Recommending with an Agenda: Active Learning of Private Attributes using Matrix Factorization

Smriti Bhagat, Udi Weinsberg, Stratis Ioannidis, Nina Taft
Technicolor, Los Altos, CA
{smriti.bhagat, udi.weinsberg, stratis.ioannidis, nina.taft}@technicolor.com

ABSTRACT

Recommender systems leverage user demographic information, such as age, gender, etc., to personalize recommendations and better place their targeted ads. Oftentimes, users do not volunteer this information due to privacy concerns, or due to a lack of initiative in filling out their online profiles. We illustrate a new threat in which a recommender learns private attributes of users who do not voluntarily disclose them. We design both passive and active attacks that solicit ratings for strategically selected items, and could thus be used by a recommender system to pursue this hidden agenda. Our methods are based on a novel usage of Bayesian matrix factorization in an active learning setting. Evaluations on multiple datasets illustrate that such attacks are indeed feasible and use significantly fewer rated items than static inference methods. Importantly, they succeed without sacrificing the quality of recommendations to users.

Categories and Subject Descriptors: H.2.8 Database Applications: Data Mining
Keywords: Recommendations, Privacy, Active Learning.

1. INTRODUCTION

Recommender systems rely on knowing their users – not just their preferences (i.e., ratings on items), but also their social and demographic information, e.g., age, gender, political affiliation, and ethnicity [14, 19]. A rich user profile allows a recommender system to better personalize its services, and at the same time enables additional monetization opportunities, such as targeted advertising.

Users of a recommender system know they are disclosing their preferences (or ratings) for movies, books, or other items (we use movies as our running example). A recommender may obtain additional social and demographic information about its users by explicitly soliciting it [14, 19]. While some users may willingly disclose such information, others may be more privacy-sensitive and elect to only reveal their ratings. Privacy research has shown that some users are uncomfortable revealing their demographic data to personalization systems [1, 15]. Even when such services provide transparency about their data collection and use practices [1], some users are unwilling to disclose personal data despite the allure of personalized services. In [14] the authors conduct a small scale user study on Amazon Turk that examines how to motivate users to disclose their demographic data.

For users who wish to withhold some demographic information, a recommender can still undermine their attempt at privacy. In previous work [30], we show that users' movie ratings can be used to predict their gender with 80% accuracy. Other studies also show the potential to infer demographics from a range of online user activities [2, 4, 18, 20].

In this work, we consider a recommender system that offers a legitimate service, yet is simultaneously malicious: it purposefully attempts to extract certain attributes from users who choose to withhold them. Unlike previous work that studies static attacks on the complete data, we consider an *active learning* setting, in which the recommender system aims to efficiently (quickly and accurately) infer a user's private attribute via interactive questioning. Recommender systems routinely ask users to rate a few items, as a means to address a "cold start" setting, or to improve the quality of recommendations. We leverage these instances of interactions with the user, alongside with the observation that item selection is at the recommender's discretion, to propose a new attack. We hypothesize that if the sequence of questions (items to rate) is carefully selected, the recommender system can quickly (so as not to be detected by the user) determine a user's private attribute with high confidence, thereby violating her privacy.

A key idea in the design of this attack is to leverage matrix factorization (MF) as the basis for inference. Most recommender systems use matrix factorization (MF) models as a building block for providing recommendations [17]. While MF is well understood for rating prediction, it has generally not been applied for inference. To the best of our knowledge, this paper is the first to leverage MF as the basis for building both (a) an inference method of private attributes using item ratings and (b) an active learning method that selects items in a way that maximizes inference confidence in the smallest number of questions.

Our contributions are as follows:

- First, we propose a novel classification method for determining a user's binary private attribute – her *type* – based upon ratings alone. In particular, we use matrix factorization to learn item profiles and type-dependent biases, and show how to incorporate this information into a clas-

RecSys'14, October 6–10, 2014, Foster City, Silicon Valley, CA, USA.

http://dx.doi.org/10.1145/2645710.2645747.

sification algorithm. This classification is consistent with Bayesian matrix factorization.

- Second, we demonstrate that the resulting classification method is well suited for learning of a user's type. A simple *passive* approach, ordering items based on a set of weights computed off-line, works quite well in many cases. Beyond this, we design an *active learning* algorithm for selecting the next item to ask a user to rate: each selection maximizes the expected confidence of the private attribute's inference. Equivalently, the selections of the active learning algorithm minimize the expected risk of misclassifying the user's private attribute.
- Third, we show that our active learning method is very efficient, as item selection can reuse computations made during previous selections. We show that this reduces the naïve solution that is cubic in the number of ratings, to one that is quadratic in the number of ratings.
- Fourth, we extensively evaluate the above classifier and selection methods on three real-world datasets: Movielens, Flixster and Politics-and-TV. We show that our methods consistently outperform other baselines; with only 10 questions, we achieve between 3-20% higher classification accuracy on different datasets. Importantly, such an attack can be carried out without any sacrifice to the recommendations made to the user.

2. RELATED WORK

A number of studies have shown that user demographics can be inferred from various types of online user activity. For example, Bhagat *et al.* [2] show that it is possible to learn age and gender information from blogs. Mislove *et al.* [20] study data from online social networks and illustrate that even when only a fraction of users provide profile attributes (such as location, interests, etc.), it is possible to infer these attributes among users who do not disclose them. Bi *et al.* [4] show how demographics can be inferred from search queries, and Kosinski *et al.* [18] show that several personality traits, including political views, sexual orientation, and drug use can be accurately predicted from Facebook "likes".

Recommender systems were shown to be exploitable by several works utilizing off-line attacks [5, 22, 30]. Closest to our setting, Weinsberg et al. [30] empirically studied how to infer a user's gender from her movie ratings using a variety of different classifiers, showing that logistic regression and SVMs succeed with an accuracy close to 80%. We depart from [30] in multiple ways. First, we introduce a novel factor-based classifier, that relies on the Bayesian assumptions behind MF. Second, we study a recommender system in an adversarial setting that actively adapts item selection to quickly learn the private attributes. Finally, we establish that our classifier is very well suited for this task.

The Bayesian model underlying MF (discussed in detail in Section 3.2) was recently employed by Silva and Carin [28] to actively learn the actual factors (*i.e.*, the user and item profiles) in MF. More specifically, the authors consider a recommender system that adaptively selects which items to ask its users to rate in order to diminish the entropy of its user and item profiles as quickly as possible. The entropy estimation is based on the Gaussian noise and prior assumptions underlying MF, which we also employ in our work. A variety of active learing objective were also studied by Sutherland et al. [29], including minimizing the prediction error on unrated items, reducing the profile uncertainty, and identifying highly rated items. We depart from the above works as the goal of our learning task is to discover a user's demographic information, captured by a categorical type, rather the above objectives motivated by rating prediction.

In the classic active learning setting [8,9], a learner wishes to disambiguate among a set of several possible hypotheses, each represented as a function over a set of inputs. Only one of the hypotheses is valid; to discover it, the learner has access to an oracle that returns the evaluation of the valid hypothesis on a given input. In the case of a *noiseless* oracle, that always returns the correct evaluation on a query, Generalized Binary Search (GBS) discovers the valid hypothesis in a number of queries within a polylogarithmic factor from the optimal [8, 9]. Our setup can be cast into the above framework in the context of a *noisy* oracle, whose evaluations may not necessarily be exact. GBS is known to yield arbitrarily suboptimal results in the presence of noise [10]. Though algorithms for restricted noise models exist (see, e.g., [23] and [10]), no algorithm with provable performance guarantees is known in the presence of an oracle with arbitrary noise. Unfortunately, none of the existing models apply to the noisy setting we encounter here.

3. SYSTEM DESCRIPTION

3.1 Problem Statement

We consider a recommender system that provides a legitimate item recommendation service, but at the same time maliciously seeks to infer a private user attribute. The system has access to a dataset, provided by non-privacy-sensitive users, that contains item ratings as well as a categorical variable, which we refer to as the user *type*. The type is a private attribute such as gender, age, political affiliation, etc. A new user, who is privacy sensitive (i.e., her type is unknown) interacts with the system. The recommender system actively presents items for the user to rate, masquerading it as a way to improve recommendations in the cold-start setting. In this context, our goal is twofold:

1. We wish to design a *type classifier* that discovers the type of the user based on her ratings. We seek to leverage the latent factor model prevalent in matrix-factorization, a technique successfully used for rating prediction by recommender systems.
2. We wish to address the problem of *actively learning* a user's type. We aim to design an item selection method, that determines the order in which items are shown to a user for her to rate. The best order finds the user's type as quickly as possible.

For the attack to be considered successful, the recommender system needs to obtain high confidence in the value of the inferred type, with a minimum number of questions posed to the user. As our classifier and item selection methods rely heavily on matrix factorization, we review this as well as the latent factor model that underlies it below.

3.2 Data Model & Matrix Factorization

We use the following notation to describe the training dataset of the recommender. The dataset comprises of ratings to m items in set $\mathcal{M} \equiv \{1, \ldots, m\}$ given by n users in set $\mathcal{N} \equiv \{1, \ldots, n\}$. We denote by r_{ij} the rating of user $i \in \mathcal{N}$ to item $j \in \mathcal{M}$, and by $\mathcal{E} \subset \mathcal{N} \times \mathcal{M}$ the set of user-item pairs (i, j), for which a rating r_{ij} is present in the dataset.

Each user is characterized by a categorical type, which captures demographic information such as gender, occupation, income category, *etc.* Focusing on binary types, we denote by $t_i \in \mathcal{T} \equiv \{+1, -1\}$ the type of user $i \in \mathcal{M}$.

We assume that the ratings are generated from the standard generative model used in matrix factorization, augmented with type-dependent biases. More specifically, there exist latent factors $u_i \in \mathbb{R}^d$, $i \in \mathcal{N}$, and $v_j \in \mathbb{R}^d$, $j \in \mathcal{M}$ (the *user* and *item profiles*, resp.) such that ratings are:

$$r_{ij} = u_i^T v_j + z_{jt_i} + \epsilon_{ij}, \quad (i,j) \in \mathcal{E} \tag{1}$$

where $\epsilon_{ij} \sim N(0, \sigma_0^2)$ are independent Gaussian noise variables and z_{jt} is a *type bias*, capturing the effect of a type on the item rating. Our model is thus parametrized by $U = [u_i^T]_{i\in\mathcal{N}} \in \mathbb{R}^{n\times d}$,$V = [v_j^T]_{j\in\mathcal{M}} \in \mathbb{R}^{m\times d}$, and $Z = [z_{j,t}]_{j\in\mathcal{M},t\in\mathcal{T}} \in \mathbb{R}^{m\times|\mathcal{T}|}$. We further assume a *prior* on user and item profiles: for all $i \in \mathcal{N}$, $j \in \mathcal{M}$,

$$u_i \sim N(\mathbf{0}, \sigma_u^2 I), \text{ and } v_j \sim N(\mathbf{0}, \sigma_v^2 I), \tag{2}$$

i.e., profiles are sampled from independent zero-mean multivariate Gaussian priors.

The Gaussian priors (2) are used in many works on so-called Bayesian matrix factorization (see, *e.g.*, [21, 25, 28]). Under (1) and (2), the maximum likelihood estimation of the model parameters reduces to the standard [16,17] minimization of the (non-convex) regularized error:[1]

$$\min_{U,V,Z} \sum_{(i,j)\in\mathcal{E}} (r_{ij} - u_i^T v_j - z_{jt_i})^2 + \lambda \sum_{i\in\mathcal{N}} \|u_i\|_2^2 + \mu \sum_{i\in\mathcal{M}} \|v_j\|_2^2 \tag{3}$$

with $\lambda = \frac{\sigma_0^2}{\sigma_u^2}$ and $\mu = \frac{\sigma_0^2}{\sigma_v^2}$. Given a dataset of ratings r_{ij}, $(i,j) \in \mathcal{E}$ and types t_i, $i \in \mathcal{N}$, the parameters U, V, Z can be computed as local minima to (3) through standard methods [17], such as gradient descent or alternating minimization, while λ and μ are computed through cross-validation.

4. A FACTOR-BASED CLASSIFIER

We now turn our attention to the following classification problem. Suppose that the recommender system, with access to the dataset of ratings and types, has computed a set of item profiles V as well as a set of biases Z, *e.g.*, by minimizing (3) through gradient descent. A new user arrives in the system and submits ratings for items in some set $A \subseteq \mathcal{M}$, *but does not submit her type.* In order to bypass the user's attempt at privacy, we need to construct a classifier to infer the type of this new user.

In this section, we present a classifier that uses the item profiles and biases (*i.e.*, the latent factors obtained through matrix factorization) to accomplish this task. We refer to this classifier as a *Factor-Based Classifier* (FBC). Crucially, FBC is consistent with the Bayesian model of matrix factorization presented in the previous section. In particular, it amounts to the maximum a-posteriori estimation of the type under the bi-linear noise model (1) and the priors (2).

4.1 Type Posterior

For $A \subset \mathcal{M}$ the set of items for which the user submits ratings, we introduce the following notation. We denote by $r_A \equiv [r_j]_{j\in A} \in \mathbb{R}^{|A|}$ the vector of ratings provided by the user, by $V_A \equiv [v_j^T]_{j\in A} \in \mathbb{R}^{|A|\times d}$ the matrix of profiles for items rated, and by $z_{At} \equiv [z_{jt}]_{j\in A} \in \mathbb{R}^{|A|}$ the vector of type-t biases for items rated.

As in the previous section, we assume the new user has an unknown profile $u \in \mathbb{R}^d$ and a type $t \in \{-1, +1\}$, such that the ratings she submits follow (1), *i.e.*,

$$r_j = u^T v_j + z_{jt} + \epsilon_j, \quad j \in A, \tag{4}$$

where $\epsilon_j \sim N(0, \sigma_0^2)$. That is, conditioned on u and t, the ratings $r_A = [r_j]_{j\in A} \in \mathbb{R}^{|A|}$ given to items in $A \subset [M]$ are distributed as follows:

$$\Pr(r_A \mid u, t) = e^{-\|r_A - V_A u - z_{At}\|_2^2 / 2\sigma_0^2} / \left(\sigma_0\sqrt{2\pi}\right)^{|A|} \tag{5}$$

where σ_0^2 is the noise variance.

Moreover, we assume as in the previous section that profile u follows a zero-mean Gaussian prior with covariance $\sigma_u^2 I$, and that the type follows a uniform prior (*i.e.*, each of the two types is equally likely), i.e.:

$$\Pr(u, t) = 0.5 e^{-\|u\|_2^2/2\sigma_u^2} / \left(\sigma_u\sqrt{2\pi}\right)^d \tag{6}$$

4.2 Classification

Under the above assumptions, it is natural to classify the incoming user using maximum a posteriori estimation of the type t. In particular, FBC amounts to

$$\hat{t}(r_A) = \arg\max_{t\in\mathcal{T}} \Pr(t \mid r_A). \tag{7}$$

Under this notation, FBC can be determined as follows:

THEOREM 1. *Under noise model* (5) *and prior* (6)*, the* FBC *classifier is given by*

$$\hat{t}(r_A) = \text{sgn}(\delta_A^T M_A \bar{r}_A) \tag{8}$$

where $\bar{r}_A \equiv r_A - \frac{z_{A+} + z_{A-}}{2}$, $\delta_A \equiv \frac{z_{A+} - z_{A-}}{2}$, $M_A \equiv I - V_A \Sigma_A^{-1} V_A^T$*, and* $\Sigma_A \equiv \lambda I + V_A^T V_A$*, for* $\lambda = \frac{\sigma_0^2}{\sigma_u^2}$.

The proof can be found in the extended version of our paper [3]. There are several important observations to be made regarding FBC, as defined by Theorem 1.

Set of Classifiers. We first note that FBC in fact defines a *set* of classifiers, each parametrized by set $A \subseteq \mathcal{M}$: each such classifier $\hat{t} : \mathbb{R}^{|A|} \to \{-1, +1\}$ takes as input any possible set of ratings $r_A \in \mathbb{R}^{|A|}$ as input. Note however that all classifiers are trained *jointly* from the ratings dataset: this "training" amounts to determining the item profiles V and the item biases Z through matrix factorization. With V and Z learned, when presented with ratings r_A, the classifier can compute the vectors $\bar{r}_A$, δ_A and the matrix M_A needed to determine the type. Indeed, the fact that training the classifier amounts to computing the latent factors/item profiles is consistent with the observation that FBC shares the same underlying Bayesian assumptions as matrix factorization.

5. LEARNING STRATEGIES

The second task in designing this threat is to find a user's type *quickly.* In what follows, we present two strategies for addressing this problem. The first is a *passive* strategy: the recommender presents items to the user in a predetermined order, computed off-line. The second is an *active* strategy: the recommender selects which item to present to the user next based on the answers she has given so far. Both strategies are extensively evaluated in Section 6.

[1]Note that, as is common practice, to ensure that the profiles U, V obtained by (3) are invariant to a translation (shift) of the ratings, we do not regularize the category biases (or, equivalently, we assume no prior on Z).

Algorithm 1 FBC-SELECTION

Input: Item profiles V, item biases Z, confidence τ

$A \leftarrow \emptyset$
$r_A \leftarrow \emptyset$
repeat
  for all $j \in \mathcal{M} \setminus A$ **do**
    Compute L_j through (9)
  $j^* \leftarrow \arg\min_{j \in \mathcal{M} \setminus A} L_j$
  Query user to obtain r_{j^*}
  $A \leftarrow A \cup \{j^*\}$, $r_A \leftarrow r_A \cup r_{j^*}$
until $\Pr(\hat{t}(r_A) \mid r_A) > \tau$

5.1 MAXGAP: A Passive Strategy

A simple, passive method for presenting items to the user is to (a) sort items $j \in \mathcal{M}$ with respect to $|\delta_j|$, the absolute value of the gap between the type biases, and (b) present items to the user in decreasing order. We call this strategy MAXGAP: intuitively, this method identifies the most discriminative items in the dataset, and solicits responses to these items first. Clearly, MAXGAP does not take into account (or adapt to) how the user rates the items presented so far. Despite this limitation, as we will see in Section 6, this simple strategy actually performs surprisingly well in many cases, especially when there exist many highly discriminative items. When this is not the case, however, an active strategy is needed, motivating our second method.

5.2 FBC-SELECTION: An Active Strategy

Our active method, FBC-SELECTION, is summarized in Algorithm 1. Let $\hat{t}$ be the FBC classifier defined by (8). Given observed ratings $r_A \equiv [r_j]_{j \in A} \in \mathbb{R}^{|A|}$, for some $A \subset \mathcal{M}$, we define the *risk* $L(\hat{t}(r_A))$ of the classifier to be 0 if the prediction is correct, and 1 otherwise. Conditioned on r_A, the expected risk is $\mathbb{E}[L(\hat{t}(r_A)) \mid r_A] = 1 - \Pr(\hat{t}(r_A) \mid r_A)$, *i.e.*, it equals the 1 minus the *confidence* of the classifier, the posterior probability of the predicted type, conditioned on the observed ratings. Since, by (7), FBC selects the type that has the maximum posterior probability, the expected risk is at most (and the confidence at least) 0.5.

FBC-SELECTION proceeds greedily, showing the item that minimizes the classifier's expected risk at each step. More specifically, let A be the set of items whose ratings have been observed so far. To select the next item to present to the user, the algorithm computes for each item $j \in \mathcal{M} \setminus A$, the expected risk $\mathbb{E}[L(\hat{t}(r_A \cup r_j)) \mid r_A]$ if rating r_j is revealed:

$$\int_{r_j \in \mathbb{R}} (1 - \Pr(\hat{t}(r_A \cup r_j)) \mid r_A \cup r_j) \Pr(r_A \cup r_j \mid r_A) dr_j.$$

This expected risk depends on the *distribution* of the unseen rating r_j *conditioned* on the ratings observed so far.

Under noise model (5) and prior (6), the expected risk for each item j can be computed in a closed form. In particular, let M_A, $\bar{r}_A$, δ_A be as defined in Theorem 1. Then, the expected risk when revealing the rating of item j is proportional to the following quantity, derived in [3]:

$$L_j = \int_{r_j} e^{-\frac{\bar{r}_{A_j}^T M_{A_j} \bar{r}_{A_j} + 2|\delta_{A_j}^T M_{A_j} \bar{r}_{A_j}| + \delta_{A_j}^T M_{A_j} \delta_{A_j}}{2\sigma_0^2}} dr_j / \sqrt{\det \Sigma_{A_j}} \quad (9)$$

where $A_j = A \cup \{j\}$. The integration above is w.r.t. r_j, *i.e.*, the predicted rating for item j. The outcome of the above integration can be computed in closed form in terms of the error function erf (i.e., *no numerical integration is necessary*). The formula can be found in [3].

Each iteration amounts to computing the "scores" L_j for each item j not selected so far, and picking the item with the lowest score (corresponding to minimum expected risk). Once the item is presented to the user, the user rates it, adding one more rating to the set of observed ratings. The process is repeated until the confidence of the classifier (or, equivalently, the expected risk) reaches a satisfactory level.

5.3 INCFBC: An Efficient Implementation

FBC-SELECTION requires the computation of the scores L_j after each interaction with the user. Each such calculation involves computing the determinant $\det(\Sigma_{A_j})$, as well as the matrix $M_{Aj} = (I - V_{A_j} \Sigma_{A_j}^{-1} V_{A_j}^T)$, both of which appear in (9). Though having a closed form formula for (9) avoids the need for integration, computing each of these matrices directly from their definition involves a considerable computational cost. In particular, the cost of computing $\Sigma_A = \lambda I + V_A^T V_A$ is $O(d^2|A|)$. Computing Σ_A^{-1} and $\det(\Sigma_{A_j})$ have a cost $O(d^3)$ multiplications using, *e.g.*, LU-decomposition, which can be dropped to $O(d^{2.807})$ using Strassen's algorithm for multiplication [7]. Finally, the computation of M_A requires $O(|A| \times d^2 + |A|^2 \times d)$ multiplications. As a result, the overall complexity of computing L_j directly is $O(|A| \times d^2 + |A|^2 \times d + d^{2.807})$.

However, the performance of these computations can be significantly reduced by constructing these matrices incrementally: M_{A_j}, $\Sigma_{A_j}^{-1}$ and $\det(\Sigma_{A_j})$ can be computed efficiently from M_A, Σ_A^{-1}, and $\det(\Sigma_A)$, exploiting the fact that $\Sigma_{A_j} = \Sigma_A + v_j v_j^T$, *i.e.*, it results from Σ_i through a *rank-one* update. We discuss this below.

Incremental computation of $\det(\Sigma_{A_j})$. The determinant can be computed incrementally using only $O(d^2)$ multiplications through the Matrix Determinant Lemma [11], namely:

$$\det(\Sigma_{A_j}) = (1 + v_j^T \Sigma_A v_j) \det(\Sigma_A). \quad (10)$$

Incremental computation of $\Sigma_{A_j}^{-1}$. The inverse of a rank-one update of a matrix can be computed through the Sherman-Morisson formula [27], which gives:

$$\Sigma_{A_j}^{-1} = \Sigma_A^{-1} - \Sigma_A^{-1} v_j v_j^T \Sigma_A^{-1} / (1 + v_j^T \Sigma_A^{-1} v_j), \quad (11)$$

and again reduces the number of multiplications to $O(d^2)$.

Incremental computation of M_{A_j}. Finally, using (11), we can also reduce the cost of computing M_{A_j}, as:

$$M_{A_j} = \begin{bmatrix} M_A + \frac{\phi\phi^T}{1 + v_j^T \Sigma_A^{-1} v_j} & -\xi \\ -\xi^T & 1 - v_j^T \xi \end{bmatrix} \quad (12)$$

where $\xi = V_A(\Sigma_{A_j}^{-1} v_j)$ and $\phi = V_A(\Sigma_A^{-1} v_j)$, which reduces the computation cost to $O(|A|^2 + d^2)$ multiplications.

In conclusion, using the above adaptive operations reduces the cost of computing L_j by one order of magnitude to $O(|A|^2 + d^2)$, which is optimal (as M_A is an $|A| \times |A|$ matrix, and Σ_A is $d \times d$). The rank-one adaptations yield such performance without sophisticated matrix inversion or multiplication algorithms, such as Strassen's algorithm. The we refer to resulting algorithm as INCFBC; we empirically compare the two implementations in Section 6.

Algorithm 2 PointEst Active Learning

Input: Item profiles V, item biases Z, classifier C, confidence τ

$A \leftarrow \emptyset$, $r_A \leftarrow \emptyset$
repeat
  $\hat{t} \leftarrow \arg\max_{t \in \mathcal{T}} \Pr_C(t \mid r_A)$
  $\hat{u} \leftarrow (\lambda I + V_A^T V_A)^{-1} V_A^T (r_A - z_{A\hat{t}})$
  for all $j \in \mathcal{M} \setminus A$ **do**
    $\hat{r}_j \leftarrow \hat{u}^T v_j + z_{j\hat{t}}$
    $L_j \leftarrow \min_{t \in \mathcal{T}} \Pr_C(t \mid r_A \cup \hat{r}_j)$
  $j^* \leftarrow \arg\min_j L_j$
  Query user to obtain r_{j^*}
  $A \leftarrow A \cup \{j^*\}$, $r_A \leftarrow r_A \cup r_{j^*}$
until $1 - L_{j^*} > \tau$

5.4 Selection Through Point Estimation

An alternative method for selection can be constructed by replacing the exact estimation of the expected risk with a "point estimate" (see also [28]). In fact, such a selection method can be easily combined with an arbitrary classifier that operates on user-provided ratings as input. This makes such an approach especially useful when the expected risk is hard to estimate in a closed form. We therefore outline this method below, noting however that several problems arise when the risk is computed through such a point estimation.

We describe the method for a general classifier C, also summarized in Algorithm 2. Given a set of ratings r_A over a set $A \subseteq \mathcal{M}$, the classifier C returns a probability $\Pr_C(t \mid r_A)$, for each type $t \in \mathcal{T}$. This is the probability that the user's type is t, conditioned on the observed ratings r_A. Given a set of observed ratings r_A, we can estimate the type of the user using the classifier C though maximum likelihood a-posteriori estimation, as $\hat{t}(r_A) = \arg\max_{t \in \mathcal{T}} \Pr_C(t \mid r_A)$. Using this estimate, we can further estimate the most likely profile $\hat{u} \in \mathbb{R}^d$ through ridge regression [12] over the observed ratings r_A and the corresponding profiles V_A (see Algorithm 2 for details). Using the estimated profile $\hat{u}$ and the estimated type $\hat{t}$, we can predict the rating of every item $j \in \mathcal{M} \setminus A$ as $\hat{r}_j = \hat{u}^T v_j + z_{j\hat{t}}$, and subsequently estimate the expected risk if the rating for item j is revealed as $\min_{t \in \mathcal{T}} \Pr_C(t \mid r_A \cup \hat{r}_j)$. We refer to this as a "point estimate", as it replaces the integration that the expected risk corresponds to with the value at a single point, namely, the predicted rating $\hat{r}_j$.

Using such estimates, selection can proceed as follows. Given the set of observed ratings A, we can estimate the risk of the classifier C for every item j in $\mathcal{M} \setminus A$ through the above estimation process, and pick the item with the minimum estimated risk. The rating of this item is subsequently revealed, and new estimates $\hat{t}$ and $\hat{u}$ can thusly be obtained, repeating the process until a desired confidence is reached.

Clearly, point estimation avoids computing the expected risk exactly, which can be advantageous when the corresponding expectation under a given classifier C can only be computed by numerical integration. This is not the case for FBC, as we have seen, but this can be the only tractable option for an arbitrary classifier. Unfortunately, this estimation can be quite inaccurate in practice, consequently leading to poor performance in selections; we observe such a performance degradation in our evaluations (Section 6). Put differently, a point estimate of the risk takes into account what the predicted rating of an item j is in *expectation*, and how this rating can potentially affect the risk; however, it does not account for how *variable* this prediction is. A highly variable prediction might have a very different expected risk; the exact computation of the expectation does take this into account whereas point estimation does not.

Dataset	Type	Users	Items	Ratings
Movielens	All	6K	3K	1M
	Gender (Female:Male)	1:2.5	-	1:3
	Age (Young:Adult)	1:1.3	-	1:1.6
PTV	All	992	50	29.9K
	Gender (Female:Male)	1.8:1	-	1.6:1
	Political Views (R:D)	1:1.6	-	1:2.1
Flixster	All	26K	9921	5.6M
	Gender (Female:Male)	1.7:1	-	1.5:1

Table 1: Dataset statistics.

6. EVALUATION

In this section we evaluate the performance of our methods using real datasets. We begin by describing the datasets and experiments, then perform a comparative analysis of both passive and active methods.

6.1 Experimental Setup

Datasets. We evaluate our method using three datasets: Movielens, Flixster [13], and Politics-and-TV (PTV) [26]. The Movielens dataset includes users' ratings for movies alongside with the users' gender and age. For simplicity, we categorize the age group of users as *young adults* (ages 18–35), or *adults* (ages 36–65). Flixster is a similar movie ratings dataset, and contains user gender information. PTV includes ratings by US users on 50 different TV-shows, along with each user's gender and political affiliation (Democrat or Republican). We preprocessed Movielens and Flixster to consider only users with at least 20 ratings, and items that were rated by at least 20 users. Since PTV includes only 50 TV-shows, we preprocessed the data to ensure that each user has at least 10 ratings. Table 1 summarizes the datasets used for evaluation. For each user type, the table shows the ratio between the number of users of one type versus the other type (as labeled in the table). Further details on the datasets can be found in [3].

Evaluation Method. In our setting, the recommender system infers user attributes from a set of strategically selected items. To understand the effectiveness of FBC compared to other classification methods in an adversarial setting, we perform the following evaluation. We first split the dataset into a training set (e.g., users that are willing to disclose the private attribute) and evaluation set (e.g., users that are privacy-sensitive), and train different classifiers on the training set – e.g., in the case of FBC we learn the item profiles and biases. Then, for each user in the evaluation set, we incrementally select items for the user to rate. In the passive methods, the selection of next item does not depend on the previous ratings provided by the user, whereas in the active methods it does.

After the user rates an item, we use the classifier to infer the private type. For any user, since we only have the rating information for the set of movies that she has rated, we limit the selection process to this set. Users may have rated different number of movies, for instance, roughly 50% of the users of Movielens have rated less than 100 movies out of 3000 (see [3]). Therefore, we limit the number of questions asked to 100 for Movielens and Flixster and all 50 for PTV. Unless specified, all evaluations of FBC were done using the efficient incremental implementation IncFBC.

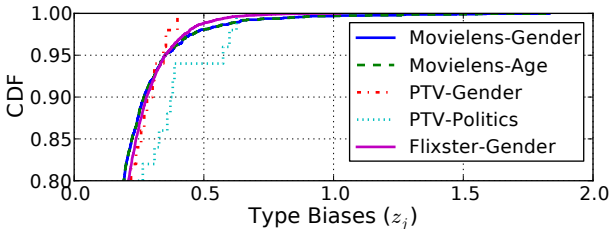


Figure 1: Cumulative distributions of the type bias ($|z_j|$) for the different datasets, zoomed to the top 20% of items.

Evaluation Metrics. We evaluate classification performance through the area under the curve (AUC) metric, and prediction performance through the root mean squared error (RMSE). If a recommender system uses our method to maliciously learn user features, it is important that such a mechanism for strategic solicitation of ratings has a minimal impact on the quality of recommendations, otherwise the user may detect its hidden agenda. We measure the quality of recommendations by holding out an evaluation set of 10 items for each user. After every 10 questions (solicited ratings) we predict the ratings on the evaluation set by applying ridge regression using the provided ratings and item profiles to learn a user profile. We predict the ratings on the evaluation set and compute the RMSE over all users.

Parameter settings. We split each dataset into training and testing and perform MF with 10-fold cross validation. We learn the item latent factors required by FBC using the training set, with type biases for age, gender and political affiliation as applicable to the three datasets. For MF, we use 20 iterations of stochastic gradient descent [17] to minimize (3), using the same regularization parameter for users and movies. Through 10-fold cross validation we determined the optimal dimension to be $d = 20$, and the optimal regularization parameter to be 0.1, for each of the biases. We also compute the optimal λ used in the classifier (8) through 10-fold cross validation to maximize the AUC, resulting in $\lambda = 100$ for gender and $\lambda = 200$ for age for the Movielens dataset, $\lambda = 10$ for gender and political views for the PTV dataset, and $\lambda = 200$ for gender for the Flixster dataset.

6.2 Passive Learning

Figure 2 shows the AUC and RMSE obtained using MaxGap, and two other passive methods – Random and Entropy. For reference, we also show the performance of our active method IncFBC. Random selection is a natural baseline as users may rate items in any arbitrary order. The second method, Entropy, presents items to the user in descending order of their rating entropy, i.e., start with items that have polarized ratings. This method was shown to be efficient in a cold-start setting in [24] as it can quickly build user profiles in a matrix factorization based recommender system.

AUC. Figure 2a shows that MaxGap performs significantly better than the other passive methods on both Movielens and PTV datasets. For the first 10 questions, which are critical when considering the need for quick inference, it is the best passive method on all datasets. Interestingly, on Movielens-Gender and PTV-Politics, MaxGap performs very similar to the adaptive, more complex IncFBC. As a result of the greedy nature of MaxGap, we expect it to perform well on datasets that have items with large biases. To better understand MaxGap's performance, Figure 1 shows the top 20% of the cumulative distributions of the type biases over the set of items in the different datasets. The plot clearly shows that the items in PTV-Gender have the lowest biases, resulting in the poorest performance of MaxGap. Conversely, in Movielens-Gender, the biases are the largest, thus MaxGap performs well, in par with the adaptive IncFBC. In PTV-Politics the biases are not overly high, but since most users rate all items, a few discriminating items are sufficient to enable MaxGap to perform well. This observation is supported by the findings of [6] that identified 5 TV-shows that immediately reveal a user's political views.

RMSE. Figure 2b plots the RMSE over increasing number of rated items, for MaxGap, Random, and Entropy, along with IncFBC. Even though IncFBC and MaxGap are designed to explore a specific attribute of the user's profile, they perform very well. Their RMSE is very close to that of Random and Entropy, with the MaxGap visibly worse only in one case, the PTV-Gender dataset. Since IncFBC and MaxBias focus on quickly learning a singe attribute of the user's profile, it is expected that they perform worse than the other methods, that aim to explore attributes more broadly. However, the figures show that IncFBC and MaxGap are almost identical to the other methods, and MaxGap only perform worse in the PTV-Gender dataset. Moreover, in all datasets, IncFBC performs close to a random selection, indicating that IncFBC does not incur significant impact on the RMSE relative to an arbitrary order in which a user may rate items. Finally, IncFBC has an RMSE similar to the entropy method, which is designed to improve the RMSE in a cold-start setting.

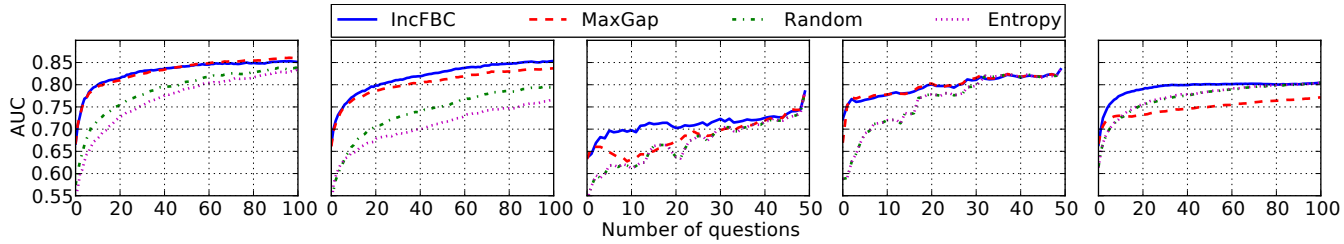
These results show that a malicious recommender system that uses IncFBC to infer a private attribute of its users can also use the solicited ratings to provide recommendations, making it difficult for users to detect the hidden agenda.

6.3 Active Learning

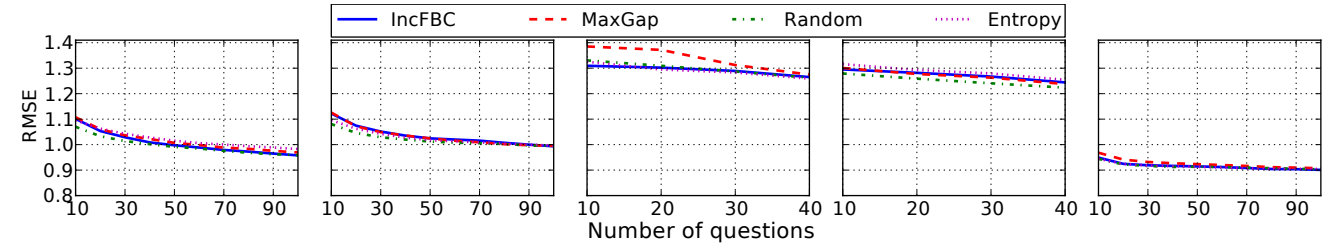
We compare our selection method to the logistic and multinomial classifiers by adapting them to an active learning setting. These classifiers were the top performing among those studied in previous work [30] for gender prediction. Following [30], we train both of these classifiers over rating vectors padded with zeros: an item not rated by a user is marked with a rating value of 0. In order to use logistic and multinomial classifiers in an active learning setting we use the point-estimate (PointEst) method as described in Section 5.4 (see Algorithm 2). For the remainder of this section we refer to PointEst with a logistic and multinomial classifiers as *Logistic* and *Multinomial*, respectively.

AUC. Figure 3a plots the AUC of classification for a given number of question using PointEst and IncFBC selection, for all datasets. PointEst selector enables us to compare FBC with the other classifiers for which we do not have a closed-form solution for selection. In all datasets, the plots show that IncFBC significantly outperforms both logistic and multinomial within a few questions, and reaches an improvement in AUC of 10–30% in the Movielens and Flixster datasets. PointEst with the other classifiers is unable to achieve a good classification accuracy.

To put this in perspective, Table 2 shows the performance of these classifiers, and that of a non-linear classifier SVM with RBF kernel, using all user ratings. Note that this table

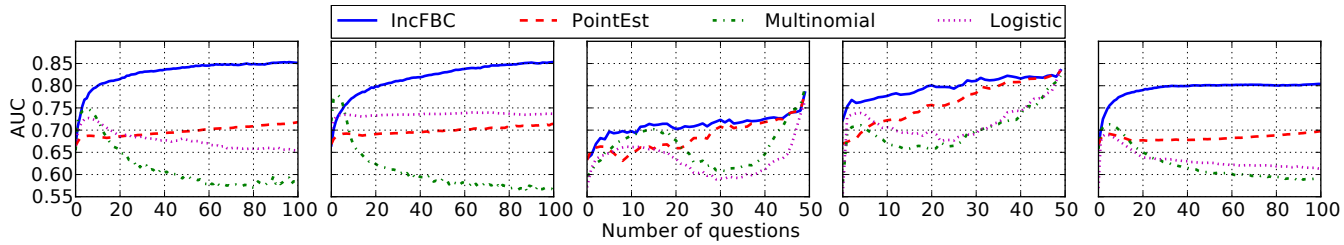


(a) AUC with passive learning

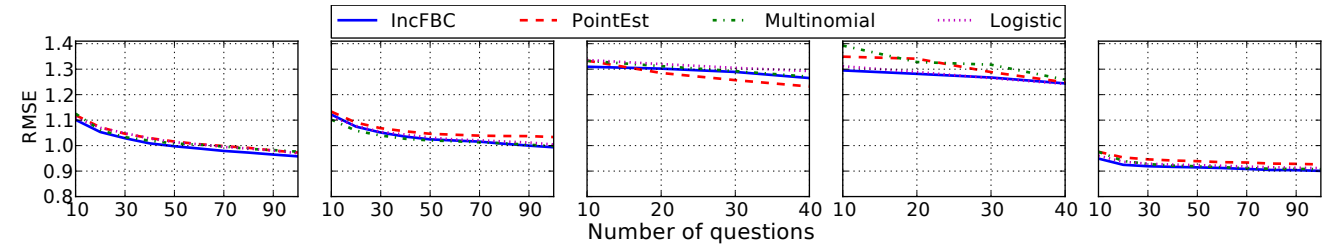


(b) RMSE with passive learning

Figure 2: Average AUC and RMSE of the FBC classifier with increasing questions for different passive selection strategies, and IncFBC for comparison. Datasets (left to right) – Movielens-Gender, Movielens-Age, PTV-Gender, PTV-Political Views, Flixster-Gender.



(a) AUC with active learning



(b) RMSE with active learning

Figure 3: Average AUC and RMSE per number of questions for the three classifiers: IncFBC, multinomial (using PointEst selector) and logistic (using PointEst selector). Datasets from left to right – Movielens-Gender, Movielens-Age, PTV-Gender, PTV-Political Views, Flixster-Gender.

considers *all* ratings performed by *all* users in each dataset, whereas the plots in Figure 3a show the average AUC computed over the users that have rated the indicated number of questions. Logistic and in some cases multinomial classifiers perform significantly better than FBC, when classifying over the entire dataset. This shows that although any of these classifiers could be used for a static attack [30], FBC is better suited to adaptive attacks with fewer available ratings. For instance, using IncFBC with just 20 movies per user we obtain a classification accuracy that is reasonably close to that obtained by static inference techniques which use the complete dataset.

	Movielens		PTV		Flixster
	Gender	Age	Gender	Politics	Gender
FBC	0.827	0.825	0.683	0.748	0.650
Logistic	0.865	0.906	0.756	0.778	0.861
Multinomial	0.810	0.817	0.753	0.758	0.747
SVM (RBF)	0.838	0.893	0.613	0.737	NA

Table 2: AUC of classification with full user history.

RMSE. For completeness, Figure 3b provides the RMSE using the different active methods, showing that IncFBC has a lower RMSE on almost all datasets.

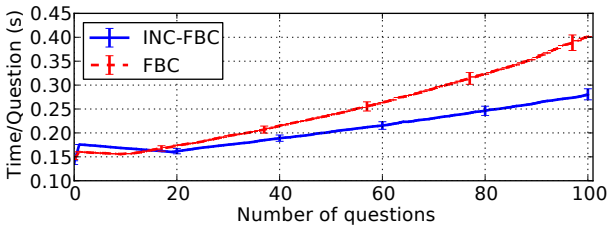


Figure 4: Running time improvement of IncFBC over FBC.

Running Time. Finally, we seek to quantify the improvement in execution time obtained by the incremental computations of IncFBC. We ran both FBC and IncFBC on a commodity server with a RAM size of 128GB and a CPU speed of 2.6GHz. Figure 4 shows the average time per movie selection for both FBC and IncFBC for increasing number of questions (movies presented to the user). The error bars depict the 95% confidence interval surrounding the mean. The plot shows that when the number of questions is small the time per question is relatively constant, and increases with the number of questions. As discussed in Section 5.3, when the number of questions is smaller than the dimension of the factor vectors (in our case $d = 20$), the complexity of the efficient algorithm is dominated by d. In the first few questions FBC is slightly faster than IncFBC as a result of the efficient implementation of inversion for small matrices. However, as the matrix becomes larger, the size of the matrix dominates the complexity and the incremental computations performed in IncFBC are significantly faster than FBC, reaching a speedup of 30%.

7. CONCLUSION AND FUTURE WORK

We presented a new attack that a recommender system could use to pursue a hidden agenda of inferring private attributes for users that do not voluntarily disclose them. Our solution, that includes a mechanism to select which question to ask a user, as well as a classifier, is efficient both in terms of the number of questions asked, and the runtime to generate each question. Moving beyond binary attributes to multi-category attributes is an interesting open question. Exploring the attack from the user's perspective, to better advise users on ways to identify and potentially mitigate such attacks is also an important future direction.

8. REFERENCES

[1] N. Awad and M. Krishnan. The personalization privacy paradox: An empirical evaluation of information transparency and the willingness to be profiled online for personalization. In *MIS Quarterly*, March 2006.

[2] S. Bhagat, I. Rozenbaum, and G. Cormode. Applying link-based classification to label blogs. In *WebKDD*, 2007.

[3] S. Bhagat, U. Weinsberg, S. Ioannidis, and N. Taft. Recommending with an agenda: Active learning of private attributes using matrix factorization, 2013. `http://arxiv.org/abs/1311.6802`.

[4] B. Bi, M. Shokouhi, M. Kosinski, and T. Graepel. Inferring the demographics of search users. WWW, 2013.

[5] J. A. Calandrino, A. Kilzer, A. Narayanan, E. W. Felten, and V. Shmatikov. "You Might Also Like:" Privacy Risks of Collaborative Filtering. IEEE SSP, 2011.

[6] F. Calmon and N. Fawaz. Privacy against statistical inference. In *Allerton*, 2012.

[7] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein. *Introduction to algorithms*. MIT press, 2001.

[8] S. Dasgupta. Analysis of a greedy active learning strategy. In *NIPS*, 2005.

[9] D. Golovin and A. Krause. Adaptive submodularity: A new approach to active learning and stochastic optimization. In *COLT*, 2010.

[10] D. Golovin, A. Krause, and D. Ray. Near-optimal Bayesian active learning with noisy observations. In *NIPS*, 2010.

[11] D. A. Harville. *Matrix Algebra From a Statistician's Perspective*. Springer-Verlag, 1997.

[12] T. Hastie, R. Tibshirani, and J. H. Friedman. *The elements of statistical learning*. Springer-Verlag, 2001.

[13] M. Jamali and M. Ester. A matrix factorization technique with trust propagation for recommendation in social networks. In *RecSys*, 2010.

[14] B. Knijnenburg and A. Kobsa. Making decisions about privacy: Information disclosure in context-aware recommender systems. *ACM Trans on Intelligent Interactive Systems*, 2013.

[15] A. Kobsa, B. Knijnenburg, and B. Livshits. Let's do it at my place instead? attitudinal and behavioral study of privacy in client-side personalization. In *CHI*, 2014.

[16] Y. Koren. Factorization meets the neighborhood: a multifaceted collaborative filtering model. In *KDD*, 2008.

[17] Y. Koren, R. Bell, and C. Volinsky. Matrix factorization techniques for recommender systems. *Computer*, 2009.

[18] M. Kosinski, D. Stillwell, and T. Graepel. Private traits and attributes are predictable from digital records of human behavior. *PNAS*, 2013.

[19] A. Levi, O. Mokryn, C. Diot, and N. Taft. Finding a needle in a haystack of reviews: Cold-start context-based recommender system. In *ACM RecSys*, 2012.

[20] A. Mislove, B. Viswanath, K. P. Gummadi, and P. Druschel. You are who you know: Inferring user profiles in Online Social Networks. In *WSDM*, 2010.

[21] S. Nakajima and M. Sugiyama. Implicit regularization in variational bayesian matrix factorization. ICML, 2010.

[22] A. Narayanan and V. Shmatikov. Robust de-anonymization of large sparse datasets. IEEE SSP, 2008.

[23] R. Nowak. The geometry of generalized binary search. *Transactions on Information Theory*, 2012.

[24] A. M. Rashid, I. Albert, D. Cosley, S. K. Lam, S. M. Mcnee, J. A. Konstan, and J. Riedl. Getting to know you: Learning new user preferences in recommender systems. In *IUI*, 2002.

[25] R. Salakhutdinov and A. Mnih. Probabilistic matrix factorization. In *NIPS*, 2008.

[26] 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 *GlobalSIP*, 2013.

[27] J. Sherman and W. J. Morrison. Adjustment of an inverse matrix corresponding to changes in the elements of a given column or a given row of the original matrix. *Annals of Mathematical Statistics*, 20:621, 1949.

[28] J. Silva and L. Carin. Active learning for online bayesian matrix factorization. KDD, 2012.

[29] D. J. Sutherland, B. Póczos, and J. Schneider. Active learning and search on low-rank matrices. In *KDD*, 2013.

[30] U. Weinsberg, S. Bhagat, S. Ioannidis, and N. Taft. Blurme: Inferring and obfuscating user gender based on ratings. In *ACM RecSys*, 2012.