Differentially Private Recommender Systems:

Building Privacy into the Netflix Prize Contenders

Frank McSherry and Ilya Mironov
Microsoft Research, Silicon Valley Campus
{mcsherry, mironov}@microsoft.com

ABSTRACT

We consider the problem of producing recommendations from collective user behavior while simultaneously providing guarantees of privacy for these users. Specifically, we consider the Netflix Prize data set, and its leading algorithms, adapted to the framework of *differential privacy*.

Unlike prior privacy work concerned with cryptographically securing the computation of recommendations, differential privacy constrains a computation in a way that precludes any inference about the underlying records from its output. Such algorithms necessarily introduce uncertainty—*i.e.,* noise—to computations, trading accuracy for privacy.

We find that several of the leading approaches in the Netflix Prize competition can be adapted to provide differential privacy, without significantly degrading their accuracy. To adapt these algorithms, we explicitly factor them into two parts, an aggregation/learning phase that can be performed with differential privacy guarantees, and an individual recommendation phase that uses the learned correlations and an individual's data to provide personalized recommendations. The adaptations are non-trivial, and involve both careful analysis of the per-record sensitivity of the algorithms to calibrate noise, as well as new post-processing steps to mitigate the impact of this noise.

We measure the empirical trade-off between accuracy and privacy in these adaptations, and find that we can provide non-trivial formal privacy guarantees while still outperforming the Cinematch baseline Netflix provides.

Categories and Subject Descriptors

H.2.8 [**Database Management**]: [Data Mining]

General Terms

Algorithms, Security, Theory

Keywords

differential privacy, netflix, recommender systems

KDD'09, June 28–July 1, 2009, Paris, France.

1. MOTIVATION

A recommender system based on collaborative filtering is a double-edged sword. By aggregating and processing preferences of multiple users it may provide relevant recommendations, boosting a web site's revenue and enhancing user experience. On the flip side, it is a potential source of leakage of private information shared by the users. The focus of this paper is on design, analysis, and experimental validation of a recommender system with built-in privacy guarantees. We measure accuracy of the system on the Netflix Prize data set, which also drives our choice of algorithms.

The goals of improving accuracy of recommender systems and providing privacy for their users are nicely aligned. They are part of a virtuous cycle where better accuracy and stronger privacy guarantees relieve anxiety associated with sharing one's private information, leading to broader and deeper participation which in turn improves accuracy and privacy in the same time.

Consider a recommender system that collects, stores, and processes information from its user base. Even if all security measures such as proper access control mechanisms, protected storage, encrypted client-server communications are in place, the system's output visible to any user (i.e., recommendations) is derived in part from other users' input. A curious or malicious user, or a coalition thereof, may attempt to make inferences about someone else's input based on their own and the view exposed through the standard interfaces of the recommender system. The threat is especially ominous in the context of open-access web sites with weak identities and greater potential for on-line active attacks, where the adversary is able to create multiple accounts, inject its own input into the recommender system, observe the changes and adapt its behavior, constrained only by the network speed and the system's turnaround time.

There are two common arguments used to deflect privacy concerns presented by recommender systems. We address these arguments in turn.

Non-sensitivity of data. In many instances the information shared by users is assumed to be non-sensitive and treated as such. We observe that sensitivity of data is contextual, heavily dependent on the user's circumstances and the attacker's axillary knowledge, and any global policy decisions ought to err on the conservative side. Unsophisticated users may not be aware of the amount of personal data made available (and often collected) in the course of routine web surfing, such as the IP address, timing information, HTTP headers, etc., with far-reaching privacy implications. Moreover, a correct determination of sensitivity of information

is difficult and itself is a moving target. Narayanan and Shmatikov [22] demonstrated a powerful and practical de-anonymization attack linking records in the Netflix Prize data set with public IMDb profiles. Taking their attack one step further, consider a scenario where an individual maintains two different profiles (say, a professional blog and a pseudonymous page on a social network) with occasional discussions of disjoint sets of movies, all rated on Netflix. It is then possible that they could be linked to the same Netflix Prize row and thus to each other. Whether the movie ratings attributed to this individual are embarrassing, revealing, or at all remarkable is irrelevant, it is the fact that the two personas are linked that can be perceived as deeply invasive and disturbing.

Implicit properties of quality recommender systems. A recommender system that is overly sensitive to one person's input would normally be considered deficient. However, when assessing security of a system, one has to consider not its typical state but rather all feasible states into which it can forced by a determined and resourceful attacker. For instance, in a recommender system based on user-user similarities, the attacker can create a fictitious profile (or many profiles) resembling an individual based on the attacker's partial knowledge and induce the recommender system to faithfully report all items highly rated by the individual. Systems that allow submission of entries (such as digg.com or stumbleupon.com) are vulnerable to similar attacks even if they relied exclusively on item-item similarities. Although complexity of this attack increases with complexity of the recommender system, which is typically kept secret, security through obscurity is a discredited security practice.

In other words, a privacy-preserving recommender system must retain its security properties against any feasible attacker (or a well-defined class of attackers) who has access to the system's design and unrestricted auxiliary information about its targets.

1.1 Contributions

The main contribution of this work is to design and analyze a realistic recommender system built to provide modern privacy guarantees. The task is non-trivial: prior recommender systems are not designed with an eye towards privacy, and prior privacy research has focused on more modest algorithms without attempts at practical validation. Recommender systems add the additional complexity that their end-to-end behavior *should* reflect each users private data, requiring us to crisply separate a privacy-preserving "learning" phase from a highly non-private "prediction" phase (conducted by the user, in the privacy of their own home).

One natural approach would follow the large volume of work that has occurred on anonymized data publication such as k-anonymity [23], where, as Netflix has done, data is released with an attempt to remove sensitive information and overly specific combinations of attributes. In addition to the uncertain privacy guarantees [18], Brickell and Shmatikov [8] find that these techniques applied to high dimensional data cause irreparable damage for data mining algorithms. Instead, we integrate the privacy protection into the computation itself, ensuring that the learned recommendations preserve privacy using the framework of differential privacy.

Our findings are that privacy does not need to come at substantial expense in accuracy. For the approaches we consider, privacy-preserving algorithms can be parameterized to essentially match the recommendation performance of their non-private analogues. While there is some specialized analysis required, the methodology itself is relatively straight forward and accessible. As an additional contribution of this note, we hope to demonstrate the integration of modern privacy technology to practical and realistic learning systems.

1.2 Related Work

We base our choice of algorithms on leading solutions to the Netflix Prize [4, 5, 6]. We adapt algorithms exemplifying two approaches that emerged as main components of Netflix Prize contenders: factor models and neighborhood models.

Several papers have recently introduced and studied the application of differential privacy to problems in learning and data mining, surveyed by Dwork [14]. Algorithms such as k-means clustering, perceptron classification, association rule mining, decision tree induction, and low rank approximation are all shown to have differentially private analogues, with theoretical bounds on their accuracy [7]. While we borrow substantially from these works for our underlying privacy technology, our focus is more on building and evaluating a full end-to-end recommender system, rather than isolated components.

The wholesale release of data with anonymized user identities by Neflix has been shown to have far-reaching privacy implications [22], establishing, in particular, that most rows can be identified with near certainty based on as few as a dozen partially known data points. Although a commercial recommender system is unlikely to willingly disclose all or substantial fraction of its underlying data, a recent work by Calandrino et al. [9] demonstrates that passive observations of Amazon.com's recommendations are sufficient to make valid inferences about individuals' purchase histories.

The focus of prior work on cryptographic solutions to the problem of secure recommender systems is on removing the single trusted party having access to everyone's data [10, 11, 2, 3]. It does not attempt to limit amount of information leaked through the system's recommendations in the course of its normal execution. Our solution can be combined with the modular approach of the Alambic framework [2].

It is important to distinguish our approach, of privacy preserving computation, from much prior work on privacy studying the release of anonymized records. One could imagine building a recommender system, or any machine learning technology, on top of anonymized data, drawing privacy properties from the anonymization rather than reproducing them itself. However, especially for rich, high-dimensional data, most anonymization techniques appear to cripple the utility of the data [8, 1]. By integrating the privacy guarantees into the application, we can provide it with unfettered access to the raw data, under the condition that its ultimate output—substantially less information that an entire data set—respect the privacy criteria.

2. RECOMMENDATION ALGORITHMS

We start with an introduction to some of the approaches applied to the Netflix prize. The approaches we consider were actual contenders at one point, but are understandably simpler than the current state of the art. While their level of accuracy has since been surpassed, we hope that by understanding their private adaptations we can derive methodol-

ogy that may continue to apply to the progressively more complex recommender systems.

The setting we consider has both users and items, with ratings for a subset of the (user, item) pairs. Given such a partial set of ratings, the goal is to predict certain held out values at specified (user, item) locations.

Global Effects. A common first step in these systems is to center the ratings by computing and subtracting average ratings for users and for items. To stabilize this computation, the average is often computed including an additional number of fictitious ratings at the global average; users and movies with many ratings drift to their correct average, but averages with small support are not allowed to overfit.

Covariance. Having factored first order effects, derived from properties of the ratings themselves, it is very common to look at correlations between items.[1] A common approach is to look at the covariance matrix of the items, whose (i, j) entry is the average product of ratings for items i and j across all users. Of course, relatively few users have actually rated both i and j, and so the average is taken across only those users who have rated both items.

Geometric Recommendation Algorithms. Oversimplifying tremendously, to a first approximation once we have computed the covariance matrix of the items we have enough information at hand to apply a large number of advanced learning and prediction algorithms. The covariance matrix encodes the complete geometric description of the items, and any geometric algorithm (eg: latent factor analysis, nearest neighbor approaches, geometric clustering, etc) can be deployed at this point. Importantly, from our perspective, these approaches can be applied for each user using only the covariance information and the user's collection of ratings. If the covariance measurement can be conducted privately, any algorithm that does not need to return to the raw data of other users can be deployed at this point with privacy guarantees. We will experiment with several, borrowing almost entirely from previous research published about the Netflix prize, but defer the discussion of the specific algorithms for now.

2.1 A Non-Private Approach

We will formalize the previous sketch into an algorithm that is non-private, but will form the skeleton of our privacy preserving approach. The steps in the algorithm may appear especially pedantic, but writing them in a simplistic form will allow us to adapt them easily to their private forms.

Following [4] we use r to refer to a collection of ratings, with the notation r_{ui} for the rating of user u for movie i and r_u for the vector of ratings for user u. We use the notation e_{ui} and e_u for the binary elements and vectors indicating the presence of ratings (allowing us to distinguish from reported zero values).

We start by subtracting the movie averages from each movie, where the average is dampened by a number β of ratings with the global average.

Movie Effects

1. For each item i, compute totals and counts:
   (a) Let $\text{MSum}_i = \sum_u r_{ui}$.
   (b) Let $\text{MCnt}_i = \sum_u e_{ui}$.
2. Compute global average $G = \sum_i \text{MSum}_i / \sum_i \text{MCnt}_i$.
3. For each item i, compute the stabilized average:
   (a) Let $\text{MAvg}_i = (\text{MSum}_i + \beta G)/(\text{MCnt}_i + \beta)$.
4. For each rating r_{ui}, subtract the appropriate average:
   (a) Set $r_{ui} = r_{ui} - \text{MAvg}_i$.

We perform exactly the same operation for the users, computing stabilized averages and subtracting the appropriate averages from each rating.

Our covariance computation is also direct, but for reasons that will become clear we will want to take a weighted combination of the contributions from each user, using a weight $0 \le w_u \le 1$ for user u:

Compute Covariance

1. For each movie-movie pair (i, j)
   (a) Let $\text{Cov}_{ij} = \sum_u w_u r_{ui} r_{uj}$.
   (b) Let $\text{Wgt}_{ij} = \sum_u w_u e_{ui} e_{uj}$.
   (c) Let $\text{Avg}_{ij} = \text{Cov}_{ij}/\text{Wgt}_{ij}$.

The matrix Avg now contains our estimate for the covariance matrix. We could then pass this matrix to one of a number of geometric approaches proposed by other researchers as being especially effective on the Netflix data set. While the choice of subsequent algorithm is obviously very important for the performance of the recommender system, we will not attempt to derive any privacy properties from their specifics. Rather, we providing them only with inputs that have been produced using differential privacy.

3. DIFFERENTIAL PRIVACY

Differential privacy [13], surveyed in [15], is a relatively recent privacy definition based on the principle that the output of a computation should not allow inference about any record's presence in or absence from the computation's input. Formally, it requires that for any outcome of a randomized computation, that outcome should be nearly equally likely with and without any one record.

We say two data sets A and B are *adjacent*, written $A \approx B$, if there is exactly one record in one but not in the other.

DEFINITION 1. *A randomized computation M satisfies ϵ-*differential privacy *if for any adjacent data sets A and B, and any subset S of possible outcomes Range(M),*

$$\mathbf{Pr}[M(A) \in S] \quad \le \quad \exp(\epsilon) \times \mathbf{Pr}[M(B) \in S] \; .$$

One interpretation of the guarantee differential privacy provides is that it bounds the ability to infer from any output event S, whether the input to the computation was A or B. From an arbitrary prior $p(A)$ and $p(B)$, we see that

$$\frac{p(A|S)}{p(B|S)} \quad = \quad \frac{p(A)}{p(B)} \times \frac{p(S|A)}{p(S|B)} \; .$$

When $A \approx B$, differential privacy bounds the update to the prior by a factor of $\exp(\epsilon)$, limiting the degree of inference possible about slight differences in the input data sets. Specifically, inference about the presence or absence (and consequently the value of) any single record is bounded by a factor of $\exp(\epsilon)$.

We stress that differential privacy is a property of the computation that produces the output, not of the output itself. At the same time, the probabilities are purely a function

[1] While user-user correlations may also be useful, they have proven less successful in the Netflix competition (and would be much more challenging to accomodate privately).

of the randomness of the computation, and not of possible randomness or uncertainty in the input.

There is a large volume of literature on privacy, and many other definitions and approaches have been suggested that provide other guarantees. Many of these, such as the popular k-anonymity, only provide syntactic guarantees on the outputs, without the semantic implications used above. Unlike the majority of these other approaches, differential privacy has proven resilient to attack, continuing to provide privacy guarantees for arbitrary prior knowledge, under repeated use, for arbitrary data types.

Several approaches have looked at weakened versions of differential privacy, exchanging the generality of the guarantee (protecting perhaps against only a subset of priors) for improved accuracy or usability [21]. Nonetheless, the techniques we outline here can provide the stronger guarantee, and it is not clear that the weakened definitions are needed (although, we will consider one of the relaxations next).

3.1 Approximate Differential Privacy

We will also consider a relaxed form of differential privacy that permits an additive term in the bound, as well as the multiplicative term, introduced in [16].

DEFINITION 2. *A randomized computation M satisfies (ϵ, δ)-*differential privacy *if for any adjacent data sets A and B, and any subset S of possible outcomes Range(M),*

$$\mathbf{Pr}[M(A) \in S] \quad \leq \quad \exp(\epsilon) \times \mathbf{Pr}[M(B) \in S] + \delta \; .$$

One interpretation of this guarantee is that the outcomes of the computation M are unlikely to provide much more information than for ϵ-differential privacy, but it is possible. For any $\gamma > \epsilon$, take S_γ to be the set of outcomes x for which

$$\frac{p(x|A)}{p(x|B)} \quad > \quad \exp(\gamma).$$

Combining this constraint with the definition of (ϵ, δ)-differential privacy, we can conclude that such outputs are unlikely:

$$p(S_\gamma|B) \quad \leq \quad p(S_\gamma|A) \quad \leq \quad \frac{\delta}{1 - \exp(\epsilon - \gamma)}.$$

While γ much larger than ϵ is possible, the probability is effectively bounded by δ. Moreover, for the privacy mechanisms we use (described in the next section), there will always be a trade-off between ϵ and δ; one can decrease either arbitrarily, at the expense of increasing the other. In a sense, the amount of information released (measured as the ratio of the two probabilities) is a random variable which is most likely to be small.

Importantly, approximate differential privacy satisfies sequential composition logic:

THEOREM 1. *If M_f and M_g satisfy (ϵ_f, δ_f) and (ϵ_g, δ_g) differential privacy, respectively, then their sequential composition satisfies $(\epsilon_f + \epsilon_g \, , \, \delta_f + \delta_g)$-differential privacy.*

We will use this theorem to be able to derive bounds on the end-to-end privacy guarantees of our recommender system, comprised of multiple independent (ϵ, δ)-differentially private computations.

3.2 Noise and Sensitivity

The simplest approach to differential privacy when computing numerical measurements is to apply random noise to the measurement, and argue that this masks the possible influence of a single record on the outcome. If we aim to compute a function $f\colon D^n \rightarrow \mathbb{R}^d$, the following results describe prior privacy results achieved through the addition of noise [17].

THEOREM 2. *Define $M(X)$ to be $f(X) + Laplace(0, \sigma)^d$. M provides ϵ-differential privacy whenever*

$$\sigma \quad \geq \quad \max_{A \approx B} \|f(A) - f(B)\|_1 / \epsilon \; .$$

We can achieve an approximate differential privacy guarantee using Gaussian noise, proportional to the smaller $\|\cdot\|_2$ distance between $f(A)$ and $f(B)$. Writing $N(\mu, \sigma^2)$ for the normal distribution with mean μ and variance σ^2. It is proven in [16] that

THEOREM 3. *Define $M(X)$ to be $f(X) + N(0, \sigma^2)^d$. M provides (ϵ, δ)-differential privacy, whenever*

$$\sigma \quad \geq \quad \sqrt{2 \ln(2/\delta)}/\epsilon \times \max_{A \approx B} \|f(A) - f(B)\|_2.$$

Notice that for any one parameter σ to the noise distribution, there are many valid settings for ϵ and δ. Specifically, for any positive ϵ there is a $\delta = \delta(\epsilon)$ associated with it. As such, we will often focus only on the σ value, without deriving specific (ϵ, δ) pairs.

3.3 Counts, Averages, and Covariances

There are relatively few statistics we will need to measure from the data to begin adapting recommendation algorithms from previous work. Global effects, such as per-movie averages and per-user averages, play an important role in prediction. Additionally, the movie-movie covariance matrix forms the basis of many geometric algorithms, and specifically the SVD factorization approaches and the kNN geometric distance approaches. Before continuing to the specifics of our approach, we see how these quantities can be measured in the previously described frameworks.

Counting and sums are relatively easy functions to analyze. If $f\colon D^n \rightarrow \mathbb{R}^d$ partitions the records (ratings) into d bins and counts the contents of each, for both $\|\cdot\|_1$ and $\|\cdot\|_2$

$$\max_{A \approx B} \|f(A) - f(B)\| \quad = \quad 1 \; .$$

Consequently, we can report counts of arbitrary partitions of the records (our interest is in ratings per movie) with appropriate additive noise providing privacy.

Sums are more complicated only in that we must explicitly constrain the range of values each element contributes. In the case of ratings, the scores initially range from 1 to 5, but this will grow and shrink as we apply various operations to the data. If a single record has maximum range at most B, then for both $\|\cdot\|_1$ and $\|\cdot\|_2$

$$\max_{A \approx B} \|f(A) - f(B)\| \quad \leq \quad B \; .$$

The most complex measurement we will take is the movie-movie covariance matrix. Simplifying a bit (specifically, ignoring weights for now), which we can the covariance matrix write as (using r_u for the rating vector for person u)

$$\text{Cov} \quad = \quad \sum_u r_u r_u^T \; .$$

This formulation makes it very clear that a single change to a single record can have a limited influence on the sum. If a single rating changes, changing from r_u^a to r_u^b, the difference between the two covariance matrices is

$$\begin{aligned}\|\mathrm{Cov}^a - \mathrm{Cov}^b\| &= \|r_u^a r_u^{aT} - r_u^b r_u^{bT}\| \\ &\leq \|r_u^a - r_u^b\| \times (\|r_u^a\| + \|r_u^b\|) \ .\end{aligned}$$

Taking $\|r_u^a - r_u^b\|$ to be one (corresponding to a single change),

$$\|\mathrm{Cov}^a - \mathrm{Cov}^b\| \leq \|r_u^a\| + \|r_u^b\| \ .$$

This bound may be large for users with many ratings, which is what leads us to introduce weights to the terms contributing to the covariance matrix. The weights will be selected to carefully normalize the contributions of each user, ensuring that the norms of the possible differences are at most a fixed constant. So normalized, we can simply compute and report the covariance matrix, with a fixed amount of additive noise applied to each of its entries. As we expect the magnitude of the values in the covariance matrix to grow linearly with the number of data points, this influence of this noise should intuitively diminish as the amount of training data grows.

4. ALGORITHM AND ANALYSIS

Our algorithm consists of several steps, measuring (with noise) progressively more challenging aspects of the data before feeding the measurements to appropriately parameterized variants of the currently top learning algorithms. We first describe the approach at a high level, before describing the sequence of precise calculations more concretely.

Global effects. We start with the noisy measurement of and baseline correction for various global effects. We first measure and publish the sum and count across all ratings to derive a global average. We then measure and publish, for each movie, the number and sum of ratings for that movie. We use these two quantities to produce a per-movie average, stabilized by including a number β_m of ratings at the global average. Finally, we remeasure the global average, as above, for upcoming use in centering each user's ratings. The algorithms and privacy implications for these steps are described in detail in Section 4.2.

We next invest some effort in preparing each user's ratings for covariance measurement. We do not want to release per-user statistics, such as the average rating for each user, as to do so with sufficient accuracy to be useful for learning would demolish our privacy guarantees. Instead, we will apply several transformations to a user's ratings before measurement, and argue that the transformations are such that privacy guarantees made of their outputs propagate to their inputs. Our specific operations include the centering of each user's ratings, again including a number of fictitious ratings at the global average, as well as a clamping of the resulting value to a more compact interval (increasing privacy, at the expense of error in outlying values).The algorithms and privacy implications for these steps are described in detail in Section 4.3.

Covariance matrix. We next measure the covariance matrix of the resulting user rating vectors. To achieve privacy, we incorporate noise into each coordinate, following [7].

As an example of the subtle nature of effectively integrating privacy, consider the computation of latent factors from geometric data. An important step in many geometric learning approaches, we might like to find a low dimensional subspace that best fits the data, when projected onto it, in terms of the mean squared error. There are several ways to compute such a space, but three otherwise equivalent approaches are to compute the SVD of the user $\times$ movie data matrix, compute the SVD of the movie $\times$ movie covariance matrix, and compute the SVD of the user $\times$ user Gram matrix.

While these three approaches are equivalent in non-private computation, they are very different when faced with the task of incorporating privacy. Consider the simple technique of adding noise to measurements to provide privacy: To mask the data matrix sufficiently, we must add noise to every entry in the matrix, in a process known as randomized response. While the independence of the noise leads to some amount of cancelation, the error in the system still grows with the number of participants. Adding noise to the covariance matrix scales with the number of movies involved, but does not need to grow as the number of participants increases, which gives the potential for arbitrarily accurate measurements for arbitrarily large populations. Working with the Gram matrix, with an entry for each pair of users, is a disaster; one must add enough noise to each entry (quadratic in the participants) proportional to the largest covariance any two users might have, linear in the number of movies. This quickly becomes unmanageably disruptive.

While the three techniques are similar without privacy constraints, there is a clear ordering on them when we need to introduce noise for privacy (covariance, data matrix, Gram matrix).

4.1 Notation

As before, r_{ui} stands for the rating of user u for movie i, r_u for the entire vector of ratings for user u, and similarly e_{ui} and e_u denote the binary elements and vectors indicating the presence of ratings (allowing us to distinguish from reported zero values). We use $c_u = \|e_u\|_1$ for the number of ratings by user u.

In our exposition, we will distinguish between private data and released data by using lower case and upper case, respectively. The reader should verify that whenever an upper case variable is assigned, it is a function only of upper case variables or lower case variables with noise added. When we add noise to a variable x, we simply write

$$\mathrm{X} = x + \mathrm{Noise},$$

where the distribution of the noise is yet unspecified. We then bound the amount by which the variable x could change under $\|\cdot\|_1$ and $\|\cdot\|_2$ with addition of a single rating to the data set, allowing for the addition of either Laplace or Gaussian noise, and providing privacy guarantees through Theorems 2 and 3 respectively.

4.2 Movie Effects

We start with a few global effects that are easy to measure and publish accurately without incurring substantial privacy cost. We first measure and publish the number of ratings present for each movie, and the sum or ratings for

each movie, with random noise added for privacy:

$$\begin{aligned}\text{GSum} &= \sum_{u,i} r_{ui} + \text{Noise},\\ \text{GCnt} &= \sum_{u,i} e_{ui} + \text{Noise}.\end{aligned}$$

We use these to derive a global average, G = GSum/GCnt. Next, we sum and count the number of ratings for each movie, using d dimensional vector sums.

$$\begin{aligned}\text{MSum} &= \sum_{u} r_{u} + \text{Noise}^d,\\ \text{MCnt} &= \sum_{u} e_{u} + \text{Noise}^d.\end{aligned}$$

We produce a stabilized per-movie average rating by introducing β_m fictitious ratings at value G for each movie:

$$\text{MAvg}_i = \frac{\text{MSum}_i + \beta_m \text{G}}{\text{MCnt}_i + \beta_m}.$$

With these averages now released, they can be incorporated arbitrarily into subsequent computation with no additional privacy cost. In particular, we can subtract the corresponding averages from the every rating to remove the per-movie global effects.

4.3 User Effects

Having published the average rating for each movie, we will subtract these averages from each rating before continuing. We then center the ratings for each user, taking an average again with a number β_p of fictitious ratings at the recomputed global average:

$$\overline{r}_u = \frac{\sum_i (r_{ui} - \text{MAvg}_i) + \beta_p \text{G}}{c_u + \beta_p}.$$

Unlike with movies, we do not report the averages, we will just subtract them from the appropriate ratings. We also clamp the resulting centered ratings to the interval $[-B, B]$, to lower the sensitivity of the measurements at the expense of the relatively few remaining large entries:

$$\widehat{r}_{ui} = \begin{cases} -B, & \text{if } r_{ui} - \overline{r}_u < -B,\\ r_{ui} - \overline{r}_u, & \text{if } -B \le r_{ui} - \overline{r}_u < B,\\ B, & \text{if } B \le r_{ui} - \overline{r}_u. \end{cases}$$

We now argue that the presence or absence of a single rating has a limited effect on this centering and clamping process.

THEOREM 4. *Let r^a and r^b differ on one rating, present in r^b. Let α be the maximum possible difference in ratings*[2]*. For centered and clamped ratings $\widehat{r}^a$ and $\widehat{r}^b$, we have*

$$\begin{aligned}\|\widehat{r}^a - \widehat{r}^b\|_1 &\le \alpha + B,\\ \|\widehat{r}^a - \widehat{r}^b\|_2^2 &\le \frac{\alpha^2}{4\beta_p} + B^2.\end{aligned}$$

PROOF. If r^a and r^b are two sets of ratings with a single new rating in r^b at r^b_{ui}, then $\widehat{r}^a$ and $\widehat{r}^b$ are everywhere equal, except for the ratings of user u. For the ratings in common between r^a and r^b, the difference is at most the difference in the subtracted averages:

$$|\overline{r}^b_u - \overline{r}^a_u| = \frac{|r_{ui} - \overline{r}^a_u|}{c^b_u + \beta_p} \le \frac{\alpha}{c^b_u + \beta_p}.$$

[2]For the Netflix Prize data set $\alpha = 4$.

For the new rating, r_{ui}, its previous contribution of zero is replaced with the new centered and clamped rating, at most B in magnitude. Therefore

$$\begin{aligned}\|\widehat{r}^a - \widehat{r}^b\|_1 &\le c^a_u \times \frac{\alpha}{c^b_u + \beta_p} + B,\\ \|\widehat{r}^a - \widehat{r}^b\|_2^2 &\le c^a_u \times \frac{\alpha^2}{(c^b_u + \beta_p)^2} + B^2.\end{aligned}$$

For $\|\cdot\|_2^2$, since $c^b_u = c^a_u + 1$ the first term is maximized at $c^a_u = \beta_p + 1$ and can be bounded from above by $\alpha^2/4\beta_p$. □

By choosing β_p sufficiently large we can drive the $\|\cdot\|_2$ difference arbitrarily close to B. The same is not true of $\|\cdot\|_1$, and we just cancel its c^a_u term with the denominator.

4.4 Covariance Matrix

The final measurement we make of the private data is the covariance of the centered and clamped user ratings vectors. However, we will want to take the average non-uniformly, using per-user weights w_u equal to the reciprocal of $\|e_u\|$. The choice of norm will determine the norm in which we derive a stability bound.

$$\begin{aligned}\text{Cov} &= \sum_u w_u \widehat{r}_u \widehat{r}_u^T + \text{Noise}^{d\times d},\\ \text{Wgt} &= \sum_u w_u e_u e_u^T + \text{Noise}^{d\times d}.\end{aligned}$$

Notice that if a single rating r_{ui} is in difference between r^a and r^b, only the terms contributed by user u contribute to a difference in the matrices. We now bound the norms of this difference using Theorem 4.

THEOREM 5. *Let ratings r^a and r^b have one rating in difference. Taking $w_u = 1/\|e_u\|_1$ we have*

$$\|w^a_u \widehat{r}^a_u \widehat{r}^{aT}_u - w^b_u \widehat{r}^b_u \widehat{r}^{bT}_u\|_1 \le 2B\alpha + 3B^2\ .$$

For β_p at least $\alpha^2/4B^2$, taking $w_u = 1/\|e_u\|_2$ we have

$$\|w^a_u \widehat{r}^a_u \widehat{r}^{aT}_u - w^b_u \widehat{r}^b_u \widehat{r}^{bT}_u\|_2 \le (1 + 2\sqrt{2})B^2\ .$$

PROOF. We rewrite the difference $w^a_u \widehat{r}^a_u \widehat{r}^{aT}_u - w^b_u \widehat{r}^b_u \widehat{r}^{bT}_u$ as

$$w^a_u \widehat{r}^a_u (\widehat{r}^a_u - \widehat{r}^b_u)^T + w^b_u (\widehat{r}^a_u - \widehat{r}^b_u) \widehat{r}^{bT}_u + (w^a_u - w^b_u) \widehat{r}^a_u \widehat{r}^{bT}_u.$$

For $\|\cdot\|_1$ and $\|\cdot\|_2$, as $\|e^b_u\| - \|e^a_u\| \le 1$, we have that

$$w^a_u - w^b_u = \frac{1}{\|e^a_u\|} - \frac{1}{\|e^b_u\|} \le \frac{1}{\|e^a_u\|\|e^b_u\|}\ .$$

The norm of the original matrix difference is bounded by

$$\left(\frac{\|\widehat{r}^a_i\|}{\|\widehat{e}^a_i\|} + \frac{\|\widehat{r}^b_i\|}{\|\widehat{e}^b_i\|}\right)\|\widehat{r}^a_i - \widehat{r}^b_i\| + \frac{\|\widehat{r}^a_i\|\|\widehat{r}^b_i\|}{\|\widehat{e}^a_i\|\|\widehat{e}^b_i\|}.$$

As $\|\widehat{r}_i\| \le \|\widehat{e}_i\| \times B$ for any norm, the normalizations cancel the norms of the ratings, giving the claim via Theorem 4. □

A similar result holds for the weight matrix, but can be optimized substantially as the e_i vectors do not undergo centering.

THEOREM 6. *Let ratings r^a and r^b have one rating in difference. Taking $w_u = 1/\|e_u\|_1$ we have*

$$\|w^a_u e^a_u e^{aT}_u - w^b_u e^b_u e^{bT}_u\|_1 \le 3\ .$$

Taking $w_u = 1/\|e_u\|_2$ we have

$$\|w^a_u e^a_u e^{aT}_u - w^b_u e^b_u e^{bT}_u\|_2 \le \sqrt{2}\ .$$

PROOF. Between the two weight matrices, $(c_p^a)^2$ entries change from w_p^a to w_p^b, and $2c_p^b - 1$ entries emerge with weight w_p^b. For $\|\cdot\|_1$, this bound is

$$\|w_u^a e_u^a e_u^{aT} - w_u^b e_u^b e_u^{bT}\|_1 \;\leq\; 3 - 2/c_p^b \;<\; 3.$$

For $\|\cdot\|_2$ and $c_p^a > 0$, we observe that the difference in weights $w_p^a - w_p^b$ is at most the derivative of $x^{-1/2}$ at c_p^a: $1/2(c_p^a)^{3/2}$, which implies

$$\begin{aligned}\|w_u^a e_u^a e_u^{aT} - w_u^b e_u^b e_u^{bT}\|_2^2 &\leq \frac{(c_p^a)^2}{4(c_p^a)^3} + \frac{(2c_p^b - 1)}{(c_p^b)^2} \\ &< 2.\end{aligned}$$

The case of $c_p^a = 0$ is handled by direct computation. □

4.5 Per-User Privacy

The mathematics we have done so far describe the amount of noise required to mask the presence or absence of a single rating. A stronger privacy guarantee would mask the presence or absence of an entire user, providing uniform privacy guarantees even for prolific movie raters.

To update the mathematics to provide per-user privacy we only need to apply a more aggressive down-weighting by the number of ratings, scaling each contribution down by $\|e_u\|$. For the contribution to sums and counts, a new user contributes exactly their weighted rating and count vectors:

$$\|\frac{r_u}{\|e_u\|}\| \;\leq\; \alpha \quad \text{and} \quad \|\frac{e_u}{\|e_u\|}\| \;\leq\; 1.$$

Likewise, the contribution to the covariance and weight matrices is exactly the new outer product of weighted vectors, whose norms are the square of the norms of the vectors:

$$\|\frac{\widehat{r}_u}{\|e_u\|}\|^2 \;\leq\; B^2 \quad \text{and} \quad \|\frac{e_u}{\|e_u\|}\|^2 \;\leq\; 1.$$

This normalization is much more aggressive than with per-rating privacy, and results in less accurate prediction and recommendation. However, it is still the case that the amount of noise remains fixed even as the number of users increases.

4.6 Cleaning the Covariance Matrix

The covariance matrix we have computed is somewhat noisy, and while we could hand it of to one of many recommendation algorithms, we will first clean it up a bit.

As a first step we apply the "shrinking to the average" method [4] with separate constants for the diagonal and off-diagonal entries, setting the matrix to

$$\overline{\text{Cov}}_{ij} \;=\; \frac{\text{Cov}_{ij} + \beta \cdot \text{avg Cov}}{\text{Wgt}_{ij} + \beta \cdot \text{avg Wgt}}.$$

There is substantial theoretical and empirical evidence that low rank matrix approximations (of the same form as the matrix factorization approaches) are highly effective at removing noise from matrices while retaining the significant linear structure. By computing a rank-k approximation to our covariance matrix, we can remove a substantial amount of "squared error" that we have introduced, without removing nearly as much from the underlying signal.

Before applying the rank-k approximation, we would like to unify, to the extent possible, the variances of the noise. Covariance entries with relatively fewer contributed terms have higher variance in their added noise (as it was divided by a smaller Wgt_{ij}). It has also been observed that the denoising of low rank approximations is most effective when the variances of the entries are equivalent (the error bounds, to a first approximation, scale with the maximum per-entry variance, and it causes no harm to scale up lesser entries while increasing the amount of "signal" each contributes).

To correct this, we borrow a technique from [12] and scale each entry Avg_{ij} upwards by a factor of $(\text{MCnt}_i\text{MCnt}_j)^{1/2}$ before applying the rank-k approximation. We then scale each entry down by the same factor, and return the results to our recommendation algorithm of choice.

One additional benefit of this step is that it produces a highly compressed representation of the covariance matrix, which can now be sent in its entirety to the client computers.

5. EVALUATION

We evaluate our approach on the Netflix Prize data set that consists of roughly 100M ratings of 17770 movies contributed by 480K people. By adjusting the parameters of the noise distributions we use, our computation will provide varying differential privacy guarantees, and its output will have measurable accuracy properties. The accuracy is measured by the root mean squared error (RMSE) on the qualifying set (3M ratings) and can be self-tested on the probe set with similar characteristics (1.5M ratings).

5.1 The Privacy v. Accuracy Tradeoff

While it is natural to parameterize differential privacy using a variety of (ϵ, δ) pairs, we simplify to a single parameter. For each measurement f_i, we will parameterize the magnitude of the noise we use as

$$\sigma_i \;=\; \max_{A \approx B} \|f_i(A) - f_i(B)\|/\theta_i \;,$$

where the θ_i are required to sum to a pre-specified value θ. In fact, we will take each θ_i to be a fixed fraction of θ, whose value we will take and vary as our single parameter.

By Theorem 2, using Laplace noise, measurement i provides ϵ_i-differential privacy for

$$\epsilon_i \;=\; \theta_i.$$

By Theorem 3, using Gaussian noise, measurement i provides (ϵ_i, δ_i)-differential privacy for

$$\epsilon_i \;=\; \theta_i \sqrt{2\ln(2/\delta_i)}.$$

As Theorem 1 tells us that ϵ and δ values add, our final guarantees have the form (for $\|\cdot\|_1$ and $\|\cdot\|_2$ respectively)

$$\epsilon \;=\; \sum_i \theta_i \quad \text{and} \quad \epsilon \;=\; \sum_i \theta_i \sqrt{2\ln(2/\delta_i)}$$

By taking a common value of δ_i, we can see that $\theta = \sum_i \theta_i$ scales the value of ϵ linearly. By varying θ, and thus the θ_i, we can add more or less noise to our measurements and provide more or less privacy, respectively. From any θ, we can reconstruct a range of (ϵ, δ) pairs.

Section 4 describes privacy-preserving computations of global effects and the covariance matrix. There are three important measurements our algorithm makes of the data: the global average, the per-movie averages, and the covariance matrix. For any θ, we will set the respective θ_i according to

$$\theta_1 \;=\; 0.02 \times \theta, \quad \theta_2 \;=\; 0.19 \times \theta, \quad \theta_3 \;=\; 0.79 \times \theta \;.$$

We choose θ_1 so small because every rating contributes to its computation, and even with substantial additive noise the resulting average is very accurate.

Given the covariance matrix and with global effects factored out, we apply the k-Nearest Neighbor (kNN) method of Bell and Koren [4] and the standard SVD-based prediction mechanism (SVD), both with ridge regression. We use the weight matrix Wgt as the similarity metric for kNN. Both methods can be preceded by the cleaning step (Section 4.6), which may improve or degrade performance depending on the value of the privacy parameter.

All global parameters are optimized for the value of $\theta = 0.15$ where both SVD and kNN preceded by the cleaning step match Netflix's Cinematch benchmark. The dimensionality for all algorithms is fixed at $k = 20$, the shrinking parameters $\beta_m = 15$ and $\beta_p = 20$, the clamping parameter $B = 1.0$.

The parameters for the cleaning step as well as for the kNN- and SVD-based recommendation mechanisms are trained for each data set and the privacy parameter separately, since it can be done with relatively few new measurements that depend on private data. We use the gradient descent method which repeatedly evaluates each mechanism with varying parameters. This fitting does not require use to re-measure the covariance matrix, and so we do not incur additional privacy cost there. However, we must evaluate the RMSE, which can be measured with excellent precision even with very low setting of the privacy parameter.

Our main findings are presented in Figure 1. As the value of θ, which is inversely proportional to σ and the qualitative amount of privacy, increases, so does accuracy of the recommendation algorithms. Both k-NN and SVD (both with cleansing) cross the Cinematch threshold at $\theta \approx 0.15$. Without the cleansing, both k-NN and SVD do pass the baseline, but with less noise and consequently privacy. While the post-processing "cleansing" of the covariance matrix helps substantially in the high noise (small θ) regime, it impairs the analysis when less noise is used. This is perhaps a consequence of optimizing the cleansing parameters for $\theta = 0.15$, and it is possible that a more delicate post processing could accommodate both regimes.

Corresponding graphs for the Laplace noise and user-level privacy appear in the full version of the paper.

5.2 Privacy v. Accuracy over Time

Our algorithms (like most differentially-private computations) introduce a fixed amount of error to any measurement, that is increasingly dominated by actual data records as the size of the data sets increase. With more and more users and ratings, we expect the additive error we introduce for any fixed value of θ to eventually vanish.

To explore how the loss due to the privacy-preserving property of the recommender mechanism decreases with the amount of available data (for the fixed value of $\theta = 0.15$), we simulated the data gathering process at different times between 2000 and 2006 (including the peculiar property of including users with fewer than 20 ratings). Consistently with the Netflix Prize data set, the probe set was the 9 most recent ratings for each user chosen with probability 1/3 each. Fig. 2 plots the difference in RMSE (as percentage points) between privacy-preserving k-NN (after scaling) and the same algorithm without privacy guards.

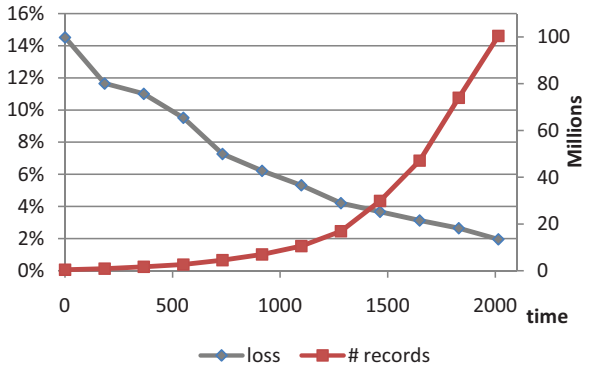

Figure 2: Left scale—accuracy loss, right scale—the number of records. The x-axis is the number of days elapsed since 7/1/2000.

6. CONCLUSIONS AND FUTURE WORK

We conclude that a recommendation system with differential privacy guarantees is feasible without taking significant hit in the recommendations accuracy. The loss in accuracy (for a fixed value of the privacy parameter) decreases as more data becomes available.

In our experiments we fixed several parameters that had the potential to vary freely, and it is natural to expect that more in-depth experimentation could lead to noticeably improved prediction accuracy. The chosen dimensionalities, smoothing weights, and distribution of "accuracy" θ_i between the measurements could be adjusted and possibly improved.

Directions for future work include efficient methods for direct privacy-preserving computations of latent factors and incorporation in the differential privacy framework of advanced methods for collaborative filtering that do not immediately admit factorization into two phases such as the integrated model of [19].

7. REFERENCES

[1] C. C. Aggarwal. On k-anonymity and the curse of dimensionality. In K. Böhm, C. S. Jensen, L. M. Haas, M. L. Kersten, P.-Å. Larson, and B. C. Ooi, editors, *VLDB*, pages 901–909. ACM, 2005.

[2] E. Aïmeur, G. Brassard, J. M. Fernandez, and F. S. M. Onana. Alambic: a privacy-preserving recommender system for electronic commerce. *Int. J. Information Security*, 7(5):307–334, 2008.

[3] E. Aïmeur, G. Brassard, J. M. Fernandez, F. S. M. Onana, and Z. Rakowski. Experimental demonstration of a hybrid privacy-preserving recommender system. In *ARES*, pages 161–170. IEEE Computer Society, 2008.

[4] R. M. Bell and Y. Koren. Scalable collaborative filtering with jointly derived neighborhood interpolation weights. In *ICDM*, pages 43–52. IEEE Computer Society, 2007.

[5] R. M. Bell, Y. Koren, and C. Volinsky. The BellKor solution to the Netflix Prize. Available from `http://www.netflixprize.com`, 2007.

[6] R. M. Bell, Y. Koren, and C. Volinsky. The BellKor 2008 solution to the Netflix Prize. Available from `http://www.netflixprize.com`, 2008.

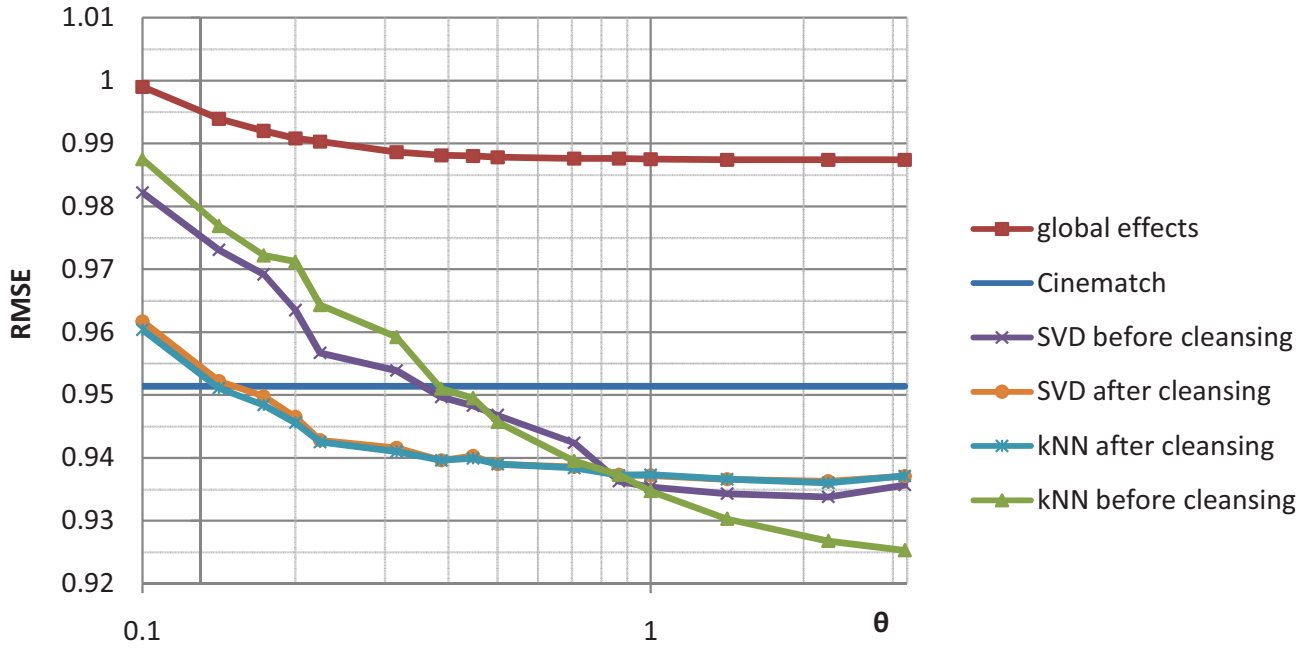

Figure 1: RMSE for four algorithms as a function of $\theta \propto 1/\sigma$ on the Netflix Prize set.

[7] A. Blum, C. Dwork, F. McSherry, and K. Nissim. Practical privacy: the SuLQ framework. In C. Li, editor, *PODS*, pages 128–138. ACM, 2005.

[8] J. Brickell and V. Shmatikov. The cost of privacy: destruction of data-mining utility in anonymized data publishing. In Li et al. [20], pages 70–78.

[9] J. Calandrino, A. Narayanan, E. Felten, and V. Shmatikov. Don't review that book: Privacy risks of collaborative filtering. Manuscript, 2009.

[10] J. F. Canny. Collaborative filtering with privacy. In *IEEE Symposium on Security and Privacy*, pages 45–57, 2002.

[11] J. F. Canny. Collaborative filtering with privacy via factor analysis. In *SIGIR*, pages 238–245. ACM, 2002.

[12] A. Dasgupta, J. E. Hopcroft, and F. McSherry. Spectral analysis of random graphs with skewed degree distributions. In *FOCS*, pages 602–610. IEEE Computer Society, 2004.

[13] C. Dwork. Differential privacy. Invited talk. In *Automata, Languages and Programming—ICALP (2)*, volume 4052 of *Lecture Notes in Computer Science*, pages 1–12. Springer, 2006.

[14] C. Dwork. An ad omnia approach to defining and achieving private data analysis. In F. Bonchi, E. Ferrari, B. Malin, and Y. Saygin, editors, *PinKDD*, volume 4890 of *Lecture Notes in Computer Science*, pages 1–13. Springer, 2007.

[15] C. Dwork. Differential privacy: A survey of results. In M. Agrawal, D.-Z. Du, Z. Duan, and A. Li, editors, *Theory and Applications of Models of Computation, 5th International Conference, TAMC 2008, Xi'an, China, April 25-29, 2008. Proceedings*, volume 4978 of *Lecture Notes in Computer Science*, pages 1–19. Springer, 2008.

[16] C. Dwork, K. Kenthapadi, F. McSherry, I. Mironov, and M. Naor. Our data, ourselves: Privacy via distributed noise generation. In S. Vaudenay, editor, *Advances in Cryptology—EUROCRYPT 2006*, volume 4004 of *Lecture Notes in Computer Science*, pages 486–503. Springer, May 2006.

[17] C. Dwork, F. McSherry, K. Nissim, and A. Smith. Calibrating noise to sensitivity in private data analysis. In S. Halevi and T. Rabin, editors, *Theory of Cryptography Conference—TCC 2006*, volume 3876 of *Lecture Notes in Computer Science*, pages 265–284. Springer, 2006.

[18] S. R. Ganta, S. P. Kasiviswanathan, and A. Smith. Composition attacks and auxiliary information in data privacy. In Li et al. [20], pages 265–273.

[19] Y. Koren. Factorization meets the neighborhood: a multifaceted collaborative filtering model. In Li et al. [20], pages 426–434.

[20] Y. Li, B. Liu, and S. Sarawagi, editors. *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Las Vegas, Nevada, USA, August 24-27, 2008*. ACM, 2008.

[21] A. Machanavajjhala, D. Kifer, J. M. Abowd, J. Gehrke, and L. Vilhuber. Privacy: Theory meets practice on the map. In *ICDE*, pages 277–286. IEEE, 2008.

[22] A. Narayanan and V. Shmatikov. Robust de-anonymization of large sparse datasets. In *IEEE Symposium on Security and Privacy*, pages 111–125. IEEE Computer Society, 2008.

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