Privacy in Data Publishing for Tailored Recommendation Scenarios

João M. Gonçalves[*,**], **Diogo Gomes**[**,***], **Rui L. Aguiar**[**,***]

[*]Portugal Telecom Inovação e Sistemas, R. Eng. José Ferreira Pinto Basto, Aveiro, 3810-106, Portugal.

[**]University of Aveiro, Campus Universitário de Santiago, Aveiro, 3810-193, Portugal.

[***]Instituto de Telecomunicações, Campus Universitário de Santiago, Aveiro, 3810-193, Portugal.

E-mail: `joao.m.goncalves@ua.pt,dgomes@av.it.pt,ruilaa@ua.pt`

Abstract. Personal information is increasingly gathered and used for providing services tailored to user preferences, but the datasets used to provide such functionality can represent serious privacy threats if not appropriately protected. Work in privacy-preserving data publishing targeted privacy guarantees that protect against record re-identification, by making records indistinguishable, or sensitive attribute value disclosure, by introducing diversity or noise in the sensitive values. However, most approaches fail in the high-dimensional case, and the ones that don't introduce a utility cost incompatible with tailored recommendation scenarios. This paper aims at a sensible trade-off between privacy and the benefits of tailored recommendations, in the context of privacy-preserving data publishing. We empirically demonstrate that significant privacy improvements can be achieved at a utility cost compatible with tailored recommendation scenarios, using a simple partition-based sanitization method.

Keywords. Recommender systems, Data anonymization and sanitization, Privacy-preserving data publishing, Rating prediction, Tailored recommendations, High-dimensional datasets

1 Introduction

The value of personal information for use in data mining is clear to everyone involved in the digital economy. Features like product recommendations and tailored advertising are usually welcomed by customers and, on the business side, consumer information is used by companies to define their strategies. However, the information required for either case is created from the analysis of personal data, which is increasingly collected and stored by businesses. This situation has originated a debate on the privacy issues of recent information gathering practices, especially driven by the widespread adoption of social networking sites and by their common practice of making available user's information to advertisers and application developers.

User data is many times made available to third parties in an supposedly anonymized form. However, the employed anonymization mechanisms are often naïve, relying on the replacement or deletion of common identifiers such as names and email addresses. Such methods are vulnerable to inference attacks because information that is apparently non-personally identifiable, such as zip code, sex and birth date, can be combined to identify the user. Sweeney determined that with those 3 attributes it is possible to identify 87% of the US population [41]. A well known case where privacy concerns were raised involved the Netflix Prize [33]. The Netflix Prize was a data mining competition whose objective was to crowd-source data mining algorithms for use in movie recommendations. Netflix released a sanitized movie ratings dataset for participants to use, suppressing the users' common identifiers, leaving only a meaningless user identification number. Using information crawled from IMDb, Narayanan and Shmatikov successfully re-identified a number of users from the Netflix dataset, showing that it was vulnerable to privacy attacks [31]. Netflix was to organize a follow up competition, Prize 2, which was cancelled due to concerns expressed by the Federal Trade Commission on the potential privacy infringements [14].

Recommendation techniques are also widely applied in other domains, such as e-commerce site items. Getting automatic recommendations for which items are worth looking at is essential when navigating a large search space. The datasets that enable this kind of analysis are designated as high-dimensional because they have a very large number of attributes - all the available movies or shopping items. They typically contain the known affinity values, such as ratings, between each user record and the attributes. However, these values are only known for a small fraction of the attributes, as each user only watches or rates a small number of available movies, meaning such datasets are typically sparse. The typical goal of data mining workloads used in such datasets is to estimate the null values, in order to provide tailored item recommendations to users.

Previous work in privacy preserving high-dimensional data publishing has targeted a static privacy goal when sanitizing such datasets. However, the utility costs of these approaches are evaluated in generic ways, with little results on how they perform in tailored recommendation scenarios. In this paper we propose a utility metric that can provide this insight. Furthermore we propose a metric to measure privacy instead of targeting a fixed goal, enabling working with scenarios where utility is not simply sacrificed for privacy, but instead the trade-off is improved. Also, a simple partitioning-based dataset sanitization method is described, applied to two high-dimensional datasets, and compared to a baseline state-of-the-art method.

This paper is organized as follows. Section 2 describes previous work in high-dimensional privacy preserving data publishing, focusing various perspectives as the assumptions and utility metrics considered. Section 3 introduces a utility metric adjusted to tailored recommendation scenarios and a privacy metric, instead of the traditional objective. In Section 4 we explain our partition-based approach to privacy preserving data publishing in tailored recommendations scenarios. In Section 5 the characteristics of the datasets are analysed and we establish reference algorithms for the experiments. In Section 6 the experimental results are described and, finally, Section 7 presents a summary of contributions and proposes future work.

2 Previous Work

2.1 Privacy-Preserving Data Publishing

Notable examples of supposedly anonymized data releases, which prompted legal and PR problems for the involved companies, include the AOL search queries [5], the Netflix Prize dataset, and Sweeney's zip code, sex and birth date example [41]. In this last one, a number of key attributes were used to infer the identity of the person to which a certain dataset row refers to. In order to address this issue, the term quasi-identifier was coined in the definition of k-anonymity [42], a reference privacy concept. Quasi-identifiers are attributes such as birth date and zip code, easy to obtain, typically part of the attackers auxiliary information, while sensitive attributes are the ones which the attacker aims to discover the value of. Sweeney protects privacy with k-anonymity by limiting re-identification based on quasi-identifiers. Informally, a dataset satisfies k-anonymity if and only if for each row of the dataset there are at least k-1 other rows with the same quasi-identifier values. In order to enforce this, the quasi-identifier values are generalized, masking their real values with more generic ones when necessary. In these groups of at least k elements (i.e. equivalence classes) the rows are indistinguishable regarding their quasi-identifiers. k-anonymity effectively targets the quasi-identifiers, improving resilience against re-identification attacks, but disregards the sensitive attributes.

A follow up approach dubbed l-diversity [26] captures this shortcoming by presenting an approach that resists to two types of attacks to which k-anonymity is vulnerable: a homogeneity and a background knowledge attack. In these attacks sensitive attributes can be leaked even if the attacker cannot associate the individual with a single row of the equivalence class, it just requires that the sensitive values are not diverse enough. Li et al. [24] further analysed these issues and proposed a privacy model that formalizes the privacy breach as the change of knowledge of the attacker as he comes in contact with a dataset. This approach, dubbed t-closeness, considers three attacker information states:

1. the attacker's prior belief (auxiliary information),

2. the attacker's belief after knowing the overall distribution on sensitive attributes in the released database,

3. the attacker's belief after knowing the distribution on sensitive attributes of the rows that match the target person.

Li et al. assume the overall distribution of sensitive attributes is very similar to the one from the global population, which configures public data, having very little information state change between 1 and 2. Consequently t-closeness focuses on protecting privacy by reducing the difference of information between information states 2 and 3, which implies approximating the distributions of sensitive attributes of each equivalence class to the overall sensitive attributes distribution. It is not enough that the sensitive attributes are diverse for each equivalence class, as l-diversity states, but that their sensitive value distribution is similar enough to the overall sensitive value distribution.

Generalization is a technique commonly used to achieve k-anonymity and similar privacy goals. However, it requires domain-specific hierarchies, making it unsuitable to be applied to every attribute without greatly reducing the dataset utility. Other value obfuscation methods have been proposed that can be applied to all the attributes, namely perturbation [4] and condensation [3]. Perturbation works by introducing error in the values according to a known density function, making it possible to recover generic statistical properties of the original data, but difficult to re-identify records since the error for a specific record is unknown. Condensation, unlike perturbation, considers the fact that attribute values are typically not independent. The data is "condensed" in a predefined number of groups and then randomly re-generated based on each group's statistical properties. This allows correlations between different groups to be preserved while making individual data records indistinguishable within the groups. Changing the number of groups allows to adjust the trade-off between privacy and data utility.

2.2 High-Dimensional Datasets

High-dimensional datasets are datasets with many attributes, such as the Netflix dataset, where ratings were given to 17 thousand different movies. High-dimensional datasets are usually sparse: each record typically has a (non-null) value defined only for a small fraction of the attributes. Also, the distribution of the attribute support is typically long-tailed: there is a small number of attributes that have non-null values for many records while there is a large number of attributes that only have non-null values for a few records. A side effect of this is that records are very distinguishable, even by merely considering which attributes are defined and which ones aren't, representing an anonymity threat even if attribute values are obfuscated or omitted [31, 1].

A wide range of work has been done towards addressing privacy in high-dimensional datasets. Xu et al. [46] formulate the privacy problem in a way that allows them to relate some amount of auxiliary information with the probability of sensitive attribute disclosure, and propose a suppression-based algorithm so that the dataset complies with that privacy requirement. In subsequent work [45], the concept of frequent itemsets are used in order to minimize the utility lost in the suppression process. Ghinita et al. [16] propose Correlation-aware Anonymization of High-dimensional Data (CAHD) to protect non-null occurrences of sensitive attributes in an high-dimensional dataset. The rationale is to form a group of similar records for each sensitive occurrence and associate the sensitive occurrence to a group rather than to a specific record. More recently, Parra-Arnau et al. [36] took a theoretical approach to high-dimensional privacy-preserving data publishing, and formulated it as an optimization problem addressed through data suppression and forgery (dubbed artificial or synthetic data in other work).

The high-dimensional case has also been targeted by privacy work that does not target data publishing, but instead interactive data access. Chawla et al. [11] presents an important base in this field by considering all attributes as dimensions of a hyper-cube of records and formulating mathematical definitions for privacy and sanitization. Also, two sanitization methods are proposed: one that relies in histograms to transmit the data - coarsely groups records to provide relevant statistical information - and another that uses perturbation to make records less identifiable (or isolated, in their terminology). Promising work has been recently done towards achieving Differential Privacy [13] in high-dimensional datasets. McSherry and Mironov [27] adapt the most common prediction techniques used for the Netflix Prize to return differentially private recommendations. The experimental results show impressive RMSE results, however the responses are differentially private regarding the detection of ratings and not users, which would require much more aggressive noise addition, as the authors themselves note. However, the interactive data access model that these approaches assume has fundamental implications on the adversary model and data applications, which differ from the data publishing model targeted in this paper.

2.3 The Quasi-Identifier Assumption

Most work in privacy-preserving data publishing [15] relies on the assumption that attributes can be classified as quasi-identifying or sensitive, where quasi-identifiers are attributes that are relatively easy to gather from other sources - which compose the adversary's auxiliary information - and sensitive attributes are the target of the privacy attack. However, in many situations the quasi-identifier and sensitive attribute separation cannot be clearly defined. Considering diverse real life scenarios, the sensitive attributes in one case may not be sensitive in another case: an individual's home address is sensitive information in a database of high profile art collectors, while it is a quasi-identifier in most other databases. Also, without assuming limitations on the adversary's access to information about an individual, any big enough set of attributes can be considered quasi-identifier as together the attributes are likely to re-identify that individual.

Re-identification attacks are usually agnostic to the semantics of the attributes and rely instead on two properties that are common to many types of personal information [32]. First is the stability of data across time, enabling datasets that are not temporally coincident to be used together in re-identification, and thus making it easier to have available auxiliary information. Secondly, if the quantity and precision of the data attributes is high enough, then it becomes highly unlikely that two individuals have the same set of values. This is especially easy in the high-dimensional case, prompting the discussion of what personally identifiable information truly is, and whether any type of information can be distinguished between personally identifiable and non-identifiable simply by its semantic.

Regarding the attributes of high-dimensional datasets, e.g. the movies rated and the items bought, while some movies/items have more potential to harm than others, it is not clear which attributes are sensitive. They are potentially all sensitive depending on the disclosure context: buddies would potentially crack jokes if they knew the rating given to some musical movie or the future employer could have second thoughts about hiring if he knew the rating given to certain ideology-charged movies, and so on.

Unlike previously discussed work [46, 45, 16], we and many other authors drop this assumption altogether in the high-dimensional case: any attribute can belong to the adversary's auxiliary information and all attributes are to be protected. Notably, Terrovitis et al. [43] proposed a new version of k-anonymity for set-valued high-dimensional data, k^m-anonymity, because k-anonymity assumes the existence of quasi-identifiers, and because the methods to achieve it do not scale to the high-dimensional case [30]. Given an adversary with auxiliary information of at most m attributes about a record, a k^m-anonymous dataset must contain at least k records undistinguishable with respect to those attributes. The authors also describe a generalization-based method to make set-valued high-dimensional datasets k^m-anonymous. This privacy guarantee is also adopted in a cluster-based generalization technique [17].

2.4 Partitioning-Based Techniques

The generalization and perturbation techniques commonly used with low-dimensional datasets are sometimes also applied to the high-dimensional case. However, as pointed out in Section 2.2, records of high-dimensional dataset are very distinguishable by merely considering which attributes are defined and which ones aren't [31, 1]. For that reason, dataset partitioning has been increasingly explored as an alternative technique for protecting privacy in the high-dimensional case. One of the advantages of partitioning is that the resulting dataset values remain unchanged - what changes is the records they are associated with.

Li et al. [25] propose *Slicing*: a vertical and horizontal partitioning method that complies with l-diversity. They argue that because vertical partitioning is used, slicing can be used in high-dimensional scenarios and test the algorithm on the Netflix dataset. However, as also noted by Terrovitis et al. [44], *Slicing* cannot handle sparse data. For the Netflix validation performed by them, the dataset's null values were replaced with the average rating of the movie, removing sparsity - the main source of distinguishability between records [31, 1]. Furthermore, the quasi-identifier and sensitive information assumption is present in this approach.

More promisingly, Chen et al. [12] describe a probabilistic top-down partitioning algorithm to generate differentially private data releases. Unlike most work done under Differential Privacy that considers an interactive approach to data access, Chen's goal is to publish a dataset via differential privacy. Also, in work following up the use of generalization in the high-dimensional case [43], Terrovitis et al. choose to use disassociation, a horizontal and vertical partitioning approach, to guarantee k^m-anonymity in a dataset of web query terms [44].

Recently Zakerzadeh et al. [47] published a vertical partitioning approach to achieve k-anonymity in high-dimensional datasets. The type of partitioning used differs from the one used by Terrovitis et al. [44], as it is applied uniformly to all records. They note that while all the theoretical difficulties of the dimensionality curse [1] remain true, their impact can be reduced by relying on common properties of real-life datasets.

2.5 Privacy for Recommender Systems

The term *collaborative filtering* was coined by the developers of one of the first recommender systems, and is commonly used to refer to such systems even if the system does not drive its users to collaborate explicitly [40]. Recommender systems can be functionally classified into three major groups [21]:

1. generic recommenders, which recommend sets of "good" items to the user;
2. utility optimization recommenders, a generic recommender tuned to the goals of the business implementing it;
3. prediction recomenders, which attempt to predict user opinion (i.e. rating) over a set of items.

Generic recommenders can rely only on aggregate data, such as average and total number of ratings, not requiring access to the full dataset. Some web sites simply keep track of the rating average of each item and the aggregate number of ratings in order to produce "popularity" driven recommendations. Thus, unsurprisingly, most literature on recommender systems focuses on prediction recommenders, which provide recommendations tailored to each user. These rely on the assumption that if some users rate some items similarly, they will also rate other items in a similar way. In order to identify these similarities, prediction recommenders analyse large ratings datasets [40], such as the one from Netflix Prize. Data mining is performed by the system on such high-dimensional datasets in order to estimate the missing values, which can be used to predict the rating that users would give to each item. Throughout this paper we will refer to recommendations given by prediction recommenders as tailored recommendations.

Pre-existing privacy work for recommenders can be classified based on the topology of the recommender system: centralized or distributed. In the centralized case, the goal of privacy work is to keep the recommender system from knowing the exact rating while still providing with useful recommendations. Similarly to the work in privacy preserving data publishing, the recommender system, which may be the adversary, has unrestricted access to the dataset after it has been published.

Privacy work addressing centralized recommenders is similar to the work seen in privacy-preserving data publishing. Approaches typically rely in the application of some perturbation to the ratings given by users before they are supplied to the centralized entity. Polat and Du [39] use perturbation, as do Berskovsky et al. [6] along with simpler obfuscation techniques. Their privacy model aims to hide from the centralized recommender, with

Privacy Scenario	Accessible Data	Recommender Applicability
Aggregate data release	Rating average and count per item	Generic recommendations
Interactive aggregate queries	Sanitized aggregate query responses	Generic recommendations
Distributed recommenders	Aggregate rating matrix	Tailored recommendations
High-dimensional data publishing	Sanitized high-dimensional dataset	Tailored recommendations

Table 1: Recommender Privacy Scenarios

sufficient probability, the real rating the user provided to each item. However, the attack models considered here do not include the use of auxiliary information.

In the case of distributed recommenders, a dataset of ratings given by numerous individual users is never collected and processed by a central entity. Users of such systems collaborate in a peer-to-peer manner in order to rate items such that the best rated items are recommended. The privacy goal here is keeping the values of ratings known only to the user that gave them. The adversaries are the other users that collaborate in rating the item. This enables new attack models, such as the ones proposed recently by [35].

A well established approach for the peer-to-peer case is the use of some homomorphic encryption scheme in the collaborative filtering protocol, as do Canny [8] and many others after him (e.g. [48, 37]). Homomorphic encryption enables a number of users sharing their encrypted ratings with each other and being able to retrieve the aggregate ratings. Furthermore, Canny [8] describes some degree of tailoring is possible using this scheme by locally correlating user preferences and the aggregate ratings model.

It's also possible to consider the interactive data access model in the context of recommender scenarios. In this case the attacker and the recommender also doesn't have access to the full dataset, only to aggregate queries performed on it. Differential Privacy, widely recognized in the data privacy community as a very strict privacy guarantee, is built on this model which is obviously capable of producing generic recommendations through aggregate results.

Table 1 synthesizes the applications of different privacy protection models to recommender scenarios. We assume that the recommendations and the adversary access data under the same model, and classify the possible recommendations under each scenario. We also add a naïve case in which a trusted centralized recommender stores only anonymous item rating averages and vote count data. Generic recommendations are possible in a number of different scenarios, some of which provide significantly stricter privacy guarantees than the ones possible in a data publishing setting.

2.6 Utility Metrics

A common approach used to measure utility in this field relies in the use of a proxy metric that quantifies the changes done to the original dataset by the sanitization algorithm.

Previous Work	Utility Metric	Type of Metric
Terrovitis et al. [43]	Normalized Certainty Penalty	Proxy metric quantifying changes to original dataset
Gkoulalas-Divanis and Loukides [17]	Generalization-minimization utility	Proxy metric quantifying changes to original dataset
Parra-Arnau et al. [36]	Forgery and suppression rate	Proxy metric quantifying changes to original dataset
Chen et al. [12]	Relative error of counting queries	Error on statistical aggregates
Terrovitis et al. [44]	Top-K deviation	Error on statistical aggregates
Terrovitis et al. [44]	Relative error in the support of term combinations	Error on statistical aggregates
Zakerzadeh el al. [47]	F-measure	Error on typical workload results

Table 2: Types of Utility Metrics in Previous Work

This is also true regarding high-dimensional privacy-preserving data publishing work that doesn't rely on the quasi-identifier assumption. Terrovitis et al. [43] estimated the impact of their generalization method by using Normalized Certainty Penalty (NCP), which captures the degree of generalization the method enforces. Similarly, Gkoulalas-Divanis and Loukides [17] rely on generalization-minimization utility measures which can be applied to non-hierarchical generalizations, and Parra-Arnau et al. [36] use forgery and suppression rate.

Another type of metrics validate statistical aggregates of the dataset. Chen et al. [12] evaluates utility through the relative error of counting queries. Terrovitis et al. [44] rely on two different metrics:

1. top-K deviation: the ratio of the top-K frequent itemsets of the original dataset that appear in the top-K frequent itemsets of the anonymized data;

2. relative error in the support of term combinations, limited to combinations of size two.

Finally, Zakerzadeh el al. [47] measure utility in terms of changes in classification accuracy, through F-measure. To the best of our knowledge, this is the only previous high-dimensional privacy-preserving data publishing work that doesn't rely on the quasi-identifier assumption and that uses an approach based on results of a typical data mining workload to measuring utility. Table 2 synthesizes the metrics used in such work, discussed in previous sections.

3 Privacy and Utility Metrics for Tailored Recommendation Scenarios

3.1 Tailoring Utility

As described in Section 2.6, privacy-preserving data publishing work typically addresses utility as a secondary objective, an optimization target having defined a privacy goal. In that context, the metrics used to quantify utility simply capture generic statistical properties of datasets or rely a proxy metric, being insufficient to conclude about real-world applicability, failing to give clear insight regarding utility for some common data mining workload. The one exception [47] has classification instead of prediction as the data mining goal.

Furthermore, as mentioned in Section 2.5, generic recommendations are possible without collection and publishing of personal data. Alternatively, tailored recommendations require complex data analysis, enabled, among other methods, by performing data mining on personal rating data. These data collection practices are justified by the benefit of tailoring, that is not possible to achieve with access to aggregate data alone. Thus, having applicability in mind and lacking existing satisfactory metrics, we consider a new utility metric to be used within the context of privacy-preserving data publishing, focused on the concept of tailored recommendations.

A realistic measure of utility in tailored recommendation scenarios is the prediction error of data mining algorithms used to perform them. This error can be measured in terms of root mean square error (RMSE) of the predicted values compared to the real values. Analogously to the workload-centric approach that Zakerzadeh el al. [47] took to a classification problem, we quantify the changes in prediction error for the original and sanitized versions of a dataset using the same prediction algorithm. Furthermore, because we work in the context of tailored recommendations, the error of naïve predictions based on aggregate data is considered as a baseline.

Let us formally define Tailoring Utility. Let P be the prediction function of a tailored recommender system, and A a naïve prediction function which predicts that all users rated items with the average rating given to that item. Let now $RMSE(P, D)$ be the RMSE resulting from applying prediction function P to dataset D, and $RMSE(A, D)$ be the RMSE of using a rating function based on aggregate data A to predict ratings in dataset D.

Definition 1. Tailoring Utility of prediction function P for dataset D, $\mu(P, D)$, captures the degree to which P adapts to the preferences of individual users in dataset D:

$$\mu(P, D) = 1 - (RMSE(P, D)/RMSE(A, D))$$

This new utility metric, Tailoring Utility, enables the comparison of sanitization processes to be applied in recommendation data publishing scenarios. $\mu(P, D)$ is positive if tailoring benefits predictions, and is proportional to the importance of tailoring in the recommendation.

Because data collection and publishing is justifiable only in tailored and not generic recommendation scenarios, we adopt a minimum acceptable value for utility: the utility provided by recommendations based on aggregate data. More formally, for a sanitization algorithm S, let $S(D) = D'$ be the sanitized dataset. If $\mu(P, D) > 0$, then, for S to be acceptable in the context of tailored recommendations, $\mu(P, D')$ must also be positive.

3.2 Measuring Privacy

Privacy-preserving data publishing work has given strong privacy foundations in sanitizing datasets, by making rows indistinguishable or protecting against attribute disclosure, while preserving some utility. However, most methods target static privacy guarantees and easily destroy utility beyond the threshold that justifies personal data collection, meaning that simply building aggregates and deleting personal data is a valid real-world alternative to them. Aiming at a sanitization method applicable and useful in real-world scenarios, we depart from the privacy preserving data publishing tradition of a static privacy guarantee. Let us consider a privacy metric instead of a guarantee, and attempt to improve the overall privacy-utility trade-off in data publishing for tailored recommendation scenarios.

A real-world adversary does not simply wish to defeat existing privacy protections. Thus, instead of protecting absolutely against one type of attack, we consider the ultimate goal of the adversary is to enrich his knowledge on the user. After gaining access to a database which may contain a record that refers to that user, the adversary attempts to match his auxiliary information against the records in the database. In case a record is found that sufficiently matches the auxiliary information, the adversary considers the re-identification successful. If not, the adversary considers that his attack failed, either because the user is indistinguishable or not present in the database.

From an applicability point if view, it is beneficial to measure privacy instead of *guarantee* it for two main reasons:

- for some scenarios the utility cost of *guaranteeing* privacy is simply too high to be applicable;
- for some scenarios it may be enough to improve privacy just to the point that it becomes economically unviable to attack it.

Tailored recommendation scenarios are an example of such cases: current privacy guarantees render entire datasets as useful as aggregate data and the disclosure of a few movie ratings is not critical. In order to fill this gap we introduce Adversary Gain, which, based on some reference re-identification attack, quantifies the adversary reward: in average, how many new attributes will an attack render.

We consider a probabilistic model, enabling us to tolerate error in the adversary's auxiliary information. The base our attack model is the re-identification of Narayanan and Shmatikov [31]. The strength of this attack model is well supported in the original paper, and has been a reference for subsequent theoretical work [28]. One of the reasons why the attack is so successful draws from the common long-tailed support distribution of sparse high-dimensional datasets: there is a small number of attributes that have non-null values for many records while there is a large number of attributes that only have non-null values for a few records. This long-tailed distribution makes records very distinguishable even by merely considering which attributes are defined and which ones aren't. This represents a privacy threat even if attribute values are obfuscated or omitted [31, 1], rendering value obfuscation techniques almost useless since the very existence of a value is often enough to convey the information necessary for a re-identification attack.

A natural metric for studying re-identification attacks in a probabilistic setting is the success probability of the re-identification. This success probability is expected to increase with the increase in auxiliary information, so we present it as a function of the amount of auxiliary information available to the adversary. However, this metric does not capture our notion of privacy breach. A trivial case where there is no privacy breach with a successful re-identification is the one in which the auxiliary information already contains all of the user's attributes that are non-null for the attacked dataset, as the adversary's data pool on the user remains unaltered. Thus, let us define Adversary Gain as our key privacy metric.

Definition 2. An adversary has access to an attacked dataset D, and to auxiliary information aux about a user - a set of ratings that user gave to items possibly present in D. Let I be his re-identification attack function, which outputs the set of ratings present in dataset D correctly identified to belong that user, and an empty set otherwise. Then, let the Adversary Gain (AG) of an attack on dataset D be the number of new attributes regarding that user rendered by the attack:

$$AG(D, aux) = |I(D, aux) \setminus aux|$$

A key benefit of this metric is that it allows us to take an economical look on an adversary's incentives. Bringing AG below certain values will render attacks economically unviable, which should be enough for a variety of practical applications, especially in high-dimensional data scenarios. Assuming the cost of performing one attack is greater than the reward of acquiring one rating, a target value of 1 for AG would be an acceptable value. However, an estimation of acceptable AG values is out of scope of this work, as it would require data on attack cost.

4 Record Fragmentation

4.1 Rationale

Most work in recommender systems attempts to identify similarities between users in order to perform recommendations. The underlying assumption is that if some users rated n items similarly, they will also rate other items similarly [40]. The characteristic that makes recommenders perform well are similarities between users, and it must be possible to process a dataset in a way to leverage those same similarities to make users more indistinguishable in the dataset, achieving better privacy at a very reduced utility cost. In this Section we will present a sanitization method aiming to validate this idea.

In the context of computer communications and networks, the concept of pseudonym has been extensively used to designate a temporary or scoped identifier of a subject [38]. An historic reference to "digital pseudonym" is made in a 1981 paper by Chaum [9] while describing the use of public key cryptography in such a way that it allows users to send verifiable messages while protecting their identity. Subsequently Chaum described the use of digital pseudonyms for interacting with multiple organizations while preventing that these organizations collude in order to build a profile of the user [10]. The pseudonym used with one organization is unlinkable with the one used with another organizations. Furthermore the user can prove the possession of some credentials obtained from one organization to another without revealing the pseudonym he uses to interact with the first. While the typical case is to use one pseudonym per organization the use of one-time pseudonyms, and more generally multiple pseudonyms per organization, is also mentioned.

In this paper we use the concept of pseudonyms in the context of privacy-preserving high-dimensional data publishing. Each record - representing an individual - is split into several records with different identifiers, i.e. pseudonyms, and the values of non-null attributes are distributed among the new records, i.e. fragments. As a direct consequence the linking between different attribute values is broken. No values are changed, inserted

UID	M1	M2	M3	M4	M5
1	3		5	4	
2		2	5	5	1
3	4			1	
4	3		3	2	4
5	4		5		4

(a) Original Dataset

Nym	M1	M2	M3	M4	M5
1	4			1	
2	4				4
3		2			1
4	3				
5			5	5	
6			5		
7			3	2	
8	3				4
9			5	4	

(b) Fragmented Dataset

Nym	UID
1	3
2	5
3	2
4	1
5	2
6	5
7	4
8	4
9	1

(c) Pseudonym Mapping

Figure 1: Record Fragmentation Example

or deleted: sets of values are simply unlinked from each other. Each record is fragmented in several pseudonymous versions of it. The linking of these fragments using pseudonym mapping information restores the original data, thus is to remain unpublished. From a data privacy perspective, record fragmentation is vertical partitioning applied per record. Previous work has employed different forms of vertical partitioning to improve privacy, however it was either applied in the dataset as a whole [47] or to horizontal partitions of the dataset [25, 44]. In our approach we consider each record to be an horizontal partition and apply vertical partitioning independently to each of them. This approach also has similarities with Gkoulalas-Divanis and Loukides' clustering-based anonymization [17]: our fragments are conceptually similar to their clusters, but instead of using generalization, only suitable for high-dimensional itemset datasets, we perform partitioning.

Record fragmentation is illustrated in Figure 1: each record of the original dataset is split in several, forming the sanitized dataset and the mapping between the identifiers of the two datasets. We assume that the sanitized dataset is accessible to the adversary while the mapping dataset is either destroyed, stored securely, or distributed among the users - each user holds the pseudonyms that refer to him.

The choice of which values are presented together and which are separated in different fragments is done based on the statistical properties of the dataset. Following the principles used in condensation approaches [3], it is desirable to keep inter-attribute correlations as much as possible in order to reduce the utility loss. In order to do so, a meaningful distance measure between dataset values is required. However, it has been argued that the distances to the nearest and farthest neighbours from a given target in high-dimensional space is almost the same for a variety of data distributions and distance functions [1] [2] [7]. For that reason some dimensionality reduction technique should be applied before using distance functions. Also, records with greater support can be fragmented more times than records with smaller support, in order to avoid the *new user* problem of recommender systems as much as possible.

On the privacy side, the aim is to reduce the amount of information conveyed by the presence of a (non-null) value. The amount of information is directly related to the frequency of the value: if only a few records have a value assigned for a specific attribute, then that attribute is more distinctive than others. Separating rare occurrences of values is key to make records less distinguishable, increasing resilience to re-identification attacks [28].

4.2 Algorithm

In order to formally describe the algorithm, the matrix model of datasets is used. Let dataset D be an N x M matrix where each row r_i is associated with an individual and each column c_j with an attribute. Record $d_{i,j}$ refers to the value that the individual associated with r_i has for the attribute associated with column c_j.

In order to estimate column distance, so that the fragmentation can be done minimizing the error, matrix factorization was used as a dimensionality reduction technique. In a preprocessing step D is factorized in f features, originating two matrices: the RF N x f matrix, showing the correlation between rows and features, and the CF f x M matrix, with the correlation between features and columns. Also during preprocessing, the support - the number of non-null values of each row or column - is respectively captured in vectors RS and CS. The algorithm then generates the dataset D', an P x M matrix, in which P is the total number of pseudonyms used, greater than N, the original number of individuals. Each row of D' is basically a fragment of an original row r_i of D.

To perform the fragmentation, values are clustered together based on their column characteristics. A number of the lowest support columns for which an original row has values defined are fixed as centroids for each new record. The number of lowest-support columns that are elected as centroids depends on the chosen privacy-utility trade-off parameters. After the centroids are assigned, a simple one-pass value assignment is performed based on a distance measure between the column of the value and the defined centroids. This is a lightweight approach to grouping allowing column-neighbour values to be kept together, especially when compared with possible alternatives which include clustering algorithms like K-Means.

The algorithm starts by iterating over the N rows r_i of D, each generating a number of new rows in D'. Given an original row r_i, the collection of j for which $d_{i,j}$ is non-null is temporarily stored and sorted in ascending order by their cardinality value cs_j. The resulting vector J is used to create the new rows iteratively. In case the cardinality value cs_j of the current iteration is below a certain threshold t, then a new row d'_p is created in D', otherwise the row creation iterations for that original row r_i stops.

Let X be the number of successful iterations, and consequently the number of assigned pseudonyms for original row r_i. For each r_i a X x f temporary centroid matrix CFi is built by assigning the column features cf_j for the X first values of J. Finally the algorithm iterates over the records $d_{i,j}$ of row r_i, assigning each of them to one of the new rows d'_p. For that the feature vector cf_j is considered and its distance is calculated to each of the rows in CFi, which represent the centroids. The record is assigned to the centroid to which it has lowest distance and assigned to the corresponding new row.

The result of the algorithm is dataset D', an P x M matrix such that P $\geq$ N, and that has the same number of non-null records as D. The distance function $dist$ used in our implementation was Euclidean Distance but other distance measures could be considered. Experimenting with different distance measures would likely slightly influence the RMSE,

Algorithm 1 Record Fragmentation Algorithm

```
initialize $D'$;
for all $r_i$ do
    initialize $CFi$;
    $x$ = 0;
    for all $cs_j$ referring to $d_{i,j}$ in $r_i$, by ascending order of values do
        if $cs_j \leq t$ then
            add row $d'_p$ assocaitated with $x$;
            add row $CFi_x$ with the values $cs_j$;
            increment $x$
        else
            break;
        end if
    end for
    for all $d_{i,j}$ in $r_i$ do
        for all $x$ in $CFi_x$ do
            $temp_x = dist(j, CFi_x)$
        end for
        $x_{min} = x$ for the value of $x$ that minimizes $temp_x$
        add record $d_{i,j}$ to row $d'_p$ associated with $x_{min}$;
    end for
end for
randomize row order of $D'$;
```

but the results obtained with Euclidean Distance were enough to demonstrate the potential of this approach, as shown by the results in Section 6. Instead, we considered more relevant to experiment with different thresholds t, since this parameter has a key influence in the privacy-utility trade-off.

We used a function to calculate the threshold value t in each iteration, instead of a fixed value. This allows us to tune the amount of created fragments per row. The considered threshold function takes in 2 parameters. The first is an estimation for a threshold attribute cardinality value, tc, indicating whether a non-null value is considered a rare occurrence. The second is a target number of fragments for a user, np, depending on the cardinality of that user. The function itself is linear: the threshold value is the estimated attribute cardinality when the number of attributed pseudonyms matches the target number of fragments, and it varies with the number of attributed pseudonyms, x.

$$Thr(tc, np, x) = tc * (x/np)$$

The second parameter is itself also a function that maps user cardinality to the target number of fragments. We used a logarithm-based function for it, which can be parametrized in order to allow increasing or reducing the target number of fragments for the same user cardinality, $|u|$, respectively leading to more privacy or more utility. We picked logarithmic over linear because it preserves the long-tail of the user cardinality frequency function, which is characteristic for this kind of datasets, otherwise the application of the algorithm would be trivial to detect. The two parameters, p_1 and p_2, allow varying the $NPseudo$ function, respectively in a linear and in a logarithmic way.

$NPseudo(p_1, p_2, |u|) = p_1 * log(1 + (|u|/p_2))$

The impact of p_1 and p_2 variation in pseudonym attribution is empirically analysed in section 6.1.

5 Experiments Description and Implementation

5.1 Datasets

The objective of the Netflix Prize [33] was to come up with a recommendation algorithm that, trained with a supplied dataset, performed 10% better than the reference algorithm, Cinematch, which scored 0.9514 RMSE on the test dataset. The Grand Prize was won by an aggregate team of previously competing teams called BellKor's Pragmatic Chaos [34]. Their algorithm scored 0.8567 RMSE, a 10.06% improvement on Cinematch.

The Netflix training dataset consists of movie ratings, from 1 to 5, submitted by Netflix users, with the submission date, organized by movie id. The movie ids are sequential and range from 1 to 17770, unlike user ids, which range from 1 to 2649429 but amount to only 480189 distinct values. The dataset accounts for a total of 100480507 ratings, which represents 1.1% of the possible rankings for the considered number of users and movies.

In order to confirm results in different datasets, another freely available well-known movie ratings dataset was considered. Movielens is a dataset made available by the University of Minnesota, crowdsourced via their web site [19], with the purpose of gathering data for research in recommendation systems and providing movie recommendations to users. There are currently three releases of the dataset made available to the public [20] which have different sizes with the biggest having 10000054 ratings - significantly smaller than the 100 million ratings from Netflix. These ratings are given by 69878 users to 10677 movies, which means Movielens is slightly less sparse than Netflix having 1.3% non-null values.

For the experiments described, the Movielens dataset was converted to the Netflix format, so that the same setup and code could be used. This included rounding the ratings up because Netflix supports integer ratings from 1 to 5 while Movielens supports 10 possible values for ratings: 0.5 to 5 with 0.5 steps. Because our goal is simply to compare the recommendation accuracy between the original and resulting datasets, the pre-experiment rounding process isn't an influencing factor.

5.2 Reference Re-Identification Algorithm

Narayanan and Shmatikov presented a generic algorithm for re-identification in sparse datasets and applied it to the Netflix dataset with interesting results [31]. The algorithm has two variations named Scoreboard and Scoreboard-RH, and they both rely in the overwhelmingly low similarity between users of the dataset. The mere information about which movies users rated makes them very distinguishable, especially because of non-mainstream movies which are not rated by many users. Even with incorrect and incomplete auxiliary data or dataset perturbation, it is possible to re-identify many users with a good probability. For this reason, Narayanan and Shmatikov assigned a weight to the different movies depending on the size of their support set, *supp*. Furthermore, they defined similarity between two rows as a kind of cosine similarity: Sim maps a pair of records to the interval $[0, 1]$. The Scoreboard-RH algorithm is a more robust version than Scoreboard and it defines a scoring function $Score$ which assigns a numerical score to each record in the database D based on how well it matches the attacker's auxiliary information over a target user, aux.

$Score(aux, d_i) = \sum_{j \in supp(aux)} wt(j) \times Sim(aux_j, d_{i,j})$

where $wt(j) = \frac{1}{\log |supp(j)|}$

After $Score$ is calculated for all the rows d_i Scoreboard-RH takes the two highest scores and calculates how different they are in relation to the standard deviation. If the value is bigger than a defined "eccentricity" parameter ϕ, then the best match is considered to be a successful re-identification. Otherwise the algorithm outputs no match.

$\frac{max1 - max2}{\sigma} > \phi$

In order to evaluate how susceptible the resulting datasets are to re-identification we implemented our version of Scoreboard-RH. Original work considered that Sim would output 1 on a pair of movies rated by different subscribers if the ratings and dates are within some threshold, and 0 otherwise. For simplicity purposes, and to consider the same data in the utility and privacy analyses, we disregard dates in our implementation of Scoreboard-RH, relying only in the ratings information. Apart from that, we instantiated our version of Scoreboard-RH as did Narayanan and Shmatikov:

- the Sim function will output 1 in case the rating of a movie in the two rows matches and 0 otherwise;
- the eccentricity parameter ϕ is set to 1.5.

We consider the re-identification successful for the fragmented datasets in case the algorithm outputs any pseudonym that refers to the user to which the supplied auxiliary information belongs to. Our Scoreboard-RH implementation uses a random sampling approach to evaluate the re-identification success: auxiliary information of a certain size referring to a random individual from the original data set is randomly sampled and used to match it in a target dataset. The estimation of the results is done for a 95% confidence interval and 2% error margin, so the sampling is repeated according to the number of samples required using the normal approximation for a binomial proportion interval (Wald interval). We found it to require 300 iterations in the best case and above 2400 in the worst case.

5.3 Reference Recommendation Algorithm

Motivated by the Netflix Prize competition, there was source code contributed by contestants that could be used to perform basic operations with the dataset. Two contributions deserved our attention: the Netflix Recommender Framework [29] and, based on it, the Kadri Framework [23]. These frameworks provide functions to efficiently process the Netflix dataset text files, as well as implementations of some recommendation algorithm primitives, namely average, matrix factorization, K-NN and prediction blending. Another function provided by these frameworks is the possibility to scrub the probe data from the dataset: the probe data is removed from the training set, effectively separating the training and test sets, increasing the reliability of the RMSE results.

Using these frameworks, we created a reference prediction algorithm by blending movie average and matrix factorization predictions. These algorithms ignore date information, relying only on movie ratings, similarly to what was done regarding the re-identification algorithm, described in Section 5.2. Our reference prediction algorithm scored 0.921299 RMSE on Netflix, a better result than the original Cinematch algorithm.

5.4 Reference Privacy Preservation Algorithm

To the extent of our knowledge, the pre-existing technique most similar to record fragmentation is disassociation [44]. As explained in Section 4.1, both methods use forms of horizontal and vertical partitioning to fulfill their objective of protecting privacy in high-dimensional datasets. The main difference is that disassociation has k^m-anonymity as its objective, a static privacy guarantee, while record fragmentation has a target number of fragments function, representing the privacy-utility trade-off.

Disassociation was chosen also because k^m-anonymity is one of the most relaxed privacy guarantees defined, and yet not relaxed enough to cope with tailored recommendation scenarios. The partitioning applied with disassociation generates a dataset where the great majority of records have a support of 1 or 2, as explained in the choice of the evaluation parameters [44, p. 952]. This may be enough for associating query some items together but not to perform tailored recommendations. In order to validate this concern, we implemented the disassociation algorithms, VERPART and HORPART [44], targeted at achieving the most relaxed privacy guarantee possible with that method: 2^2-anonymity. Because we set $m = 2$, we didn't need to implement the REFINE algorithm, as it only has an impact for $m > 2$.

5.5 Implementation Detail

A Netfix Commons library was created in Java to provide quick and memory-efficient access to the dataset. Heuristics were used to improve access times to data without requiring more memory for the representation. Both our Scoreboard-RH, Record Fragmentation and disassociation [44] implementations use this library.

The preprocessing and main steps of the algorithm described in Section 4.2 were implemented separately. The preprocessing step that involves matrix factorization and attribute cardinality count, which in the Netflix case is the number of ratings per user, was implemented in C++ using the Kadri Framework. This generates 3 text files each containing a matrix: user-features, movie-features and user ratings. The main step of the algorithm was implemented in Java. It requires access to the 3 text files from preprocessing and to the original data set, and it generates the fragmented version of the dataset in the same format as the original one, a probe file in accordance to the resulting dataset, and a pseudonym mapping file to match the attributed pseudonyms to the original user ids for evaluation purposes. For performance purposes, a safe cardinality value for movies was introduced in the implementation. If the movie has a number of ratings above this value then it is not considered to be a centroid. This significantly reduces the number of movies that we need to sort by cardinality, consequently reducing the time our quicksort takes.

All the implemented source code (record fragmentation, disassociation, rating prediction and RMSE calculation) is available on GitHub [18].

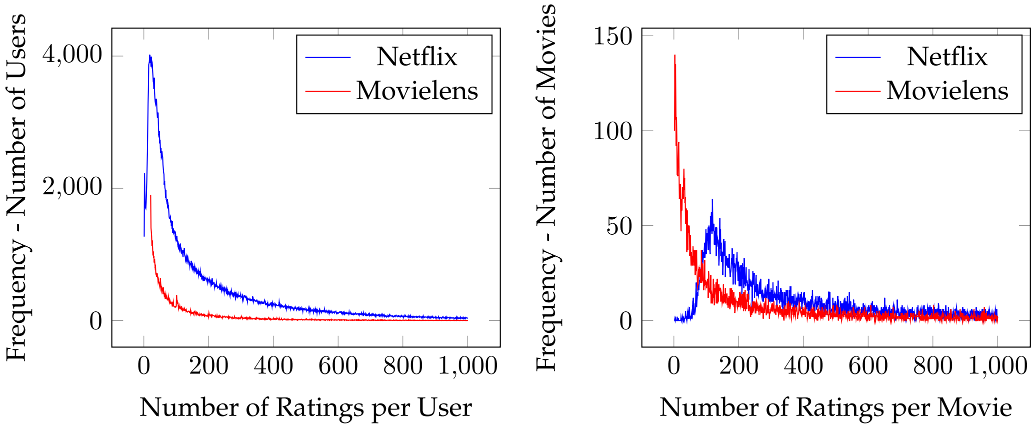

Figure 2: Support Analysis of Netflix and Movielens Datasets

6 Results

6.1 Partitioning Analysis

Support analysis to both datasets shows similar long tail patterns both regarding users and movies: a few movies/users have many (non-null) ratings while the majority of users/movies have only a few ratings, as shown in Figure 2. Netflix shows frequency maxima in users and movies for low cardinalities: 18 and 119 respectively. Movielens has their maximum frequency values for the lowest cardinalities possible: 20 for users and 1 for movies - the Movielens site imposes that each user rates at least 20 movies on registration so that it can deliver meaningful recommendations, avoiding the *new user* problem.

The algorithm preprocessing step described in 4.2 was run only once on each of the original datasets to create the required matrix files. Then the main algorithm was run several times, each of the runs generating a fragmented version of the original dataset. Different privacy-utility trade-off parameters were used in each run, resulting in different fragmentation levels. As described in Section 4.2, the considered trade-off parameters were:

- the movie cardinality safe value, until which movies are considered to initialize group centroids;
- the target movie cardinality value, the last value for which a movie is elected as a centroid in case the user was assigned exactly the target number of pseudonyms;
- the target number of fragments function, for which we assumed different parameters for the logarithm function.

The first two parameters were set based on the movie cardinality of the dataset. The movie cardinality safe value was set to a conservative value of 5000, merely for improving the algorithm processing times, expected not to influence the number of created fragments. The target movie cardinality value was set to 500 by looking at the Netflix movie cardinality distribution, as it represents the start of the long tail and divides the movie domain in half: approximately 48% of the movies have less than 500 ratings. In order to simplify, since both an increase in the target movie cardinality and a linear increase on the target fragments

Dataset	Netflix	Movielens
Original	480189	69878
Frag1	746153	157421
Frag2	1441715	330961
Frag3	3057962	735668
Frag4	5039491	1324378
Frag5	7292946	2008289
Frag6	9111456	2676293
2Dis	82685159	8437233

Table 3: Number of Rows in the Original and Fragmented Datasets

is equivalent, we consider these first two parameters to be fixed and we vary the target number of fragments function for the trade-off.

The fragmentation experiments were run on a virtual machine with 1 virtual core and 4GB of RAM allocated, running Java HotSpot VM on Ubuntu Linux, hosted by a Intel Core i7 machine. The measured run-times for the Netflix dataset were approximately 10 minutes for the least aggressive fragmentation and 1 hour for the most aggressive one. For the Movielens dataset the run-times were from 2 to 11 minutes. These results confirm the expected linear run-time scalability with respect to the target number of fragments, as well as an overhead related to the size of the dataset for input/output operations. In the same conditions, our implementation of 2^2-anonymous disassociation took orders of magnitude longer to be completed: approximately 9 hours for Movielens and 95 hours for Netflix. Note that these results suggest that the algorithm run-time grows linearly with the number of ratings, as Netflix has approximately 10 times more ratings than Movielens, but conclusive results would require further work.

The target number of fragments function, as described in Section 4.2 is based on a logarithm function to which two parameters are applied, one allowing to vary the function linearly and other logarithmically. Six fragmented datasets were generated from applying the algorithm with different parameters, and numbered according to their aggressiveness, being 1 the most utility-friendly and 6 the most privacy friendly. Table 3 shows the number of rows of the resulting datasets, both for the Netflix and Movielens cases, compared with the number of rows resulting from 2^2-anonymous Disassociation. As expected (see Section 5.4) the number of rows generated by 2^2-anonymous disassociation comes very close to the number of total ratings. The average row support of 2^2disassociated Netflix is 1.215, and for Movielens is 1.185, illustrating the strictness of the method.

6.2 Utility Evaluation

In order to evaluate the results in terms of utility, the recommendation algorithm implementation described in Section 5.3 was run in the original and fragmented datasets. As explained in Section 3.1, the RMSE value itself does not convey a direct understanding on the utility loss in the context of tailored recommendations. For that reason the Tailored Utility (μ) metric is used. In order to calculate it, we consider the naïve prediction function to be the average movie rating, and take the following RMSE values as reference: 1.05282 for Netflix and 0.946021 for Movielens. These RMSE values of the naïve prediction are the same for all datasets because both record fragmentation and Terrovitis' Disassociation both rely exclusively on partitioning, leaving the ratings themselves and their association

Dataset	Netflix		Movielens	
	RMSE	μ	RMSE	μ
Average	1.05282	0	0.946021	0
Original	0.921299	0.124923	0.874797	0.075288
Frag1	0.931068	0.115643	0.855595	0.095585
Frag2	0.944055	0.103308	0.850197	0.101292
Frag3	0.978027	0.071041	0.852461	0.098899
Frag4	0.979276	0.069854	0.862424	0.088368
Frag5	0.981908	0.067355	0.869907	0.080458
Frag6	0.985127	0.064297	0.875914	0.074108
2Dis	1.053909	-0.001034	0.948689	-0.00282

Table 4: RMSE of the Predictions for the Different Datasets

to movies unaltered.

The RMSE results, as well as the our utility metric μ, are depicted in Table 4. It can be seen that, generally, the higher the fragmentation, the more significant is the utility loss. However this is not true for the more utility-friendly Movielens generated datasets, where a RMSE reduction is observed. This is explained by a side-effect of the algorithm and chosen metrics. Similarly to what is observed for the condensation method, where the classification accuracy improves due to a noise reduction effect [3], in the fragmentation case keeping nearest movies together initially increases prediction accuracy. The minimum error is reached at certain fragmentation level, where the information being removed from the dataset is no longer mostly noise and starts to be useful information, observed to be close the average of 5 fragments per record. After that point the utility-improvement effect fades as the fragmentation becomes more aggressive, exhibiting degraded utility at the most fragmented case tested. Disassociation, because it strictly enforces a privacy guarantee, even its most relaxed instantiation, 2^2-anonymity completely destroys μ. Otherwise successful prediction algorithms become less useful, when applied to a disassociated dataset, than considering the movie average for prediction.

6.3 Re-Identification Evaluation

To evaluate the risk of re-identification, Scoreboard-RH was run using the implementation described in Section 5.2, with the auxiliary information being built randomly from the original dataset. This enables the evaluation of re-identification success with increasing sizes of auxiliary information, at the cost of some generality: for higher values of auxiliary information the random sampling becomes skewed as not all users can be considered, only the ones that have at least the required number of ratings.

The re-identification success of the fragmented datasets behaves similarly as it does on the original dataset, as seen in Figure 3: re-identification success rises rapidly as available auxiliary information size increases. The re-identification success reaches its maxima and stabilizes for auxiliary information sizes of 20 to 25 for all the versions of the Netflix dataset. The difference is the maxima value: while for the original dataset the success rate reaches 100%, as previously shown by Narayanan and Shmatikov, for the fragmented ones the maxima is lower, depending on how aggressive fragmentation was.

Similar behaviour is observed for the original Movielens dataset (Figure 4), with Scoreboard-RH reaching 100% re-identification success for the same values of auxiliary information

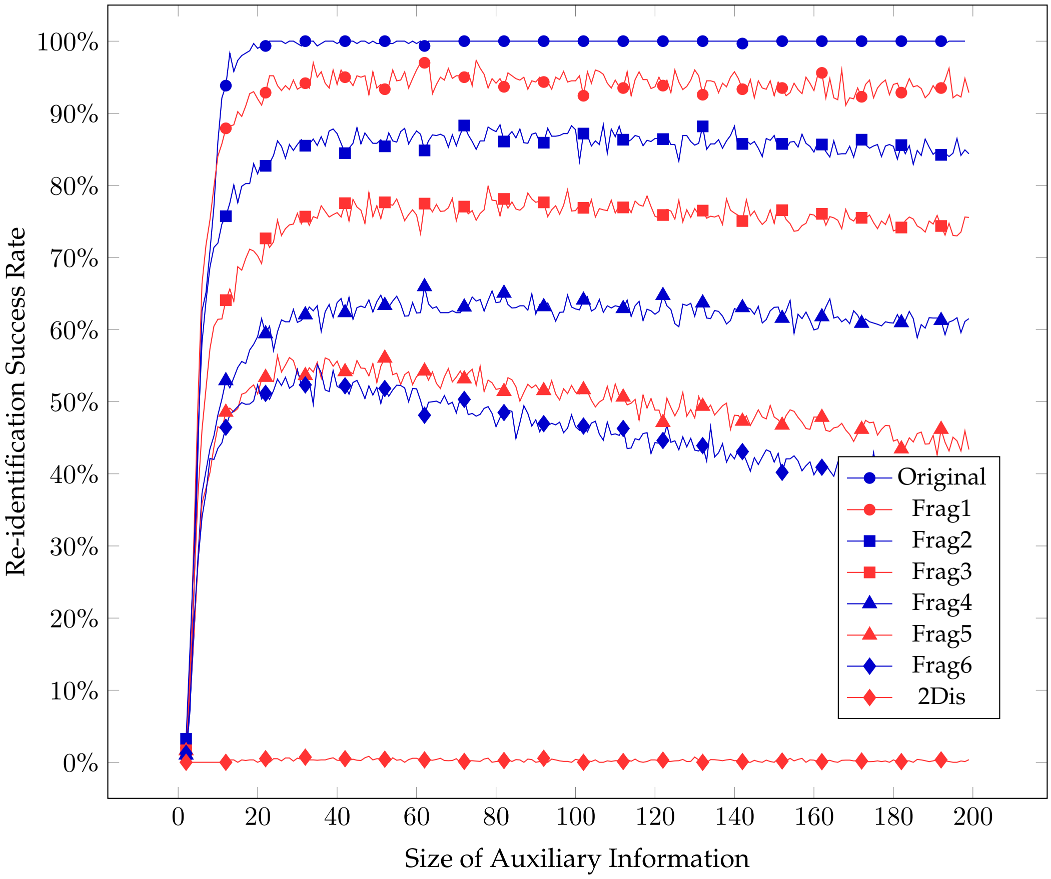

Figure 3: Re-identification Success Rate on Original and Generated Netflix Datasets by Size of Available Auxiliary Information

size. The fragmented datasets also exhibit comparable behaviour with increases in re-identification success for increasing size of auxiliary information until the maxima are reached at approximately the same values. However in this case, as fragmentation increases, the success rate doesn't plateau near the maxima but decreases instead: more auxiliary information apparently leads to less re-identification success. This is originated by the skew effect previously referred: auxiliary information generation is more skewed towards users with more ratings for higher sizes. The decrease of re-identification success with the increase of auxiliary information size merely shows that users that originally had more ratings are better protected against re-identification. Although this can be observed more clearly in Movielens fragmented datasets, it is also noticeable in Netflix's most aggressively fragmented datasets.

Because 2^2-anonymity only guarantees protection against an adversary with knowledge of at most 2 items, as the auxiliary information size increases the probability of it containing more than 2 non-disassociated items also increases. Consequently, the 2^2-disassociated dataset behaves in an inverse manner to the fragmented ones, with re-identification probability steadily rising with increases in auxiliary information, but at very low values.

Because of the auxiliary information sampling skew effect, we estimate the real re-identi-

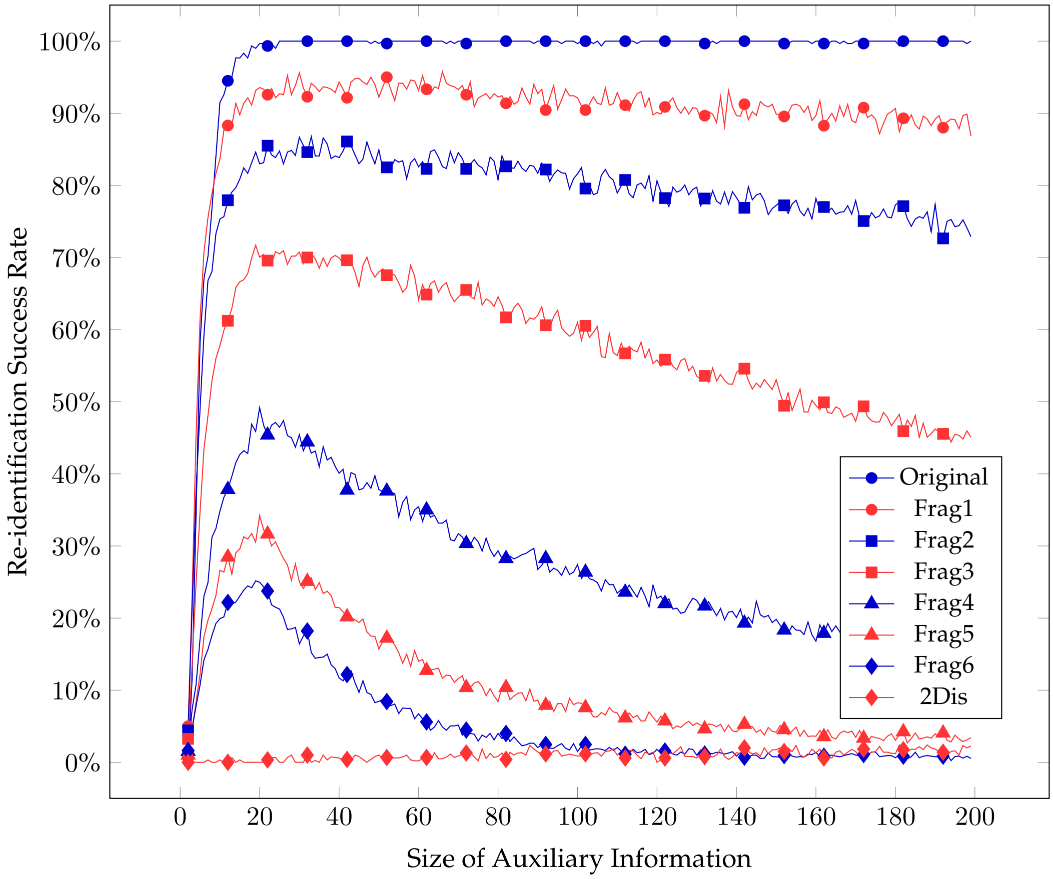

Figure 4: Re-identification Success Rate on Original and Generated Movielens Datasets by Size of Available Auxiliary Information

fication success based on the values for the auxiliary information size interval where global success maxima are found for most aggressively fragmented datasets - 20 to 25 items. This maxima interval coincides with the re-identification success plateuing for the least aggressively fragmented datasets. In Figure 5 the re-identification results - success, inconclusive or wrong result - for each of the datasets in the success maxima interval are shown. Inconclusive results grow faster for low fragmentation datasets and then stabilize around 30%. Scoreboard-RH shows significant resilience to wrong results for low fragmentation datasets, but as fragmentation increases wrong outputs become more noticeable, especially after inconclusive results stabilize. Netflix and Movielens datasets show similar behaviour, with the Movielens dataset showing itself as more privacy-friendly than Netflix.

Unsurprisingly, 2^2-disassociation is extremely effective, returning an inconclusive result around 95% of the times for both datasets. However, Scoreboard-RH does manage to successfully re-identify the target in a few occurrences, demonstrating its strength. Furthermore, in real-life scenarios it may be beneficial that the adversary isn't able to trivially detect the use of a sanitization algorithm. While AG shows no difference between wrong and inconclusive results, from an economical point of view it's worse for an adversary to have false positives (wrong results) than true negatives (inconclusive results) [22].

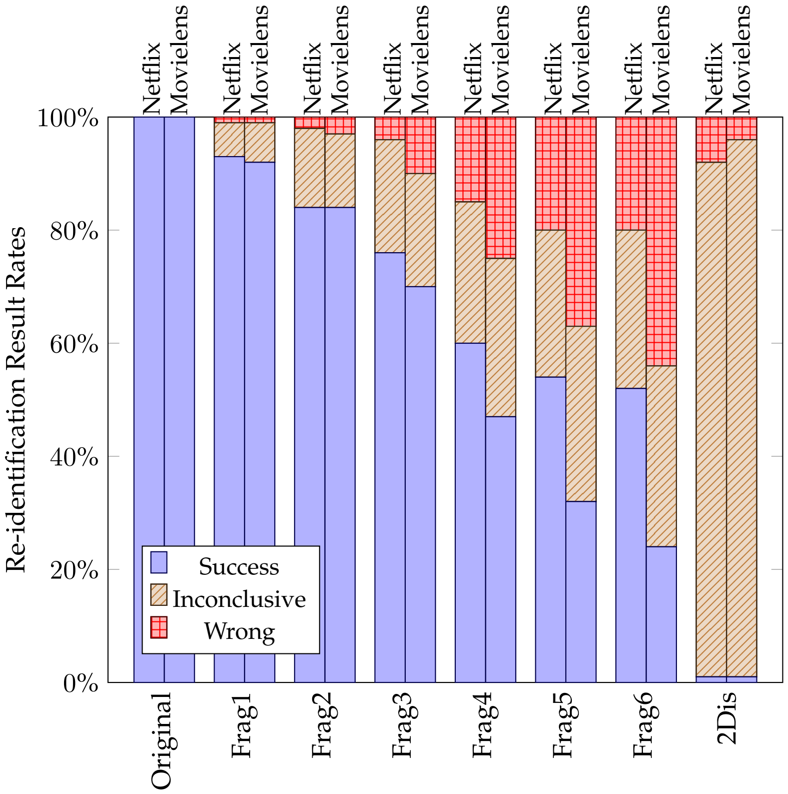

Figure 5: Re-identification Results at Success Maxima Interval per Dataset

6.4 Adversary Gain Evaluation

As argued in Section 3.2, because re-identification rate is an incomplete privacy metric, we use it to calculate the AG metric. This metric estimates the worth of the attack from the adversary's point of view - higher values mean less privacy. The goal of a sanitization algorithm doesn't have to be no AG, but a low enough value that makes attacks economically unviable.

Figure 6 shows the AG significantly decreases for fragmented datasets, especially for the Movielens case. Generally, relative to the original case, the reduction in AG is significantly greater than the reduction of μ for the user. This indicates that record fragmentation has a positive effect on the overall privacy-utility trade-off. 2^2-disassociation renders residual AG, as positive AG can only occur for successful re-identification of rows with more than 2 items, which comprise of a very small fraction of the disassociated dataset.

7 Conclusion

Previous work in privacy preserving data publishing for the high-dimensional case is currently not useful for tailored recommendation scenarios. Existing privacy guarantees destroy the benefits that can be harnessed by publishing datasets because recommendations of similar utility are possible by simply accessing aggregate data. However, the user pat-

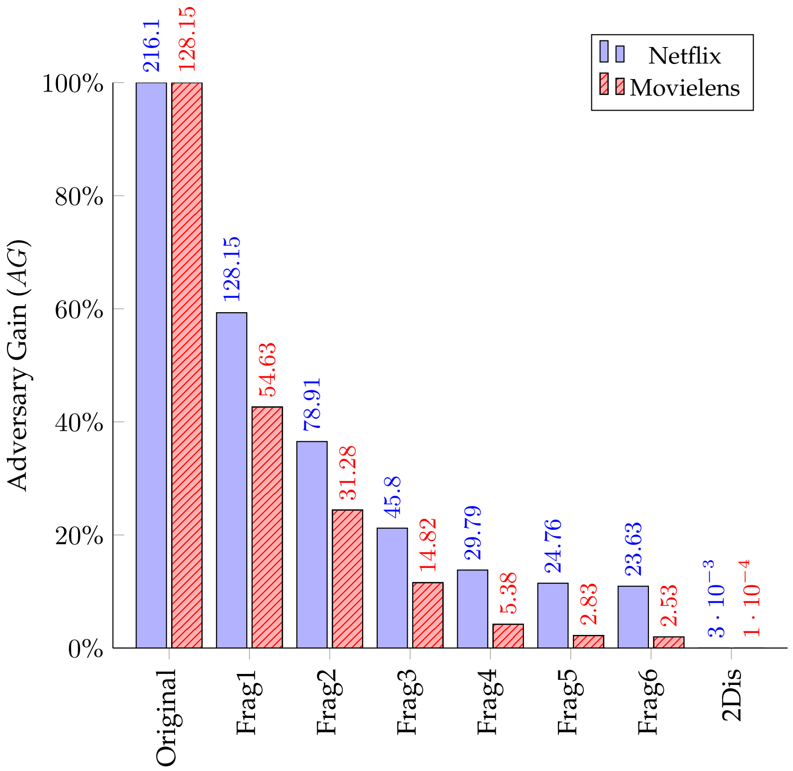

Figure 6: Adversary Gain (AG) at Success Maxima Interval per Dataset

terns that make tailored recommendations possible may also provide some degree of privacy protection to those users. Driven by this idea, we present a new utility metric, to be used in tailored recommendation scenarios, and a privacy metric instead of a static guarantee. Also we present a sanitization method that relies on per-record vertical partitioning, inspired by distributed systems work in pseudonyms, that aims to validate that is possible to significantly improve privacy while maintaining tailoring capabilities.

The metrics were defined with applicability in mind. Tailoring Utility measures how well a database-algorithm pair adapts tailored recommendations to the preferences of individual users, compared to generic popularity-based recommendations. Adversary Gain quantifies the adversary reward per attack: in average, how many new attributes will an attack render.

The presented sanitization method limits the probability of re-identification attack success independently of the available auxiliary information size. The method performs well regarding both metrics, reducing very significantly Adversary Gain at a reduced Tailoring Utility cost. The method is completely truthful: all values are unchanged (no generalization or noise), there are no artificial values included in the result (no synthetic data), and no original value is omitted (no suppression). Instead it unlinks sets of values that belong to the same record, fragmenting them into several records based on simple metrics of privacy and utility optimization: separation of rare occurrences and link preservation of "neighbour" attributes, based on some distance function. While similar fragmentation methods are employed in related work, some methods rely in the quasi-identifier assumption [16, 25], and the others aim to achieve privacy guarantees that are too destructive for tailored recommendations [12, 43, 44, 47].

The pseudonym mapping which results from the process can be destroyed or stored in a secure, inaccessible location, because it isn't required to perform recommendations. It can also be distributed among the users' recommendation clients, enabling the combination of several estimations belonging to pseudonyms of the same user locally. Possible directions for future work include the development of an interactive version of the algorithm, enabling new ratings to be added to a sanitized dataset, as well as a mathematical analysis of the re-identification limits under the use of record fragmentation, analogous to what Parra-Arnau, Rebollo and Forn [36] did for suppression and forgery.

Acknowledgements

We are grateful to Mário Antunes for the useful discussions about clustering methods, matrix factorization and distances. The work presented in this paper was partially supported by the European Commission via the ICT FP7 SOCIETIES Integrated Project (No. 257493).

References

[1] Charu C. Aggarwal. On k-anonymity and the curse of dimensionality. In *31st International Conference on Very Large Databases (VLDB)*, pages 901–909, 2005.

[2] Charu C. Aggarwal, Alexander Hinneburg, and David A. Keim. On the surprising behavior of distance metrics in high dimensional space. In *8th International Conference on Database Theory (ICDT)*, pages 420–434, 2001.

[3] Charu C. Aggarwal and Philip Yu. A condensation approach to privacy preserving data mining. *Advances in Database Technology - EDBT 2004*, pages 183–199, 2004.

[4] Dakshi Agrawal and Charu C. Aggarwal. On the design and quantification of privacy preserving data mining algorithms. In *20th ACM Symposium on Principles of Database Systems (SIGMOD-SIGACT-SIGART)*, pages 247–255, 2001.

[5] Michael Barbaro and Tom Zeller, Jr. A Face Is Exposed for AOL Searcher No. 4417749. http://select.nytimes.com/gst/abstract.html?res=F10612FC345B0C7A8CDDA10894DE404482,2006. Accessed: 23-11-2015.

[6] Shlomo Berkovsky, Yaniv Eytani, Tsvi Kuflik, and Francesco Ricci. Enhancing privacy and preserving accuracy of a distributed collaborative filtering. In *Proceedings of the 2007 ACM conference on Recommender systems*, 2007.

[7] Kevin Beyer, Jonathan Goldstein, Raghu Ramakrishnan, and Uri Shaft. When is nearest neighbor meaningful? In *7th International Conference on Database Theory (ICDT)*, pages 217–235, 1999.

[8] John Canny. Collaborative Filtering with Privacy. In *Proceedings of the IEEE Symposium on Security and Privacy*, 2002.

[9] David L. Chaum. Untraceable electronic mail, return addresses, and digital pseudonyms. *Communications of the ACM*, 24(2):84–90, 1981.

[10] David L. Chaum. Security without identification: Transaction systems to make big brother obsolete. *Communications of the ACM*, 28(10):1030–1044, 1985.

[11] Shuchi Chawla, Cynthia Dwork, Frank McSherry, Adam Smith, and Hoeteck Wee. Toward Privacy in Public Databases. *Theory of Cryptography*, pages 363–385, 2005.

[12] Rui Chen, Noman Mohammed, Benjamin CM Fung, Bipin C. Desai, and Li Xiong. Publishing set-valued data via differential privacy. *Proceedings of the VLDB Endowment*, 4(11):1087–1098, 2011.

[13] Cynthia Dwork. Differential privacy. *Automata, languages and programming*, pages 1–12, 2006.

[14] Federal Trade Commission. Closing Letter to Reed Freeman, Esq., Counsel for Netflix, Inc. Technical report, Federal Trade Commission, 2010. Accessed: 23-11-2015.

[15] Benjamin Fung, Ke Wang, Rui Chen, and Philip Yu. Privacy-Preserving Data Publishing: A Survey on Recent Developments. *ACM Computing Surveys (CSUR)*, 2010.

[16] Gabriel Ghinita, Yufei Tao, and Panos Kalnis. On the Anonymization of Sparse High-Dimensional Data. In *IEEE 24th International Conference on Data Engineering (ICDE)*, pages 715–724, 2008.

[17] Aris Gkoulalas-Divanis and Grigorios Loukides. Utility-guided Clustering-based Transaction Data Anonymization. *Transactions on Data Privacy*, 5:223–251, 2012.

[18] João Miguel Gonçalves. jmgoncalves/netflix-pseudonymizer. `https://github.com/jmgoncalves/netflix-pseudonymizer`, 2013. Accessed: 23-11-2015.

[19] GroupLens. MovieLens: movie recommendations. `http://movielens.umn.edu/login`, 1997. Accessed: 23-11-2015.

[20] GroupLens. MovieLens Data Sets. `http://grouplens.org/datasets/movielens/`, 2011. Accessed: 23-11-2015.

[21] Asela Gunawardana and Guy Shani. A survey of accuracy evaluation metrics of recommendation tasks. *The Journal of Machine Learning Research*, 10:2935–2962, 2009.

[22] Cormac Herley. Why do nigerian scammers say they are from nigeria? In *Workshop on the Economics of Information Security*, 2012.

[23] Saqib Kadri. Kadri Framework C++ Source Code (Pre-Processing, Dates, Blending). `http://www.netflixprize.com/community/viewtopic.php?pid=9202`, 2008. Accessed: 23-11-2015.

[24] Ninghui Li, Tiancheng Li, and Suresh Venkatasubramanian. t-closeness: Privacy beyond k-anonymity and l-diversity. In *IEEE 23rd International Conference on Data Engineering (ICDE)*, pages 106–115, 2007.

[25] Tiancheng Li, Ninghui Li, Jian Zhang, and Ian Molloy. Slicing: A New Approach for Privacy Preserving Data Publishing. *IEEE Transactions on Knowledge and Data Engineering*, 24(3):561–574, 2012.

[26] Ashwin Machanavajjhala, Daniel Kifer, Johannes Gehrke, and Muthuramakrishnan Venkitasubramaniam. l-diversity: Privacy beyond k-anonymity. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 1(1):3, 2007.

[27] Frank McSherry and Ilya Mironov. Differentially private recommender systems: building privacy into the net. In *15th ACM International Conference on Knowledge Discovery and Data Mining (ACM SIGKDD)*, pages 627–636, 2009.

[28] Martin M. Merener. Theoretical results on de-anonymization via linkage attacks. *Transactions on Data Privacy*, 5:377–402, 2012.

[29] Benjamin Meyer. Netflix Recommender Framework. `http://www.netflixprize.com/community/viewtopic.php?id=352`, 2006. Accessed: 23-11-2015.

[30] Adam Meyerson and Ryan Williams. On the complexity of optimal k-anonymity. In *Proceedings of the twenty-third ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems*, pages 223–228, 2004.

[31] Arvind Narayanan and Vitaly Shmatikov. Robust De-anonymization of Large Sparse Datasets. *IEEE Symposium on Security and Privacy (SP)*, pages 111–125, 2008.

[32] Arvind Narayanan and Vitaly Shmatikov. Myths and fallacies of personally identifiable infor-

mation. *Communications of the ACM*, 53(6):24–26, 2010.

[33] Netflix. Netflix Prize: Home. `http://www.netflixprize.com/`, 2006. Accessed: 23-11-2015.

[34] Netflix. Grand Prize awarded to team BellKors Pragmatic Chaos. `http://www.netflixprize.com/community/viewtopic.php?id=1537`, 2009. Accessed: 23-11-2015.

[35] Murat Okkalioglu, Mehmet Koc, and Huseyin Polat. On the Privacy of Horizontally Partitioned Binary Data-based Privacy-Preserving Collaborative Filtering. In *10th International Workshop on Data Privacy Management*, 2015.

[36] Javier Parra-Arnau, David Rebollo-Monedero, and Jordi Forné. Optimal forgery and suppression of ratings for privacy enhancement in recommendation systems. *Entropy*, 16(3):1586–1631, 2014.

[37] Manas A. Pathak and Bhiksha Raj. Efficient Protocols for Principal Eigenvector Computation over Private Data. *Transactions on Data Privacy*, 4:129–146, 2011.

[38] Andreas Pfitzmann and Marit Hansen. A terminology for talking about privacy by data minimization: Anonymity, Unlinkability, Undetectability, Unobservability, Pseudonymity, and Identity Management. `http://dud.inf.tu-dresden.de/Anon_Terminology.shtml`, 2010. Accessed: 23-11-2015.

[39] Huseyin Polat and Wenliang Du. Privacy-preserving collaborative filtering using randomized perturbation techniques. In *Proceedings of the Third IEEE International Conference on Data Mining*, 2003.

[40] Xiaoyuan Su and Taghi M. Khoshgoftaar. A survey of collaborative filtering techniques. *Advances in artificial intelligence*, 4, 2009.

[41] Latanya Sweeney. Simple demographics often identify people uniquely. *Health (San Francisco)*, 671:1–34, 2000.

[42] Latanya Sweeney. k-anonymity: A model for protecting privacy. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 10(05):557–570, 2002.

[43] Manolis Terrovitis, Nikos Mamoulis, and Panos Kalnis. Privacy-preserving anonymization of set-valued data. *Proceedings of the VLDB Endowment*, 1(1):115–125, 2008.

[44] Manolis Terrovitis, Nikos Mamoulis, John Liagouris, and Spiros Skiadopoulos. Privacy preservation by disassociation. *Proceedings of the VLDB Endowment*, 5(10):944–955, 2012.

[45] Yabo Xu, Benjamin Fung, Ke Wang, Ada Fu, and Jian Pei. Publishing Sensitive Transactions for Itemset Utility. In *IEEE 8th International Conference on Data Mining (ICDM)*, pages 1109–1114, 2008.

[46] Yabu Xu, Ke Wang, Ada Fu, and Philip Yu. Anonymizing Transaction Databases for Publication. In *14th ACM International Conference on Knowledge Discovery and Data Mining (SIGKDD)*, pages 767–775, 2008.

[47] Hessam Zakerzadeh, Charu C. Aggarwal, and Ken Barker. Towards Breaking the Curse of Dimensionality for High-Dimensional Privacy. In *SIAM International Conference on Data Mining (SDM)*, 2014.

[48] Justin Zhan, Chia-Lung Hsieh, I-Cheng Wang, Tsan-Sheng Hsu, Churn-Jung Liau, and Da-Wei Wang. Privacy-preserving collaborative recommender systems. *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, 40(4):472–476, 2010.