A Privacy-Protecting Architecture for Collaborative Filtering via Forgery and Suppression of Ratings

Javier Parra-Arnau, David Rebollo-Monedero, and Jordi Forné

Department of Telematics Engineering, Universitat Politècnica de Catalunya (UPC), E-08034 Barcelona, Spain
{javier.parra,david.rebollo,jforne}@entel.upc.edu*

Abstract. Recommendation systems are information-filtering systems that help users deal with information overload. Unfortunately, current recommendation systems prompt serious privacy concerns. In this work, we propose an architecture that protects user privacy in such collaborative-filtering systems, in which users are profiled on the basis of their ratings. Our approach capitalizes on the combination of two perturbative techniques, namely the forgery and the suppression of ratings. In our scenario, users rate those items they have an opinion on. However, in order to avoid privacy risks, they may want to refrain from rating some of those items, and/or rate some items that do not reflect their actual preferences. On the other hand, forgery and suppression may degrade the quality of the recommendation system. Motivated by this, we describe the implementation details of the proposed architecture and present a formulation of the optimal trade-off among privacy, forgery rate and suppression rate. Finally, we provide a numerical example that illustrates our formulation.

1 Introduction

From the advent of the Internet and the World Wide Web (WWW), the amount of information available to users has grown exponentially. Today, due to this information overload, users feel they have to separate the wheat from the chaff. Recommendation systems are a type of information-filtering systems that assist users in this task by suggesting information items they may be interested in. Among the existing recommendation systems, some of the most successful ones are based on collaborative filtering (CF) algorithms [1, 2]. Examples of CF-based systems include recommending books, music, and other products at Amazon.com [3], movies by MovieLens [4] and Netflix [5], and news at Digg [6].

* This work was supported in part by the Spanish Government through Projects CONSOLIDER INGENIO 2010 CSD2007-00004 "ARES" and TEC2010-20572-C02-02 "CONSEQUENCE", and by the Catalan Government under Grant 2009 SGR 1362.

One of the most popular forms of interaction in recommendation systems is that users communicate their preferences by rating items. This is the case of Movielens, where users assign *ratings* to movies they have already watched. Other strategies to capture users' interests include asking them to sort a number of items by order of predilection, or suggesting that they mark the items they like. On the other hand, recommendation systems may collect data from users without requiring them to explicitly convey their interests [7]. Such practices include observing the items clicked by users in an online store, analyzing the time it takes users to examine an item, or simply keeping a record of the purchased items.

The prolonged collection of these data allows the system to extract an accurate snapshot of user interests or *user profiles*. Once this information has been captured, the recommendation system applies an algorithm that returns a prediction of users' interests for those items they have not yet considered. For example, Movielens and Digg apply CF algorithms to predict the rating that a user would give to a movie and to create a personalized list of recommended news, respectively. Fig. 1 illustrates the case of Movielens and provides an example of user profile.

Despite the many advantages recommendation systems are bringing to users, the information collected, processed and stored by these systems prompts serious privacy concerns. One of the main privacy risks perceived by users is that of a computer "figuring things out" about them [8]. Namely, many users are worried about the idea that their profiles may reveal sensitive information such as health-related issues, political affiliation, salary or religion. On the other hand, other users are concerned that the system's predictions may be totally erroneous and be later used to defame them. The latter situation is illustrated in [9], where the accuracy of the predictions provided by TiVo digital video recorder and Amazon is questioned. Specifically, the author describes several real cases in which the recommender makes dubious, and in some cases aberrant, inferences about users' sexual preferences. Lastly, other privacy risks embrace unsolicited marketing, information leaked to other users of the same computer, court subpoenas, and government surveillance [8].

Therefore, it is not surprising that some users are reticent to disclose their interests. In fact, [10] reports that the 24% of Internet users surveyed provided false information in order to avoid giving private information to a Web site. In this line, another study [11] finds that 95% of the respondents refused, at some point, to provide personal information when requested by a Web site. In a nutshell, these studies seem to indicate that submitting

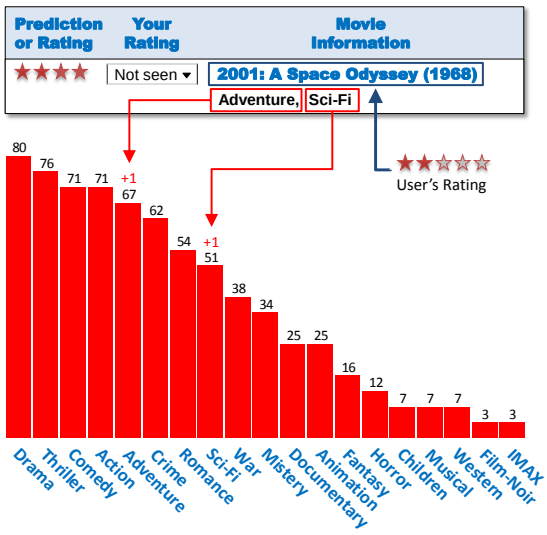


Fig. 1. The profile of a user is modeled in Movielens as a histogram of absolute frequencies of ratings within a set of movie genres (bottom). Based on this profile, the recommender predicts the rating that the user would probably give to a movie (top). After having watched the movie, the user rates it and their profile is updated.

false information and refusing to give private information are strategies accepted by users concerned with their privacy.

1.1 Contribution and Plan of this Paper

In this work, we tackle the problem of protecting user profiles in recommendation systems based on CF algorithms. Specifically, we propose an architecture aimed at preserving user privacy in those systems in which users are profiled on the basis of their ratings. Our approach relies upon the combination of two conceptually simple mechanisms: forgery and suppression of ratings. More accurately, in our scenario, users rate those items they have an opinion on. However, in order to prevent privacy risks, they may wish to refrain from rating some of those items, and/or rate items that are unknown to them and do not reflect their actual preferences. In order to hinder privacy attackers in their efforts to profile users' interests, the proposed architecture helps users decide which items should be rated and which should not. Consequently, this approach guarantees user privacy to a certain extent, at the cost of some processing overhead and a degradation in the accuracy of the prediction, but without having to trust the recommendation system or the network operator.

In addition, we present an information-theoretic, mathematical formulation of the trade-off among privacy, forgery rate and suppression rate. More specifically, in this formulation we measure privacy as the entropy of the user's apparent profile, which is the profile observed by the system, after the forgery and suppression of ratings. Our formulation results in a convex optimization problem for which there exist efficient numerical methods to solve it. Finally, we would like to add that our approach could benefit from the combination with other alternatives in the literature.

Sec. 2 reviews some relevant approaches aimed at preserving user privacy in CF-based recommendation systems. Sec. 3 describes a privacy-protecting architecture based on the forgery and suppression of ratings. In addition, this section presents the model of user profile assumed, the adversarial model and our privacy measure. Sec. 4 introduces a formulation of the trade-off among privacy, forgery rate and suppression rate. Sec. 5 shows a simple but insightful example that illustrates this formulation. Conclusions are drawn in Sec. 6.

2 State of the Art

Numerous approaches have been proposed to protect user privacy in the context of recommendation systems using CF techniques. These approaches basically suggest three main strategies: perturbing the information provided by users, using cryptographic techniques, and distributing the information stored by recommenders.

In the case of perturbative methods for recommendation systems, [12] proposes that users add random values to their ratings and then submit these perturbed ratings to the recommender. After receiving these ratings, the system executes an algorithm and sends the users some information that allows them to compute the prediction. When the number of participating users is sufficiently large, the authors find that user privacy is protected to a certain extent and the system reaches a decent level of accuracy. However, even though a user disguises all their ratings, it is evident that the items themselves may uncover sensitive information. In other words, the simple fact of showing interest in a certain item may be more revealing than the ratings assigned to that item. For instance, a user rating a book called "How to Overcome Depression" indicates a clear interest in depression, regardless of the score assigned to this book. Apart from this critique, other works [13, 14] stress that the use of *randomized* data distortion techniques might not able to preserve privacy.

In line with this work, [15] applies the same perturbative technique to CF algorithms based on singular-value decomposition (SVD). More specifically, the authors focus on the impact that their technique has on privacy. For this purpose, they use the privacy metric proposed by [16], which is essentially equivalent to *differential entropy*, and conduct some experiments with data sets from Movielens and Jester [17]. The results show the trade-off curve between accuracy in recommendations and privacy. In particular, they measure accuracy as the mean absolute error between the predicted values from the original ratings and the predictions obtained from the perturbed ratings.

At this point, we would like to remark that the use of perturbative techniques is by no means new in other application scenarios such as private information retrieval (PIR). In this scenario, users send general-purpose queries to an information service provider. An example would be a user sending the query:"What was George Orwell's real name?". A perturbative approach to protect user profiles in this context consists in combining genuine with false queries. In this sense, [18] proposes a *non-randomized* method for query forgery and investigates the trade-off between privacy and the additional traffic overhead. Naturally, the perturbation of user profiles for privacy protection may be carried out not only by means of insertion of bogus activity, but also by suppression. This is exactly the alternative proposed in [19], aimed at protecting user privacy in the scenario of the semantic Web. More accurately, this approach recommends that users refrain from tagging some resources when their privacy is being compromised.

Regarding the use of cryptographic techniques, [20, 21] propose a method that enables a community of users to calculate a public aggregate of their profiles without revealing them on an individual basis. In particular, the authors use a homomorphic encryption scheme and a peer-to-peer (P2P) communication protocol for the recommender to perform this calculation. Once the aggregated profile is computed, the system sends it to users, who finally use local computation to obtain personalized recommendations. This proposal prevents the system or any external attacker from ascertaining the individual user profiles. However, its main handicap is assuming that an acceptable number of users is online and willing to participate in the protocol. In line with this, [22] uses a variant of Pailliers' homomorphic cryptosystem which improves the efficiency in the communication protocol. Another solution [23] presents an algorithm aimed at providing more efficiency by using the scalar product protocol.

In order to mitigate the potential privacy risks derived from the fact that users' private information is kept in a single repository, some approaches suggest that this information be stored in a distributed way. This is the case of [24], which presents a CF algorithm called PocketLens, specifically designed to be deployed to a P2P scenario. The algorithm in question enables users to decide which private information should exchange with other users of the P2P community. In addition, the authors provide several architectures for the problem of locating neighbors. Another alternative assumes a pure decentralized P2P scenario and proposes the use of several perturbative strategies [25]. In essence, this scheme could be regarded as a combination of the approaches in [24] and [12]. Namely, the mentioned scheme recommends replacing the actual ratings by fixed, predefined values, by uniformly distributed random values, and by a bell-curve distribution imitating the distribution of the population's ratings.

3 An Architecture for Privacy Protection in CF-based Recommendation Systems

In this section, we present the main contribution of this work: an architecture for the protection of user profiles in recommendation systems relying on CF algorithms. Specifically, we consider the case in which users' preferences are exclusively derived from the ratings they assign to items. Bearing this in mind, we shall hereafter refer to the user's *known items* as those items they have an opinion on. In the case of Movielens, for example, the known items of a particular user would be those movies the user has already watched. Analogously, we shall refer to the user's *unknown items* as those items the user is not in the position to rate. For instance, this could be the case of a movie the user is not aware of or a movie the user has heard about, but has not watched yet.

Our approach is based on the combined use of two perturbative techniques, namely the submission of ratings of unknown items and the suppression of ratings of known items. For the sake of brevity, we shall occasionally refer to these techniques simply as the forgery and suppression of ratings, respectively. According to these mechanisms, in our scenario users rate known items. However, in order to avoid privacy risks, they may want to refrain from rating some of those known items, and/or rate some unknown items. Having said this, we would like to mention that the fact that forgery only applies to unknown items is basically because users may be reluctant to assign false ratings to known items. Despite the above, our approach could also give the user the option to forge ratings of known

items. However, for brevity, we only describe the case where forgery applies just to unknown items. Lastly, we would like to say that our approach could be integrated with other systems, like for example, with some of the approaches mentioned in Sec. 2, and those using pseudonyms [26,27].

In the rest of this section, we provide further insight into our proposal. Concretely, we propose a mathematical model of user profiles in Sec. 3.1. Afterwards, Sec. 3.2 examines the assumed adversarial model. Next, our privacy criterion is presented and justified in Sec. 3.3. Lastly, we delve into our architecture and analyze each of its internal components in Sec. 3.4.

3.1 User Profile

We pointed out in Sec. 1 that Movielens uses histograms of absolute frequencies to show user profiles. Other recommendation systems represent this information by means of a tag cloud, which may be regarded as another kind of histogram.

According to these examples, and as used in [18,19,28], we propose a tractable model of user profile as a probability mass function (PMF), that is, a histogram of relative frequencies of ratings within a predefined set of categories of interest. We would like to remark that, under this model, user profiles do not capture the particular scores given to items, but what we consider to be more sensitive: the categories these items belong to. This corresponds to the case of Movielens, which we illustrate in Fig. 1. In this example, a user assigns two stars to a movie, meaning that they consider it to be "fairly bad". However, the recommender updates their profile based only on the categories this movie belongs to.

Having assumed the above model, now we focus on how to estimate the profile of a user from their ratings. The reason is that our approach requires this information to help users decide which items should be rated and which should not. Clearly, the easiest way to obtain a user profile is by asking the recommender. Movielens users, for instance, can do that. Unfortunately, in most recommendation systems users do not have access to this information. In order to cope with this, we suggest an alternative for extracting users' preferences from their rating activity.

We consider two possible cases for the information that a system shows about its items. The trivial case is when the recommender provides users with a categorization of all of its items. In this situation, it is straightforward to keep a histogram based on these categories. This is the case of Netflix or Movielens, where the genres of all movies are available to users. On the contrary, it may happen that this categorization is not at the disposal of users. This applies to Digg, where the only information

that the recommender provides about news is the headline, the first lines of the news and the source of information. In systems like this, the categorization of items may be accomplished by exploring web pages with information about those items. Specifically, this process could be carried out by using the vector space model [29], as normally done in information retrieval, to represent these web pages as tuples containing their most representative terms. Namely, the term frequency-inverse document frequency (TF-IDF) could be applied to calculate the weights of each term appearing in a web page. Next, the most weighted terms of each web page could be combined in order to create a category and assign it to the item. After obtaining the categories associated with all the items rated by a user, their profile would be computed as a histogram across these categories.

3.2 Adversarial Model

In our scenario, we suppose users interact with recommendation systems that infer their preferences based only on their ratings. This supposition is reinforced by the tractability of the model considered and also by the fact that implicit mechanisms are often less accurate than explicit ratings [30].

Under this assumption, we consider an adversarial model in which users submitting their ratings are observed by a passive attacker who is able to ascertain which ratings are associated with which items. Concretely, this could be the case of the recommendation system itself or, in general, any privacy attacker able to crawl through this information.

Bearing in mind the model of user profile assumed in Sec. 3.1, after the rating of a sufficiently large number of items, the attacker can compute a histogram with the actual interests of a particular user. However, when this user adheres to the forgery and suppression of ratings, the attacker observes a perturbed version of this histogram, which makes it more difficult for the attacker to discover the user's actual preferences. We shall refer to this perturbed profile as the user's *apparent* profile. Last but not least, we suppose that the attacker is unaware of or ignores the fact that the user is adopting our strategy, thereby assuming that the apparent profile reflects genuine interests.

3.3 Privacy Metric

Any optimized mechanism aimed at protecting the privacy of users necessarily requires to evaluate the extent to which it is effective. In this work, just as in [18, 19], we use an information-theoretic quantity to emphasize

that an attacker will have gained some information about a user whenever their preferences are biased towards certain categories of interest.

Specifically, we measure privacy as the Shannon entropy [31] of the user's apparent distribution. Recall that the entropy is formulated in the following terms. Consider a random variable (r.v.) distributed according to a PMF t and taking on values in the alphabet $\{1, \ldots, n\}$. The entropy of this probability distribution is defined as $\mathrm{H}(t) = \sum_{i=1}^{n} t_i \log_2 t_i$, which may be interpreted as a measure of the uncertainty of the outcome of that random variable, and also regarded as a special case of Kullback-Leibler (KL) divergence [32]. An interesting property of the entropy is that it is maximized, among all distributions on that alphabet, by the uniform distribution $u_i = 1/n$ for all i. This allows us to capture the intuitive observation that an attacker will have compromised user privacy as long as the user's apparent profile diverges from the uniform profile.

Having defined our measure of privacy, later in Sec. 4 we formulate the optimization problem given by the maximization of the entropy of the user's apparent distribution for a given forgery rate and a suppression rate. Precisely, our privacy criterion is justified by the rationale behind entropy maximization methods [33,34]. Namely, some of the arguments in favor of these methods are related to the highest number of permutations with repeated elements associated with an empirical distribution [33], or more generally, the method of types and large deviation theory [32, §11].

In addition, we would like to stress that, although our privacy criterion is based on a fundamental quantity in information theory, the convergence of these two fields is by no means new. In fact, Shannon's work in the fifties introduced the concept of *equivocation* as the conditional entropy of a private message given an observed cryptogram [35], later used in the formulation of the problem of the wiretap channel [36,37] as a measure of confidentiality. In addition, recent studies [38,39] reassert the suitability and applicability of the concept of entropy as a measure of privacy.

3.4 Architecture

In this section, we describe an architecture that helps users decide which unknown items should be rated and which known items should not in order to hinder privacy attackers in their efforts to profile users' interests. Our architecture is conceived to be implemented by a software application running on the user's local machine. Fig. 2 shows the proposed architecture, which consists of a number of modules, each of them performing a specific task. Next, we provide a functional description of all of its modules and examine the details of a practical implementation.

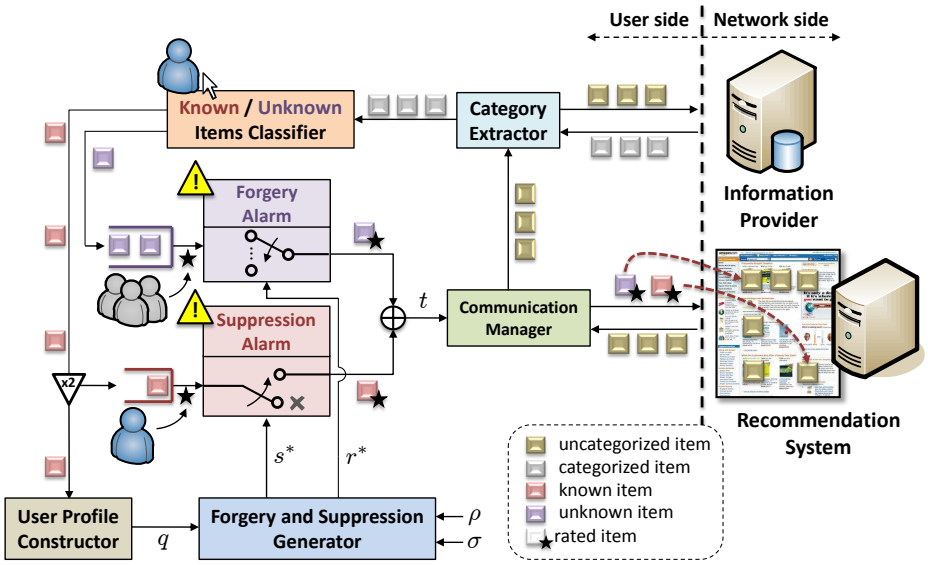


Fig. 2. Block diagram of the proposed architecture.

Communication Manager. This module is in charge of interacting with the recommendation system. Specifically, it downloads information about the items the user finds when browsing the recommender's web site. This information may include a description about the items, the ratings that other users assigned to them, and the categories of interest these items belong to. In Amazon, for instance, all this information is available to users. However, as commented on in Sec. 3.1, this is not always the case. For this reason, our approach incorporates modules intended to retrieve the population's ratings and categorize all the items that the user explores.

On the other hand, this module receives the ratings of unknown items suggested by the *forgery alarm generator* and the ratings of known items sent by the *suppression alarm generator*. Afterwards, the module submits these ratings to the recommendation system.

Category Extractor. This component is responsible for obtaining the categories the items belong to. To this end, the module uses the information provided by the communication manager. Should this information not be enough, the module will have to get additional data by searching the Web or by querying an information provider. Afterwards, the categorization of these items is carried out by using the vector space model and the TF-IDF weights as commented on in Sec. 3.1. In a last stage,

this module sends the items and their corresponding categories to the *known/unknown items classifier*.

Known/Unknown Items Classifier. This module requires the active involvement of the user. Namely, it shows the user the items categorized by the category extractor module, and then asks the user to classify them as known or unknown. Evidently, this module will have previously checked whether these items have already been rated by the user. Should this be the case, the rated items would not be shown to the user, since these items would be classified as known items. For this purpose, the module keeps a record of all the items that the user rates. Once these items have been classified as known or unknown, they are sent to the *forgery alarm generator* and the *suppression alarm generator*, respectively. In addition, the known items are submitted to the *user profile constructor*.

User Profile Constructor. This module is responsible for the estimation of the user profile. To this end, the module is provided with the user's known items, i.e., those items capturing their preferences. Based on these items, it generates the user profile as described in Sec. 3.1. Obviously, during this process, the module discards those rated items that were already considered in the histogram computation.

Forgery and Suppression Generator. This block is the centerpiece of the architecture as it is directly responsible for the user privacy. First, the block is provided with the user profile. In addition, the user specifies a forgery rate ρ and a suppression rate σ. The former is the fraction of ratings of unknown items that the user is willing to submit. The latter is the relative frequency of ratings of known items that the user is disposed to eliminate. Having specified these two rates, the module computes the optimum tuples of forgery r^* and suppression s^*, which contain information about the ratings that should be forged and suppressed, respectively. More accurately, the component r_i is the percentage of ratings of unknown items that our architecture suggests submitting in the category i. The component s_i is defined analogously for suppression. In the end, the tuples r^* and s^* are sent to the forgery alarm generator and the suppression alarm generator, respectively. Later in Sec. 4, we provide a more detailed specification of this module by using a formulation of the trade-off among privacy, forgery rate and suppression rate.

Suppression Alarm Generator. This module is responsible for warning the user when their privacy is being compromised. Concretely, this module receives the tuple s^* and stores the known items provided by the known/unknown items classifier. These items are stored in an array. When the user decides to assign a rating to one of these items, the se-

lected item is removed from the array. The user then rates this item, and the module proceeds as follows: if s^* has a positive component in at least one of the categories the item belongs to, a privacy alarm is generated to alert the user, and it is then for the user to decide whether to eliminate the rating or not. However, if s^* is zero for all components, our architecture does not become aware of any privacy risk and the rating is sent to the communication manager module. This process is repeated provided that the array is not empty.

Forgery Alarm Generator. Our approach also relies on the forgery of ratings. Precisely, this module selects, on the one hand, which unknown items should be rated, and on the other hand, which particular ratings should be assigned to these unknown items. With regard to the ratings to be given to the items, we follow a method similar to the one pointed out in [20]. Namely, our approach assigns each unknown item a random rating, drawn according to the distribution of the other users' ratings to that item. Alternatively, we could also contemplate the distribution of ratings of a user with similar preferences, and the distribution of all ratings. In order to obtain this information, the module will have to query information providers or explore other recommenders. In the case of Amazon, for example, this is not necessary since users are provided with the population's ratings.

In parallel, the module receives unknown items and stores them in an array. After getting the tuple r^*, the module proceeds as follows: if r^* has positive components in one or several categories, a privacy alarm is triggered. Our architecture encourages then the user to submit a random rating to one of the unknown items in the array which belong to these categories. However, it is the user who finally decides whether to send this rating or not. If the user accepts the recommendation, then the rating is sent to the communication manager module, and the unknown item is removed from the array. This whole process is repeated provided that r^* has at least one positive component.

After having explored each of the modules of the architecture, next we shall describe how our approach would work. Initially, the user would browse the recommendation system's web site and would find some items. In order for the user to obtain future recommendations from the system, they would have to rate some of those items. Before proceeding, though, our approach would retrieve information about the items and extract the categories they belong to. Afterwards, the user would be asked to classify the items as known or unknown. The known items would allow the proposed architecture to build the user profile. After computing the

tuples r^* and s^*, our approach could suggest submitting a random rating to one of the unknown items. Should this be the case, the user would have to decide whether to send the rating or not. Next, the user would start rating the known items. At a certain point, the user could receive a privacy alarm when trying to rate one of these items. Should this be the case, it would be up to the user to decide whether to eliminate the rating or not.

4 Formulation of the Trade-Off among Privacy, Forgery Rate and Suppression Rate

In this section, we present a formulation of the trade-off among privacy, forgery rate and suppression rate. As we shall show later, this formulation will allow us to go into the details of one of the functional blocks of the proposed architecture.

Next, we formalize some of the concepts that we introduced in previous sections. Specifically, we model the *items* in a recommendation system as r.v.'s taking on values in a common finite alphabet of categories, namely the set $\{1, \ldots, n\}$ for some $n \in \mathbb{Z}^+$. Accordingly, we define q as the probability distribution of the known items of a particular *user*, that is, the distribution capturing the actual preferences of the user. In line with Sec. 3.4, we introduce a *rating forgery* rate $\rho \in [0, 1)$, which is the ratio of forged items. Analogously, we define a *rating suppression* rate $\sigma \in [0, 1)$, modeling the proportion of items that the user consents to eliminate. Bearing this in mind, we define the user's *apparent* item distribution t as $\frac{q+r-s}{1+\rho-\sigma}$ for some forgery strategy $r = (r_1, \ldots, r_n)$ and some suppression strategy $s = (s_1, \ldots, s_n)$, satisfying, on the one hand, $r_i \geqslant 0$ and $\sum r_i = \rho$ for $i = 1, \ldots, n$, and on the other hand, $q_i \geqslant s_i \geqslant 0$ and $\sum s_i = \sigma$ for $i = 1, \ldots, n$. In light of this definition, the user's apparent item distribution may be interpreted as the result of the suppression of some genuine ratings from the actual user profile and the posterior addition of some forged ratings. Afterwards, this is normalized by $\frac{1}{1+\rho-\sigma}$ so that $\sum_i t_i = 1$.

According to the justification provided in Sec. 3.3, we use Shannon's entropy to quantify user privacy. More precisely, we measure privacy as the entropy of the user's apparent item distribution. Consistently with this measure, we define the *privacy-forgery-suppression* function

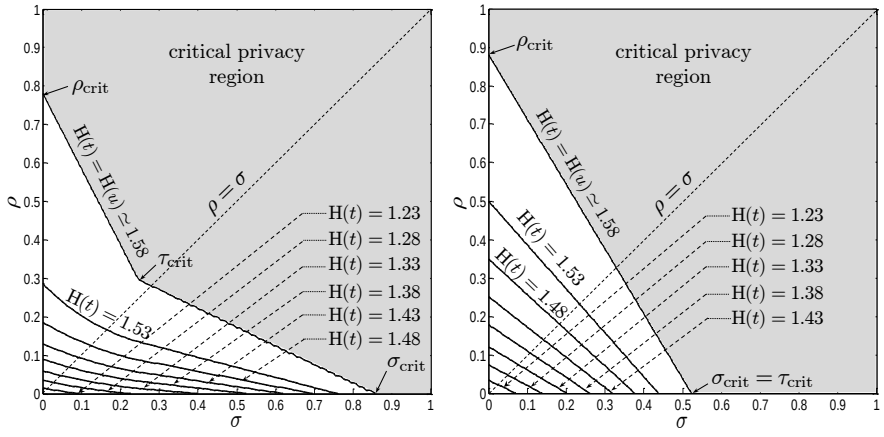
$$\mathcal{P}(\rho, \sigma) = \max_{\substack{r,s \\ r_i \geqslant 0, \sum r_i = \rho \\ q_i \geqslant s_i \geqslant 0, \sum s_i = \sigma}} \mathrm{H}\left(\frac{q + r - s}{1 + \rho - \sigma}\right), \tag{1}$$

which characterizes the optimal trade-off among privacy, forgery rate and suppression rate, and enables us to specify the module *forgery and suppression generator* in Sec. 3.4. More accurately, this functional block will be in charge of solving the optimization problem in (1). Last but not least, we would like to remark that this optimization problem is convex [40], which in practice means that there are powerful and efficient numerical methods to solve it.

5 Numerical Example

This section provides a simple yet insightful numerical example that illustrates the formulation in Sec. 4 and sheds some light into the benefits from combining the forgery and suppression of ratings.

In this example, we assume two user's profiles $q_1 = (0.05, 0.35, 0.60)$ and $q_2 = (0.15, 0.15, 0.70)$, across three categories of interest. We define the user's *critical privacy* region as the set $\{(\rho, \sigma) : \mathcal{P}(\rho, \sigma) = \mathrm{H}(u)\}$, and the *critical forgery* rate ρ_{crit} as $\min\{\rho : \mathcal{P}(\rho, 0) = \mathrm{H}(u)\}$. Analogously, we define the *critical suppression* rate σ_{crit}. Fig. 3 shows the critical privacy region and the critical rates for these two users. In addition, we depicted several contour lines of function (1).



(a) $q_1 = (0.05, 0.35, 0.60)$ (b) $q_2 = (0.15, 0.15, 0.70)$

Fig. 3. We represent the critical privacy region and several contour lines of the privacy-forgery-suppression function, for two users with actual profiles q_1 and q_2.

One important observation that emerges from these figures is that the combined use of forgery and suppression may be more effective than

the sole application of one of these techniques. To illustrate this, consider the *cost* $\tau = \rho + \sigma$, where the impact of forgery and suppression is balanced. According to this, we define the *critical cost* τ_{crit} as $\min\{\tau : \mathcal{P}(\rho, \sigma) = \mathrm{H}(u), \tau = \rho + \sigma\}$. Now we contemplate two possible strategies for the user: the *mixed* strategy, where forgery and suppression are used in conjunction, and the *pure* strategy, consisting in the application of one of these two mechanisms. In Fig. 3(a), we can appreciate a significant difference between these two strategies. Namely, when the user is not willing to eliminate any of their ratings, $\tau_{\text{crit}}|_{\sigma=0} = \rho_{\text{crit}} \simeq 0.78$. Similarly, when forgery is not applied, $\tau_{\text{crit}}|_{\rho=0} = \sigma_{\text{crit}} \simeq 0.85$. However, when the user adopts the mixed strategy, it turns out that $\tau_{\text{crit}} \simeq 0.55$. Unfortunately, this is not always the case. For example, in Fig. 3(b) we find that $\tau_{\text{crit}} = \sigma_{\text{crit}}$, i.e., the pure strategy leads to the minimum cost. In a nutshell, the combination of forgery and suppression may result in a synergy that can help users protect their privacy more efficiently.

6 Concluding Remarks

There exist numerous proposals for the protection of user privacy in CF-based recommendation systems. Within these approaches, the forgery and suppression of ratings arise as two simple mechanisms in terms of infrastructure requirements, as users need not trust the recommender. However, the application of these mechanisms comes at the cost of some processing overhead and, more importantly, at the expense of a degradation in the accuracy of the recommendations.

Our main contribution is an architecture that implements the forgery and suppression of ratings in those recommendation systems that profile users exclusively from their ratings. We describe the functionality of the internal modules of this architecture. The centerpiece of our approach is a module responsible for computing a pair of tuples containing information about which ratings should be forged and which ones should be eliminated. Our architecture uses then this information to warn the user when their privacy is being compromised. The user is who finally decides whether to follow the recommendations made by our approach or not.

We present a mathematical formulation of the optimal trade-off among privacy, forgery rate and suppression rate, which arises from the definition of our privacy criterion. This formulation allows us to specify the module responsible for user privacy. Lastly, we illustrate the formulation with a simple albeit insightful numerical example.

References

1. D. Goldberg, D. Nichols, B. M. Oki, and D. Terry, "Using collaborative filtering to weave an information tapestry," *Commun. ACM*, vol. 35, no. 12, pp. 61–70, Dec. 1992.
2. X. Su and T. M. Khoshgoftaar, "A survey of collaborative filtering techniques," *Adv. Artif. Intell.*, vol. 2009, Jan. 2009.
3. "Amazon.com." [Online]. Available: http://www.amazon.com
4. "Movielens." [Online]. Available: http://movielens.umn.edu
5. "Netflix." [Online]. Available: http://www.netflix.com
6. "Digg." [Online]. Available: http://digg.com
7. D. Oard and J. Kim, "Implicit feedback for recommender systems," in *Proc. AAAI Workshop Recommender Syst.*, 1998, pp. 81–83.
8. L. F. Cranor, ""I didn't buy it for myself". Privacy and e-commerce personalization," in *Proc. ACM Workshop on Privacy in the Electron. Society*, Washington, DC, 2003, pp. 111–117.
9. J. Zaslow, "If TiVo thinks you are gay, here's how to set it straight," Nov. 2002. [Online]. Available: http://online.wsj.com/article_email/SB1038261936872356908.html
10. S. Fox, "Trust and privacy online: Why americans want to rewrite the rules," Pew Internet and Amer. Life Project, Res. Rep., Aug. 2000.
11. D. L. Hoffman, T. P. Novak, and M. Peralta, "Building consumer trust online," *Commun. ACM*, vol. 42, no. 4, pp. 80–85, Apr. 1999.
12. H. Polat and W. Du, "Privacy-preserving collaborative filtering using randomized perturbation techniques," in *Proc. SIAM Int. Conf. Data Min. (SDM)*. IEEE Comput. Soc., 2003.
13. H. Kargupta, S. Datta, Q. Wang, and K. Sivakumar, "On the privacy preserving properties of random data perturbation techniques," in *Proc. IEEE Int. Conf. Data Min. (ICDM)*. Washington, DC: IEEE Comput. Soc., 2003, pp. 99–106.
14. Z. Huang, W. Du, and B. Chen, "Deriving private information from randomized data," in *Proc. ACM SIGMOD Int. Conf. Manage. Data*. ACM, 2005, pp. 37–48.
15. H. Polat and W. Du, "SVD-based collaborative filtering with privacy," in *Proc. ACM Int. Symp. Appl. Comput. (SASC)*. ACM, 2005, pp. 791–795.
16. D. Agrawal and C. C. Aggarwal, "On the design and quantification of privacy preserving data mining algorithms," in *Proc. ACM SIGMOD Int. Conf. Manage. Data*, Santa Barbara, CA, 2001, pp. 247–255.
17. "Jester: The online joke recommender." [Online]. Available: http://eigentaste.berkeley.edu/
18. D. Rebollo-Monedero and J. Forné, "Optimal query forgery for private information retrieval," *IEEE Trans. Inform. Theory*, vol. 56, no. 9, pp. 4631–4642, 2010.
19. J. Parra-Arnau, D. Rebollo-Monedero, and J. Forné, "A privacy-preserving architecture for the semantic web based on tag suppression," in *Proc. Int. Conf. Trust, Privacy, Security, Digit. Bus. (TRUSTBUS)*, Bilbao, Spain, Aug. 2010.
20. J. Canny, "Collaborative filtering with privacy via factor analysis," in *Proc. ACM SIGIR Conf. Res., Develop. Inform. Retrieval*. Tampere, Finland: ACM, 2002, pp. 238–245.
21. J. F. Canny, "Collaborative filtering with privacy," in *Proc. IEEE Symp. Security, Privacy (SP)*, 2002, pp. 45–57.
22. W. Ahmad and A. Khokhar, "An architecture for privacy preserving collaborative filtering on web portals," in *Proc. IEEE Int. Symp. Inform. Assurance, Security (IAS)*. Washington, DC: IEEE Comput. Soc., 2007, pp. 273–278.

23. J. Zhan, C. L. Hsieh, I. C. Wang, T. S. Hsu, C. J. Liau, and D. W. Wang, "Privacy-preserving collaborative recommender systems," *IEEE Trans. Syst. Man, Cybern.*, vol. 40, no. 4, pp. 472–476, Jul. 2010.
24. B. Miller, N. Bradley, and J. A. K. J. Riedl, "Pocketlens: Toward a personal recommender system," *ACM Trans. Inform. Syst.*, vol. 22, no. 3, pp. 437–476, Jul. 2004.
25. S. Berkovsky, Y. Eytani, T. Kuflik, and F. Ricci, "Enhancing privacy and preserving accuracy of a distributed collaborative filtering," in *Proc. ACM Conf. Recommender Syst. (RecSys)*. ACM, 2007, pp. 9–16.
26. G. Bianchi, M. Bonola, V. Falletta, F. S. Proto, and S. Teofili, "The SPARTA pseudonym and authorization system," *Sci. Comput. Program.*, vol. 74, no. 1–2, pp. 23–33, 2008.
27. V. Benjumea, J. López, and J. M. T. Linero, "Specification of a framework for the anonymous use of privileges," *Telemat., Informat.*, vol. 23, no. 3, pp. 179–195, Aug. 2006.
28. J. Domingo-Ferrer, "Coprivacy: Towards a theory of sustainable privacy," in *Privacy Stat. Databases (PSD)*, ser. Lecture Notes Comput. Sci. (LNCS), vol. 6344. Corfu, Greece: Springer-Verlag, Sep. 2010, pp. 258–268.
29. G. Salton, A. Wong, and C. S. Yang, "A vector space model for automatic indexing," *Commun. ACM*, vol. 18, no. 11, pp. 613–620, 1975.
30. G. Adomavicius and A. Tuzhilin, "Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions," *IEEE Trans. Knowl. Data Eng.*, vol. 17, no. 6, pp. 734–749, 2005.
31. C. E. Shannon, "A mathematical theory of communication," Bell Syst., Tech. J. 27, 1948.
32. T. M. Cover and J. A. Thomas, *Elements of Information Theory*, 2nd ed. New York: Wiley, 2006.
33. E. T. Jaynes, "On the rationale of maximum-entropy methods," *Proc. IEEE*, vol. 70, no. 9, pp. 939–952, Sep. 1982.
34. ——, "Information theory and statistical mechanics II," *Phys. Review Ser. II*, vol. 108, no. 2, pp. 171–190, 1957.
35. C. E. Shannon, "Communication theory of secrecy systems," Bell Syst., Tech. J., 1949.
36. A. Wyner, "The wiretap channel," Bell Syst., Tech. J. 54, 1975.
37. I. Csiszár and J. Körner, "Broadcast channels with confidential messages," *IEEE Trans. Inform. Theory*, vol. 24, pp. 339–348, May 1978.
38. C. Díaz, S. Seys, J. Claessens, and B. Preneel, "Towards measuring anonymity," in *Proc. Workshop Privacy Enhanc. Technol. (PET)*, ser. Lecture Notes Comput. Sci. (LNCS), vol. 2482. Springer-Verlag, Apr. 2002.
39. C. Díaz, "Anonymity and privacy in electronic services," Ph.D. dissertation, Katholieke Univ. Leuven, Dec. 2005.
40. S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge, UK: Cambridge University Press, 2004.