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

Smriti Bhagat, Udi Weinsberg, Stratis Ioannidis, Nina Taft

{smriti.bhagat, udi.weinsberg, stratis.ioannidis, nina.taft}@technicolor.com
Technicolor
Palo Alto

Abstract

Recommender systems leverage user social and demographic information, *e.g.*, age, gender and political affiliation, to personalize content and monetize on their users. Oftentimes, users do not volunteer this information due to privacy concerns or to the lack of initiative in filling out their profile information. In this work, we illustrate a new threat in which the system can nevertheless learn the private attribute for those users who do not voluntarily disclose them. We design an active attack that solicit ratings for strategically selected items, and could thus be used by a recommender system to pursue its hidden agenda. Our method is based on a novel usage of Bayesian matrix factorization in an active learning setting. Evaluations, on multiple datasets, illustrate that such an attack is indeed feasible and can be carried out using significantly fewer rated items than the previously proposed static inference methods. Importantly, this threat can succeed without sacrificing the quality of the regular recommendations made to the user.

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. A rich user profile allows a recommender system to better personalize its service, 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). In order for a recommender system to obtain additional social and demographic information about its users, it can choose to explicitly ask users for this information. While some users may willingly disclose it, others may be more privacy-sensitive and may explicitly elect not volunteer any information beyond their ratings. Users are increasingly becoming privacy-conscious. Even on Facebook, a social network in which people are likely to reveal intimate details about themselves, studies show that almost 30% of users do not disclose their age, 20% do not disclose their gender, and more than 95% do not disclose their political affiliation [19, 28]. We expect these figures to be much higher for recommender systems, where the benefits of disclosing private information may not be obvious to a user.

Although users may wish to withhold some demographic information, a recommender systems can still undermine users' attempt at privacy. In our previous work [29], we showed that users' movie ratings can be used to predict their gender with 80% accuracy. Other studies have also shown the potential to infer demographics from a range of online user activities [1, 2, 15, 16]. In this work, we consider a recommender system that offers a legitimate service, yet is simultaneously malicious in the sense that it purposefully attempts to extract certain attributes from those who choose to withhold them. Unlike previous work that study 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 often ask users to rate a few items, as means to assist them in 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 recommender systems do not disclose how they choose the order of items that a user is asked to rate to propose

a new active 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 [14]. 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 classification algorithm. This classification method is consistent with the underlying assumptions employed by matrix factorization.
- Second, we demonstrate that the resulting classification method is well suited for the task of active learning of a user's type. We design a method to select the next item to ask a user to rate, so that each subsequent selection is made to maximize the expected confidence of the inference, equivalently, minimize the expected risk of misclassifying the user's private attribute.
- Third, we show that our active learning method can be implemented efficiently, 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 classifier and selection methods on three real-world datasets: Movielens, Flixster and Politics-and-TV. We show that our method consistently outperforms baseline methods; 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.

In the remainder of the paper, we first review related work (**??**) and formulate our problem (**??**). We then present our factor-based classifier (**??**) and our active learning method (**??**), and conclude after presenting empirical results (**??**).

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.* [1] show that it is possible to learn age and gender information from blogs. Mislove *et al.* [16] 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.* [2] illustrated how demographics can be inferred from search queries.

Recommender systems were shown to be exploitable by several works utilizing passive, off-line attacks. Most recently, Goyal and Lakshmanan [8] show that recommender systems can be used for targeted advertising to larger audience by carefully selecting the set of seed users that are presented with the product's ad. Calandrino *et al.* [3] propose a method for inferring transactions made by users based on temporal changes that can be observed from the output of recommender systems. In an extreme case of inference, Narayanan *et al.* [18] famously showed that movie ratings can be used for full de-anonymization (through a linkage attack), thereby enabling the unique identification of users.

In previous work [29], we empirically studied how to infer a user's gender from her movie ratings using a variety of different classifiers. We showed that logistic regression and SVMs succeed with an accuracy close to 80%. In this paper, we depart from [29] 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 factor-based classifier is very well suited for this task.

The Bayesian model underlying MF (discussed in detail in **??**) was recently employed by Silva and Carin [27] 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

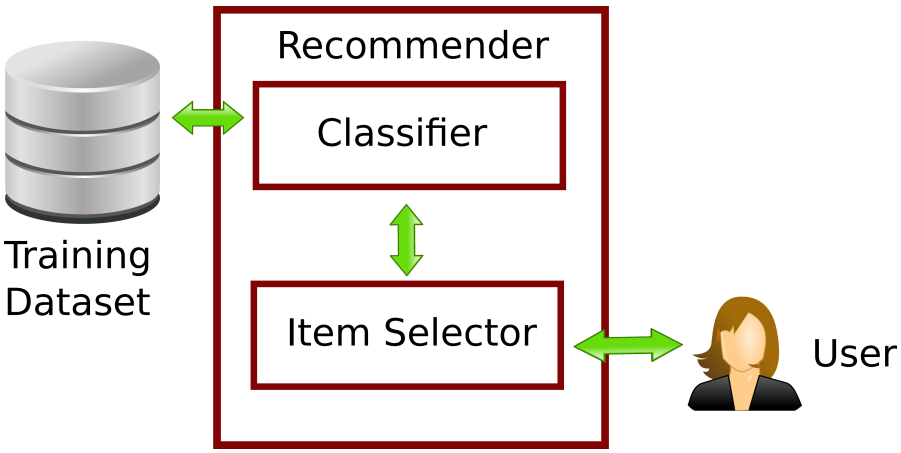


Figure 1: System description. The recommender system uses a dataset of user ratings to train a type classifier. An item selection process proposes items to the user, which she subsequently rates; these ratings are used to infer her type.

noise and prior assumptions underlying MF, which we also employ in our work. Nevertheless, we depart from the above work as the goal of our learning task is to discover a user's demographic information, captured by a categorical type, rather the user and item factors. In turn, this leads to a significantly different item selection process, built upon on the factor-based classifier we introduce, rather than on entropy minimization.

Biases have been used extensively to improve prediction performance in MF [13, 14]. Our introduction of demographic-specific biases is not for improving prediction per se–though this does happen; rather, incorporating such biases allows us to use the classic MF model as a basis for classification and, subsequently, active learning.

In the classic active learning setting [5, 6], 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. The goal of the learner is to adaptively select which function evaluations to conduct – *i.e.*, which input queries to submit to the oracle – in order to discover the valid hypothesis as quickly as possible. In the case of a *noiseless* oracle, that always returns the correct evaluation on a query, Dasgupta [5] proposes an algorithm called Generalized Binary Search (GBS) that discovers the valid hypothesis in a number of queries within a polylogarithmic factor from the optimal. Recently, Golovin *et al.* improved upon this bound (in terms of constants) by casting active learning in the context of adaptively submodular optimization [6].

Our setup can be cast into the above framework in the context of a noisy oracle, whose evaluations may not necessarily be exact: the hypothesis space comprises linear maps, each corresponding to a unique user profile and type; the query space comprises the items to be ratings, and ratings can be seen as noisy evaluations of the linear map over the query space. GBS is known to yield arbitrarily suboptimal results in the presence of noise [7]. In general, though algorithms with general performance guarantees are not known in the presence of an oracle with arbitrary noise, such guarantees can be provided for restricted noise models (see, *e.g.*, [20] and [7]). Unfortunately, none of these models directly apply to the noise setting we encounter here.

3 System Description

3.1 Problem Statement

We consider a recommender system, depicted in **??**, 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.

We consider these two goals because in order 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, so that the user is unaware of the hidden agenda.

As both our classifier and item selection methods will rely heavily on matrix factorization, we review this as well as the latent factor model that belies it in the remainder of this section.

3.2 Data Model

We use the following notation to describe the training dataset accessible to the recommender system. 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 user $i \in \mathcal{N}$ gives 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 in the dataset 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, we assume that 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 given by the following bi-linear relationship

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

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 (**??**) are used in many works on so-called Bayesian matrix factorization (see, *e.g.* [11, 17, 22, 23, 25, 27]), and have several advantages. First, maximum a-posteriori estimation of the latent factors (*i.e.* user and item profiles and biases) under these priors reduces to a least squares minimization augmented with ℓ_2 regularization terms, discussed further in the next section. This estimation method is known to perform very well in determining parameters that yield good rating prediction results, thereby motivating the selection of the above priors. Second, endowing model parameters with such priors allows the analysis of matrix factorization from a Bayesian perspective; we exploit this point of view both in the development of our factor-based classifier as well as our active learning algorithm.

3.3 Matrix Factorization as Maximum A-Posteriori Estimation

Using a dataset of ratings as described above, a recommender system can extract user and item profiles for the purpose of predicting future user ratings. The technique typically used to learn these profiles from a training set, namely, matrix factorization, can be viewed as maximum a posteriori estimation based on the noise and prior assumptions presented in the previous section.

More specifically, under (**??**) and (**??**), the maximum likelihood estimation of the model parameters reduces

to the minimization of the following (non-convex) regularized square 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 (**??**) through standard methods [14], such as gradient descent or alternating minimization, while λ and μ are typically computed through cross-validation [10].

The last two terms of (**??**) are called the *regularization terms*. In practice, they are introduced to avoid overfitting. Beyond the Bayesian perspective described above, another motivation behind the introduction of such terms is the prior belief that the model ought to be simple; the regularization terms penalize the complexity of the parametrized model (through the penalty on the ℓ_2-norms of profiles). As such, they act as "Occam's razor", favoring parsimonious or simpler models over models that better fit the observed data. The Bayesian point of view also agrees with this intuition, as the Gaussian priors indeed bias the parameter selection to profiles with small norm.

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 (**??**) 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 (**??**) and the priors (**??**).

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 (**??**), *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:

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

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). Hence, the joint prior distribution of parameters $u \in \mathbb{R}^d$ and $t \in \mathcal{T}$ is:

$$\Pr(u, t) = 0.5 \times \frac{1}{(\sigma_u\sqrt{2\pi})^d} e^{-\|u\|_2^2 / 2\sigma_u^2} \tag{6}$$

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

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* (**??**) *and prior* (**??**)*, the* FBC *classifier is given by*

$$\hat{t}(r_A) = \begin{cases} +1, & \text{if } \delta_A^T M_A \bar{r}_A \geq 0 \\ -1, & \text{o.w.} \end{cases} \tag{8}$$

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* $\lambda = \frac{\sigma_0^2}{\sigma_u^2}$.

Proof. Under model (**??**) and prior (**??**), conditioned on type t, the ratings r_A a user gives items in a set $A \subseteq [M]$ are distributed according to:

$$\begin{aligned} \Pr(r_A \mid t) &= \frac{\int_{u \in \mathbb{R}^d} e^{-\frac{\|r_A - V_A u - z_{At}\|^2}{2\sigma_0^2} - \frac{\|u\|_2^2}{2\sigma_u^2}} du}{(\sigma_0\sqrt{2\pi})^{|A|}(\sigma_u\sqrt{2\pi})^d} \\ &= \frac{e^{\frac{(r_A - z_{At})^T \left(V_A \Sigma_A^{-1} V_A^T - I\right)(r_A - z_{At})}{2\sigma_0^2}}}{(\sigma_0\sqrt{2\pi})^{|A|}(\sigma_u/\sigma_0)^d \sqrt{\det(\Sigma_A)}} \end{aligned} \tag{9}$$

where $\Sigma_A \equiv \lambda I + V_A^T V_A$ and $\lambda \equiv \frac{\sigma_0^2}{\sigma_u^2}$. Hence, the posterior probability of the user's type is given by:

$$\Pr(t \mid r_A) = \frac{e^{(r_A - z_{At})^T \left(V_A \Sigma_A^{-1} V_A^T - I\right)(r_A - z_{At})/2\sigma_0^2}}{\sum_{t' \in \mathcal{T}} e^{(r_A - z_{At'})^T \left(V_A \Sigma_A^{-1} V_A^T - I\right)(r_A - z_{At'})/2\sigma_0^2}} \tag{10}$$

Type $t = +1$ is thus most likely if

$$\begin{aligned} (r_A - z_{A+})^T \left(V_A \Sigma_A^{-1} V_A^T - I\right)(r_A - z_{A+}) - \\ (r_A - z_{A-})^T \left(V_A \Sigma_A^{-1} V_A^T - I\right)(r_A - z_{A-}) \geq 0 \end{aligned}$$

and it is easy to verify that (**??**) follows. □

4.3 Discussion

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.

Relationship to LDA. Second, for a given set of items A, the classifier defined over ratings r_A is a *linear* classifier. In particular, (**??**) defines a hyperplane in $\mathbb{R}^{|A|}$ above which the user type is classified as $+1$ and below which the type is classified as -1. In fact, when restricted to a specific set of items A, (**??**) can be seen as classification through Linear Discriminant Analysis (LDA) [10]. More formally, the proof of Theorem 1 uses the fact that the ratings $r_A \in \mathbb{R}^{|A|}$ are normally distributed with a mean that depends on the user type (*cf* eq. (**??**)) and a covariance $M_A = (I - V_A \Sigma_A^{-1} V_A)$, as defined in Theorem 1. As such, given a uniform prior on the types, the most likely type can indeed be determined through LDA, which yields a decision boundary precisely as in (**??**) (see, *e.g.*, eq. (4.9) pp. 108 of [10]). Nevertheless, FBC significantly departs from classical LDA in that all classifiers across all sets $A \subseteq \mathcal{M}$ are trained jointly.

Algorithm 1 FBC Active Learning

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 (**??**)
  $j^* \leftarrow \arg\min_{j \in \mathcal{M} \setminus A} L_j$
  Query user to obtan 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 Active Learning

The second task in designing this threat is to find a user's type *quickly*. We seek to design an algorithm that adaptively selects items to show to the user, who subsequently rates them. The selection of the next item to show is based on the ratings that the user has provided so far, and aims to select an item whose rating would be most informative. More precisely, the active learning algorithm we propose selects an item at each step whose rating increases the confidence of the classifier the most; maximizing classifier confidence is the same as minimizing the risk of misclassification (we formally define both confidence and risk below). Our approach has several important advantages. First, the expected risk of FBC can be computed exactly in closed form. This allows us to incorporate both the effect that a rating has on the performance of the classifier, and how variable our estimate is. Second, we demonstrate how this computation can be performed incrementally, thus significantly reducing the computational complexity of item selection operations. Finally, one might be tempted to consider a simple point estimate of risk, *i.e.*, the risk for a single predicted rating, rather than computing the expected risk over a probability distribution of predicted ratings. Although we do not need the point estimate method with FBC, we discuss it here and use it in comparative evaluations since it can be used with any classifier, for which a closed-form solution is not trivial or might not even exist.

5.1 FBC Selection Strategy

We use the data model and the FBC classifier presented in **??** to propose a method for selecting which items to present to the user. Let $\hat{t}$ be the FBC classifier defined by (**??**). 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. Note that, 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 (**??**), FBC selects the type that has the maximum posterior probability, the expected risk is at most (and the confidence at least) 0.5.

Our active learning algorithm 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, given by:

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

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

Under noise model (**??**) and prior (**??**), 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 the appendix:

$$L_j = \frac{\int_{r_j} e^{-\frac{\bar{r}_{A_j}^T M_{A_j} \bar{r}_{A_j} + 2|\delta_{A_j}^T M_{A_j} \bar{r}_{A_j}| + \delta_{A_j}^T M_{A_j} \delta_{A_j}}{2\sigma_0^2}} dr_j}{\sqrt{\det \Sigma_{A_j}}} \tag{11}$$

where $A_j = A \cup \{j\}$. We note that 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, and *no numerical integration is necessary.* We also present the corresponding closed form formula in terms of the error function erf in the appendix. The active learning/item selection algorithm is summarized in Algorithm 1. 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.

5.2 IncFBC Efficient Implementation of FBC Selection Strategy

The algorithm presented in the previous section 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 (**??**). Though having a closed form formula for (**??**) 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 [4]. 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

$$O(|A| \times d^2 + |A|^2 \times d + d^{2.807}). \tag{12}$$

However, the performance of these computations can be significantly reduced by constructing these matrices incrementally. In particular, M_{A_j}, $\Sigma_{A_j}^{-1}$ and $\det(\Sigma_{A_j})$ can be computed very efficiently from M_A, Σ_A^{-1}, and $\det(\Sigma_A)$. These calculations exploit the fact that

$$\Sigma_{A_j} = \Sigma_A + v_j v_j^T,$$

i.e., results from Σ_i through a *rank-one* update. We discuss how each of these can be computed below.

Incremental computation of $\det(\Sigma_{A_j})$**.** The determinant can be computed incrementally through the Matrix Determinant Lemma [9], which relates the determinant of a matrix to the determinant of a rank-one update. In particular:

$$\det(\Sigma_{A_j}) = (1 + v_j^T \Sigma_A v_j) \det(\Sigma_A). \tag{13}$$

This computation requires only $O(d^2)$ multiplications.

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 [26], which gives:

$$\Sigma_{A_j}^{-1} = \Sigma_A^{-1} - \frac{\Sigma_A^{-1} v_j v_j^T \Sigma_A^{-1}}{1 + v_j^T \Sigma_A^{-1} v_j}, \tag{14}$$

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

Incremental computation of M_{A_j}**.** Finally, using (**??**), we can also reduce the cost of computing M_{A_j}, as it can be written as:

$$M_{A_j} = \begin{bmatrix} M_A + \frac{\phi \phi^T}{1 + v_j^T \Sigma_A^{-1} v_j} & -\xi \\ -\xi^T & 1 - v_j^T \xi \end{bmatrix} \tag{15}$$

Algorithm 2 IncFBC Active Learning

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

$A \leftarrow \emptyset$, $r_A \leftarrow \emptyset$
$\Sigma_A^{-1} \leftarrow \lambda^{-1} I$, $\det(\Sigma_A) \leftarrow \lambda^{-d}$, $M_A \leftarrow \emptyset$,
repeat
  for all $j \in \mathcal{M} \setminus A$ **do**
    $\det(\Sigma_{A_j}) \leftarrow (1 + v_j^T \Sigma_A v_j) \det(\Sigma_A)$
    $\Sigma_{A_j}^{-1} \leftarrow \Sigma_A^{-1} - \frac{\Sigma_A^{-1} v_j v_j^T \Sigma_A^{-1}}{1 + v_j^T \Sigma_A^{-1} v_j}$
    $\xi \leftarrow V_A(\Sigma_{A_j}^{-1} v_j)$, $\phi \leftarrow V_A(\Sigma_A^{-1} v_j)$
    $M_{A_j} = \begin{bmatrix} M_A + \frac{\phi\phi^T}{1 + v_j^T \Sigma_A^{-1} v_j} & -\xi \\ -\xi^T & 1 - v_j^T \xi \end{bmatrix}$
    Compute L_j through (**??**)
  $j^* \leftarrow \arg\min_{j \in \mathcal{M} \setminus A} L_j$
  Query user to obtan r_{j^*}
  $A \leftarrow A \cup \{j^*\}$, $r_A \leftarrow r_A \cup r_{j^*}$
  $\det(\Sigma_A) \leftarrow \det(\Sigma_{A_{j^*}})$, $\Sigma_A^{-1} \leftarrow \Sigma_{A_{j^*}}^{-1}$, $M_A \leftarrow M_{A_{j^*}}$
until $\Pr(\hat{t}(r_A) \mid r_A) > \tau$

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 from (**??**) 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 permit such performance without the use of sophisticated matrix inversion or multiplication algorithms, such as Strassen's algorithm. The resulting algorithm is outlined in Algorithm 2, and an empirical comparison of the two implementations is shown in **??**.

5.3 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 [27]). 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 3. 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 [10] over the observed ratings r_A and the corresponding profiles V_A (see Algorithm 3 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

$$\hat{r}_j = \hat{u}^T v_j + z_{j\hat{t}},$$

Algorithm 3 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 $1 - L_{j^*} > \tau$

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 of the risk 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 is hard to obtain in a closed form. 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 (**??**). 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.

6 Evaluation

6.1 Datasets and Experimental Setup

We evaluate our method using three datasets: Movielens[2], Flixster [12], and Politics-and-TV (PTV) [24]. 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 35–65). Flixster is a similar movie ratings dataset, also containing 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. **??** 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 (labeled in the table). **??** shows the cumulative distribution function (CDF) of the number of ratings per user across the three preprocessed datasets. We see that for the Movielens and Flixster datasets, there are many users with hundreds of items rated in their profile.

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 [14] to

[2] `http://www.grouplens.org/node/73`

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: Datasets statistics

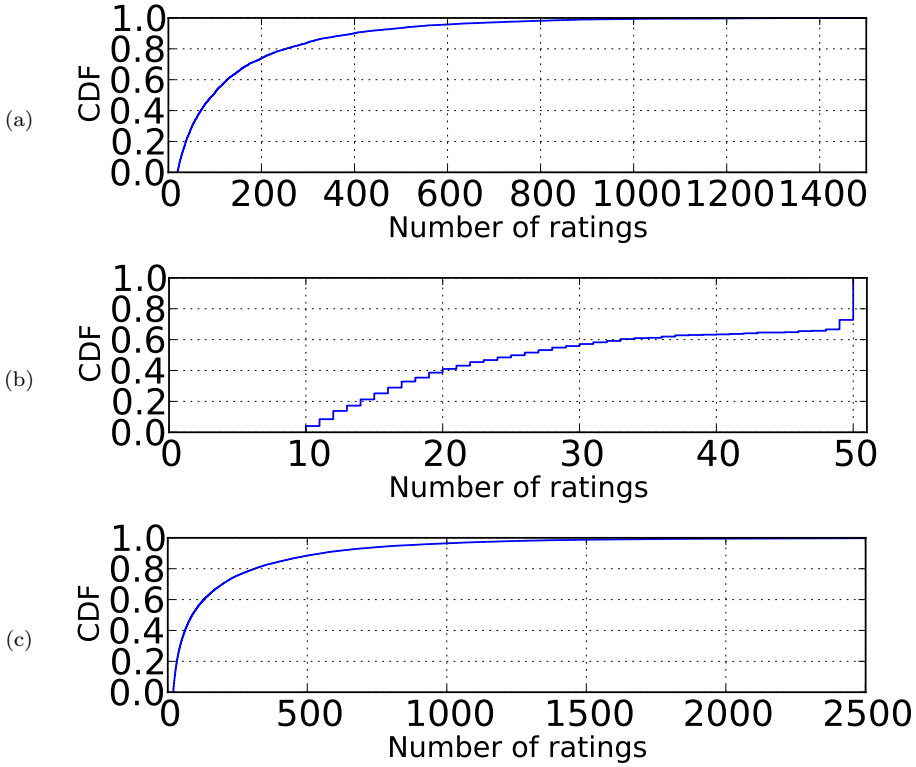


Figure 2: Cumulative distribution of number of ratings per user for (a) Movielens (b) Politics-and-TV (c) Flixster

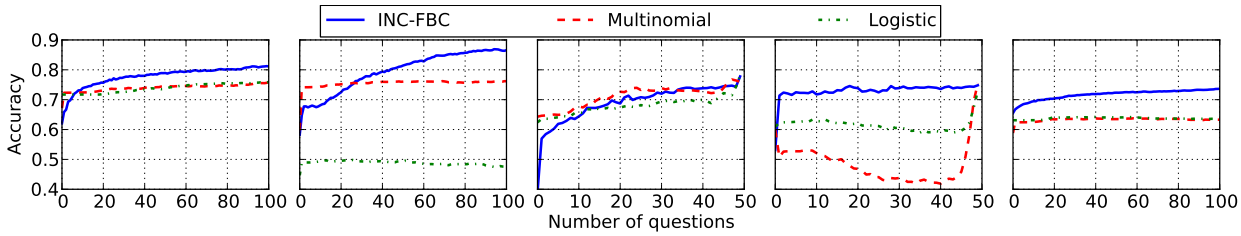
minimize (**??**), using the same regularization parameter for users and movies. Through 10-fold cross validation we determined the optimal dimension to be $d = 20$, and the optimal regularization parameter to be 0.1, for each of the biases. We also compute the optimal parameter λ used in the classifier (**??**) through 10-fold cross validation, selecting the parameter that maximizes the area under the curve (AUC) obtaining values $\lambda = 100$ for gender and $\lambda = 200$ for age for the Movielens dataset, $\lambda = 1$ for gender and $\lambda = 10$ for political views for the PTV dataset, and $\lambda = 200$ for gender for the Flixster dataset.

6.2 Classification

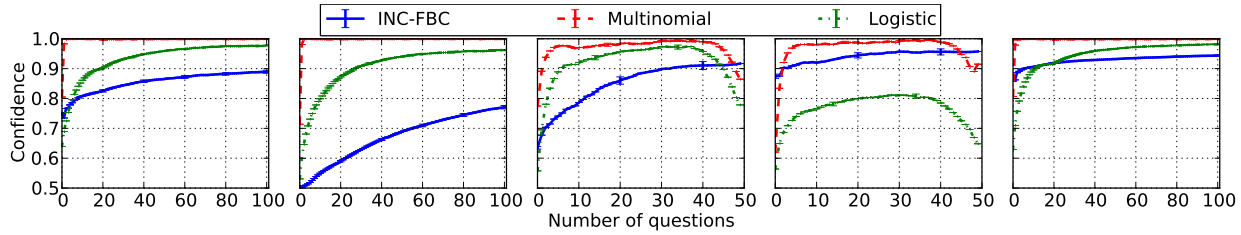
We first look at the performance of static classifiers that have access to the entire user history, in some cases with hundreds of ratings. In a regular recommender system, this history can take many months to accumulate. For this static classification task, we compare FBC to the state-of-the-art, in **??**, and show both the AUC and the accuracy (fraction of users classified correctly) for FBC, logistic regression and multinomial classifier. The latter two were the top performing among the classifiers studied in our previous work [29] in the context of predicting gender. Following [29], 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. Overall, the table shows that the performance of

	Movielens				Politics-and-TV				Flixster	
	Gender		Age		Gender		Political Views		Gender	
	AUC	Accuracy	AUC	Accuracy	AUC	Accuracy	AUC	Accuracy	AUC	Accuracy
FBC	0.827	0.773	0.825	0.751	0.683	0.646	0.748	0.695	0.650	0.606
Logistic	0.865	0.804	0.906	0.827	0.756	0.705	0.778	0.707	0.861	0.789
Multinomial	0.810	0.764	0.817	0.761	0.753	0.709	0.758	0.703	0.747	0.725

Table 2: Classification with full user history; item selection not needed



(a) Accuracy



(b) Confidence

Figure 3: Average accuracy and confidence 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

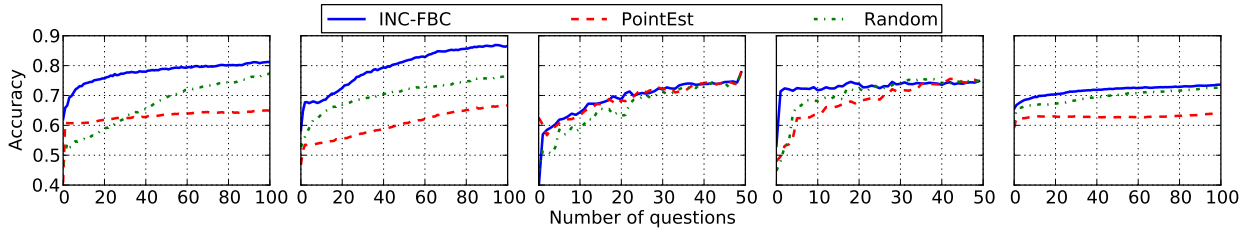


Figure 4: Average accuracy of the FBC classifier per number of questions with different selection strategies. Datasets from left to right – Movielens-Gender, Movielens-Age, PTV-Gender, PTV-Political Views, Flixster-Gender

FBC is close to the state-of-the art on the Movielens dataset, and slightly worse on the other two datasets. Although any of these classifiers could be used for a static attack, we will see below that FBC is better suited to adaptive attacks with fewer ratings.

6.3 Active Learning

In our active learning 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 active learning 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 a classifier on the training set – in the case of FBC we learn the item profiles and biases. Then, for each user in the evaluation set, we select items to show that the user rates incrementally. We use the selector to pick an item (we refer to the item, either a movie or a TV-show, as a "question" that is presented to the user), limiting

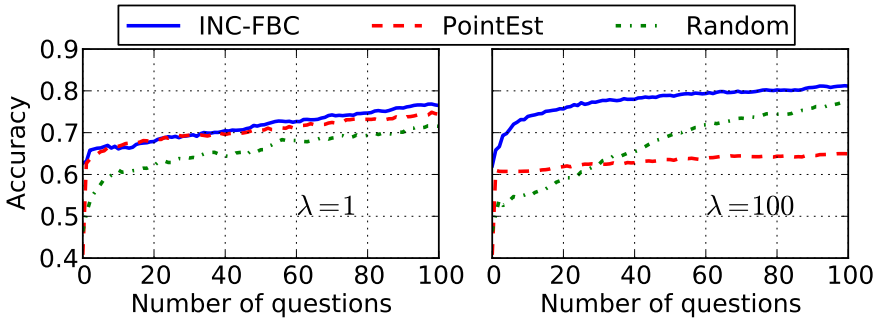


Figure 5: Effect of λ on PointEst in Movielens-Gender

our selections only to the set of the movies that the user has rated. We then emulate a user "response", by assigning the true user rating to the selected movie. At each iteration, we use the classifier to infer the private type and also derive the confidence in this classification. We note that we must limit the selector to the set of rated movies and not the entire movie catalog because we have no rating information regarding movies that the user did not actually rate.

Using this evaluation process we compute the accuracy and the average confidence of the classifier across all users, as a function of the number of questions presented to the users. Notice that not all users have rated the same number of movies. For instance, as seen in **??**, roughly 50% of the users of Movielens have rated less than 100 movies out of 3000. Therefore, we limit the number of questions asked to 100 for Movielens and Flixster and all 50 for PTV.

We seek to compare the performance of FBC with other classification methods. Unless specified, all evaluations were done using the efficient incremental implementation IncFBC. We compare our selection method to the logistic and multinomial classifiers with the point-estimate (PointEst) active learning method described in **??** (see Algorithm 3). For the remainder of this section we refer to PointEst with a logistic and multinomial classifiers as *Logistic* and *Multinomial*, respectively.

Accuracy and Confidence. **??** plots the accuracy of classification for a given number of ratings ("questions") ordered using PointEst, 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. Note that these plots are not directly comparable with **??**, because the table considers *all* ratings performed by *all* users in each dataset, whereas the plots show the average accuracy computed over the users that have rated no more than the indicated number of questions. In all datasets, the plots show that IncFBC either outperforms or is very close to both logistic and multinomial within a few questions, and reaches an improvement in accuracy of over 10% in the Movielens and Flixster datasets. Using IncFBC with just 20 movies per user we obtain a classification accuracy that is close or even better than that obtained by static inference techniques which use the complete dataset. PointEst with the other two classifiers starts in some cases with a relatively good accuracy (around 0.7), however, these are essentially the class priors, as can be seen in **??**. Furthermore, the improvement in accuracy is extremely slow, indicating that these classifiers do not substantially improve beyond the class priors, whereas IncFBC exhibits a much faster increase in accuracy. We note that this is in stark contrast with the performance of these classifiers over the entire dataset (**??**), where both classifiers perform significantly better.

?? shows the average corresponding classification confidence. Compared to **??**, we see that both logistic and multinomial are "over-confident", in that they assign high confidence to users even while failing to classify them correctly. Moreover, in the PTV dataset, the final TV-shows selected by both logistic and multinomial actually reduce their confidence. These are TV-shows that the selector would not ask to rate in the presence of better options, but are eventually selected because of the limited number of shows available (50). In contrast, IncFBC is consistent and robust to the addition of such items, increasing both in terms of confidence and accuracy.

Comparing Selectors. Given the relative complexity of the IncFBC selector, a natural question is how well does the FBC classifier perform with simpler selectors. We thus compare the results of IncFBC, to PointEst, which essentially approximates the optimal selector, and to a random selector that selects items uniformly at random from the set of rated items. **??** illustrates the results of this comparison across different datasets. Clearly, IncFBC consistently outperforms other selectors by achieving $3-20\%$ higher accuracy in fewer questions.

We observe that for the PTV dataset, the performance of PointEst is very close to IncFBC, however, for the other datasets, it is significantly lower than IncFBC. On careful investigation, we find that the performance

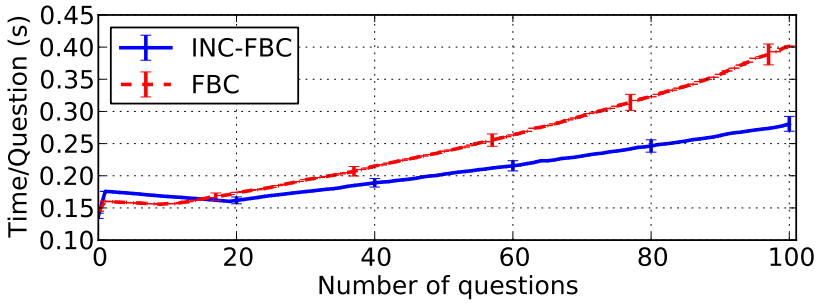


Figure 6: Running time improvement of IncFBC over FBC

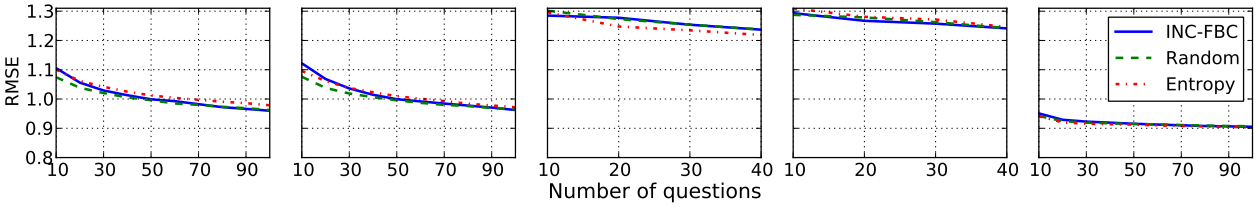


Figure 7: Impact on recommendation as captured by the Root Mean Squared Error (RMSE) of predicted ratings on datasets (left to right) Movielens-Gender, Movielens-Age, PTV-Gender, PTV-Political Views, Flixster-Gender

of PointEst is greatly affected by the amount of noise in the dataset. **??** shows the results of this comparison using two different values of λ, 1 and 100, on the Movielens-Gender dataset. Recall that the λ parameter is used to estimate the amount of noise that exists in the users' ratings; a small λ assumes that there is little noise, therefore PointEst should perform close to IncFBC. When the noise is large, PointEst fails to correctly estimate the expected risk, and performs significantly worse than IncFBC. This is precisely the behavior we observe in **??**, the PointEst selection closely follows IncFBC for $\lambda = 1$, and fails to do so for $\lambda = 100$. For all datasets, the optimal value of lambda is found through cross validation ($\lambda = 100$ for Movielens age and gender), and correctly captures the inherent noise in the data. Similarly, the optimal λ was 200 for Flixster, and 1 and 10 for PTV-Gender and PTV-Political Views, respectively.

Running Time. Next, we seek to quantify the improvement in execution time obtained by the incremental computations of IncFBC. To this end, we ran both FBC and IncFBC on a commodity server with a RAM size of 128GB and a CPU speed of 2.6GHz. **??** 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 as the number of questions increases the time per question also increases. As discussed in **??**, 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. This is the 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%.

Impact on Recommendations. Finally, if a recommender system uses FBC 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 the 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. Using the predicted ratings, we compute the root mean squared error (RMSE) over the evaluation sets of all users. In order to understand how IncFBC affects the RMSE, we compare it with two other item selection methods. The first is a random selection, which we consider to be a natural baseline as users may rate items in any arbitrary order. The second method 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 [21] as it can quickly build

user profiles in a matrix factorization based recommender system.

?? plots the RMSE for an increasing number of rated items, using IncFBC, random, and entropy-based ordering. The figures show that in all datasets, IncFBC performs almost identical 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. Furthermore, IncFBC has an RMSE similar to the entropy method, which is directly designed to improve the RMSE in a cold-start setting. We note that while entropy-based method may have lower RMSE than IncFBC when considering very few questions (*e.g.*, less than 5) its performance is similar as questions increase. Overall, 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.

7 Conclusion and Future Work

In this paper, we present 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. After evaluation on 5 datasets, we showed that we consistently outperform other baseline methods in terms of classification accuracy, and that we do so with far fewer questions.

We plan to pursue some important extensions to broaden the scope our our work. First, we intend to move beyond binary attributes to multi-category attributes. Second, we plan to study the problem of selecting questions to infer multiple private attributes (*e.g.*, age and gender) simultaneously. In addition, we want to consider broader use cases, such as the impact of our attack for users with pre-existing profiles (having already rated some items), where the recommender system's goal is to improve its confidence in the inferred attribute. This could be useful when a subset of the users are hard to classify. Finally, we seek to explore the attack from the user's perspective, to better advise users on ways to identify and potentially mitigate such attacks.

APPENDIX

Derivation of Equation (??) For $A_j = A \cup \{j\}$ and t^c the binary complement of t, the expected risk $\mathbb{E}[L(\hat{t}(r_{A_j})) \mid r_A]$, if the rating for movie j is revealed is

$$\begin{aligned}
&\int_{r_j \in \mathbb{R}} \Pr(\hat{t}^c(r_{A_j}) \mid r_{A_j}) \Pr(r_{A_j} \mid r_A) dr_j \\
&= \int_{r_j \in \mathbb{R}} \Pr(\bar{b}(r_{A_j}) \mid r_{A_j}) \frac{\Pr(r_{A_j})}{\Pr(r_A)} dr_j \\
&\overset{(??),(??)}{=} C \frac{\int_{r_j \in \mathbb{R}} e^{y_{\hat{t}^c(r_{A_j})}^T \left(V_{A_j} \Sigma_{A_j}^{-1} V_{A_j}^T - I\right) y_{\bar{b}^c(r_{A_j})} / 2\sigma_0^2} dr_j}{\sqrt{\det(\Sigma_{A_j})}}
\end{aligned}$$

where $y_{\hat{t}(r_{A_j})} = r_{A_j} - z_{A_j \hat{t}^c(r_{A_j})}$, $\Sigma_{A_j} = \lambda I + V_{A_j}^T V_{A_j}$, and C a term that does not depend on j. The expected risk is thus proportional to:

$$L_j = \frac{\int_{r_j \in \mathbb{R}} e^{-(r_{A_j} - z_{A_j \hat{t}^c(r_{A_j})})^T M_{A_j} (r_{A_j} - z_{A_j \hat{t}^c(r_{A_j})}) / 2\sigma_1^2} dr_j}{\sqrt{|\Sigma_{A_j}|}}$$

where $|\Sigma_{A_j}| = \det(\Sigma_{A_j})$ $M_{A_j} = I - V_{A_j} \Sigma_{A_j}^{-1} V_{A_j}^T$ and $\hat{t}^c(r_{A_j})$ the complement of prediction under r_{A_j}. Under Theorem 1, as $\hat{t}$ is given by (**??**), L_j simplifies to (**??**).

Closed Form of (??). Let $\xi = V_A(\Sigma_{A_j}^{-1} v_j)$, $\phi = V_A(\Sigma_A^{-1} v_j)$, $\mu_1 = M_A + \frac{\phi\phi^T}{1 + v_j^T \Sigma_A^{-1} v_j}$ and $\mu_2 = 1 - v_j^T \Sigma_{A_j}^{-1} v_j$.

Then, from (**??**) we get that:

$$\begin{aligned}\bar{r}_{A_j}^T M_{A_j}\bar{r}_{A_j} + 2|\delta_{A_j}^T M_{A_j}\bar{r}_{A_j}| + \delta_{A_j}^T M_{A_j}\delta_{A_j} = \\ \mu_2\bar{r}_j^2 - 2\bar{r}_A^T\xi\bar{r}_j + |(\delta_j\mu_2 - \delta_A^T\xi)\bar{r}_j + \delta_A^T\mu_1\bar{r}_A - \delta_j\xi^T\bar{r}_A| \\ + \bar{r}_A^T\mu_1\bar{r}_A + \delta_{A_j}^T M_{A_j}\delta_{A_j}\end{aligned}$$

For simplicity of exposition, let $\alpha_1 = \mu_2/\sigma_1^2, \alpha_2 = -2r_A^T\xi/\sigma_1^2, \alpha_3 = (z_j\mu_2 - z_A^T\xi)/\sigma_1^2, \alpha_4 = (z_A^T\mu_1 - z_j\xi^T)r_A/\sigma_1^2, \alpha_5 = (r_A^T\mu_1 r_A + z_{A_j}^T M_{A_j} z_{A_j})/\sigma_1^2$ If $\alpha_3 > 0$, substituting these in the risk, we have,

$$\begin{aligned}L_j = \frac{1}{\sqrt{|\Sigma_{A_j}|}}\Big(\int_{r_j=-\infty}^{-\frac{\alpha_4}{\alpha_3}} e^{-(\alpha_1 r_j^2 + (\alpha_2-\alpha_3)r_j - \alpha_4 + \alpha_5)/2} dr_j \\ + \int_{r_j=-\frac{\alpha_4}{\alpha_3}}^{\infty} e^{-(\alpha_1 r_j^2 + (\alpha_2+\alpha_3)r_j + \alpha_4 + \alpha_5)/2} dr_j\Big)\end{aligned}$$

Letting $x = \sqrt{\alpha_1}r_j + \frac{\alpha_2-\alpha_3}{2\sqrt{\alpha_1}}$ and $y = \sqrt{\alpha_1}r_j + \frac{\alpha_2+\alpha_3}{2\sqrt{\alpha_1}}$, and substituting $dx = dy = \sqrt{\alpha_1}dr_j$ we can rewrite the above as,

$$\begin{aligned}L_j = \frac{1}{\sqrt{\alpha_1|\Sigma_{A_j}|}}\Big(e^{\frac{\frac{(\alpha_2-\alpha_3)^2}{4\alpha_1} - \alpha_5 + \alpha_4}{2}} h\left(\frac{\alpha_2-\alpha_3}{2\sqrt{\alpha_1}} - \frac{\alpha_4\sqrt{\alpha_1}}{\alpha_3}\right) + \\ e^{\frac{\frac{(\alpha_2+\alpha_3)^2}{4\alpha_1} - \alpha_5 - \alpha_4}{2}} h\left(-\frac{\alpha_2+\alpha_3}{2\sqrt{\alpha_1}} + \frac{\alpha_4\sqrt{\alpha_1}}{\alpha_3}\right)\Big)\end{aligned}$$

where $h(x) = \int_{-\infty}^{x} e^{-x^2/2}dx$, which can be expressed in terms of the error function (erf). A similar derivation applies if $\alpha_3 \leq 0$. □

References

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

[2] B. Bi, M. Shokouhi, M. Kosinski, and T. Graepel. Inferring the demographics of search users. *Prof. of World Wide Web (WWW)*, 2013.

[3] J. A. Calandrino, A. Kilzer, A. Narayanan, E. W. Felten, and V. Shmatikov. "You Might Also Like:" Privacy Risks of Collaborative Filtering. In *IEEE Symposium on Security and Privacy*, 2011.

[4] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein. MIT press, 2001.

[5] S. Dasgupta. Analysis of a greedy active learning strategy. *Advances in neural information processing systems*, 17:337–344, 2005.

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

[7] D. Golovin, A. Krause, and D. Ray. Near-optimal Bayesian active learning with noisy observations. *Arxiv preprint arXiv:1010.3091*, 2010.

[8] A. Goyal and L. V. S. Lakshmanan. Recmax: exploiting recommender systems for fun and profit. In *KDD*, 2012.

[9] D. A. Harville. Springer-Verlag, 1997.

[10] T. Hastie, R. Tibshirani, and J. H. Friedman. *The elements of statistical learning: data mining, inference, and prediction*. New York: Springer-Verlag, 2001.

[11] M. Hoffman, D. Blei, and P. Cook. Bayesian nonparametric matrix factorization for recorded music. In *Proc. ICML*, pages 439–446, 2010.

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

[13] Y. Koren. Factorization meets the neighborhood: a multifaceted collaborative filtering model. In *14th ACM SIGKDD international conference on Knowledge discovery and data mining (KDD)*, 2008.

[14] Y. Koren, R. Bell, and C. Volinsky. Matrix factorization techniques for recommender systems. *IEEE Computer*, 42(8):30–37, 2009.

[15] M. Kosinski, D. Stillwell, and T. Graepel. Private traits and attributes are predictable from digital records of human behavior. *Proc. of the National Academy of Sciences of USA (PNAS)*, 2013.

[16] 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.

[17] S. Nakajima and M. Sugiyama. Implicit regularization in variational bayesian matrix factorization. In *27th International Conference on Machine Learning (ICML)*, 2010.

[18] A. Narayanan and V. Shmatikov. Robust de-anonymization of large sparse datasets. In *IEEE Symposium on Security and Privacy*, 2008.

[19] A. Nosko, E. Wood, and S. Molema. All about me: Disclosure in online social networking profiles: The case of Facebook. *Computers in Human Behavior*, 26(3):406–418, May 2010.

[20] R. Nowak. The geometry of generalized binary search. *Transactions on Information Theory*, 5, 2012.

[21] 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*, pages 127–134. ACM Press, 2002.

[22] R. Salakhutdinov and A. Mnih. Bayesian probabilistic matrix factorization using markov chain monte carlo. In *Proceedings of the 25th international conference on Machine learning*, pages 880–887. ACM, 2008.

[23] R. Salakhutdinov and A. Mnih. Probabilistic matrix factorization. In *Advances in Neural Information Processing Systems*, volume 20, 2008.

[24] 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.

[25] M. N. Schmidt, O. Winther, and L. K. Hansen. Bayesian non-negative matrix factorization. In *Independent Component Analysis and Signal Separation*, pages 540–547. Springer, 2009.

[26] 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.

[27] J. Silva and L. Carin. Active learning for online bayesian matrix factorization. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 325–333. ACM, 2012.

[28] F. Stutzman, R. Gross, and A. Acquisti. Silent listeners: The evolution of privacy and disclosure on facebook. *Journal of Privacy and Confidentiality*, 4, 2012.

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