PriView: Media Consumption and Recommendation Meet Privacy Against Inference Attacks

Amy Zhang$^{(1)}$, Sandilya Bhamidipati$^{(2)}$, Nadia Fawaz$^{(2)}$, Branislav Kveton$^{(2)}$
$^{(1)}$ UC San Diego, CA, amy.zhang@ucsd.edu
$^{(2)}$ Technicolor, Palo Alto, CA, {sandilya.bhamidipati, nadia.fawaz, branislav.kveton }@technicolor.com

***Abstract*—We propose PriView, an interactive privacy-preserving personalized video consumption system, that protects a user's privacy while delivering relevant content recommendations to the user. PriView provides the user with three functionalities: transparency on privacy risk, control of privacy risk, and personalized content recommendations. PriView bridges privacy theory and practice: it successfully implements an information theoretic framework to design a utility-aware privacy-preserving mapping that perturbs a user's video ratings to prevent inference of user private attributes, e.g. political views, age, gender, while maintaining the utility of the released perturbed ratings for recommendation. Our model uses convex optimization to learn a probability mapping from actual ratings to perturbed ratings that minimizes distortion subject to a privacy constraint. One practical challenge of the optimization is scalability, when the size of the underlying alphabet of the user data is very large, e.g. due to a large number of features representing the data. To reduce the optimization size, we introduce a quantization step that allows to control the number of optimization variables, and explore using low rank approximations of the rating matrix. Evaluations on the Politics and TV dataset show that these methods can achieve perfect privacy with little change in recommendation quality.**

I. Introduction

With the advent of targeted advertising and the popularity of mining user data, users find their privacy threatened. To address this rising concern, many privacy-preserving mechanisms have been proposed [1]. Most of these mechanisms have strong theoretical guarantees, but often lack practicality. For instance, reaching a sufficiently high level of privacy often requires that the user data be distorted to the point where it is not usable. PriView demonstrates an interactive privacy system, which brings together theory and practice, and shows how information-theoretic privacy can lead to practical policies for protecting user profiles, while maintaining the utility of sanitized data.

We consider the setting where the user has two kinds of data: some data A that should remain private, such as the user's political views, age, gender; and some data B that the user wishes to release to a service provider in exchange for some utility, such as video ratings to get content recommendations. As these two kinds of data are correlated, releasing video ratings could potentially lead to revealing user's private data through inference attacks. Indeed, large scale surveys [2], [3] have shown that the audiences for a number of TV shows can be distinctly characterized.

A. Contributions

PriView is an interactive privacy-preserving system for video consumption and recommendation that provides a user with **privacy transparency** and **control**, while maintaining the **quality of recommendations** the user receives. The PriView system shows the risk of releasing data related to media preferences (e.g. tv show viewing) with respect to private attributes (e.g. political views, age, gender), and how these risks can be controlled while providing relevant personalized recommendations to a user. The modules of PriView:

- help the user monitor his privacy status,
- inform the user, before he releases data, of the potential threat that he may incur by releasing such data,
- give means to the user to control the information leaked on private attributes by the released data, and inform him of the measures taken to ensure his privacy,
- maintain the quality and relevance of personalized recommendations while preserving privacy.

PriView has the potential to be interfaced with online video services, as well as TV and VoD services. It could also be extended to other media content, e.g. music, books, news, and to other products, services, or locations rated online by users.

Implementing each module of PriView required solving practical challenges. More precisely, PriView implements an information-theoretic privacy-utility framework [4], that successfully reduces the privacy risk to zero while maintaining utility of the sanitized data released by the user. The first practical challenge, encountered when implementing this privacy-utility framework, is that of scalability. The framework relies on convex optimization to generate a probability mapping that perturb video ratings, however the number of optimization variables in the original framework grows quadratically with the number of possible rating vectors to map to. For instance, for rating vectors of length 50 shows, and a 0-5 star rating scale, there exist 6^{50} possible rating vectors, thus 6^{100} optimization variables. We reduce the number of optimization variables by introducing a quantization step prior to the optimization [5]. Clustering techniques, e.g. K-means clustering, are used to quantize rating vectors into K clusters, prior to solving the privacy optimization, in order to reduce the size of the optimization to K^2. The second challenge is that the framework requires estimating the prior distribution between private data A and data to be released B, yet the *Politics and TV* dataset that we use to estimate the prior distribution is a relatively small dataset. Thanks to the quantization step, we adapted the framework to use the prior distribution between private data and quantized data, and used kernel density estimation to smooth our distribution estimate.

We analyze the performance of our system in terms of privacy and utility. First, the mutual information between the user's private data and perturbed data $I(A;\hat{B})$ is used as a privacy measure. We show that our privacy mapping from B to $\hat{B}$ successfully breaks the existing correlation between A and B by bringing the mutual information $I(A;\hat{B})$ to 0, which is equivalent to statistical independence between the private variable A and the distorted data $\hat{B}$. Consequently, inference attacks that try to infer A from $\hat{B}$ fail, as shown by our evaluations. The second performance metric measures the utility at the recommendation end, by comparing the root mean square error (RMSE) of rating prediction based on actual and privatized ratings. Our evaluations show that PriView succeeds in ensuring perfect privacy while maintaining the quality of recommendations.

B. Related Work

To the best of our knowledge, PriView is the first practical interactive system which *informs* a user of the privacy risks for multiple sensitive attributes (political views, age, gender) from *any inference* attack *prior* to the release of the user's data (ratings) to a service provider, and that gives means to the user to continuously monitor and control his privacy risk through data perturbation. PriView is the first practical system to implement an information-theoretic privacy-utility framework [4] all the way from the user end to the recommender end–and thus to be supported by strong theoretical guarantees against any inference attack algorithm. PriView successfully runs a matrix-factorization recommender system on top of privatized data.

Privacy-utility tradeoffs have been studied under either a local privacy setting, or a centralized privacy setting.

In the local privacy setting, users do not trust the entity aggregating data. Thus, each user holds her data locally, and processes it according to a privacy-preserving mechanism before releasing it to the aggregator. Local privacy dates back to randomized response in surveys [6], and has been considered in privacy for data mining and statistics [4], [5], [7]–[14]. The setup we consider falls under the local privacy setting, since the service provider is assumed to be untrusted, and users wish to protect against statistical inference of private information from data they release to the service provider. Local privacy has also been considered in the differential privacy [15], [16] corpus, e.g. for learning concept classes [12], clustering [13], and statistical parameter estimation [14]. These works are concerned with the problem of learning aggregate statistical properties from the data of several users. In contrast, we focus on devising content recommendations for an individual user while maintaining the privacy of this individual user's attributes.

In the centralized privacy setting, a trusted entity aggregates data from users in a database, while an untrusted analyst asks queries on the database. The trusted aggregator jointly processes data from multiple users according to a centralized privacy-preserving mechanism to produce a privatized answer to the query, that is released to the analyst. The centralized privacy setting is less stringent than the local privacy setting. Information theoretic frameworks have been used to analyze privacy-utility tradeoffs in the centralized database setting. One line of work [17], [18] asymptotically characterizes rate-distortion-equivocation regions as the number of data samples grows large. Traditionally, many differential privacy works assumed a centralized setting with a trusted database owner, and focused on making the output of an application running on the database differentially private, e.g. data mining [19], social recommendations [20] and recommender systems [21]. More specifically, [21] considers the case of a trusted recommender system who has access to ratings from privacy-conscious users, and addresses the challenge of training a differentially-private recommendation algorithm based on these original ratings. In contrast, we study a local privacy setup where the recommender system is not trusted by privacy-conscious users, who wish to protect against statistical inference of private information from data they release to the recommender. We also assume that the recommender system already owns a recommendation algorithm, trained on ratings from non-privacy conscious users, and we address the privacy challenges faced by any privacy-conscious user who wishes to use the recommender system.

II. Setting And System

A. Privacy-Utility Framework

We consider the local privacy setting described in [4], [5], [10], where a user has two types of data: some data that he would like to remain private, e.g. his political views, age, gender, and some data that he is willing to release publicly and from which he will derive some utility, for example the release of his media preferences (TV show ratings) to a service provider allows the user to receive content recommendations. We denote by A the vector of personal attributes that the user wants to keep private, and by B the vector of data he is willing to make public. We assume that the user private attributes A are linked to his data B by the joint probability distribution $p_{A,B}$. Thus, an adversary— the service provider or a third party with whom he may exchange data— who would observe B could infer some information about A from B.

To reduce this inference threat, instead of releasing B, the user will release a *distorted version* of B, denoted $\hat{B}$, generated according to a conditional probabilistic mapping $p_{\hat{B}|B}$, called the *privacy-preserving mapping*. The privacy-preserving mapping $p_{\hat{B}|B}$ should be designed in such a way that it renders any statistical inference of A based on the observation of $\hat{B}$ harder, yet, at the same time, preserves some utility to the released data $\hat{B}$, by limiting the distortion generated by the mapping. We adopt the privacy-utility framework in [4], where the privacy-preserving mapping is designed to control the privacy leakage, modeled as the mutual information $I(A;\hat{B})$ between the private attributes A and the publicly released data $\hat{B}$, subject to a utility requirement, modeled by a constraint on the average distortion $E_{B,\hat{B}}[d(B,\hat{B})]$. We focus on *perfect privacy* $I(A;\hat{B}) = 0$: the privacy-preserving mapping $p_{\hat{B}|B}$ renders the released data $\hat{B}$ statistically independent from the private data A, and *any* inference algorithm that tries to infer the private data A from the released data $\hat{B}$ cannot outperform an uninformed random guess. The privacy and utility metrics, and the design of the privacy-preserving mapping, are discussed in greater details in Section III.

We would like to point out that in the local setting, perfect privacy $I(A;\hat{B}) = 0$ is equivalent to statistical independence between A and $\hat{B}$, i.e. $p_{\hat{B}|A}(\hat{b}|a) = p_{\hat{B}|A}(\hat{b}|a') = p_{\hat{B}}(\hat{b})$, for all a, a' and $\hat{b}$, which in turn is equivalent to $\hat{B}$ being locally 0-differential private with respect to A. Indeed, in the local setting, on one hand the local database A is of size 1 as it contains only the data of a single individual user, thus all databases a, a' are neighboring databases [12], [14]; on the other hand, the service provider asks for the query B which due to its correlation with A can be considered as a randomized function of A, and receives the sanitized version $\hat{B}$. Thus, in the local privacy setting at perfect privacy, the information theoretic privacy metric and the differential privacy metric are equivalent with respect to private data A [22].

B. System functionalities

Based on research on information theoretic privacy, PriView showcases a service that allows the user to release data about his media consumption (e.g. TV viewing habits) to get content recommendations, while ensuring that attributes he deems sensitive (e.g. political views) and wants to remain private, are protected against inference attacks. PriView system provides the following functionalities:

- **Transparency**: On one hand, the system allows the user to monitor his privacy status through a privacy dashboard. On the other hand, the system informs the user about the potential increase in privacy risk from releasing additional pieces of data, e.g. new ratings.
- **Control**: First, the system allows the user to select which attributes (political views, age, gender) he would like to remain private. Second, the system implements a privacy-preserving mechanism for the release of the user TV show ratings to a service provider, that ensures perfect privacy $(I(A;\hat{B}) = 0)$ against statistical inference of his private features [4], while at the same time minimizing the distortion to the released data. Third, a TV show history log allows the user to know at all time his true rating for a show, and what the distorted released rating was, to protect his privacy.
- **Personalized Recommendations**: the service provider sends content recommendations to the user, based on the released ratings. The

demonstration shows how the recommendations obtained when privacy is activated compare with the recommendations the user would have obtained if he did not activate privacy protection. It demonstrates that utility can be maintained while ensuring privacy.

Providing these functionalities require addressing technical challenges, that we describe in details in Section III.

C. Dataset

The PriView system makes use of the *Politics-and-TV* dataset [5], which contains data on political views and TV preferences of viewers in the USA in Fall 2012. The dataset contains entries for 1,218 users, broken into 744 Democrats, and 474 Republicans. For each user, the dataset entry is a vector $[\text{age}, \text{gender}, \text{State}, \text{politics}, B_1, \dots B_{50}]$ where $B_i \in \{0, 1, \dots, 5\}$ is the user's 5-star rating for TV show i if the user rated the show, and 0 otherwise, for a total of 50 TV shows in 6 categories: Sitcoms, Reality Shows, TV series, Talk Shows, News, and Sports.

D. System Architecture

The system consists of three components: a user client, a privacy server, and a recommendation server. The client is a web interface written in HTML5 and javascript. The servers are written in flask, which is a python based micro web framework. The user client has three roles: Let the user interact with various privacy settings; Let the user watch and provide ratings for TV shows; Display recommendations based on the user's privacy settings and privatized ratings. Both the privacy and the recommendation servers serve client requests (web pages), and store and fetch data from databases (user and privacy mapping data for the privacy server, content and recommender system data for the recommendation server). Additionally, the privacy server performs rating privatization based on the user's privacy settings, and send privatized ratings to the recommendation server, and to the user client. On the other hand, the recommendation server generates recommendations based on the user's privatized ratings, and send them to the user client.

Finally, four types of data collections (tables) are stored in MongoDB databases. One collection stores the user privacy settings and user interactions with the content (e.g. ratings), while another collection stores data related to the privacy mapping. Both are accessed by the privacy server. A third collection stores the content metadata used to display on the web interface at the client side, while the last collection stores content profiles for recommendation purposes. These two collections are accessed by the recommendation server.

III. PriView Overview

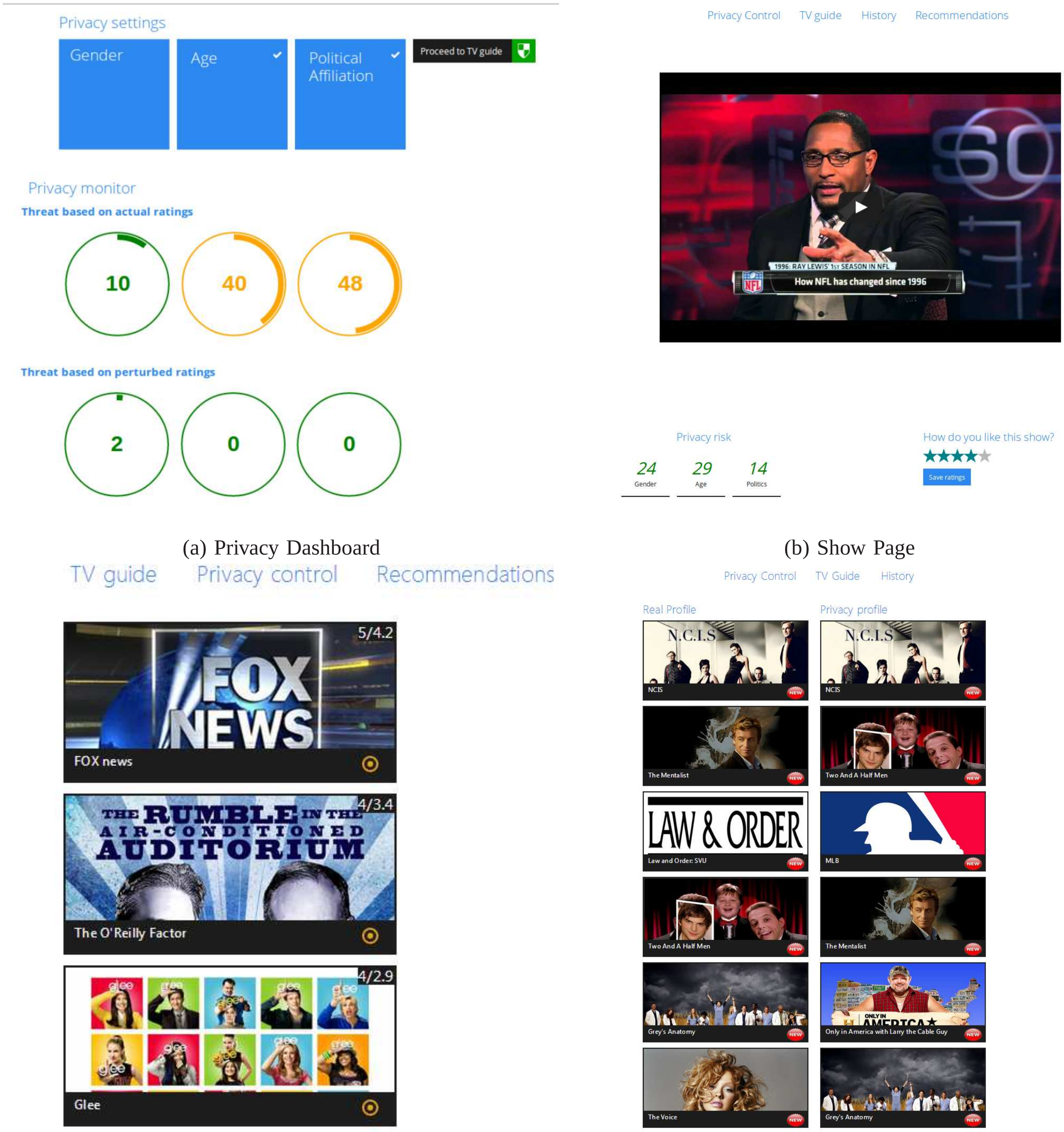
A. Transparency: Informing users about privacy risks

Privacy monitoring dashboard: The privacy dashboard, in Fig. 1a, contains the privacy settings of the user, and the privacy monitor. The *privacy settings* allow the user to select any combination of three attributes— age, gender, and political views— that he deems private and would like to protect. It should be noted that the user does not need to reveal what his political view, age, or gender are, but only whether he considers any of these features as sensitive information that he wants to remain private. The *privacy monitor* shows the inference threat for each private attribute from the actual TV show ratings of the user, and from the distorted privacy-preserving ratings. Thus, the privacy monitor allows the user to compare what his risk would have been if he did not activate privacy protection, with his risk after the privacy-preserving mechanism sanitized his ratings.

To model the inference threat for each private attribute from a particular rating vector representing the user history of ratings, we propose a privacy risk metric on a scale [0,100]. For a private attribute A and a specific vector of ratings $B = b$, we define the privacy risk by

$$\text{Risk}(A, b) = \left(1 - \frac{H(A|B = b)}{H(A)}\right)^+ * 100, \qquad (1)$$

where $H(A) = -\sum_a p_A(a) \log p_A(a)$ denotes the entropy of the variable A distributed according to $p_A(a)$, and represents the inherent uncertainty on A. Similarly, $H(A|B = b) = -\sum_a p_{A|b}(a|b) \log p_{A|b}(a|b)$ denotes the remaining entropy of A given the observation $B = b$, and represents the remaining uncertainty on A after observing $B = b$. Intuitively, the privacy risk $\text{Risk}(A, b)$ measures the percentage by which the uncertainty on A decreases due to the observation of $B = b$, relative to the original uncertainty prior to observing B. A privacy $\text{Risk}(A, b) = 0$ means that the rating vector $B = b$ does not provide any information about the private attribute A, while a $\text{Risk}(A, b) = 100$ implies that no uncertainty is left about the attribute A from observing the rating vector $B = b$. The privacy risk based on the user's actual rating vector $B = b$ is $\text{Risk}(A, b)$, while the privacy risk based on the distorted ratings $\hat{B} = \hat{b}$ is $\text{Risk}(A, \hat{b})$, and is obtained by replacing $B = b$ in (1) by $\hat{B} = \hat{b}$. Note that the mutual information between the private data A



(a) Privacy Dashboard

(b) Show Page

(c) History Page

(d) Recommendations Page

Fig. 1: Sample PriView Screenshots

and the distorted data $\hat{B}$ is

$$I(A;\hat{B}) = H(A)\left(1 - E_{\hat{B}}\left[\frac{H(A|\hat{B}=\hat{b})}{H(A)}\right]\right)$$

which is related to the average of the privacy risks over all possible distorted rating vectors $\hat{B}$. Achieving perfect privacy $(I(A;\hat{B}) = 0)$ ensures a 0-privacy risk, meaning that any inference algorithm that would try to infer A from $\hat{B}$ would not outperform an uninformed random guess.

Instantaneous privacy risk information: After completing his privacy settings, the user can move to the TV guide (not shown for the sake of conciseness), and pick a show that he would like to watch. On each TV show page, e.g. Fig. 1b, the user can give a star rating to the show. Prior to rating this new show, a privacy risk tool reminds the user of his privacy risk based on his current history of actual ratings. When the user hovers above the stars for this new show, for each possible rating in {1,..,5}, the privacy risk tool dynamically updates its numbers to inform the user of how his privacy risk would evolve if he added a particular rating. It should be noted that this privacy risk tool shows the risk based on actual ratings, before sanitization. Once the user picks a rating, and submits it to the system, the privacy-preserving mechanism operates on the rating vector to sanitize it. The privacy dashboard mentioned earlier allows the user to check that the privacy risk after distortion of the ratings is 0 for the attributes he selected as private.

B. Control: privacy-preserving mechanism

Privacy-preserving mechanism: Based on the privacy-utility framework in [4], the system implements a privacy-preserving mechanism for the release of the user TV show ratings to a service provider, that ensures perfect privacy $(I(A;\hat{B}) = 0)$ against statistical inference of his private features [4], while at the same time minimizing the distortion to the released data. The TV show history page in Fig. 1c shows the user's actual ratings and the perturbed ratings generated by PriView. While implementing the privacy-utility framework, we encountered technical challenges, that we describe below and that required adapting the framework.

Challenge: Scalability: Designing the privacy-preserving mapping $p_{\hat{B}|B}$ requires characterizing the value of $p_{\hat{B}|B}(\hat{b}|b)$ for all possible pairs $(b,\hat{b}) \in \mathcal{B} \times \hat{\mathcal{B}}$, i.e. solving the convex optimization problem over $|\mathcal{B}||\hat{\mathcal{B}}|$ variables. When $\hat{\mathcal{B}} = \mathcal{B}$, and the size of the alphabet $|\mathcal{B}| = 6^{50}$ is large, solving the convex optimization over $|\mathcal{B}|^2$ variables may become intractable. Quantization was proposed in [5] as a method to reduce the number of optimization variables, from $|\mathcal{B}|^2$ to K^2, where K denotes the number of quantization levels. It should be noted that the choice of K is a tradeoff between the size of the optimization, and the additional distortion introduced by quantization.

Algorithm 1 Quantized privacy-preserving mapping

```
Input: prior $p_{A,C}$
Solve: convex optimization
    $\underset{p_{\hat{B}|C}}{\text{minimize}} \quad \mathbb{E}_{p_{C,\hat{B}}}\left[d(C,\hat{B})\right]$
    subject to $\quad I(A;\hat{B}) \leq \epsilon, \quad$ and $\quad p_{\hat{B}|C} \in \text{Simplex}$
Remap : $p_{\hat{B}|B} \Leftarrow p_{\hat{B}|C(B)}$
Output: mapping $p_{\hat{B}|B}$
```

Quantization assumes that vectors B lie in a metric space. Directly applying quantization on the original rating vector B as in [5], where unrated shows are assigned a 0 rating, would make our model perceive unrated shows as strongly disliked by the user, when they actually may not be disliked, but simply unknown to the user for example. To circumvent this issue, we propose to first complete the rating vector B into B_c using low rank matrix factorization, a standard collaborative filtering technique. We then feed the completed rating vector B_c to the quantization module that maps B_c to a cluster center C. For quantization, we used K-means clustering, with $K = 75$ cluster centers, where our choice of K was guided empirically. The cluster center C is then fed to the privacy optimization algorithm, that finally outputs a distorted rating vector $\hat{B}$. In summary, the design of the privacy-preserving mapping, described in Algorithm 1, follows the Markov chain $A \rightarrow B \rightarrow B_c \rightarrow C \rightarrow \hat{B}$.

Challenge: Estimating the prior distribution: Computing the privacy $Risk(A,b)$, as well as finding the privacy-preserving mapping as the solution to the privacy convex optimization in [4], rely on the fundamental assumption that the prior distribution $p_{A,B}$ that links private attributes A and data B is known and can be fed as an input to the algorithm. In practice, the true distribution may not be known, but may rather be estimated from a set of sample data that can be observed, for example from a set of users who do not have privacy concerns and publicly release both their attributes A and their original data B. However, the dataset may contain a small number of samples,

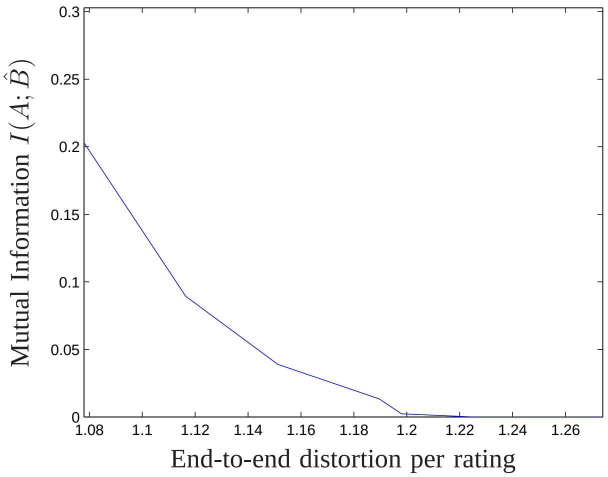


Fig. 2: Privacy-Utility Tradeoff

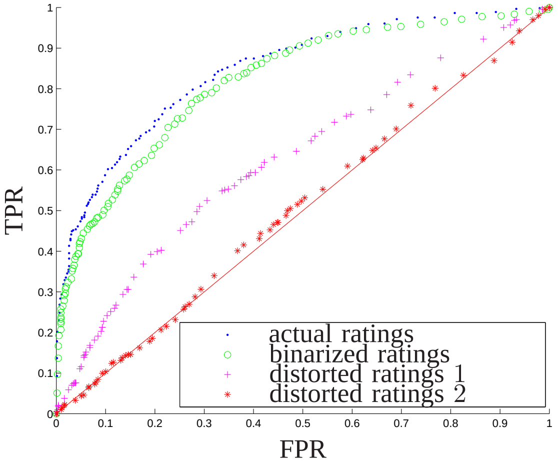


Fig. 3: ROC curve Logistic Regression of political views from TV show ratings

or be incomplete, which makes the estimation of the prior distribution challenging. Thanks to the completion and quantization step, we adapted the framework in [4] to use the prior distribution between private data and quantized completed data in Algorithm 1. We estimated the distribution using Kernel Density Estimation, with a Gaussian kernel with width $\sigma = 9.5$.

Evaluation: In Algorithm 1, ϵ bounds the amount of information about the private data A that is leaked by the distorted data $\hat{B}$, and thus represents the level of privacy requirement on the user side. Varying ϵ allows to study the tradeoff between privacy requirement and distortion. Fig. 2 shows the privacy-utility tradeoff: mutual information $I(A;\hat{B})$ against end-to-end distortion (quantization + privacy mapping) per rating. K-means quantization introduces a distortion 1.08 per rating and yields a mutual information $I(A;C) = 0.2$. With 0.14 additional distortion, the privacy-preserving mapping achieves perfect privacy $I(A;\hat{B}) = 0$ for an end-to-end distortion of 1.22.

As mentioned previously, PriView system focuses on perfect privacy $I(A;\hat{B}) = 0$, thus on ϵ close to 0. At perfect privacy, any inference algorithm that tries to infer A from $\hat{B}$ can only perform as well as an uninformed random guess. Intuitively, $\hat{B}$ is statistically independent from A, thus the privacy mapping statistically 'erased' any information about the private data A from $\hat{B}$, and an inference algorithm that tries to infer A from $\hat{B}$ can only perform as well as an uninformed inference algorithm that would try to infer A without knowledge of $\hat{B}$. Fig. 3 is an ROC curve showing the performance of an example logistic regression classifier, that tries to infer the user's political views from the original rating vector (blue curve), from a binarized version of the rating vector where ratings $>= 4$ are mapped to 1 (like), and ratings $<= 3$ are mapped to zero (dislike), or from rating vectors distorted according to a privacy-preserving mapping with average distortion $<= 1$ (pink curve), or distortion $<= 2$ (red curve). We used 10-fold cross validation, and plot the false positive rate (Democrats falsely classified as Republicans) against the true positive rate (Republicans correctly classified). The blue curve illustrates the privacy risk on inferring the political views from the original rating vectors. The green curve is close to the blue curve, and shows that merely binarizing the ratings is not enough to ensure privacy. The red curve is very close to the red diagonal line, which represents an uninformed random guess: this proves that with distortion $<= 2$, the privacy-preserving mechanism successfully ensures perfect privacy against logistic regression of political views from distorted ratings. We conducted further inference attacks with other classifiers, including Naive Bayes, and SVM, and observed similar results, as predicted by theory.

C. Utility: maintaining the quality of service running on privatized data

A natural question is whether the relevance of recommendations can be preserved when recommendations are obtained based on ratings distorted for privacy. The recommendations page of PriView, a sample of which is shown in Fig. 1d, allows to compare the top 6 TV show recommendations, based on the actual ratings, and based on the distorted ratings for privacy. The recommendation engine implemented in PriView uses low rank matrix factorization (MF) [24], a standard collaborative filtering method, to predict missing show ratings from ratings provided by the user for some other shows. We trained the MF recommender engine by alternating regularized least square [24]. Fig. 1d shows an overlap of 4 out of 6 recommendations without and with privacy, which illustrates that PriView manages to maintain utility while protecting user privacy.

We conducted further testing, to illustrate that PriView is able to eliminate the privacy threat from $\hat{B}$ for chosen attributes A with little effect on the quality of recommendations. We used 5-fold cross validation, to split our dataset into a training set containing 80% of the data, and a test set containing the remaining 20% of the data on which we tested the MF recommender engine, both with and without privacy activated to compare the relevance of recommendations in these two cases. The random splitting into training and test sets was performed 5 times, as shown in the first row of Table I. More precisely, in each test set, we randomly removed and tried to predict 10% of the ratings. Table I shows the RMSE in rating prediction based on actual ratings, and on distorted ratings. $\hat{r}$ denotes predicted ratings based on the the actual ratings provided by users for other shows, while $\hat{\hat{r}}$ denotes predicted ratings based on the ratings distorted for privacy. The prediction RMSE for $\hat{r}$ (RMSE1, privacy not activated) and for $\hat{\hat{r}}$ (RMSE2, privacy activated) are calculated on the 10% of ratings that we removed. Table I shows that the RMSE for rating prediction does not degrade much when privacy protection is activated, with respect to rating prediction without privacy. Note that these results are for the case of perfect privacy $(I(A;\hat{B}) = 0)$, meaning that any inference algorithm that would try to infer A, e.g. political views, from ratings $\hat{B}$ would not outperform an uninformed random guess. If the privacy requirements were less stringent, e.g. $(I(A;\hat{B}) \leq \epsilon)$, for some $\epsilon > 0$, then the RMSE for rating prediction with privacy protection would be even closer to the RMSE without privacy. Finally, we would like to point out that using a more advanced and optimized recommendation engine, instead of the aforementioned standard MF recommendation engine, could only yield better rating prediction quality both without and with privacy protection.

TABLE I: Rating prediction RMSE

Set	1	2	3	4	5
RMSE1 $(\hat{r})$	1.2434	1.3208	1.2657	1.3359	1.2928
RMSE2 $(\hat{\hat{r}})$	1.3469	1.3522	1.4182	1.3969	1.3708

IV. Extensions and Perspective

PriView has been implemented for video consumptions and recommendation, and it has the potential to be interfaced with online video services, as well as TV and VoD services. Future work includes extending PriView to other media content, e.g. music, books, news, and to other products, services, or locations rated or reviewed online by users. PriView could also be adapted to protect privacy in the context of social networks: users could be informed of the privacy risks of actions such as likes, connecting to friends... prior to taking those actions, and provided means to control these risks. In such a context, data distortion could for example amount to simply avoiding to take some actions, or avoiding the release of some data. Extensions also include broadening the set of private attributes that can be deemed sensitive by users, and analyzing the temporal dynamics of privacy and utility in a real-time setting in a system such as PriView.

The original privacy-utility framework in [4] assumes that the true prior distribution $p_{A,B}$ is known by both the adversary and the privacy agent. A natural question is how the privacy-utility tradeoff is impacted when the adversary and the privacy agent have different knowledge of the statistical properties of A and B. In the case of a weaker adversary whose knowledge of the statistical properties of the prior distribution $p_{A,B}$ is less accurate than that of the privacy agent, the privacy-utility tradeoff derived assuming a stronger adversary still holds. Indeed, a weaker adversary who would try to infer private data A based on a less accurate knowledge of the statistics of A and B cannot outperform a stronger adversary who would try to infer A based on a more accurate statistical model. The general case of a stronger adversary, who has a more accurate knowledge of the statistics of A and B than the privacy agent, is an interesting open problem in general. The case of a mismatched prior distribution, where an estimated prior distribution $q_{A,B}$ that differs from the true distribution $p_{A,B}$ is fed to the privacy-utility optimization in [4] was addressed in [23]. More precisely, the mismatch was measured in terms of the l_1 distance between the true prior and the mismatched prior, and bounds on the impact of the mismatch on the privacy-utility tradeoff were derived. The design of privacy mappings under partial knowledge of the prior distribution $p_{A,B}$, such as knowledge of marginal distributions, or statistical moments of the prior distribution, was addressed in [10].

V. Conclusion

We propose PriView, an interactive privacy-preserving system for video consumption and recommendation that provides a user with privacy transparency and control, while maintaining the quality of recommendations the user receives. PriView informs the user about the risk of releasing data related to media preferences (e.g. tv show viewing) with respect to private attributes (e.g. political views, age, gender) prior to the release, and gives means to the user to control and monitor these risks, while maintaining the relevance of personalized recommendations based on the released sanitized data. PriView bridges privacy theory and practice: the privacy mappings implemented by PriView ensures perfect privacy against statistical inference of private attributes from the sanitized data. PriView has the potential to be interfaced with online video services, as well as TV and VoD services, and to be extended to other products or services, e.g. music, books, news, locations rated online by users.

References

[1] B. Fung and K. Wang and R. Chen and P. Yu, "Privacy Preserving Data Publishing: A Survey of Recent Developments," *ACM Computing Surveys*, 2010.

[2] "What your favorite tv shows and networks say about your politics," http://www.buzzfeed.com/rubycramer/, 2012.

[3] "Simmons Consumer Segmentations: PublicPersonas," http://www.experian.com, 2012.

[4] F. Calmon and N. Fawaz, "Privacy against statistical inference," in *Allerton*, 2012. [Online]. Available: http://arxiv.org/abs/1210.2123

[5] S. Salamatian, A. Zhang, F. du Pin Calmon, S. Bhamidipati, N. Fawaz, B. Kveton, P. Oliveira, and N. Taft, "How to hide the elephant- or the donkey- in the room: Practical privacy against statistical inference for large data," in *IEEE GlobalSIP*, 2013.

[6] S. L. Warner, "Randomized response: A survey technique for eliminating evasive answer bias," *Journal of the American Statistical Association*, vol. 60, no. 309, 1965.

[7] R. Agrawal and R. Srikant, "Privacy-preserving data mining," *ACM Sigmod Record*, vol. 29, no. 2, pp. 439–450, 2000.

[8] N. Mishra and M. Sandler, "Privacy via pseudorandom sketches," in *PODS*, 2006.

[9] A. Evfimievski, J. Gehrke, and R. Srikant, "Limiting privacy breaches in privacy preserving data mining," in *PODS*, 2003.

[10] A. Makhdoumi and N. Fawaz, "Privacy-utility tradeoff under statistical uncertainty," in *Allerton*, 2013.

[11] D. Rebollo-Monedero, J. Forné, and J. Domingo-Ferrer, "From t-closeness-like privacy to postrandomization via information theory," *IEEE Trans. on Knowledge and Data Engineering*, 2010.

[12] S. P. Kasiviswanathan, H. K. Lee, K. Nissim, S. Raskhodnikova, and A. Smith, "What can we learn privately?" *SIAM Journal on Computing*, 2011.

[13] S. Banerjee, N. Hegde, and L. Massoulié, "The price of privacy in untrusted recommendation engines," in *Allerton*, 2012.

[14] J. C. Duchi, M. I. Jordan, and M. J. Wainwright, "Local privacy and statistical minimax rates," in *FOCS*, 2013.

[15] C. Dwork, F. Mcsherry, K. Nissim, and A. Smith, "Calibrating noise to sensitivity in private data analysis," in *TCC*, 2006. [Online]. Available: http://www.cs.bgu.ac.il/\~{}kobbi/papers/sensitivity-tcc-final.pdf

[16] C. Dwork, "Differential privacy," in *Automata, Languages and Programming*. Springer, 2006, vol. 4052, pp. 1–12.

[17] H. Yamamoto, "A source coding problem for sources with additional outputs to keep secret from the receiver of wiretappers," *IEEE Trans. Information Theory*, vol. 29, no. 6, 1983.

[18] L. Sankar, S. R. Rajagopalan, and H. V. Poor, "Utility-privacy tradeoff in databases: An information-theoretic approach," *IEEE Trans. Inf. Forensics Security*, 2013. [Online]. Available: http://arxiv.org/abs/1102.3751

[19] A. Friedman and A. Schuster, "Data mining with differential privacy," in *KDD*, 2010.

[20] A. Machanavajjhala, A. Korolova, and A. D. Sarma, "Personalized social recommendations - accurate or private?" *PVLDB*, vol. 4, no. 7, 2011.

[21] F. McSherry and I. Mironov, "Differentially private recommender systems: building privacy into the net," in *KDD*, 2009.

[22] N. Fawaz, F. Calmon, A. Makhdoumi, and S. Salamatian, "From differential privacy to divergence privacy," in *ITA*, 2014.

[23] S. Salamatian, A. Zhang, F. d. Calmon, S. Bhamidipati, N. Fawaz, B. Kveton, P. Oliveira, and N. Taft, "Managing your Private and Public Data: Bringing down Inference Attacks against your Privacy," *ArXiv e-prints*, 2013. [Online]. Available: http://arxiv.org/abs/

[24] Y. Koren, R. Bell, and C. Volinsky, "Matrix factorization techniques for recommender systems," *IEEE Computer*, vol. 42, no. 8, 2009.