie viewing history has implications for her *future* privacy. In network security, "forward secrecy" is important: even if the attacker manages to compromise a session key, this should not help him much in compromising the keys of future sessions. Similarly, one may state the "forward privacy" property: if someone's privacy is breached (*e.g.*, her anonymous online records have been linked to her real identity), future privacy breaches should not become easier. Consider a Netflix subscriber Alice whose entire movie viewing history has been revealed. Even if in the future Alice creates a brand-new virtual identity (call her Ecila), Ecila will *never* be able to disclose any non-trivial information about the movies that she had rated within Netflix because any such information can be traced back to her real identity via the Netflix Prize dataset. In general, once any piece of data has been linked to a person's *real* identity, any association between this data and a *virtual* identity breaks anonymity of the latter.

Finally, the Video Privacy Protection Act of 1988 [13] lays down strong provisions against disclosure of personally identifiable rental records of "prerecorded video cassette tapes or similar audio visual material." While the Netflix Prize dataset does not *explicitly* include personally identifiable information, the issue of whether the implicit disclosure demonstrated by our analysis runs afoul of the law or not is a legal question to be considered.

How did Netflix release and sanitize the data? Figs. 2 and 3 plot the number of ratings X against the number of subscribers in the released dataset who have at least X ratings. The tail is surprisingly thick: thousands of subscribers have rated more than a thousand movies. Netflix claims that the subscribers in the released dataset have been "randomly chosen." Whatever the selection algorithm was, it was not uniformly random. Common sense suggests that with uniform subscriber selection, the curve would be monotonically decreasing (as most people rate very few movies or none at all), and that there would be no sharp discontinuities.

We conjecture that some fraction of subscribers with more than 20 ratings were sampled, and the points on the graph to the left of $X = 20$ are the result of some movies being deleted after sampling.

We requested the rating history as presented on the Netflix website from some of our acquaintances, and based on this data (which is effectively drawn from Netflix's *original*, non-anonymous dataset, since we know the names associated with these records), located two of them in the Netflix Prize dataset. Netflix's claim that the data were perturbed does not appear to be borne out. One of the subscribers had 1 of 306 ratings altered, and the other had 5 of 229 altered. (These are upper bounds, because the subscribers may have changed their ratings after Netflix took the 2005 snapshot that was released.) In any case, the level of noise is far too small to affect our de-anonymization algorithm, which has been specifically designed to withstand this kind of imprecision. We have no way of determining how many dates were altered and how many ratings were deleted, but we conjecture that very little perturbation has been applied.

It is important that the Netflix Prize dataset has been released to support development of better recommendation algorithms. A significant perturbation of individual attributes would have affected cross-attribute correlations and significantly decreased the dataset's utility for creating new recommendation algorithms, defeating the entire purpose of the Netflix Prize competition.

Note that the Netflix Prize dataset clearly has not been k-anonymized for any value of $k > 1$.

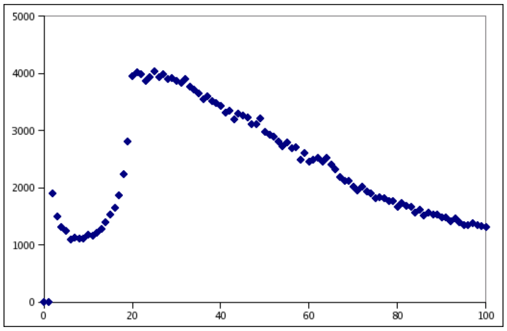

Figure 2. For each $X \leq 100$, the number of subscribers with X ratings in the released dataset.

De-anonymizing the Netflix Prize dataset. We apply Algorithm Scoreboard-RH from section 4. The similarity measure Sim on attributes is a threshold function: Sim returns 1 if and only if the two attribute values are within a certain threshold of each other. For movie ratings, which in the case of Netflix are on the 1-5 scale, we consider the thresholds of 0 (corresponding to exact match) and 1, and for the rating dates, 3 days, 14 days, or ∞. The latter means that the adversary has no information about the date when the movie was rated.

Some of the attribute values known to the attacker may be completely wrong. We say that aux of a record

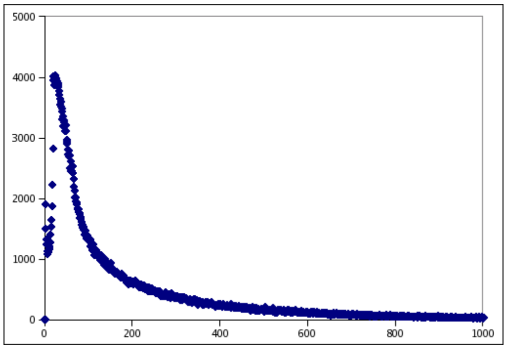

Figure 3. For each $X \leq 1000$, the number of subscribers with X ratings in the released dataset.

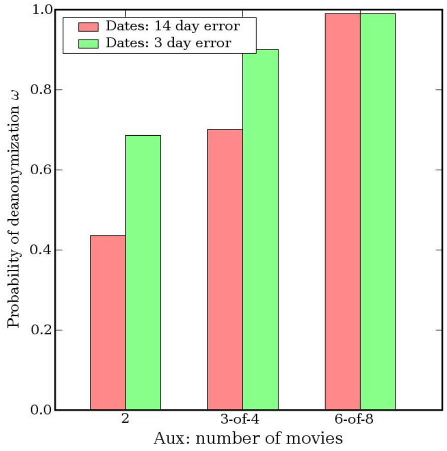

Figure 4. Adversary knows exact ratings and approximate dates.

The red (darker) bars represent the probability of correctly detecting that the record is not in the sample

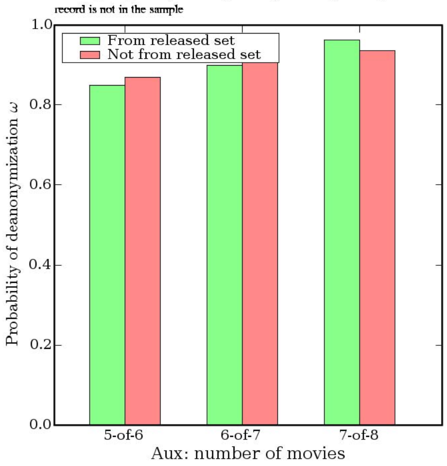

Figure 5. Same parameters as Fig. 4, but the adversary must also detect when the target record is not in the sample.

r consists of m movies out of m' if $|\mathsf{aux}| = m'$, r_i is non-null for each aux_i, and $\sum_i \mathsf{Sim}(\mathsf{aux}_i, r_i) \geq m$. We instantiate the scoring function as follows:

$$\mathsf{Score}(\mathsf{aux}, r') = \sum_{i \in \mathsf{supp}(\mathsf{aux})} \mathsf{wt}(i)(e^{\frac{\rho_i - \rho_i'}{\rho_0}} + e^{\frac{d_i - d_i'}{d_0}})$$

where $\mathsf{wt}(i) = \frac{1}{\log |\mathsf{supp}(i)|}$ ($|\mathsf{supp}(i)|$ is the number of subscribers who have rated movie i), ρ_i and d_i are the rating and date, respectively, of movie i in the auxiliary information, and ρ_i' and d_i' are the rating and date in the candidate record r'.[5] As explained in section 4, this scoring function was chosen to favor statistically unlikely matches and thus minimize accidental false positives. The parameters ρ_0 and d_0 are 1.5 and 30 days, respectively. These were chosen heuristically, as they gave the best results in our experiments,[6] and used throughout, regardless of the amount of noise in Aux. The eccentricity parameter was set to $\phi = 1.5$, *i.e.*, the algorithm declares there is no match if and only if the difference between the highest and the second highest scores is no more than 1.5 times the standard deviation. (A constant value of ϕ does not always give the equal error rate, but it is a close enough approximation.)

Didn't Netflix publish only a sample of the data? Because Netflix published less than $\frac{1}{10}$ of its 2005 database, we need to be concerned about the possibility that when our algorithm finds a record matching aux in the published sample, this may be a false match and the real record has not been released at all.

Algorithm $\mathsf{Scoreboard\text{-}RH}$ is specifically designed to detect when the record corresponding to aux is *not* in the sample. We ran the following experiment. First, we gave aux from a random record to the algorithm and ran it on the dataset. Then we *removed* this record from the dataset and re-ran the algorithm. In the former case, the algorithm should find the record; in the latter, declare that it is not in the dataset. As shown in Fig. 5, the algorithm succeeds with high probability in both cases.

It is possible, although *extremely* unlikely, that the original Netflix dataset is not as sparse as the published sample, *i.e.*, it contains clusters of records which are close to each other, but only one representative of each cluster has been released in the Prize dataset. A dataset with such a structure would be exceptionally unusual and theoretically problematic (see Theorem 4).

If the adversary has less auxiliary information than shown in Fig. 5, false positives cannot be ruled out *a priori*, but there is a lot of extra information in the dataset that can be used to eliminate them. For example, if the start date and total number of movies in a record are part of the auxiliary information (*e.g.*, the adversary knows approximately when his target first joined Netflix), they

[5] $\mathsf{wt}(i)$ is undefined when $|\mathsf{supp}(i)| = 0$, but this is not a concern since every movie is rated by at least 4 subscribers.

[6] It may seem that tuning the parameters to the specific dataset may have unfairly improved our results, but an actual adversary would have performed the same tuning. We do not claim that these numerical parameters should be used for other instances of our algorithm; they must be derived by trial and error for each target dataset.

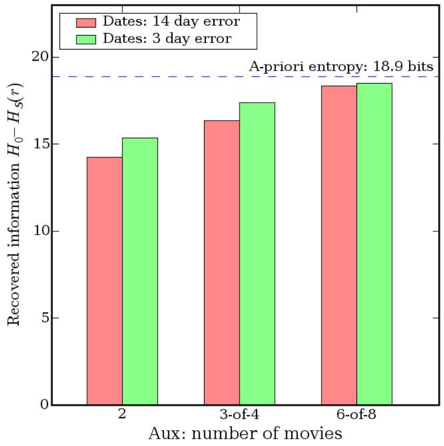

Figure 6. Entropic de-anonymization: same parameters as in Fig. 4.

can be used to eliminate candidate records.

Results of de-anonymization. We carried out the experiments summarized in the following table:

Fig	Ratings	Dates	Type	Aux selection
4	Exact	$\pm 3/\pm 14$	Best-guess	Uniform
5	Exact	$\pm 3/\pm 14$	Best-guess	Uniform
6	Exact	$\pm 3/\pm 14$	Entropic	Uniform
8	Exact	No info.	Best-guess	Not 100/500
9	± 1	± 14	Best-guess	Uniform
10	± 1	± 14	Best-guess	Uniform
11	Exact	No info.	Entropic	Not 100/500
12	± 1	± 14	Best-guess	Uniform

Our conclusion is that very little auxiliary information is needed for de-anonymize an average subscriber record from the Netflix Prize dataset. With 8 movie ratings (of which 2 may be completely wrong) and dates that may have a 14-day error, 99% of records can be uniquely identified in the dataset. For 68%, *two* ratings and dates (with a 3-day error) are sufficient (Fig. 4). Even for the other 32%, the number of possible candidates is brought down dramatically. In terms of entropy, the additional information required for complete de-anonymization is around 3 bits in the latter case (with no auxiliary information, this number is 19 bits). When the adversary knows 6 movies correctly and 2 incorrectly, the extra information he needs for complete de-anonymization is a fraction of a bit (Fig. 6).

Even without any dates, a substantial privacy breach occurs, especially when the auxiliary information consists of movies that are not blockbusters. In Fig. 7, we

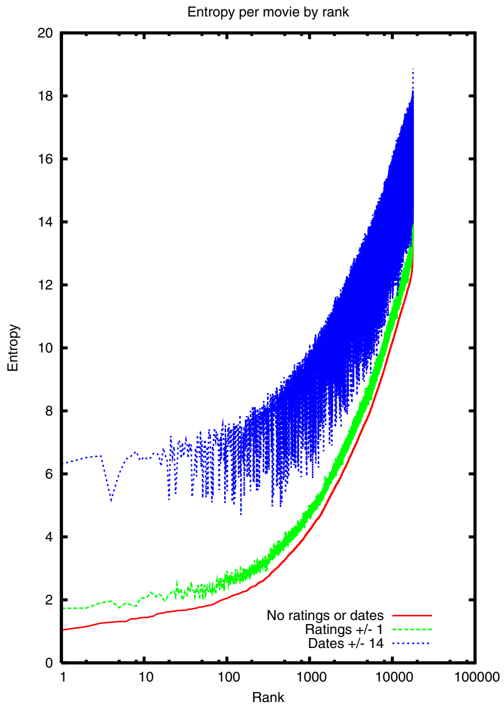

Figure 7. Entropy of movie by rank

demonstrate how much information the adversary gains about his target just from the knowledge that the target watched a particular movie as a function of the rank of the movie.[7] Because there are correlations between the lists of subscribers who watched various movies, we cannot simply multiply the information gain per movie by the number of movies. Therefore, Fig. 7 cannot be used to infer how many movies the adversary needs to know for successful de-anonymization.

As shown in Fig. 8, two movies are no longer sufficient for de-anonymization, but 84% of subscribers present in the dataset can be uniquely identified if the adversary knows 6 out of 8 moves outside the top 500. To show that this is not a significant limitation, consider that most subscribers rate fairly rare movies:

Not in X most rated	% of subscribers who rated …		
	≥ 1 movie	≥ 5	≥ 10
$X = 100$	100%	97%	93%
$X = 500$	99%	90%	80%
$X = 1000$	97%	83%	70%

Fig. 9 shows that the effect of relative popularity of movies known to the adversary is not dramatic.

In Fig. 10, we add even more noise to the auxiliary

[7] We measure the rank of a movie by the number of subscribers who have rated it.

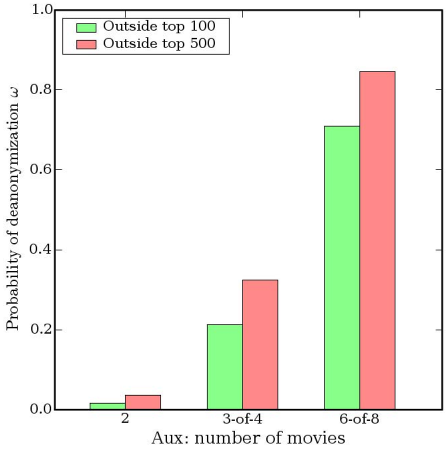

Figure 8. Adversary knows exact ratings but does not know dates at all.

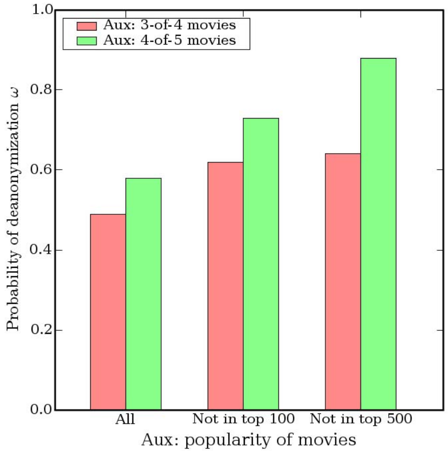

Figure 9. Effect of knowing less popular movies rated by victim. Adversary knows approximate ratings (± 1) and dates (14-day error).

information, allowing mistakes about *which* movies the target watched, and not just their ratings and dates.

Fig. 11 shows that even when the adversary's probability to correctly learn the attributes of the target record is low, he gains a lot of information about the target record. Even in the worst scenario, the additional information needed to to complete the de-anonymization has been reduced to less than half of its original value.

Fig. 12 shows why even partial de-anonymization can be very dangerous. There are many things the adversary might know about his target that are not captured by our formal model, such as the approximate number of movies rated, the date when they joined Netflix and so on. Once a candidate set of records is available, further automated analysis or human inspection might be sufficient to complete the de-anonymization. Fig. 12 shows that in some cases, knowing the number of movies the target has rated (even with a 50% error!) can more than double the probability of complete de-anonymization.

Obtaining the auxiliary information. Given how little auxiliary information is needed to de-anonymize the average subscriber record from the Netflix Prize dataset, a determined adversary who targets a specific individual may not find it difficult to obtain such information, especially since it need not be precise. We emphasize that massive collection of data on thousands of subscribers is not the only or even the most important threat. A water-cooler conversation with an office colleague about her cinematographic likes and dislikes may yield enough information, especially if at least a few of the movies mentioned are outside the top 100 most rated Netflix movies. This information can also be gleaned from personal blogs, Google searches, and so on.

One possible source of a large number of personal movie ratings is the Internet Movie Database (IMDb) [17]. We expect that for Netflix subscribers who use IMDb, there is a strong correlation between their private Netflix ratings and their public IMDb ratings.[8] Our attack does not require that all movies rated by the subscriber in the Netflix system be also rated in IMDb, or vice versa. In many cases, even a handful of movies that are rated by a subscriber in both services would be sufficient to identify his or her record in the Netflix Prize dataset (if present among the released records) with enough statistical confidence to rule out the possibility of a false match except for a negligible probability.

Due to the restrictions on crawling IMDb imposed by IMDb's terms of service (of course, a real adversary may not comply with these restrictions), we worked with a very small sample of around 50 IMDb users. Our results should thus be viewed as a proof of concept. They do not imply anything about the percentage of IMDb users who can be identified in the Netflix Prize dataset.

The auxiliary information obtained from IMDb is quite noisy. First, a significant fraction of the movies rated on IMDb are not in Netflix, and vice versa, *e.g.*,

[8] We are *not* claiming that a large fraction of Netflix subscribers use IMDb, or that many IMDb users use Netflix.

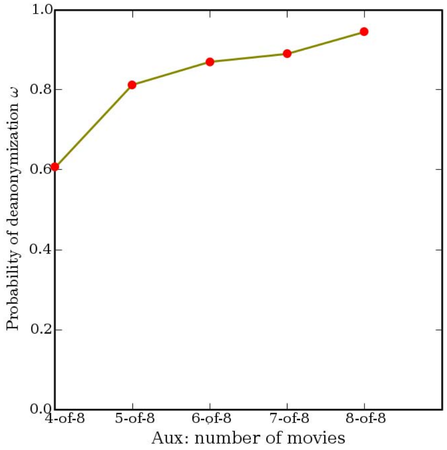

Figure 10. Effect of increasing error in Aux.

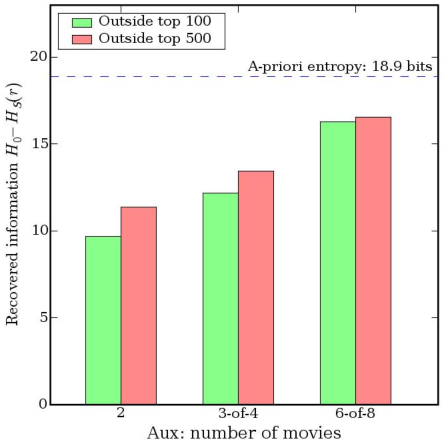

Figure 11. Entropic de-anonymization: same parameters as in Fig. 6.

movies that have not been released in the US. Second, some of the ratings on IMDb are missing (*i.e.*, the user entered only a comment, not a numerical rating). Such data are still useful for de-anonymization because an average user has rated only a tiny fraction of all movies, so the mere fact that a person has watched a given movie tremendously reduces the number of anonymous Netflix records that could possibly belong to that user. Finally, IMDb users among Netflix subscribers fall into a continuum of categories with respect to rating dates, separated by two extremes: some meticulously rate movies on both IMDb and Netflix at the same time, and others rate them whenever they have free time (which means the dates may not be correlated at all). Somewhat offsetting these disadvantages is the fact that we can use all of the user's ratings publicly available on IMDb.

Because we have no "oracle" to tell us whether the record our algorithm has found in the Netflix Prize dataset based on the ratings of some IMDb user indeed belongs to that user, we need to guarantee a very low false positive rate. Given our small sample of IMDb users, our algorithm identified the records of two users in the Netflix Prize dataset with eccentricities of around 28 and 15, respectively. These are exceptionally strong matches, which are highly unlikely to be false positives: the records in questions are **28 standard deviations** (respectively, 15 standard deviations) away from the second-best candidate. Interestingly, the first user was de-anonymized mainly from the ratings and the second mainly from the dates. For nearly all the other IMDb users we tested, the eccentricity was no more than 2.

Let us summarize what our algorithm achieves. Given a user's *public* IMDb ratings, which the user posted voluntarily to reveal *some* of his (or her; but we'll use the male pronoun without loss of generality) movie likes and dislikes, we discover *all* ratings that he entered *privately* into the Netflix system. Why would someone who rates movies on IMDb—often under his or her real name—care about privacy of his Netflix ratings? Consider the information that we have been able to deduce by locating one of these users' entire movie viewing history in the Netflix Prize dataset and that *cannot* be deduced from his public IMDb ratings.

First, his political orientation may be revealed by his strong opinions about "Power and Terror: Noam Chomsky in Our Times" and "Fahrenheit 9/11," and his religious views by his ratings on "Jesus of Nazareth" and "The Gospel of John." Even though one should not make inferences solely from someone's movie preferences, in many workplaces and social settings opinions about movies with predominantly gay themes such as "Bent" and "Queer as folk" (both present and rated in this person's Netflix record) would be considered sensitive. In any case, it should be for the individual and not for Netflix to decide whether to reveal them publicly.

6 Conclusions

We have presented a de-anonymization methodology for sparse micro-data, and demonstrated its prac-

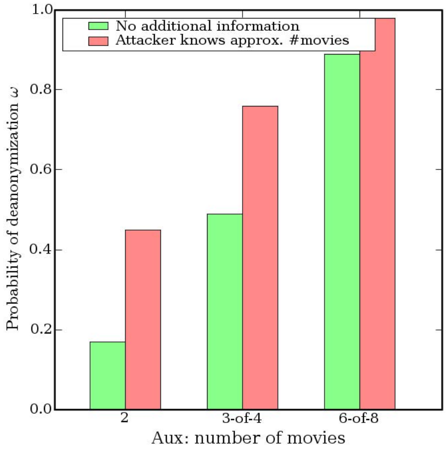

Figure 12. Effect of knowing approximate number of movies rated by victim ($\pm 50\%$). Adversary knows approximate ratings (± 1) and dates (14-day error).

tical applicability by showing how to de-anonymize movie viewing records released in the Netflix Prize dataset. Our de-anonymization algorithm Scoreboard-RH works under very general assumptions about the distribution from which the data are drawn, and is robust to data perturbation and mistakes in the adversary's knowledge. Therefore, we expect that it can be successfully used against any dataset containing anonymous multi-dimensional records such as individual transactions, preferences, and so on.

We conjecture that the amount of perturbation that must be applied to the data to defeat our algorithm will completely destroy their utility for collaborative filtering. Sanitization techniques from the k-anonymity literature such as generalization and suppression [27, 9, 20] do not provide meaningful privacy guarantees, and in any case fail on high-dimensional data. Furthermore, for most records simply knowing *which* columns are non-null reveals as much information as knowing the specific values of these columns. Therefore, any technique such as generalization and suppression which leaves sensitive attributes untouched does not help.

Other possible countermeasures include interactive mechanisms for privacy-protecting data mining such as [5, 12], as well as more recent non-interactive techniques [6]. Both support only limited classes of computations such as statistical queries and learning halfspaces. By contrast, in scenarios such as the Netflix Prize, the purpose of the data release is precisely to foster computations on the data that have not even been foreseen at the time of release [9], and are vastly more sophisticated than the computations that we know how to perform in a privacy-preserving manner.

An intriguing possibility was suggested by Matthew Wright via personal communication: to release the records without the column identifiers (*i.e.*, movie names in the case of the Netflix Prize dataset). It is not clear how much worse the current data mining algorithms would perform under this restriction. Furthermore, this does not appear to make de-anonymization impossible, but merely harder. Nevertheless, it is an interesting countermeasure to investigate.

Acknowledgements. This material is based upon work supported by the NSF grant IIS-0534198, and the ARO grant W911NF-06-1-0316.

The authors would like to thank Ilya Mironov for many insightful suggestions and discussions and Matt Wright for suggesting an interesting anonymization technique. We are also grateful to Justin Brickell, Shuchi Chawla, Jason Davis, Cynthia Dwork, and Frank McSherry for productive conversations.

References

[1] N. Adam and J. Worthmann. Security-control methods for statistical databases: A comparative study. *ACM Computing Surveys*, 21(4), 1989.

[2] C. Aggarwal. On k-anonymity and the curse of dimensionality. In *VLDB*, 2005.

[3] R. Agrawal and R. Srikant. Privacy-preserving data mining. In *SIGMOD*, 2000.

[4] C. Anderson. *The Long Tail: Why the Future of Business Is Selling Less of More*. Hyperion, 2006.

[5] A. Blum, C. Dwork, F. McSherry, and K. Nissim. Practical privacy: The SuLQ framework. In *PODS*, 2005.

[6] A. Blum, K. Ligett, and A. Roth. A learning theory approach to non-interactive database privacy. In *STOC*, 2008.

[7] E. Brynjolfsson, Y. Hu, and M. Smith. Consumer surplus in the digital economy. *Management Science*, 49(11), 2003.

[9] As of February 2008, the current best algorithm in the Netflix Prize competition is a combination of 107 different techniques.

[8] S. Chawla, C. Dwork, F. McSherry, A. Smith, and H. Wee. Towards privacy in public databases. In *TCC*, 2005.

[9] V. Ciriani, S. De Capitani di Vimercati, S. Foresti, and P. Samarati. k-anonymity. *Secure Data Management in Decentralized Systems*, 2007.

[10] C. Díaz, S. Seys, J. Claessens, and B. Preneel. Towards measuring anonymity. In *PET*, 2003.

[11] C. Dwork. Differential privacy. In *ICALP*, 2006.

[12] C. Dwork, F. McSherry, K. Nissim, and A. Smith. Calibrating noise to sensitivity in private data analysis. In *TCC*, 2006.

[13] Electronic Privacy Information Center. The Video Privacy Protection Act (VPPA). `http://epic.org/privacy/vppa/`, 2002.

[14] D. Frankowski, D. Cosley, S. Sen, L. Terveen, and J. Riedl. You are what you say: privacy risks of public mentions. In *SIGIR*, 2006.

[15] K. Hafner. And if you liked the movie, a Netflix contest may reward you handsomely. New York Times, Oct 2 2006.

[16] S. Hansell. AOL removes search data on vast group of web users. New York Times, Aug 8 2006.

[17] IMDb. The Internet Movie Database. `http://www.imdb.com/`, 2007.

[18] J. L. W. V. Jensen. Sur les fonctions convexes et les inégalités entre les valeurs moyennes. *Acta Mathematica*, 30(1), 1906.

[19] J. Leskovec, L. Adamic, and B. Huberman. The dynamics of viral marketing. In *EC*, 2006.

[20] A. Machanavajjhala, J. Gehrke, D. Kifer, and M. Venkitasubramaniam. l-diversity: Privacy beyond k-anonymity. In *ICDE*, 2006.

[21] A. Machanavajjhala, D. Martin, D. Kifer, J. Gehrke, and J. Halpern. Worst case background knowledge. In *ICDE*, 2007.

[22] B. Malin and L. Sweeney. How (not) to protect genomic data privacy in a distributed network: using trail re-identification to evaluate and design anonymity protection systems. *J. of Biomedical Informatics*, 37(3), 2004.

[23] Netflix. Netflix Prize: FAQ. `http://www.netflixprize.com/faq`, Downloaded on Oct 17 2006.

[24] A. Serjantov and G. Danezis. Towards an information theoretic metric for anonymity. In *PET*, 2003.

[25] L. Sweeney. Weaving technology and policy together to maintain confidentiality. *J. of Law, Medicine and Ethics*, 25(2–3), 1997.

[26] L. Sweeney. Achieving k-anonymity privacy protection using generalization and suppression. *International J. of Uncertainty, Fuzziness and Knowledge-based Systems*, 10(5), 2002.

[27] L. Sweeney. k-anonymity: A model for protecting privacy. *International J. of Uncertainty, Fuzziness and Knowledge-based Systems*, 10(5):557–570, 2002.

[28] J. Thornton. Collaborative filtering research papers. `http://jamesthornton.com/cf/`, 2006.

A Glossary of terms

Symbol	Meaning
D	Database
$\hat{D}$	Released sample
N	Number of rows
M	Number of columns
m	Size of aux
X	Domain of attributes
$\perp$	Null attribute
$\mathsf{supp}(.)$	Set of non-null attributes in a row/column
Sim	Similarity measure
Aux	Auxiliary information sampler
aux	Auxiliary information
Score	Scoring function
ϵ	Sparsity threshold
δ	Sparsity probability
θ	Closeness of de-anonymized record
ω	Probability that de-anonymization succeeds
r, r'	Record
Π	P.d.f over records
H_S	Shannon entropy
H	De-anonymization entropy
ϕ	Eccentricity