The Utility and Privacy Effects of a Click

Rachid Guerraoui
EPFL
rachid.guerraoui@epfl.ch

Anne-Marie Kermarrec
Inria
Anne-Marie.kermarrec@inria.fr

Mahsa Taziki
EPFL
mahsa.taziki@epfl.ch

ABSTRACT

Recommenders are becoming one of the main ways to navigate the Internet. They recommend appropriate items to users based on their *clicks*, i.e., *likes*, *ratings*, *purchases*, etc. These clicks are key to providing relevant recommendations and, in this sense, have a significant *utility*. Since clicks reflect the preferences of users, they also raise *privacy* concerns. At first glance, there seems to be an inherent trade-off between the utility and privacy effects of a click. Nevertheless, a closer look reveals that the situation is more subtle: some clicks do improve utility without compromising privacy, whereas others decrease utility while hampering privacy.

In this paper, for the first time, we propose a way to quantify the exact utility and privacy effects of each user click. More specifically, we show how to compute the privacy effect (*disclosure risk*) of a click using an *information-theoretic* approach, as well as its *utility*, using a *commonality-based* approach. We determine precisely when utility and privacy are antagonist and when they are not. To illustrate our metrics, we apply them to recommendation traces from Movielens and Jester datasets. We show, for instance, that, considering the Movielens dataset, 5.94% of the clicks improve the recommender utility without loss of privacy, whereas 16.43% of the clicks induce a high privacy risk without any utility gain.

An appealing application of our metrics is what we call a *click-advisor*, a visual user-aware clicking platform that helps users decide whether it is actually worth clicking on an item or not (after evaluating its potential utility and privacy effects using our techniques). Using a *game-theoretic* approach, we evaluate several user clicking strategies. We highlight in particular what we define as a *smart* strategy, leading to a Nash equilibrium, where every user reaches the maximum possible privacy while preserving the average overall recommender utility for all users (with respect to the case where user clicks are based solely on their genuine preferences, i.e., without consulting the click-advisor).

1 INTRODUCTION

The growth of data available online makes it difficult for individuals to extract information relevant to their interests. *Recommenders* do the job for them: they build profiles [35] representing user interests, and recommend items to users based on those profiles [23], typically using *collaborative filtering* (CF) schemes [38].

The profiles of users are derived from their *clicks*, e.g., their ratings in terms of *likes* (or *dislikes*) [35], their purchases, the pages they spend time on, etc. On the one hand, the clicks have an important effect on the recommender *utility* for a user. On the other hand, clicks may also *disclose private information* about users [34].[1] At first glance, a user might face a dilemma: *To click or not to click?* The click could improve the utility of a recommender for that specific user, yet might also disclose private information. But is there really always a trade-off between utility and privacy?

Consider as a first illustration, the case of Alice, a user who clicks on "Game of thrones". By doing so, Alice improves utility by helping the recommender find similar users to her. Assuming indeed that a large number of users have watched "Game of thrones", Alice's click makes her also less distinguishable among those users (than before she clicked). This click improves her privacy. There is no trade-off in this case between utility and privacy. If Alice instead had clicked on an esoteric movie, she would have revealed a lot. A curious user (an attacker), knowing through the item profiles that only one user liked that esoteric movie, could eventually deduce the entire profile of Alice. (The curious user could, through a KNN attack, create fake profiles containing the esoteric movie and would be recommended the entire profile of Alice.). The motivation of this work is to determine exactly when a click induces a trade-off and when it does not, by precisely quantifying the effects of every click on both utility and privacy.

In this paper, we compute the effect of a click on utility by introducing the notion of *commonality* of a user profile, i.e., representing through a number how close the taste of a user is to that of other users (which helps a recommender suggest relevant items that are likely to match the user's preferences). This notion captures *precision*, the classical well-known measure of the quality of recommenders [32]. Whereas the idea of precision has been so far considered as an empirical measure of the utility of a recommender for all the users, we compute commonality, theoretically, and for every individual user (in a user-centric manner).[2] The difference between the commonality of a user profile, before and after the click, is what we define as the *utility* of the click.

We compute the privacy effect of a click through the concept of *disclosure degree* of a user profile, using an *information-theoretic* approach. The disclosure degree corresponds to the amount of information stored in a user profile, also known as *entropy* [15]. Roughly speaking, the larger the amount of information in a user profile, the higher the disclosure degree of the user profile. If the disclosure degree of a user profile is low, then the user is not easily distinguishable from others. We capture the *disclosure risk* of a click, and hence its privacy effect, as the difference between the

SIGIR '17, August 07-11, 2017, Shinjuku, Tokyo, Japan

978-1-4503-5022-8/17/08…$15.00
DOI: http://dx.doi.org/10.1145/3077136.3080783

[1] This is without even considering the recommender itself as a threat, but only other curious users who could deduce other profiles through what is recommended to them.
[2] It is important to note at this point that two user profiles might be very different, and yet might have the same commonality, basically meaning that they could be recommended the same *number* of relevant items but not necessarily the same items.

	i_1	i_2	i_3	i_4
u_1	✓	𝔸	✓	
u_2			✓	✓
u_3		𝒳	**B**	✓
u_4	✓		ℂ	𝒳

Table 1: Clicks of Users

disclosure degree of a user profile before and after the click. Interestingly, we prove that after user u's click on item i, the disclosure degree of u is never increased by the clicks of other users on i.[3]

To illustrate our notions of utility and privacy effects of the clicks beyond Alice's example above, consider the example depicted in Table 1, involving 4 items and 4 users. (We will use this example throughout the paper.) A bold style (resp. an outlined style) depicts a like (resp. dislike) click on an item in the corresponding (user) row and (item) column, respectively. The check-marks and cross-marks represent the clicks already performed by the users and the capital letters represent the following three clicks that could be performed by the users:

𝔸 denotes a click by u_1 to dislike i_2. 𝔸 improves the recommender utility for u_1 as 𝔸 indicates that the preference of u_1 is close to that of u_3. Hence, the recommender can suggest relevant items to u_1 after 𝔸 (e.g., i_4). However, 𝔸 compromises the privacy for u_1 in the sense that u_1 becomes more distinguishable among other users after 𝔸. Indeed, the group of users who dislike i_2 after 𝔸 (u_1 and u_3) is smaller than the group of users who did not click on i_2 before 𝔸 (u_1, u_2 and u_4); 𝔸 induces a utility-privacy trade-off.

B denotes a click to like i_3 by u_3. **B** improves the privacy for u_3 in the sense that after this click, by knowing the preference of u_3 for i_3, u_3 becomes indistinguishable from a group of other users who liked i_3 (this group consists of two users out of the total of three users, other than u_3). Moreover, **B** helps the recommender find relevant items to u_3 because i_3 is similar to i_4 and i_1, and user u_3 has not clicked on i_1 yet; the recommender can now propose i_1 to u_3. Hence, **B** improves both utility and privacy for u_3.

ℂ is a click by u_4 to dislike i_3 that neither improves privacy for u_4 nor helps the recommender suggest relevant items to u_4; before ℂ, the recommender could figure out that the preference of u_4 is close to u_1 (neither to u_2 nor u_3). However, ℂ indicates that the preference of u_4 toward i_3 is also opposite to u_1. Hence, ℂ does not improve the recommender utility for u_4; worst, ℂ compromises the privacy for u_4.

Clearly, this example contradicts the traditional belief [29] of an inherent trade-off between utility and privacy, meaning that a user necessarily improves recommendation utility at the expense of compromising privacy (or vice versa). There are clicks that improve utility without decreasing privacy and, at the other extreme, there are clicks that hamper privacy without improving utility.

An interesting application of our work is what we call the *click-advisor*, a virtual platform enabling users to decide whether or not to actually click on an item. The process behind a click-advisor is as follows: a user (a) *pre-clicks* on an item, (b) gets a quick feedback (in constant time, i.e. $O(1)$, as shwon in the paper) on the utility and privacy effects of that pre-click and then (c) decides to *confirm*, *cancel*, or even *change* the pre-click.

We use a game-theoretic approach to explore several user clicking strategies. For example, a user may follow a *careful* strategy to confirm a pre-click iff the pre-click does not hamper privacy without improving utility (and to cancel the pre-click otherwise). We highlight in particular a Nash equilibrium strategy, which we call *smart*. If all users follow the smart strategy, they minimize their disclosure degrees (maximize their privacy) while ensuring that the expected value of the overall recommender utility for all users remains the same as the case without consulting the click-advisor.

We illustrate our notions of utility and disclosure risk of a click as well as the effects of various clicking strategies through experiments on real datasets from Movielens [2] and Jester [19]. We show that our notion of commonality of a user profile indeed conveys the classical concept of *precision* of a recommender [32], restricted to a user profile. We also show for instance that, according to Movielens, 5.23% of the clicks improve utility without loss of privacy–at the other extreme, 16.48% of clicks induce a high privacy risk without a utility gain. Finally, we also show that the smart clicking strategy does not impact utility (while maximizing privacy).

The rest of the paper is organized as follows. In Section 2 and Section 3, we define and show how to compute the effects of a click on utility and privacy, respectively. Section 4 discusses the relation between utility and privacy, and introduces our notion of a click-advisor. Section 5 analyzes several clicking strategies based on a game-theoretic approach. We report on our measurements with datasets from Movielens and Jester in Section 6. Section 7 discusses related work and Section 8 concludes the paper with remarks about future work. For space limitations, we defer some algorithms and discussions about the click-advisor as well as the proofs of our theorems and lemmas to a companion technical report [1].

2 UTILITY

2.1 Recommender Model

We consider a general model of a CF recommender scheme [23]. The set of user clicks is modeled as a matrix, denoted by E. The profile of user u, denoted by $\mathcal{U}$, corresponds to a row of E. Each column of E is related to an item. For the sake of presentation simplicity, but without loss of generality, we model each click as a like/dislike action.[4] The result of user u clicking on item i is either a "like" or a "dislike", represented by 1 and -1 in the corresponding cell of E, $e(u,i)$, respectively. Also, we mark 0 in $e(u,i)$ if user u has not clicked on item i.

We denote by M the size of the set of items and by N the number of users. A user profile, $\mathcal{U} \in \{-1,0,1\}^M$, is a vector of size M, in which the cell corresponding to item i is $e(u,i)$:

$$e(u,i) = \begin{cases} 1 & u \text{ has clicked on } i \text{ and likes it;} \\ 0 & u \text{ has not clicked on } i; \\ -1 & u \text{ has clicked on } i \text{ and dislikes it.} \end{cases}$$

For each item i, we denote by $N_{Like}(i)$ and $N_{Dislike}(i)$ the number of users who like and dislike i, respectively. The *item profile*

[3] Three remarks are in order here. First, our notion of disclosure degree is a privacy measure of a user profile, unlike k-anonymity [39] and differential privacy [12], which are privacy measures of a dataset and an algorithm, respectively. In Section 6, we show the extend to which our notion of disclosure degree is correlated to differential privacy and k-anonymity. Second, (just like commonality) the disclosure degree of a user profile depends on other profiles. The fact that Bob is the only one to click on his esoteric movie is what make his disclosure degree high. Third, a low disclosure degree conveys a protection against possible attacks of curious users, but not against a recommender, which is trusted.

[4] This can model binary as well as non-binary types of rating: a low rating as a dislike, and a high rating as a like.

Notations	
Items	The set of all items in the system
Users	The set of all users in the system
E	A recommender dataset
u	A user
$\mathcal{U}$	A user profile
i	An item
$\mathcal{I}$	An item profile
I	The set of all item profiles
M	The number of items in the system
N	The number of users in the system
$N_{Like}(i)$	The number of users who like item i
$N_{Dislike}(i)$	The number of users who dislike item i
$N_{NotClicked}(i)$	The number of users who have not clicked on item i
$Clicked(u)$	The set of items clicked by user u
U	A random created user profile based on the information in item profiles

Table 2: Notation Table

Item	Item Profile			Popularity	Preferability
	N_{Like}	$N_{Dislike}$	$N_{NotClicked}$		
i_1	2	0	2	0.5	0.5
i_2	0	1	3	0.25	-0.25
i_3	2	0	2	0.5	0.5
i_4	2	1	1	0.75	0.25

Table 3: Profiles, Popularity and Preferability of Items in Table 1

of i, denoted by $\mathcal{I}$, contains $N_{Like}(i)$, $N_{Dislike}(i)$ as well as the number of users who have not clicked on i yet, $N_{NotClicked}(i)$.[5] Although user profiles are *private*, item profiles are typically *public*[6] (e.g., in IMDb and Movielens). Table 2 summarizes the notations we use in the paper.

2.2 Commonality of a User Profile

To define our notion of *commonality* of a user profile (or simply, of a user), we first go through the concepts of *popularity* and *preferability* of items.

Definition 1. (Popularity)

$$popularity(i) = \frac{N_{Like}(i) + N_{Dislike}(i)}{N}.$$

Definition 2. (Preferability, i.e., average of preferences of users toward an item)

$$preferability(i) = \frac{N_{Like}(i) - N_{Dislike}(i)}{N}.$$

The item profiles as well as popularity and preferability of items in Table 1 are represented in Table 3.

Definition 3. (Mainstream preference) The mainstream preference of users for item i, denoted by $m(i)$, is computed as $popularity(i) \cdot preferability(i)$. For all items managed by a recommender, the mainstream preference of users is stored in a vector of size M in which the corresponding cell to item i is $m(i)$. We denote the mainstream vector by $m \in [-1, 1]^M$.

We now introduce the concept of commonality of user u, (with respect to other users) denoted by *commonality(u)*. Roughly speaking, we capture by a number, how close the user profile $\mathcal{U}$ is to other users profiles in the system in general.

[5]In other words, an item profile, $\mathcal{I}$, includes the average rating of i as well as the number of users who like/dislike i.

[6]Item profiles (the average rating, the percentages of users who have used/purchased an item, etc) play an important role in convincing a user to actually purchase an item.

	i_1	i_2	i_3	i_4	Commonality
u_1	✓		✓		$0.5 \times 0.5 + 0 + 0.5 \times 0.5 + 0 = 0.5$
u_2			✓	✓	$0 + 0 + 0.5 \times 0.5 + 0.75 \times 0.25 = 0.4375$
u_3		✗		✓	$0 - 0.25 \times (-0.25) + 0 + 0.75 \times 0.25 = 0.25$
u_4	✓			✗	$0.5 \times 0.5 + 0 + 0 - 0.75 \times 0.25 = 0.0625$

Table 4: Commonalities of Users in Table 1

Definition 4. (Commonality)

$$commonality(u) = \sum_{i \in I} popularity(i) \cdot preferability(i) \cdot e(u, i).$$

Remark 1. $commonality(u) = \mathcal{U} \cdot m^T$.

Remark 2. Basically, the commonality of a user represents how close the direction of the vector $\mathcal{U}$ is to the direction of the mainstream vector, m, by relatively computing the cosine of the angle between vector $\mathcal{U}$ and vector m.

The commonalities of the users in Table 1 are computed in Table 4 using Table 3 and Definition 4. As all the users clicked on exactly two items in Table 1, Table 4 shows that commonality is not proportional to the number of user clicks. Instead, commonality is a weighted function of user clicks based on how well the clicks help the recommender connect users (or items) to suggest them new items. For instance, the commonality of u_1 is higher than the commonality of u_4, therefore the recommender should provide more accurate recommendations to u_1 than to u_4. In Section 6, we emprically show that the commonality captures the quality of not only SVD-based recommenders but also KNN-based ones. [7]

2.3 Utility of a Click

The utility of a click by user u is the difference between $commonality(u)$ before and after that click. To distinguish both cases, we use the prime notation for the latter: $commonality'(u)$ denotes u's commonality after the click.

Definition 5. (Utility of a click by user u)

$$\Delta commonality(u) = commonality'(u) - commonality(u).$$

3 PRIVACY

We define the notion of *disclosure degree* of a user profile (or simply, of a user) based on the classical concept of *entropy* in information theory [15]. Basically, the disclosure degree of user u, which we denote by δ_u, corresponds to the amount of information that item profiles contain about u.[8] Remember that our goal here is to protect users from other (curious) users. We assume that the recommender, which stores all the user and item profiles, is trusted.

3.1 Disclosure Degree of a User Profile

We address the situation where an intruder (i.e., a curious user), given public item profiles (I), tries to disclose information about user profiles with a disclosure probabilistic model. The considered privacy disclosure conveys the intruder's ability to uniquely identify a user using this probabilistic model [3]. One could interpret the disclosure degree of u as the number of bits of information that the intruder has gained in order to uniquely identify u, given I.

The disclosure probabilistic model determines the probabilities that users have clicked on items; these probabilities are assigned by the intruder to a district random user profile, U. Let U have the

[7]The commonality of a user profile can be further specified for a CF recommender based on a factorization method with latent factors.

[8]We exclude users without any click from our study as those have no privacy concern.

User	Commonality	Disclosure Degree
u_1	0.5	$-\log(0.50 \times 0.75 \times 0.50 \times 0.25) = 1.329$
u_2	0.4375	$-\log(0.50 \times 0.75 \times 0.50 \times 0.50) = 1.028$
u_3	0.25	$-\log(0.50 \times 0.25 \times 0.50 \times 0.50) = 1.505$
u_4	0.0625	$-\log(0.50 \times 0.75 \times 0.50 \times 0.25) = 1.329$

Table 5: Disclosure Degrees of Users in Table 1

conditional disclosure probability function $Q(\mathcal{U}) = Pr(U = \mathcal{U}|I)$, in which $\mathcal{U}$ represents each possible user profile in the system. For a given disclosure probability distribution, the concept of disclosure degree provides a measure of the information (entropy in information theory) about user profile $\mathcal{U}$ stored in I. In this case, we denote by $\delta_u = H(\mathcal{U})$ the entropy of the user profile given the public item profiles, what we call the disclosure degree.

Naturally, the lower $Q(\mathcal{U}) = Pr(U = \mathcal{U}|I)$, the more unique user profile $\mathcal{U}$ is (the higher δ_u). When the interests of user u are determined by click on an item by u, most of the time, this click decreases $Pr(U = \mathcal{U}|I)$ and compromises the privacy for u. In other words, after a click, the user usually provides more information in the system and makes it hard for the recommender to hide this information (we show however this is not always the case).

To compute the disclosure degree of a user profile, we compute the probability of a click on an item i in U as follows:

$$E_i = \begin{cases} 1 & \text{with probability } \frac{N_{Like}(i)}{N} \\ & (U \text{ likes } i); \\ 0 & \text{with probability } 1 - \frac{N_{Like}(i)+N_{Dislike}(i)}{N} \\ & (U \text{ does not click on } i); \\ -1 & \text{with probability } \frac{N_{Dislike}(i)}{N} \\ & (U \text{ dislikes } i). \end{cases}$$

The following remark highlights the distribution of our disclosure probabilistic model for each possible user profile given I.

Remark 3. For any possible user profile $\mathcal{U}$:

$$Q(\mathcal{U}) = Pr(U = \mathcal{U}|I) = \prod_{i \in Items} Pr(E_i = e(u, i)).$$

We use the above disclosure model to define δ_u, the disclosure degree of user u:

Definition 6. (Disclosure degree)

$$\delta_u = -\log(Pr(U = \mathcal{U}|I)).$$

Note that δ_u captures the amount of information about a user profile stored in the public item profiles. In Section 6, we simulate a KNN attack [6] to illustrate the protection level of different users with different disclosure degrees. We show how accurately an intruder can disclose information about users with different disclosure degrees. The following remark highlights how to precisely compute δ_u. Later, we use Remark 4 to compute the privacy effect of a click by user u.

Remark 4. $\delta_u = -\sum_{i \in Items} \log(Pr(E_i = e(u, i)))$.

The disclosure degrees of the users in Table 1 are computed in Table 5 using Definition 6. The commonality and disclosure degree[9] of each user are shown in the corresponding columns in Table 5. Despite the fact that u_4 has a higher disclosure degree than u_2, the recommender can provide better recommendations to u_2 than to u_4 because the commonality of u_2 is higher than that of u_4. This shows that a higher disclosure degree does not necessarily lead to a better utility. Actually, not only does the click of u_4 on i_4 compromise u_4's privacy but neither does it help her get good recommendations.

[9]A user with high disclosure degree has a low level of privacy.

	$\Delta commonality(u) > 0$	$\Delta commonality(u) < 0$
$\Delta\delta_u > 0$	Trade-off	Dangerous/Deleterious
$\Delta\delta_u < 0$	Safe	Trade-off

Table 6: Conditions for Each Click Zone

3.2 Disclosure Risk of a Click

We fix a user u for whom we compute the effect of a click on privacy. We denote the difference of the disclosure degree of user u before and after the click by $\Delta\delta_u$ which can be expressed as the disclosure risk of the click.

Definition 7. (Disclosure risk of a click by user u)

$$\Delta\delta_u = \delta'_u - \delta_u.$$

Note that a click with a positive disclosure risk compromises the privacy for a user. The larger the absolute value of the disclosure risk of a click, the greater the privacy effect of that click. The following remarks can then be easily derived:

Remark 5. Whenever u clicks on i, the disclosure degree of u does not increase over time by clicks of other users on i

Remark 5 basically says that the disclosure risk of a click on i by u is actually an upper bound of all changes in the disclosure degree of u corresponding to item i after the time of the click.

4 CLICK-ADVISOR

In traditional recommenders, when a user clicks on an item, the information about that click is directly propagated to the system. We propose to leverage our utility and privacy metrics to build a user-aware clicking tool: the *click-advisor*. After getting a new recommendation, a user *pre-clicks* on the item and previews the utility and privacy effects of the pre-click through the click-advisor. Based on the click-advisor feedback, the user finalizes the decision on whether to click or not.

We categorize the clicks into four different zones depending on the sign of utility and privacy effects of the click as well as its reverse[10]:

- Safe zone: improves both utility and privacy.
- Trade-off zone: induces a utility-privacy trade-off.
- Dangerous zone: compromises both utility and privacy but the reverse of the click improves privacy.[11]
- Deleterious zone: Both the click and its reverse compromise utility as well as privacy.

In Section 6, we show that, besides trade-off clicks, safe, deleterious and dangerous clicks indeed exist in real-world datasets. Table 6 describes the corresponding zones for all possible cases for the disclosure risk and utility of a click. Note that there are two cases in which the clicks induce a trade-off: clicks with positive utility and positive disclosure risk, as well as clicks with negative utility and negative disclosure risk. To distinguish deleterious and dangerous zones for a click, we compute the disclosure risk of the reverse of the click, $\Delta\delta^*(u)$. If $\Delta\delta^*(u) > 0$, the click is in a deleterious zone, otherwise, it is in a dangerous zone.

Table 7 shows the utility effect ($\Delta commonality$) and privacy effect ($\Delta\delta$) of each of the following clicks. The zone of a click is

[10]The reverse of a like click is a dislike click and vice versa.

[11]Note that the reverse of the click always compromises the utility for the user as it is not based on the real preferences of that user.

	i_1	i_2	i_3	i_4	Click	$\Delta commonality$	$\Delta\delta$	$\Delta\delta^*$
u_1	✓	○	✓		D_2^1	0.25	0.176	0.477
u_2		○	✓	✓	L_2^2	0	0.477	0.176
u_3		✗	○	✓	L_3^3	0.5625	-0.176	0.301
u_4	✓		○	✗	D_3^4	-0.1875	0.301	-0.176

Table 7: The Effects of Clicks

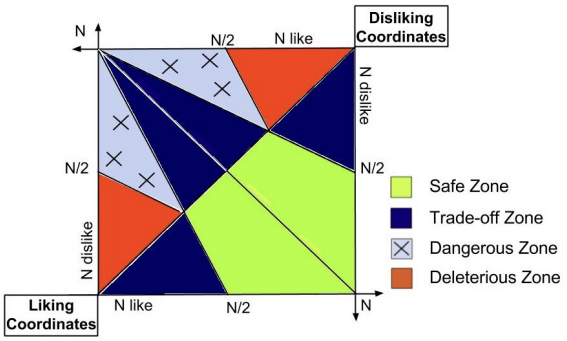


Figure 1: Click-advisor

determined by the sign of the utility and disclosure risk of that click as described in Table 6:

D_2^1 (the dislike click on i_2 by u_1): has a positive utility and a positive disclosure risk. Hence, D_2^1 is in the trade-off zone.

L_2^2 (the like click on i_2 by u_2): has a negative effect on the recommender utility for u_2. Moreover, not only L_2^2 has a positive disclosure risk but the reverse of L_2^2 also does. Hence, changing L_2^2 does not improve the privacy for u_2; L_2^2 is in a deleterious zone.

L_3^3 (the like click on i_3 by u_3): has a positive utility and a negative disclosure risk. L_3^3 thus improves both utility and privacy for u_3 and is thus a safe click.

D_3^4 (the dislike click on i_3 by u_4): compromises both utility and privacy for u_4; D_3^4 is an unsafe click for u_4. However, considering the reverse of D_3^4, even though changing D_3^4 decreases the recommender utility for u_4, the reverse of D_3^4 improves the privacy for u_4 ($\Delta\delta^*$), as shown in Table 7. That makes D_3^4 a dangerous click so u_4 may prefer to change this click.

The following theorem determines the exact signs of $\Delta\delta_u$, $\Delta\delta^*(u)$ and $\Delta commonality(u)$.

Theorem 1. When user u clicks on item i', the sign of the changes in $commonality(u)$ and δ_u are as follows:

$$sgn(\Delta\delta_u) = sgn(2 - 3popularity(i) - e(u,i') \cdot preferability(i)),$$

$$sgn(\Delta commonality(u)) = sgn(e(u,i') \cdot preferability(i)),$$

$$sgn(\Delta\delta^*(u)) = sgn(2 - 3popularity(i) + e(u,i') \cdot preferability(i)).$$

Figure 1 represents the zones in the click-advisor in general. At any point in time the click-advisor of a user represents the zones as well as the points corresponding to all pre-clicks of the user. A pre-click is presented as a point in one of the zones based on the privacy and utility effects of a click and its reverse using Theorem 1 (which we prove in our technical report [1]) and Table 6. The location of a pre-click is defined by the corresponding coordinates based on whether the pre-click is a like or dislike. For example, to locate the corresponding point to a click on item i in order to like it, we use the liking coordinates.

Furthermore, the place of a pre-click in the click-advisor represents the amount of disclosure risk and utility of that pre-click and can be computed in $O(1)$ as shown in Algorithm 1. At the time

Algorithm 1 : Computations of Commonality and Disclosure Degree in $O(M)$ and Computation of the Utility and Privacy Effects and Updates after a Click in $O(1)$

```
1:  procedure GetCommonality
2:      commonality ← 0.
3:      δ[u] ← 0.
4:      for i : Items do
5:          if u likes i then
6:              commonality ← commonality + m[i].
7:              δ[u] ← δ[u] − log(Pr(E_i = 1)).
8:          else
9:              if u dislikes i then
10:                 commonality ← commonality − m[i].
11:                 δ[u] ← δ[u] − log(Pr(E_i = −1)).
12:             else
13:                 δ[u] ← δ[u] − log(Pr(E_i = 0)).
        return (commonality, δ[u]);
14: procedure UpdateClick
15:     m[i] ← m[i] + Δm[i].
16:     ∀r ∈ {−1, 0, 1} : Pr(E_i = r) ← Pr(E_i = r) + ΔPr(E_i = r).
```

of a click on item i, Algorithm 1 only computes and updates $m[i]$, $Pr(E_i = 1)$, $Pr(E_i = 0)$ and $Pr(E_i = -1)$ in a constant time to be employed by the click-advisor. Also, Algorithm 1 computes the commonality (using Remark 1) and the disclosure degree (using Remark 4) of a user profile, linearly in the number of items.

The click-advisor solely uses the public item information to locate the pre-clicks of a users. Hence, the click-advisor can be implemented in the user-end to inform users about the utility and privacy effects of their pre-clicks without introducing any new privacy risks for the users. It is important to note here that the click-adviosr is not a privacy-preserving platform, but rather the visually informative one. More discussion about the click-advisor is available in our companion technical report [1]. For example, we show that safe pre-clicks never end up being in a dangerous or deleterious zone in the future because of the clicks of other users.

5 THE CLICKING GAME

To analyze the behavior of the users, we model the act of clicking as a game, which we call the *clicking game*. The players of this game are the users of the system (providing recommendations as well as a click-advisor) who want to maximize a *reward function*, described in the following, for each click.

5.1 Reward Function

The reward function is denoted by $\phi : Users \rightarrow \mathcal{R}$. Without loss of generality, we assume that $\forall u \in Users$, $\phi(u) = 0$ at the beginning. A click by user u with utility and privacy effects $\alpha = \Delta commonality(u)$ and $\beta = -\Delta\delta_u$ modifies $\phi(u)$ as follows:

$$\phi(u) \leftarrow \phi(u) + f(\alpha, \beta),$$

in which function $f : \mathcal{R}^2 \rightarrow \mathcal{R}$ has the following properties:

1. f is an increasing function over α.
2. f has the single crossing property over $\max(\alpha, \beta)$.

Property 1 of f implies that the higher the recommender utility for a user, the bigger the reward function for that user. The single crossing property (defined in [4]) merely defines the sign of f. More precisely, Property 2 means that the sign of $\max(\alpha, \beta)$ is the

same as the sign of $f(\alpha, \beta)$. Property 2 of f implies the following naturally desired sub-properties for $\phi(u)$:

2a. If user u cancels a click, $\phi(u)$ remains the same.
2b. Compromising utility and privacy by a click of user u leads to decreasing $\phi(u)$ (because of a negative $f(\alpha, \beta)$).

Upon getting a new recommendation, a user pre-clicks on the item and consults the click-advisor for the effects of that pre-click. Then, the user chooses the desired action on the pre-click in order to maximize the reward function through the clicking game.

5.2 User Actions

Consider N users, playing the clicking game. If a pre-click of a user is in a safe or trade-off zone, then either the privacy or the utility improves for that user after confirming that pre-click. However, confirming a pre-click in either a dangerous or a deleterious zone compromises both utility and privacy. The user may then decide to change or cancel a deleterious or dangerous pre-click. Basically, we can consider three possible actions for the user:

- Confirm: When the click follows the initial preference of the user (i.e., the user confirms the pre-click), the utility and disclosure risk of the click are the same as what we computed in Theorems 1 and 2, respectively.
- Change (for better privacy): The user decides to change the pre-clicks (i.e., clicking to dislike an item instead of clicking to like it) to improve privacy. In this case, the disclosure risk of the click is the same as what is computed in Theorem 2. However, the utility of the click would change from what we measured in Theorem 1 because this click is not the real preference of the user, but actually the opposite to it. So, the utility of the click is the opposite of the utility computed in Theorem 1.[12]
- Cancel: In this case, the user decides not to click on an item to preserve privacy. By canceling a pre-click on the item, a user preserves privacy (which is the opposite of what is calculated based on Theorem 2 assuming the pre-click was confirmed) at the expense of missing a potential utility. With the same approach, the lost utility is what the user would get with the assumption (of confirming the pre-click) in Theorem 1.

5.3 User Strategies

After being recommended a new item, a user considers one of the above actions to gain the maximum possible reward over time. There are various possible user clicking strategies. We highlight few of them here.

- Basic: A user always confirms the pre-click based on her preference. In this strategy, the click-advisor is ignored.
- Careful: The user only cancels a pre-click in a dangerous or deleterious zone. Otherwise, the user confirms it.
- Smart: The user always confirms a pre-click in a safe or trade-off zone, cancels the deleterious pre-click and reverses the pre-click in a dangerous zone.

The following theorem says that the smart strategy leads to a Nash equilibrium in the clicking game.

[12]Note that the utility of a recommender for users depends on the real preferences of users. Hence, the action of changing a pre-click by a user always decreases the recommender utility for that user.

Theorem 2. All users playing smart is a Nash equilibrium, i.e., playing the smart strategy maximizes the reward function ϕ for a user u, if every user other than u plays smart.

Lemma 1. If all users play smart, for item i:
(i) If preferability$(i) < 0$, all users clicking on i disliked it.
(ii) If preferability$(i) > 0$, all users clicking on i liked it.

The proofs of Theorem 2 and Lemma 1 are given in our technical report [1]. From Lemma 1, we can conclude that playing the smart strategy by all users guarantees the maximum possible privacy for all users. However, it can hamper the recommender utility for users. We consider the sum of commonality for all users as an overall recommender utility. In the following, we compare the effect of the smart strategy and the basic strategy (the strategy of clicking based on the real preferences of users) on the overall recommender utility for users. By playing the smart strategy, the users may change some of their clicks. If all users play the smart strategy, the changes of the clicks depends on the initial clicks on items. In that case, we compute the average of the overall recommender utility for users for all possible initial clicks on items (in order to compare with the overall recommender utility for users playing the basic strategy in Theorem 3). The set of all possible initial clicks on all items is denoted by C. For user u who is playing the smart strategy and a possible $c \in C$, $commonality_{Smart(c)}(u)$ represents the commonality of the real profile of u when all users play the smart strategy and the initial clicks on items is c.

The following theorem says that the expected value of the average of the commonalities of the users in the case where they all play the smart strategy is the same as the case where all users play the basic strategy (users click on items based on their real preferences).

Theorem 3.

$$\sum_{u \in Users} commonality_{Basic}(u) = \mathbb{E}\Big(\sum_{u \in Users} commonality_{Smart(c)}(u)\Big).$$

In other words, Theorem 3 (which we prove in our technical report [1]) says that the overall average of the recommender utility for all users playing the smart strategy is the same as the case where users click on items based on their real preferences. However, with regard to privacy, the smart strategy provides better privacy compared to the basic strategy.

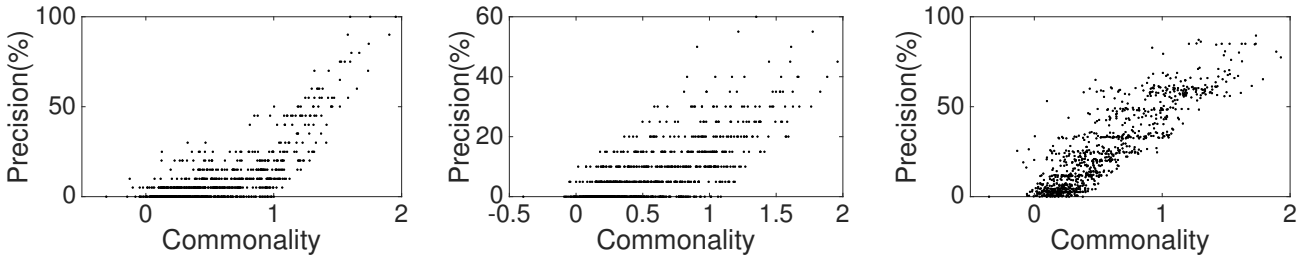
6 EXPERIMENTAL EVALUATION

We evaluate the utility and privacy metrics, as well as their application to build a click-advisor, on two real-world datasets: the Movielens 100K [2] and Jester [19] datasets. Movielens (ML) consists of 100,000 ratings given by 943 users over 1682 movies. Jester involves the clicks of 24,983 users on 100 items [10]. (As Jester does not provide timestamps, we perform the time-based evaluations for Movielens only).

6.1 Commonality as a Utility Measure

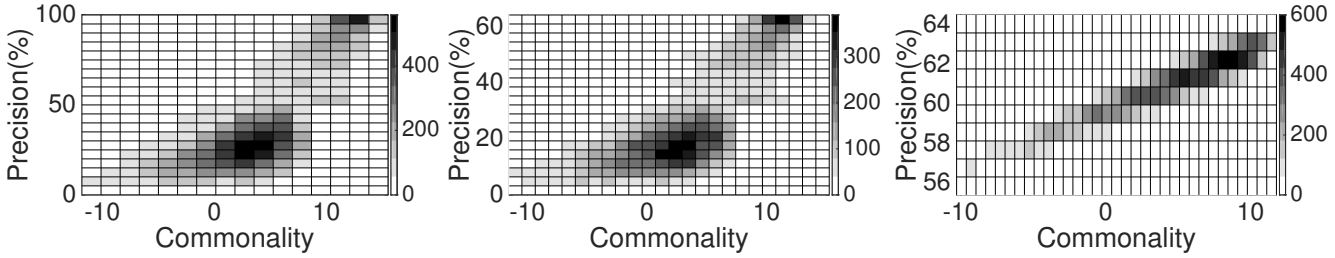
We first show here empirically that the notion of commonality of a user profile indeed expresses the quality of a recommender for that user. We more specifically highlight the positive linear correlation between commonality and the classical notion of precision [32].

We measure the precision of a recommender for each user as follows: we divide the dataset into a training set and a test set. We put each rating from the original dataset into the test set with



(a) Matrix Reconstruction as in [36] **(b) SVD-based Neighborhood as in** [24] **(c) K Nearest Neighbors as in** [38]

Figure 2: The Relation between Commonality and Precision in Movielens for Different Recommenders



(a) Matrix Reconstruction as in [36] **(b) SVD-based Neighborhood as in** [24] **(c) K Nearest Neighbors as in** [38]

Figure 3: The Relation between Commonality and Precision in Jester for Different Recommenders

probability 20% and *hide* all the ratings in the test set from the training set. For a user u, we put $N_u = 0.2 \cdot |Clicked(u)|$ and determine the top-N_u recommendations for u based on state-of-the-art recommender algorithms, namely a KNN (K Nearest Neighbors) algorithm [38], a matrix reconstruction approach [36] and a SVD-based recommender leveraging neighborhood [24].

Precision is measured in terms of standard classification accuracy metrics (CAM) [32]. More precisely, we evaluate how well a recommender can predict the context of the test set of a user u using each of the mentioned algorithms on the training set. We compare the top-N_u recommendations for user u with the hidden part of user profile u (test set). We compute *Precision(u)* as the classification accuracy metric used on top-N_u recommenders computed in [10]. *Precision(u)* is then the ratio of the number of relevant recommended items to the total number of recommended items to user u. Actually, $Precision(u)$ computes the recommendation quality for u.

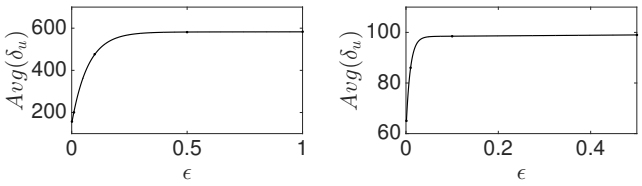
We report here on our results for both Movielens and Jester. As conveyed in Figures 2a and 2b, the commonality and precision follow a linear fashion in Movielens for a SVD-based recommenders using matrix reconstruction and neighborhood approaches. Mathematically evaluated, the correlation between commonality and precision is 0.6557 for SVD-based recommender using matrix reconstruction and the correlation increases to 0.6838 for a SVD-based recommender leveraging neighborhood. For Jester, because of the massive number of users, the direct correlation between commonality and precision in Jester for SVD-based recommenders is presented using pseudocolor plots in Figures 3a and 3b. Hence, the color of each block represents the number of users in it. (A darker color of a block means more number of users in that block.) For KNN recommender, Figures 2c and 3c depict a direct proportionality (more precisely, linear for both Movielens and Jester) between the commonality of a user profile and the precision of KNN recommender for the corresponding user. This basically means that users with high commonality get high precision for their recommendations in contrast to users with low commonality.

6.2 Disclosure Degree as a Privacy Measure

In this section, we use the Movielens and Jester datasets to show the extend to which our notion of disclosure degree (defined for each user profile) captures other well-known privacy concepts such as differential privacy [12, 31] (defined for algorithms) and k-anonymity [39, 8] (defined for datasets).

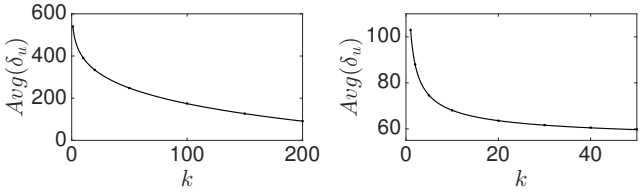
6.2.1 Differential Privacy. This guarantees that the presence or absence of a record in a dataset will not significantly affect the final output of the algorithm [12]. In the following, we study the disclosure degree as a privacy parameter of user profiles, and determine the relation between differential privacy and disclosure degree. We employ the approach proposed in [31] to provide differential privacy to recommenders by adding different levels of Laplacian noise to the Movielens dataset. Each noisy dataset corresponds to a level of differential privacy. Figure 4 represents the average of the disclosure degrees of user profiles in each of these noisy datasets. We observe that the users in a dataset with a high level of noise (i.e., a low ϵ) have smaller disclosure degrees in average compared to users in a dataset with a low level of noise. Intuitively, the presence (or absence) of a rare click (i.e., with high disclosure risk) highly affects the output of the recommender compared to the presence (or absence) of a regular expected click.

6.2.2 K-anonymity. F. Casino et. al. applied the concept of k-anonymity to collaborative filtering [8]. Figure 5 represents the average of disclosure degrees of user profiles in Movielens as well as Jester with different levels of k-anonymity. As depicted in Figure 5, increasing k (i.e., stronger anonymity for users) results in



(a) Movielens (b) Jester

Figure 4: Disclosure Degree and Differential Privacy



(a) Movielens (b) Jester

Figure 5: Disclosure Degree and K-anonymity

decreasing the average of disclosure degrees of user profiles.[13] Our concept of disclosure degree of a user profile captures how unique a user profile is in the system. Intuitively, the higher the disclosure degree of a user profile, the more distinguishable the user is among others (the higher the risk of identifying the corresponding user among all the users in the system).

6.2.3 KNN Attack. We consider here the KNN attack described in [6]. An attacker creates K fake profiles with some auxiliary information about a target user u_T and requests for recommendations. In the simple case of a KNN recommender, the K nearest neighbors of a fake profile are u_T and $K-1$ other fake profiles. When a fake profile gets a new item as a recommendation, this likely comes from the items which are already clicked by u_T. The reason is that the only items clicked by the K nearest neighbors of a fake profile which are not included in that fake profile are the items in the profile of u_T. Upon a recommendation for any of the fake profiles, all the fake profiles click on the recommended item in order to add the new recommended item to their profiles and be updated to get further information about u_T. The attack is performed through several iterations. In each iteration, the attacker accumulates new information about u_T. To figure out the relation between the disclosure degree of a user profile and the accuracy of the information an attacker can disclose about that user profile, we trigger this KNN attack for the users with different disclosure degrees in Movielens and Jester.

For this simulation, we consider the Movielens as well as the Jester dataset. We consider an attacker simulating a KNN attack to disclose information about a given target user. We denote the disclosure degree of the target user by δ_{u_T}. We illustrate the accuracy and the amount of extracted information (*%EI*) by the attacker about different target users with different disclosure degrees.

For our simulations, we put $K = 10$. Table 8 shows how accurate information an attacker can disclose by simulating the attack for users with different disclosure degrees after 10 iterations. Table 8

[13] As k anonymity is a privacy parameter for a dataset, not for a user profile, to compare k-anonymity and disclosure degree, we use the average disclosure degrees of all user profiles as an overall privacy measure for users.

Dataset	u_T	Auxiliary Information	δ_{u_T}	*%EI*	Accuracy
MLV	3	[1, 2, 3]	235.09	7.33	97.14
MLV	86	[1, 2, 3]	191.66	13.00	80.00
MLV	200	[1, 2, 3]	118.07	15.33	65.71
Jester	7	[0, 3, 6, 7, 5]	118.57	8.24	98.00
Jester	62	[1, 3, 4, 5, 0]	93.21	11.63	85.71
Jester	303	[5, 6, 7, 9, 10]	70.23	20.91	65.71

Table 8: Result of a KNN Attack for Jester and Movielens after 5 Iterations

(a) Pre-clicks (b) Click-advisor

Figure 6: An Screen-shot of the Implemented Click-advisor

shows that the higher the disclosure degree of a user, the more accurate the information disclosed by the attacker about that user.

We show that our notion of disclosure degree is compatible with state-of-the-art privacy metrics (i.e., k-anonymity and differential privacy). To illustrate the disclosure degree further, we show that a user with low disclosure degree is more resistant to a KNN attack, than another user with high disclosure degree. It is important to notice though that our concept of disclosure degree is independent of any particular attack. Also, remember that an attacker here has only access to the output of a recommender, not an (anonymized) version of the dataset. Therefore, de-anonymization attacks cannot be applied in this case.

6.3 Click Zones

We compute the utility and privacy effects as well as the zones of the clicks in Movielens, as discussed in Sections 2 and 3 and 4. (Jester does not provide timestamps.) We observe that most of the clicks (77.63%) induce a trade-off between utility and privacy. Yet, there are clicks which do not induce a utility-privacy trade-off: others are safe (5.94%), dangerous (2.00%) and deleterious (14.43%).

6.4 Click-advisor

We implemented the click-advisor for a SVD-based neighborhood recommender (as in [24]) using the Movielens dataset. Figure 6 represents a screen-shot of our click-advisor. At the beginning, three recommended items are provided to a user. The genuine pre-clicks (like or dislike) of a user on the recommended items are shown in Figure 6a. In Figure 6b, the zone of each pre-click (shown as grey circles) corresponds to the utility and privacy effects of that

	Confirm	Change	Cancel	Ask Me
Safe	100 %	0%	0%	0%
Trade-off	96.97 %	0 %	0 %	3.03 %
Dangerous	3.03 %	39.4 %	30.30 %	6.07 %
Deleterious	3.03 %	0 %	93.94 %	3.03 %

Table 9: The Preferred Default Action in the Click-advisor by the Participants in the Survey

Strategy	Smart	Careful	Basic
Precision(%)	27.02	24.63	28.87
Accuracy of a KNN Attack (%)	55.67	60.53	82.12

Table 10: Recommender Precision and KNN Attack Accuracy for Different User Strategies

pre-click on the user profile. Consulting the click-advisor, the user would decide an action for each pre-click.

The link to our click-advisor as well as a survey to gather the opinions of users on the click-advisor were distributed. More than 95% of respondents agreed that the click-advisor increases their confidence in clicking and their trust to click on more items. Also, we asked about the preferred default strategy in the survey. Table 9 represents the percentage of the respondents who prefer a default action (corresponding to a column) in the case of each types of a click (corresponding to a row). As shown in Table 9, the users chose the presented strategies as their default strategy as follows: Basic (3.03%), Careful (30.30%), and Smart (39.4%). The rest of respondents (6.06%) preferred to be asked as default for some types of clicks. Knowing that the Smart strategy is a Nash equilibrium, we can advertise the Smart strategy as the default strategy. However, the users may adopt their actions as they prefer.

6.5 Clicking Strategies

Section 5 discusses different user strategies for the clicking game. We consider here the *basic*, *careful*, and *smart* strategies and apply each of them to the Movielens dataset. Based on the timestamps in Movielens[14], we consider a click as a pre-click and place the pre-click of the user in the click-advisor instantaneously. Based on the position of the pre-click and the user strategy, we apply an action to the pre-click (confirm, cancel or change). Applying the same strategy to all the users, we create a new updated dataset and compute the precision of the recommender for this updated dataset. Table 10 shows the precision of the recommender applying each of the strategies to all the users. As predicted by Theorem 5, the precision is almost the same for the basic and smart strategies.

We also apply the KNN attack to the updated datasets for each user strategies. We consider the KNN attack with auxiliary information as what described in Section 6.2.3 for Movielens. For this experiment, we average the accuracy of the KNN attack over all the cases in which every single user is the target user. Table 10 shows that the users become more resistant to the KNN attack as they choose the Smart strategy over the Careful and Basic ones. [15]

7 RELATED WORK

CF recommenders. Although it has been shown that CF recommenders perform well, they have some limitations for sparse datasets [38]. Sarwar, et. al. applied Singular Value Decomposition (SVD) to reduce the dimensionality of sparse datasets of collaborative filtering recommenders [36]. Recently, many collaborative filtering recommenders used matrix factorization methods to cope with the sparsity of data, items correlations and dynamic recommendation domains [26, 24, 25, 28]. Our utility and privacy measures are compatible with matrix factorization-based CF recommenders as well as neighborhood-based ones.

Privacy in IR and recommenders. Privacy risks of user posts in online communities were studied in [5] for textual contents, unlike this paper which studies the privacy risks of a single click. Moreover, several papers proposed privacy-preserving methods in the context of recommenders [33, 30, 22]. Perturbative methods were proposed in [33], where users submit perturbed ratings to the recommender. However, even showing perturbed interest in a certain item may also disclose the preferences of users. For example, the perturbed rating of a user on a comedy movie still highlights the interest of the user in watching comedy movies. Furthermore, some works [17, 21] indicate that the use of randomized data distortion techniques might not able to preserve privacy. Regarding the use of cryptographic techniques, Canny proposed a method that enables a group of users to calculate a public aggregate of their profiles without revealing them on an individual basis [41, 7]. The major downside of this method is, however, the assumption of an acceptable number of users is online and willing to participate in the protocol. None of these privacy preserving approaches measured the privacy effect of a single click in a recommender (which we do in this paper).

Differential privacy. The notion of differential privacy was introduced by Dwork in the context of databases [12] and later adapted to recommender *algorithms* [31, 20, 16]. In contrast, our concept of disclosure degree studies the privacy for a specific user regardless of the recommender algorithm. While in differential privacy the recommendations are considered the only observation of an intruder, we take into account the public item profiles as another source of available information for the intruder (unlike differential privacy, we seek to assign a privacy parameter to each user profile in the system, regardless of the recommender algorithm).

A noise addition technique was described in [31] to apply differential privacy to recommenders. In Section 6, we analyze the relation between (a) the different levels of additional noise to dataset as a differential privacy approach in [31] and (b) the disclosure degree of users in the obfuscated datasets with different ϵ. We empirically show that the higher the level of noise in an obfuscated dataset (higher differential privacy), the lower the disclosure degrees of users in average.

K-anonymity. The concept of k-anonymity as a privacy protection method, was first formulated in [39], and recently applied to collaborative filtering recommenders in [8]. In Section 6, we compare the disclosure degree of users in different obfuscated datasets for different k, and we show that the average disclosure degree of users in a dataset actually captures the level of anonymity of that recommender dataset.

Degree of anonymity. Chaum introduced the notion of anonymity set in order to model the security of Dining Cryptographers' networks in [9]. Serjantov and Danezis raised some issues about anonymity sets [37]. For example, anonymity sets do not take into account the risk of inferring some sensitive attributes of

[14]The absence of timestamps makes this simulation impossible in Jester.

[15]A higher accuracy of a KNN attack means the users are less resistant to that attack.

the users in a set. To address these issues, a general measure to quantify the degree of anonymity for message passing was proposed in [37, 11]. Both papers independently proposed the use of entropy as the basis for formally measuring anonymity. The anonymity degree provided by the system quantifies the amount of information the system is leaking. We apply the idea of computing the amount of information stored in a user profile as a privacy parameter (disclosure degree) in the context of recommender datasets.

Disclosure risk. In the literature, disclosure risk measures have been classified as measures for record re-identification or confidential value disclosure [27, 14]. The latter focuses on measuring the risk of compromising a confidential value of a particular individual while the former focuses on measuring the risk of inferring the identity of an individual. In both cases, the disclosure risk measures may be applied to the database as a whole, or to individual records.

Alfalayleh and Brankovic measured the disclosure risk based on entropy [3]. However, if we use the same approach as in [3] to compute the disclosure risk of a user profile, we would get the same result for all users. Indeed, that approach provides a general metric for all users while our measure of disclosure degree is a specified metric for a user profile.

Recommender utility. The utility of recommenders has been measured differently in [18, 10, 13]. These measures, empirically evaluate the global quality of recommenders. In contrast, our commonality measure predicts the quality of recommenders for a user based on the user profile and an actual click. In Section 6, we show that commonality and precision indeed have a positive correlation.

Privacy-utility relation. The utility-privacy trade-off in databases was studied by Sankar et. al. in [29]. Their utility parameter is not applicable to recommenders for it computes the rate-distortion [40] of the database, which is related to the input of the recommender, not to the quality of its output. As we show in this paper, in the case of clicks in recommenders, there is not always a trade-off between utility and privacy.

8 CONCLUDING REMARKS

This paper is the first to precisely quantify the effects of a user click both in terms of utility and privacy in the context of recommenders. We show that all clicks do not have the same effect, neither in terms of privacy nor in terms of utility. In contrast to a common belief, we also show that there is not always a trade-off between utility and privacy. We introduce the idea of a click-advisor, which makes users aware of the effects of their clicks. Consulting the click-advisor, users can formulate their own informed clicks. Considering a reward function for the users in a clicking game, we highlight in particular a smart clicking strategy that leads to a Nash equilibrium. We prove that if all users follow this strategy, their privacy improves without compromising the average overall recommender utility (compared to the case where users do not use the click-advisor).

Our utility and privacy metrics could be further extended to latent factors of users and items in a SVD-based recommender. It would also be interesting to incorporate a recommender that optimizes the utility and privacy for all users solving a linear optimization problem for their commonalities and disclosure degrees.

Acknowledgements. This work has been supported in part by the European ERC Grant 339539 - AOC.

REFERENCES

[1] The utility and privacy effects of a click (technical report), 2017. http://go.epfl.ch/clickadvisor-technical-report.
[2] M. 100K. Movielens dataset, 2003.
[3] M. Alfalayleh and L. Brankovic. Quantifying privacy: A novel entropy-based measure of disclosure risk. In *IWOCA*, 2015.
[4] S. Athey. Monotone comparative statics under uncertainty. *QJE*, 2002.
[5] J. A. Biega, K. P. Gummadi, I. Mele, D. Milchevski, C. Tryfonopoulos, and G. Weikum. R-susceptibility: An ir-centric approach to assessing privacy risks for users in online communities. In *SIGIR*, 2016.
[6] J. A. Cal, A. Kilzer, A. Narayanan, E. W. Felten, and V. Shmatikov. You might also like: Privacy risks of collaborative filtering. In *S&P*, 2011.
[7] J. Canny. Collaborative filtering with privacy. In *S&P*, 2002.
[8] F. Casinoa, J. Domingo-Ferrerb, C. Patsakisc, D. Puigb, and A. Solanasa. A k-anonymous approach to privacy preserving collaborative filtering. *JCSS*, 2015.
[9] D. Chaum. The dining cryptographers problem: Unconditional sender and recipient untraceability. *JCRYPTOL*, 1988.
[10] P. Cremonesi, Y. Koren, and R. Turrin. Performance of recommender algorithms on top-n recommendation tasks. *RecSys*, 2010.
[11] C. Diaz, S. Seys, J. Claessens, and B. Preneel. Towards measuring anonymity. In *PET*, 2003.
[12] C. Dwork. Differential privacy. In *ICALP*, 2006.
[13] M. Ge, C. Delgado-Battenfeld, and D. Jannach. Beyond accuracy: evaluating recommender systems by coverage and serendipity. *RecSys*, 2010.
[14] D. L. George T. Duncan. Disclosure-limited data dissemination. *JASA*, 1986.
[15] R. M. Gray. *Entropy and information theory*. Springer, 2011.
[16] R. Guerraoui, A. Kermarrec, R. Patra, and M. Taziki. D2p: distance-based differential privacy in recommenders. *VLDB*, 2015.
[17] Z. Huang, W. Du, and B. Chen. Deriving private information from randomized data. In *SIGMOD*, 2005.
[18] M. Z. Islam, P. Barnaghi, and L. Brankovic. Measuring data quality: Predictive accuracy vs. similarity of decision trees. In *ICCIT*, 2003.
[19] Jester. Online joke recommender system, 2001.
[20] Z. Jorgensen, T. Yu, and G. Cormode. Conservative or liberal? personalized differential privacy. In *ICDE*, 2015.
[21] H. Kargupta, S. Datta, Q. Wang, and K. Sivakumar. On the privacy preserving properties of random data perturbation techniques. In *ICDM*, 2003.
[22] S. Katzenbeisser and M. Petkovic. Privacy-preserving recommendation systems for consumer healthcare services. *ARES*, 2008.
[23] J. A. Konstan and J. Riedl. Recommender systems: from algorithms to user experience. *UMUAI*, 2012.
[24] Y. Koren. Factorization meets the neighborhood: A multifaceted collaborative filtering model. In *KDD*, 2008.
[25] Y. Koren, R. Bell, and C. Volinsky. Matrix factorization techniques for recommender systems. *Computer*, 2009.
[26] D. kumar Bokde, S. Girase, and D. Mukhopadhyay. Role of matrix factorization model in collaborative filtering algorithm: A survey. *IJAFRC*, 2015.
[27] D. Lambert. Measures of disclosure risk and harm. *JOS*, 1993.
[28] S. Li, A. Karatzoglou, and C. Gentile. Collaborative filtering bandits. In *SIGIR*, 2016.
[29] L.Sankar, S. Rajagopalan, and H. Poor. Utility-privacy tradeoffs in databases: An information-theoretic approach. *IFS*, 2013.
[30] Y. Luo, J. Le, and H. Chen. A privacy-preserving book recommendation model based on multi-agent. *WCSE*, 2009.
[31] F. McSherry and I. Mironov. Differentially private recommender systems: Building privacy into the netflix prize contenders. In *KDD*, 2009.
[32] Y. K. P. Cremonesi and R. Turrin. Performance of recommender algorithms on top-n recommendation tasks. *RecSys*, 2010.
[33] H. Polat and W. Du. Privacy-preserving collaborative filtering using randomized perturbation techniques. *ICDM*, 2003.
[34] N. Ramakrishnan, B. J. Keller, B. J. Mirza, A. Y. Grama, and G. Karypis. Privacy risks in recommender systems. *Internet Computing*, 2001.
[35] 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, 2002.
[36] B. M. Sarwar, G. Karypis, J. A. Konstan, and J. T. Riedl. Application of dimensionality reduction in recommender system – a case study. In *ACM WEBKDD Workshop*, 2000.
[37] A. Serjantov and G. Danezis. Towards an information theoretic metric for anonymity. In *PET*, 2002.
[38] X. Su and T. M. Khoshgoftaar. A survey of collaborative filtering techniques. *AAI*, 2009.
[39] L. Sweeney. K-anonymity: a model for protecting privacy. *IJUFKS*, 2002.
[40] E. Tuncel, P. Koulgi, and K. Rose. Rate-distortion approach to databases: storage and content-based retrieval. *IT*, 2004.
[41] J. Canny. Collaborative filtering with privacy via factor analysis. In *SIGIR*, 2002.