User Model User-Adap Inter (2016) 26:425–458
DOI 10.1007/s11257-016-9177-7

A differential privacy framework for matrix factorization recommender systems

Arik Friedman[1] · **Shlomo Berkovsky**[2] · **Mohamed Ali Kaafar**[2]

Received: 11 April 2016 / Accepted in revised form: 24 July 2016 /
Published online: 16 August 2016

Abstract Recommender systems rely on personal information about user behavior for the recommendation generation purposes. Thus, they inherently have the potential to hamper user privacy and disclose sensitive information. Several works studied how neighborhood-based recommendation methods can incorporate user privacy protection. However, privacy preserving latent factor models, in particular, those represented by matrix factorization techniques, the state-of-the-art in recommender systems, have received little attention. In this paper, we address the problem of privacy preserving matrix factorization by utilizing differential privacy, a rigorous and provable approach to privacy in statistical databases. We propose a generic framework and evaluate several ways, in which differential privacy can be applied to matrix factorization. By doing so, we specifically address the privacy-accuracy trade-off offered by each of the algorithms. We show that, of all the algorithms considered, input perturbation results in the best recommendation accuracy, while guaranteeing a solid level of privacy protection against attacks that aim to gain knowledge about either specific user ratings or even the existence of these ratings. Our analysis additionally highlights the system aspects that should be addressed when applying differential privacy in practice, and when considering potential privacy preserving solutions.

✉ Shlomo Berkovsky
shlomo.berkovsky@data61.csiro.au

Arik Friedman
arik.friedman@gmail.com

Mohamed Ali Kaafar
dali.kaafar@data61.csiro.au

[1] Mobile Systems Research Group, National ICT Australia (NICTA), Sydney, Australia

[2] Data61, Commonwealth Scientific and Industrial Research Organisation (CSIRO), Eveleigh, Australia

1 Introduction

In the last decade, personalization technologies and recommender systems have become a fundamental tool in online applications and services. Recommenders are leveraged to improve sales (e.g., by retail Web sites like Amazon or eBay), to enhance user experience with online services, and to simplify decision making and information access for Web sites and services, like Netflix and Last.fm (Ricci et al. 2015). Typically, recommender systems utilize user information encapsulated in the user models, in order to recommend items that users are likely to consume, i.e., predict which items would be preferred by users and prioritize these when a user interacts with the system.

One of the dominant approaches in recommender systems is *collaborative filtering*, which predicts user preferences by utilizing user-generated signals that were gathered from other users of the system. Such signals may consist of implicit feedback, e.g., past user behavior or browsing logs, or explicit feedback, e.g., feedback on past services or product ratings, which reflect whether a user was interested in an item. Collaborative recommendation methods can be partitioned into two families. *Neighborhood methods* use the collected signals to learn correlations between items or between users (Ning et al. 2015). Item-to-item correlations can then be used to predict that a user will like items similar to those that the user liked in the past. Alternately, user-to-user correlations can be used to predict that the user will be interested in items that attracted the interest of similar users. *Latent factor models* offer a solid alternative to neighborhood methods (Koren and Bell 2015). These approaches process the collected signals and generate latent vectors that characterize users and items with respect to a small number of factors.

In particular, *Matrix Factorization* methods (Koren et al. 2009) have become the prominent technique to infer latent factor models: a matrix of user ratings is factorized into two low-dimensional matrices, which capture latent factors of users and items, respectively. Essentially, a good recommendation reflects a high degree of correspondence between an item's and a user's latent factors. Matrix factorization techniques have been shown to provide a higher predictive accuracy than the neighborhood methods, they are computationally cheaper, and they are easy to extend, e.g., by taking into account temporal effects or inputs with varying confidence levels (Koren and Bell 2015). Therefore, matrix factorization techniques have become the state-of-the-art technique in collaborative filtering recommender systems.

Since collaborative recommenders rely on personal user information to generate recommendations to users, they inherently raise legitimate privacy concerns related to the misuse of the collected data for the purpose of inferring users' sensitive information (Friedman et al. 2015; Jeckmans et al. 2013; Lam et al. 2006). The raw rating data used by memory-based collaborative recommendation methods, even if anonymized, poses an immediate privacy risk: it can be de-anonymized using auxiliary information obtained from other sources, e.g., ratings provided on other sites (Narayanan and Shmatikov 2008). In turn, this information can be used to infer personal information, like gender, political views, or other potentially sensitive data (Netflix xxxx; Kosinski et al. 2013; Weinsberg et al. 2012). Even without the direct access to user ratings, personal information could be inferred from the recommendations provided by the system

to other users, public lists of relevant items, or item-to-item covariance matrix (Calandrino et al. 2011). Although model-based recommendation methods were shown to be more robust than their memory-based counterparts (Bilge et al. 2014), exposure of certain parameters of the model was used to design specific and highly-effective attacks (Cheng and Hurley 2009). These inherent privacy risks of recommender systems have motivated an increasing research of the privacy-personalization trade-off in general, and of privacy-preserving recommender systems in particular. However, this body of research has primarily focused on memory-based recommendation methods (Berkovsky et al. 2012; Machanavajjhala et al. 2011; Parameswaran and Blough 2007; McSherry and Mironov 2009), and little prior work thoroughly investigated privacy preserving model-based recommenders (McSherry and Mironov 2009; Nikolaenko et al. 2013).

In this paper, we extend our earlier work (Berlioz et al. 2015), and target the problem of developing a sound privacy preserving matrix factorization mechanism utilizing the concept of differential privacy (Dwork et al. 2006). Differential privacy is a rigorous and provable approach to privacy in statistical databases, which has already been studied in several instances of collaborative filtering recommendation methods (Machanavajjhala et al. 2011; McSherry and Mironov 2009). While differential privacy sets constraints on privacy preserving computations, different algorithms may conduct the private computation in various ways that would result in different privacy-personalization trade-offs. We propose a number of approaches to modify the matrix factorization mechanism, so that it maintains differential privacy guarantees, under the assumption that a user wishes to protect either the values of specific item ratings, or the mere existence of such ratings, as both can potentially leak private information.

We study the privacy guarantees that can be achieved by the following approaches: (i) by obfuscating the input data before applying the matrix factorization algorithm; (ii) by adding noise within the stochastic gradient descent solver of the matrix factorization problem; and (iii) by obfuscating the output of an alternating least squares matrix factorization mechanism. For all the proposed approaches, we both provide a theoretical analysis of the (calibrated) noise level introduced by the algorithms, and evaluate the resulting privacy-personalization trade-offs by observing the effect of the noise on the accuracy of the generated rating predictions. We present our approaches in a great detail and evaluate them using a number of publicly accessible recommender systems' datasets, which allows transparency and reproducibility.

The contributions of our work are, therefore, as follows. We provide an analysis and experimental evaluation of three **differentially private matrix factorization** mechanisms. Our results show that, for the MovieLens and Netflix datasets used in our evaluation, the best performing method that results in the highest prediction accuracy while still ensuring privacy protection, is the one that obfuscates data at the input of the recommendation process. We further conduct an **investigation of the design choices** that affect the resulting privacy-personalization trade-off, showing the impact of the pre-processing of the data, how characteristics of the dataset (size, density, number of inputs per user/item) affect the choice of the algorithm, and the influence of the privacy constraints on the parameter tuning decisions. Finally, we **experimentally compare the predictive accuracy** of recommendations generated by a differentially

private matrix factorization system with that of a privacy-preserving neighborhood based method. The obtained results demonstrate that neighborhood methods are more resilient to the noise introduced by the privacy preserving algorithm, and are, therefore, more appropriate when a high level of privacy protection is required. However, when weaker privacy levels are acceptable, privacy preserving matrix factorization techniques can achieve higher predictive accuracy levels than those achievable with their neighborhood methods counterparts.

The paper is organized as follows. Section 2 surveys prior work on privacy in recommender systems. Section 3 provides background on matrix factorization techniques and differential privacy. Then, we outline in Sect. 4 several ways to apply differential privacy to matrix factorization. Section 5 describes the evaluation of each implemented method with a number of datasets. In Sect. 6, we discuss the broader implications that the presented results have for the application of differential privacy to recommender systems. Finally, Sect. 7 concludes our work.

2 Related work

User personalization and recommender systems inherently bring to the fore the issue of user privacy (Kobsa 2007). Privacy hazards in recommender systems are aggravated by the fact that generation of high-quality recommendations requires large amounts of personal user data. For instance, the accuracy of collaborative filtering recommendations was shown to correlate with both the number of users in the system and the number of their ratings (Sarwar et al. 2000). Hence, there is a trade-off between the accuracy of the personalization provided to users by recommenders and the degree of user privacy.

2.1 Privacy in recommender systems

Privacy concerns have triggered an increasing research into privacy and recommender systems. One of the first works demonstrating the difficulty of guaranteeing privacy in rating-based transaction records common in recommender systems was conducted by Narayanan and Shmatikov (2008). They considered an inference attack, where an adversary knows a subset of user attributes, e.g., items that were rated by the user, ratings that were assigned, or the time of the ratings. The de-anonymization algorithms assess the similarity of the anonymized data to the available auxiliary information and, due to the sparsity of data, can accurately uncover the original user information. Calandrino et al. (2011) studied the privacy risks imposed by three recommenders deployed by Hunch, Last.fm, and Amazon. The authors used background information to construct a profile similar to that of a target user, i.e., effectively cloned the profile with some noise. They relied on the idea that a new transaction of a user will mainly affect the recommendations given to similar users, tracked changes in the recommendations to the fake user, and used these changes to detect transactions carried out by the target user.

Although outside the direct data exposure context, Bilge et al. (2014) assessed the robustness of privacy-preserving collaborative recommendation algorithms to six

types of shilling attacks. Specifically, the authors compared memory- and model-based algorithms in terms of their ability to recover from shilling attacks and produce recommendations similar to those produced before the attacks. The results showed that model-based privacy-preserving algorithms were more robust to the attacks than their memory-based counterparts. That said, attacks involving injections of fake ratings for unpopular items, i.e., items with no solid rating evidence in the data, did affect the generated recommendations. To the best of our knowledge, no work has directly investigated data inference attacks on model-based collaborative recommenders. However, latent user and item vectors can be considered as a source of auxiliary information and be leveraged for attacks similar to those studied in Narayanan and Shmatikov (2008) and Calandrino et al. (2011).

Several works investigated privacy enhanced recommender systems. We divide these into two broad categories: distributed recommenders and data modification techniques. In the *distributed recommenders* group, user profiles are stored across several repositories. Canny (2002) proposed a decentralized storage of user profiles, which required the adversary to compromise multiple systems when attacking a distributed recommender. Individual users controlled their data and were grouped into communities representing a public aggregation of their ratings. The recommendations were generated by exposing only the aggregated community data, without exposing the ratings of individual users. Berkovsky et al. (2006) considered a similar hierarchical setting, where the requests for collaborative recommendations and aggregated ratings of underlying users propagated, respectively, down and up the hierarchy. However, both approaches required an a-priori formation of the communities or the hierarchy, limiting the responsiveness of the recommender to dynamic changes. Vallet et al. showed how matrix factorization techniques can be leveraged to allow a central server to provide accurate recommendations without retaining user data, i.e., storing latent user profiles on the client side. This work, however, did not exploit differential privacy, but rather assumed that the user profiles were not accessed by an attacker, even if the recommender system was attacked and personal data was compromised (Vallet et al. 2014).

In contrast, *data modification* techniques include approaches such as encryption (Nikolaenko et al. 2013), obfuscation (Berkovsky et al. 2012), access control (Sandhu et al. 1996), randomization (Polat and Du 2006), anonymization (Klösgen 1995), k-anonymization (Sweeney 2002), and differential privacy (McSherry and Mironov 2009). For example, Polat and Du (2006) proposed to add uncertainty to the stored user ratings through randomized data perturbation techniques. There, users could substitute part of the ratings in their profiles with fake ratings resembling, to a certain extent, the real ones. Hence, if the recommender was attacked and user data exposed to an adversary, not the original, but partially modified ratings would leak. However, an a-priori defined data perturbation policy would not preclude the adversary from recovering the original ratings. In the encryption space, Nikolaenko et al. (2013) showed how secure multiparty computation could be utilized in matrix factorization recommendations, so that the recommender learns only the item profiles, but not the user ratings. It should be highlighted that such techniques cannot prevent the inference of user ratings from the output of the matrix factorization computation, and are

orthogonal to the techniques studied in this paper, as they address a different threat model.

2.2 Applying differential privacy to recommender systems

Differential privacy has recently drawn much research attention; it makes no assumptions about the adversary's background knowledge and computation power, and provides formal and provable privacy guarantees (Dwork 2008; Dwork et al. 2006). Although differential privacy has been widely used for generic data and pattern mining purposes (Bhaskar et al. 2010; Friedman and Schuster 2010), to the best of our knowledge, only two prior works have investigated its application to recommender systems. Machanavajjhala et al. (2011) studied the problem of privacy-preserving social recommendations on the basis of a graph linking between users and items, e.g., products purchased by users. A utility vector derived from the graph captures the utility of each product for the target user, and the goal is to induce a probability distribution over the products, such as to maximize the utility for the user, while keeping the utility vector private. The authors provided a theoretical analysis of the problem and proposed differentially-private algorithms exploiting the Laplace noise mechanism. It was found that good recommendations were achievable only under weak privacy parameters, or only for a small fraction of users.

McSherry and Mironov (2009) considered the application of differential privacy to collaborative filtering recommenders. They used the Laplace mechanism to derive noisy counts and sums over the input rankings, and to compute a differentially-private variant of the item-to-item covariance matrix. The noisy covariance matrix could then be used to generate the differentially-private k-nearest neighbors and SVD recommendations. An essential component of the proposed solution was breaking the recommendation process into a learning phase, in which the private covariance matrix was derived, and a recommendation phase, in which private user ratings were combined with the aggregate data to derive predictions for the target user. In contrast, we consider in this work direct (privacy-preserving) derivation of the latent factor models. While this comes at a cost in prediction accuracy, this method allows to maintain privacy guarantees also for the user vectors, and can take advantage of the flexibility and efficiency benefits offered by matrix factorization methods.

In our own prior work, we proposed a framework for the application of differential privacy to matrix factorization (Berlioz et al. 2015). There we considered only bounded differential privacy, i.e., we focused only on the modification of existing item ratings and disregarded their mere presence, which may also leak some sensitive information. For example, in case of explicit or violent video content, users may primarily want to prevent the disclosure of the fact that such content was consumed by them, let alone the rating assigned to the content. Here, we also consider and evaluate the unbounded differential privacy hiding the presence of ratings or the consumption of items. In addition, we extend our previously reported evaluation and present the results obtained using a larger Netflix dataset, which adds validity to our experimental evidence. Overall, the current paper substantially extends our prior work (Berlioz et al. 2015), and explores additional approaches, beyond those investigated in McSherry and Mironov (2009) and Machanavajjhala et al. (2011).

3 Preliminaries

3.1 Recommendations with matrix factorization

Recent growth in the volume of online information has fueled the information overload problem. This motivated the development of recommender systems, which provide personalized suggestions for content and products. Recommender systems are used successfully in numerous application domains, e.g., entertainment, eCommerce, and eHealth, and many Web sites deploy recommenders in order to enhance user experience and increase revenues. Collaborative filtering recommenders derive their recommendations based on information collected from a community of users. They identify similarities between users, and recommend to a user items that were liked by like-minded users (Ning et al. 2015).

Matrix factorization (MF) (Koren et al. 2009) is considered the state-of-the-art variant of collaborative filtering, due to its computational scalability, predictive accuracy, and applicability to a range of recommendation tasks (Koren and Bell 2015). In its basic form, MF represents both users and items as vectors of latent factors, such that high similarity between the factors of a user and an item results in a recommendation. The input to MF is typically a rating matrix $R_{n\times m}$, containing the ratings of n users for m items, given on a predefined scale. Each matrix element r_{ui} reflects the rating of user u for item i. The matrix is typically very sparse, as users normally rate only a small subset of items. The recommendation algorithm factorizes $R_{n\times m}$ into two latent matrices: the user-factor matrix $P_{n\times d}$ and the item-factor matrix $Q_{m\times d}$. Each row p_u in P (and q_i in Q) represents the relation between the user u (item i) and the latent factor. The dimension of the latent matrices, d, is an external factorization parameter.

The factorization is done such that R is approximated as a product of P and Q, i.e., each known rating r_{ui} is approximated by $\hat{r}_{ui} = p_u \cdot q_i^{\mathsf{T}}$. To obtain P and Q, MF minimizes the regularized squared error for all the available ratings:

$$\min_{P,Q} \sum_{r_{ui}\in R} \left[\left(r_{ui} - p_u q_i^{\mathsf{T}}\right)^2 + \lambda \left(||p_u||^2 + ||q_i||^2 \right) \right]. \tag{1}$$

The constant λ regularizes the learned factors and prevents overfitting. Two common ways to solve the resulting non-convex optimization problem are stochastic gradient descent (SGD) and alternating least squares (ALS).

In SGD, the factors are learned by iteratively evaluating the error $e_{ui} = r_{ui} - p_u q_i^{\mathsf{T}}$ for each rating r_{ui}, and simultaneously updating the user and item vectors by taking a step in the direction opposite to the gradient of the regularized loss function:

$$\begin{aligned} p_u &\leftarrow p_u + \gamma \left(e_{ui} q_i - \lambda p_u\right), \\ q_i &\leftarrow q_i + \gamma \left(e_{ui} p_u - \lambda q_i\right). \end{aligned} \tag{2}$$

The constant γ determines the rate of minimizing the error and is often referred to as the learning rate. In ALS, the optimization problem is solved by updating the user and item latent vectors iteratively. That is, in each iteration, one of the factor matrices (say,

P) is fixed, resulting in a convex optimization problem, where the optimal solution (in this case, for Q) can be found analytically. Then the other factor matrix (Q) is fixed, and the optimization problem is solved again (this time for P). These steps are repeated until the convergence of the latent vectors.

Both in SGD and ALS, once the factorization converges, the latent matrices P and Q are used to predict unknown user ratings $\hat{r}_{ui}$. Namely, the resulting latent vectors p_u and q_i are multiplied, $\hat{r}_{ui} = q_i^{\mathsf{T}} p_u$, which produces the predicted rating of user u for item i. At the final step, the computed prediction are converted into recommendations, e.g., a subset of items with the highest predicted rating is recommended to the user.

3.2 Differential privacy

Differential privacy (Dwork 2008) is a provable privacy model based on the principle that the output of a computation should not allow inference about any particular record in the input. This is achieved by requiring that the probability of any computation outcome is insensitive to small input changes. We denote two datasets A and B as *adjacent*, $A \approx B$, if they are identical except for one record. More formally, there exist a user u and an item i such that $A = B \backslash \{r_{ui}\} \cup \{r'_{ui}\}$, where r'_{ui} and r_{ui} are the ratings that u assigned to i in A and B, respectively. The guaranteed level of privacy is measured by a parameter ϵ. Thus, a randomized computation K maintains ϵ-differential privacy if for any two datasets $A \approx B$, and any subset S of possible outcomes in Range(K),

$$Pr[K(A) \in S] \leq exp(\epsilon) \times Pr[K(B) \in S], \tag{3}$$

where the probability is over the randomness of K.

Low values of ϵ correspond to a high degree of privacy. Setting the bounds for the acceptable value of ϵ is an open question, which may be influenced, for example, by the data owner's threat analysis or by the users' privacy concerns. In the literature, privacy settings of $\epsilon = \ln 2$ or $\epsilon = \ln 3$ are considered as providing acceptable levels of privacy, although Dwork (2008) suggested that in some cases much higher values of ϵ could provide meaningful privacy guarantees.

Different variants of differential privacy can be derived from slightly different definitions of the *adjacent* datasets, for which the privacy constraints should hold (Kifer and Machanavajjhala 2011):

- A and B are adjacent if they contain the same ratings, but the value of one rating is different. Formally, there exist a user u and an item i such that $A = B \backslash \{r_{ui}\} \cup \{r'_{ui}\}$, where r'_{ui} and r_{ui} are the ratings that u assigned to i in datasets A and B, respectively. This variant is referred to in the literature as *bounded* differential privacy.
- A and B are adjacent if they are the same, except for one rating that appears in one dataset only. Formally, there exist a user u and an item i such that $A \backslash B = \{r_{ui}\}$ or $B \backslash A = \{r_{ui}\}$. This variant is referred to as *unbounded* differential privacy.

It should be highlighted that the bounded and unbounded variants are substantially different in the privacy guarantee they offer. On the one hand, any

algorithm which is ϵ-unbounded differentially private is 2ϵ-bounded differentially private (change of a rating could be obtained by removing and adding a record). On the other hand, there are algorithms that maintain bounded differentially privacy but break unbounded differential privacy. For example, revealing the fact that user u rated item i does not break bounded differential privacy if the value of the rating is not revealed, which is not the case for unbounded differential privacy.

From the privacy perspective, it could be argued that the main privacy risk in recommender systems stems from the disclosure of the list of items associated with a user, regardless of their exact ratings. Knowing which items interested the user may reveal a lot about the user's tastes, no matter whether the user rated those items high or low. Consequently, masking the influence of the ratings, as in the bounded differential privacy, does not necessarily hide the fact that the user rated those items, or did not rate other items. In contrast, unbounded differential privacy ensures that there is no distinction between the rated and unrated items. From the personalization perspective, hiding which items were rated by a user may be unacceptable. Since the ratings matrix is typically sparse, the stronger privacy protection of unbounded differential privacy may require more noise than masking the rated records alone. Consequently, the recommender's accuracy will drop, which may be a high price to pay for hiding the information whether a user rated an item or not.

In this work we evaluate both the bounded and unbounded variants of differential privacy, and the different privacy-personalization trade-offs that they offer. For each of the proposed algorithms, we first present and analyse the bounded variant, and then discuss the changes required to address also the unbounded variant.

3.3 The Laplace mechanism

One of the common ways to obtain differential privacy is by applying random noise to the data. The amount of noise added depends on the L_1-sensitivity of the evaluated function, i.e., the largest possible change in the measurement given a change in a single record in the dataset. In general, the L_k-sensitivity of a function g is given by:

$$S_k(g) = \max_{A \approx B} ||g(A) - g(B)||_k, \tag{4}$$

where $||\cdot||_k$ denotes the L_k-norm.

The *Laplace mechanism* (Dwork et al. 2006) obtains ϵ-differential privacy by adding noise sampled from Laplace distribution, with a calibrated scale b. The probability density function of Laplace distribution with mean 0 and scale b ($x \sim Laplace(b)$) is $f_b(x) = \frac{1}{2b} \exp(-\frac{|x|}{b})$.

Theorem 1 *Given a function $g : \mathcal{D} \to \mathbb{R}^d$, the following computation maintains ϵ-differential privacy:*

$$K(x) = g(x) + (Laplace\,(S_1(g)/\epsilon))^d\,. \tag{5}$$

For example, the function $\text{COUNT}_{cond}(A)$, which counts the number of records in dataset A that satisfy the condition $cond$, has sensitivity 1, because changing a single record could affect the count by at most 1. Hence, $K(A) = \text{COUNT}_{cond}(A) + Laplace(1/\epsilon)$ maintains ϵ-differential privacy. The function $\text{SUM}(A)$, where $a_i \in [0, \Lambda]$, has sensitivity Λ, which is the maximal change in the sum given a change in one element of A. Hence, $K(A) = \text{SUM}(A) + Laplace(\Lambda/\epsilon)$ maintains ϵ-differential privacy.

We also rely in this work on the K-norm mechanism (Hardt and Talwar 2010), which allows to calibrate noise to the L_2-sensitivity of the evaluated function:

Theorem 2 *Given a function $g : \mathcal{D} \rightarrow \mathbb{R}^d$, the following computation maintains ϵ-differential privacy:*

$$K(x) = g(x) + r\alpha, \tag{6}$$

where r is a d-dimensional vector uniformly sampled from a d-dimensional sphere with radius 1, and $\alpha \sim \Gamma(d, S_2(g)/\epsilon)$.

4 Differentially private matrix factorization

Differential privacy sets the conditions that a mechanism should maintain to preserve privacy, but within these constraints it is often possible to implement different mechanisms that would evaluate the same computation, resulting in different privacy-accuracy trade-offs. Considering the steps comprising the recommendation generation process of the MF algorithm, resulting in either rating predictions or items to be recommended to the target user, there is a number of possible approaches to add differentially private noise, as shown in Fig. 1.

– *Input perturbation* The original ratings are perturbed with a calibrated noise and then the MF algorithm is trained using the noisy input ratings. Since input perturbation is performed before training the recommender, the perturbation can be followed by any recommendation algorithm, and, in particular, by MF.

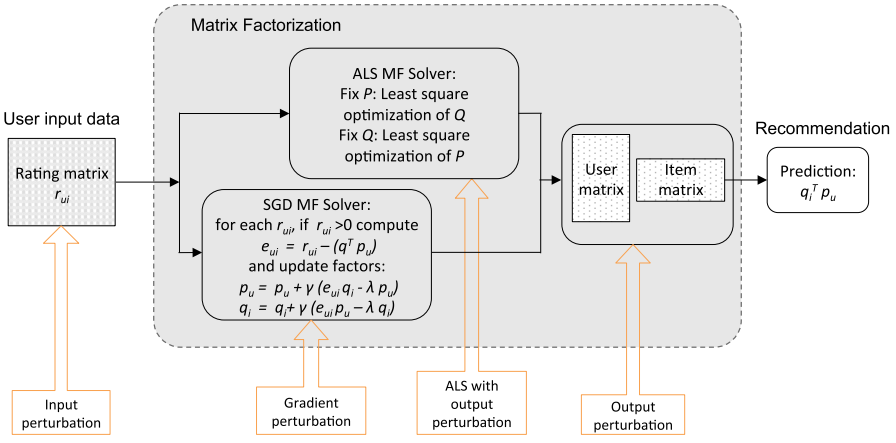

Fig. 1 Different noise application points in regards to the input, output and the solver within the matrix factorization mechanism

- *In-process mechanisms* In this approach, the algorithms used to factorize the original rating matrix into the latent user and item factor matrices are adapted to maintain differential privacy. In this work we consider two factorization algorithms, and propose their differentially private variants:
  1. *Stochastic gradient perturbation (SGD)* In the training process of MF with SGD, in each iteration, the gradient of the regularized loss function determines the direction of the update and its magnitude. In the gradient perturbation approach the gradient is perturbed with noise in each iteration.
  2. *ALS with output perturbation* In each step of the ALS algorithm, two convex optimization problems are solved to update the user and item factor matrices. These empirical risk minimization problems can be solved in a differentially-private manner using the techniques studied by Chaudhuri et al. (2011). In particular, we apply the output perturbation approach to obtain noisy versions of the factor matrices.
- *Output perturbation* In this approach, a non-private MF algorithm is executed, and then the resulting latent factors are perturbed to maintain differential privacy. Unfortunately, the optimization problem in MF is non-convex, as a small change in the input could lead to a large change in the factors. Consequently, the sensitivity of the optimization problem would require introducing large noise, potentially resulting in poor utility.

Hence, in this work we focus on the first three variants of differentially private MF, and study the privacy-accuracy trade-offs that they offer: input perturbation (where noise is added to the input ratings), SGD, and ALS with output perturbation (both including the addition of noise within the MF solver components). We do not consider the output perturbation approach, where noise is added directly to the latent factors after a non-private MF algorithm processes the inputs and produces the factors, due to its non-convex nature.

We first outline the data pre-processing steps that were taken before applying any of these approaches, and then describe each of these approaches in detail. For the pre-processing, we utilize the private versions of several aggregate values, based on the training dataset (as described in detail in Sect. 5.1). These are *global average* $GAvg(R)$—the average rating of all users to all items; *item average* $IAvg(i)$—the average rating assigned by all users to item i; and *user average* $UAvg(u)$—the average rating assigned by user u to all items; and *global effects*—the combination of average user and item ratings computed by $\text{IAvg}(i) + \text{UAvg}(u)$.

4.1 Data preprocessing and private global effects

Prior to executing any of the proposed algorithms, we preprocess the inputs following the approach outlined in McSherry and Mironov (2009). Namely, the preprocessing consists of three steps: evaluation of global and item averages, evaluation of user averages, and clamping of the resulting ratings. A notable difference is that we incorporate the user averages in rating predictions, as it was found that this allows to derive more accurate predictions when using the MF approach. Below we detail the preprocessing steps:

Input: $R = \{r_{ui}\}$ – ratings of n users for m movies,
β_m – stabilization parameter,
ϵ_1 – global average privacy parameter,
ϵ_2 – item average privacy parameter
Output: Item averages IAvg(i)

1: GAvg $= \frac{(\sum_R r_{ui}) + \text{Laplace}(\Delta r/\epsilon_1)}{|R|}$
2: **for** $j = 1$ to m **do**
3: Let $R_j = \{r_{ui} \in R | i = j\}$
4: IAvg(j) $= \frac{(\sum_{R_j} r_{ui}) + \beta_m \cdot \text{GAvg} + \text{Laplace}(\Delta r/\epsilon_2)}{|R_j| + \beta_m}$
5: Clamp IAvg(j) to $[r_{\min}, r_{\max}]$.

Algorithm 1: Evaluation of item averages

Input: $R = \{r_{ui}\}$ – ratings of n users for m movies,
β_u – stabilization parameter,
ϵ_1 – global average privacy parameter,
ϵ_2 – user average privacy parameter
Output: User averages UAvg(u)

1: Let $R' = \{r_{ui} - \text{IAvg}(i) | r_{ui} \in R\}$
2: GAvg$' = \frac{(\sum_{R'} r'_{ui}) + \text{Laplace}(\Delta r/\epsilon_1)}{|R'|}$
3: **for** $v = 1$ to n **do**
4: Let $R_v = \{r'_{ui} \in R' | u = v\}$
5: UAvg(v) $= \frac{(\sum_{R_v} r'_{ui}) + \beta_u \cdot \text{GAvg}' + \text{Laplace}(\Delta r/\epsilon_2)}{|R_v| + \beta_u}$
6: Clamp UAvg(v) to $[-2, 2]$

Algorithm 2: Evaluation of user averages

- *Private global and item averages* First, we compute the differentially-private average item ratings according to the process described in Algorithm 1. We add a number of fictitious ratings β_m with the global average $GAvg$ to stabilize the items averages: this would limit the effect of noise for items with few ratings, while only slightly affecting the average for items with many ratings. In case the added noise causes the item average to go out of the range of input ratings $[r_{\min}, r_{\max}]$, the item average is clamped to fit this range. Differential privacy is guaranteed by adding noise calibrated to the L_1-sensitivity of the ratings given by $\Delta r = r_{\max} - r_{\min}$.
- *Private user averages* Next we follow the same technique to compute the user average ratings, as outlined in Algorithm 2. The basis for evaluating the user averages is the ratings after the item averages were discounted. We stabilize the user effects with the addition of β_u fictitious ratings with the newly computed global average. The the user averages are then clamped to a bounded range (in our experiments we used the range $[-2, 2]$ for the user averages).
- *Clamping* Finally, the item and user averages are discounted from the rating matrix R, and the resulting ratings are clamped. The clamping reduces the L_1-sensitivity of the computations conducted during the MF process, and therefore results in lower magnitudes of noise being introduced to the differentially private computation. We denote the clamping parameter by B ($B = 1$ in our experiments) and set it as follows:

$$r_{ui} = \begin{cases} -B & \text{if } r_{ui} < -B \\ r_{ui} & \text{if } -B \leq r_{ui} \leq B \\ B & \text{if } r_{ui} > B \end{cases}. \tag{7}$$

The pre-processed matrix R is then passed to one of the factorization algorithms, which derives the latent matrices P and Q in the usual non-private manner. Finally, predicted item ratings are obtained through $\hat{r}_{ui} = \text{IAvg}(i) + \text{UAvg}(u) + p_u q_i^{\mathsf{T}}$.

Note that it is also possible to predict a rating using only the user and item averages: $\hat{r}_{ui} = \text{IAvg}(i) + \text{UAvg}(u)$. When clean item and user averages are calculated, this technique offers the most simplistic way to generate personalised recommendations (see comparison baselines in Sect. 5.1). We can also consider an analogous differentially-private simplistic recommendation method using the private counterparts of user and item averaged computed using the above three preprocessing steps. In this case, as we do not use MF, the privacy budget is split between the three computations: global averages, item averages, and user averages. We refer to this technique as *Private Global Effects*.

When introducing noise to a differentially-private computation, the privacy budget is divided between the parts of the computation. However, differential privacy maintains the *composability* property: if each computation in a series of computations is ϵ_i-differentially private, then the overall algorithm will be $\sum_i \epsilon_i = \epsilon$-differentially private. Accordingly, in the differentially-private implementations of MF the overall privacy budget ϵ is divided between the computations of global averages, item effects, user effects, and finally the factorization process itself.

It should also be highlighted that the above process considers the bounded variant of differential privacy, in which the number of ratings $|R|$ is known. In the unbounded case, the number of available ratings should be computed privately with the Laplace mechanism, e.g., $|R| + Laplace(1/\epsilon)$. Again, the composability property allows the privacy budget ϵ assigned to the calculations of average ratings to be split between the noisy sum calculation and the noisy count, to ensure that the entire computation is ϵ-differentially private.

4.2 Matrix factorization with private input perturbation

In the input perturbation approach, each rating is considered independently of the rest, and is perturbed in a way that maintains differential privacy. The magnitude of noise that is applied to the rating matrix is determined according to the Laplace Mechanism. In particular, the range of ratings in the data dictates the global sensitivity of the ratings. Following the data preprocessing, the ratings are in the range $r_{ui} \in [r_{\min}, r_{\max}]$, such that the global sensitivity of the ratings is $\Delta r = r_{\max} - r_{\min} = 2B$, due to clamping. Once the global sensitivity is derived the level of noise can be determined using Eq. (5). For bounded differential privacy, perturbing each of the ratings with noise sampled from the distribution Laplace$(\Delta r/\epsilon)$ ensures ϵ-differential privacy. The noisy ratings can then be clamped again, to limit the influence of excessive noise. Algorithm 3 summarizes this process.

Input: $R = \{r_{ui}\}$ – preprocessed ratings of n users for m movies,
d – number of factors,
λ – regularization parameter,
B – clamping parameter,
ϵ – privacy parameter
Output: Approximate factor matrices $P_{n\times d}$ and $Q_{m\times d}$

1: Let $R' = \{r_{ui} + \text{Laplace}(\frac{\Delta r}{\epsilon}) | r_{ui} \in R\}$
2: Clamp the ratings in R' to the range $[-B, B]$
3: $(P, Q) = \min_{P,Q} \sum_{R'} [(r'_{u,i} - p_u q_i^\intercal)^2 + \lambda(\|q_i\|^2 + \|p_u\|^2)]$
4: **return** P and Q

Algorithm 3: Matrix Factorization with Input Perturbation

Theorem 3 *Algorithm 3 maintains ϵ-differential privacy.*

Proof The global sensitivity of the input ratings is $\Delta r = (r_{\max} - r_{\min})$. According to the Laplace mechanism, this algorithm maintains ϵ-differential privacy. □

Let us now consider the unbounded privacy, where we have to protect also the existence of ratings. Following the data preprocessing, missing ratings are assigned a default value of 0 (equivalent to a rating combining the user and item biases). Therefore, the sensitivity of each of the inputs is the maximal difference between a given rating and the 0 rating, which amounts to $\Delta r \leq r_{\max} = B$. Consequently, we have to add noise to each entry of the input matrix R from the distribution Laplace(B/ϵ), effectively turning missing ratings into "fake ratings". Adding so many new ratings does not only increase computation time, but also undermines prediction accuracy. To mitigate this effect, we apply an additional clamping step after the introduction of noise, using the clamping parameter α:

$$r_{ui} = \begin{cases} -B & \text{if } r_{ui} < -B \\ 0 & \text{if } |r_{ui}| \leq \alpha \\ r_{ui} & \text{if } \alpha < |r_{ui}| \leq B \\ B & \text{if } r_{ui} > B \end{cases}. \tag{8}$$

This clamping to 0 for ratings in the range $[-\alpha, \alpha]$ removes more "fake ratings" than actual ratings and, thus, improves the prediction accuracy. The value of α used in the experiments was 0.6.

4.3 Private stochastic gradient perturbation

In MF with SGD, the input of the algorithm consists of user ratings, a learning rate γ, and the regularization parameter λ. In each iteration, the training samples are used to evaluate the prediction error resulting from the current factor matrices, and then the matrices are modified in a direction opposite to the gradient, with magnitude proportional to the learning rate γ. The private SGD perturbation approach, outlined in Algorithm 4, guarantees privacy throughout the MF process by introducing noise to the gradient descent step in each iteration of the algorithm. The error calculation conducted

Input: $R = \{r_{ui}\}$ – (preprocessed) user ratings,
d – number of factors,
γ – learning rate parameter,
λ – regularization parameter,
k – number of gradient descent iterations,
$e_{\max}$ – upper bound on per-rating error,
ϵ – privacy parameter
Output: Approximate factor matrices $P_{n\times d}$ and $Q_{m\times d}$

1: Initialize random factor matrices P and Q.
2: **for** k iterations **do**
3: **for** each $r_{ui} \in R$ **do**
4: $e'_{ui} = r_{ui} - p_u q_i^{\mathsf{T}} + \text{Laplace}(k\Delta r/\epsilon)$
5: $e'_{ui} = \begin{cases} -e_{\max} & \text{if } e'_{ui} < -e_{\max} \\ e'_{ui} & \text{if } |e'_{ui}| \leq e_{\max} \\ e_{\max} & \text{if } e'_{ui} > e_{\max} \end{cases}$
6: $q_i \leftarrow q_i + \gamma(e'_{ui} \cdot p_u^{\mathsf{T}} - \lambda \cdot q_i)$
7: $p_u \leftarrow p_u + \gamma(e'_{ui} \cdot q_i^{\mathsf{T}} - \lambda \cdot p_u)$
8: **return** final P and Q.

Algorithm 4: Matrix Factorization with bounded differentially private Stochastic Gradient Descent

in each step is carried out with the Laplace mechanism to maintain differential privacy, and consequently the gradient descent step maintains differential privacy. Optionally, the noisy error can then be clamped to constrain the effect of noise (in our experiments we used $e_{\max} = 2$). The number of iterations k is known in advance, so the noise introduced in each iteration can be calibrated to maintain ϵ/k-differential privacy. Composability ensures that the k iterations maintain the overall bound of ϵ-differential privacy.

Theorem 4 *Algorithm 4 maintains bounded ϵ-differential privacy.*

Proof Algorithm 4 accesses the input ratings only when calculating the error in line 4. The error for each rating is given by $e_{ui} = r_{ui} - p_u q_i^{\mathsf{T}}$. The L_1-sensitivity of the rating error is:

$$\max_{A \approx B} |e_{ui}(A) - e_{ui}(B)| \leq \max | \left(r_{ui} - p_u q_i^{\mathsf{T}}\right) - \left(r'_{ui} - p_u q_i^{\mathsf{T}}\right) | \leq \Delta r. \tag{9}$$

According to the Laplace mechanism, line 4 preserves ϵ/k-differential privacy for each of the input ratings. Because the overall number of iterations is k, composability ensures that the whole algorithm maintains ϵ-differential privacy. □

In the unbounded differential privacy case, we have to protect the existence of a rating in the data set. As outlined in Algorithm 4, the gradient descent is done over all the existing ratings. A new rating r_{ui} would induce another update for the vector p_u and q_i (steps 4 to 7 in Algorithm 4). In order to mask the existence of such an operation, we can evaluate the maximum change that this operation could induce in the user-vector p_u and the item-vector q_i. We note by s_p (and, respectively, s_q), the sensitivity of a user-vector q_u (respectively, item-vector q_i):

$$s_p \leq \max ||\gamma \left(e'_{ui} \cdot p_u^{\mathsf{T}} - \lambda \cdot q_i\right)||_2 = \gamma \left(e_{max} \cdot p_{max} + \lambda \cdot q_{max}\right) \tag{10}$$

$$s_q \leq \max ||\gamma \left(e'_{ui} \cdot q_i^{\mathsf{T}} - \lambda \cdot p_u\right)||_2 = \gamma \left(e_{max} \cdot q_{max} + \lambda \cdot p_{max}\right) \tag{11}$$

Input: $R = \{r_{ui}\}$ – user ratings,
d – number of factors,
γ – learning rate parameter,
λ – regularization parameter,
k – number of gradient descent iterations,
$e_{\max}$ – upper bound on per-rating error,
$q_{\max}$ – upper bound on the norm of an item vector q_i,
$p_{\max}$ – upper bound on the norm of a user vector p_u,
ϵ – privacy parameter
Output: Approximate factor matrices $P_{n\times d}$ and $Q_{m\times d}$

$s_q = \gamma(e_{\max} \cdot p_{\max} + \lambda \cdot q_{\max})$
$s_p = \gamma(e_{\max} \cdot q_{\max} + \lambda \cdot p_{\max})$
Initialize random factor matrices P and Q.
for k iterations **do**
  for each $r_{ui} \in R$ **do**
    $e_{ui} = \max\{r_{ui} - q_i^\intercal p_u, e_{\max}\}$
    $q_i \leftarrow q_i + \gamma(e'_{ui} \cdot p_u^\intercal - \lambda \cdot q_i)$
    $p_u \leftarrow p_u + \gamma(e'_{ui} \cdot q_i^\intercal - \lambda \cdot p_u)$
  endfor
  for each user u **do**
    Sample noise vector np with pdf $f(b) \propto \exp\left(-\frac{\epsilon \cdot ||b||_2}{2k \cdot s_p}\right)$
    $p_u \leftarrow p_u + np$
    if $||p_u||_2 > p_{\max}$ **then** $p_u \leftarrow p_u \cdot \frac{p_{\max}}{||p_u||_2}$
  for each item i **do**
    Sample noise vector nq with pdf $f(b) \propto \exp\left(-\frac{\epsilon \cdot ||b||_2}{2k \cdot s_q}\right)$
    $q_i \leftarrow q_i + nq$
    if $||q_i||_2 > q_{\max}$ **then** $q_i \leftarrow q_i \cdot \frac{q_{\max}}{||q_i||_2}$
endfor
return final P and Q.

Algorithm 5: Matrix Factorization with unbounded differentially private Stochastic Gradient Descent

where $e_{max} = r_{max} + p_{max} \cdot q_{max}$.

To mask the existence of a single rating of user u for item i, noise will be added to all the vectors p_u of P and to all the vectors q_i of Q at the output of each SGD iteration. Since we have evaluated the $L2$-norm sensitivities, the noise will be taken from the Gamma distribution following Theorem 2. The noise to be added to a user-vector p_u will be sampled with scale $(2s_p \cdot iter)/\epsilon$. For an item-vector q_i, the noise will be sampled with scale $(2s_q \cdot iter)/\epsilon$. Algorithm 5 outlines the resulting process for the unbounded differentially private SGD.

4.4 ALS with output perturbation

Zhou et al. (2008) have proposed solving the MF problem using the alternating-least-squares algorithm with weighted-λ-regularization (ALS-WR). In this section, we consider a differentially private version of that algorithm.

The essential idea of ALS is to alternately fix one of the the latent matrices P and Q, and optimize the regularized loss function for the other non-fixed matrix. Once one of the matrices is fixed, the optimization problem becomes convex, and it can be solved analytically. For example, once the item-factor matrix Q is fixed, the overall regularized loss function can be minimized by considering for each user u the following loss function defined over the subset of ratings $R_u = \{r_{vi} \in R | v = u\}$:

$$J_Q(p_u, R) = \left[\sum_{R_u} \left(r_{ui} - p_u q_i^{\mathsf{T}} \right)^2 \right] + n_u \lambda \| p_u \|^2, \tag{12}$$

where $n_u = |R_u|$. Each user vector p_u is then obtained by solving the risk minimization problem

$$p_u(R, Q) = \arg\min_{p_u} J_Q(p_u, R). \tag{13}$$

The problem of differentially private empirical risk minimization (ERM) was studied by Chaudhuri et al. (2011), where the authors explored how to choose in a privacy-preserving manner a vector that minimizes a regularized empirical loss function. In this section we rely on the following definition and theorem from Chaudhuri et al. (2011):

Definition 1 Function $H(x)$ over $x \in \mathbb{R}^d$ is λ-*strongly convex* if for all $\alpha \in (0, 1)$, x_1 and x_2:

$$H(\alpha x_1 + (1-\alpha)x_2) \leq \alpha H(x) + (1-\alpha)H(x_2) - \frac{1}{2}\lambda\alpha(1-\alpha)\|x_1 - x_2\|_2^2. \tag{14}$$

Theorem 5 *Let $f(x)$ and $g(x)$ be two vector-valued functions, which are continuous and differentiable at all points. Moreover, let $f(x)$ and $f(x) + g(x)$ be λ-strongly convex. If $x_1 = \arg\min_x f(x)$ and $x_2 = \arg\min_x f(x) + g(x)$, then*

$$\|x_1 - x_2\|_2 \leq \frac{1}{\lambda} \max_x \|\nabla_g(x)\|_2. \tag{15}$$

□

Chaudhuri et al. (2011) used Theorem 5 to evaluate Δ, the L_2-sensitivity of the optimization problem. Then a differentially private solution for the ERM problem could be obtained by adding a noise vector b, sampled according to the density function $f(b) \propto \exp(-\frac{\epsilon\|b\|_2}{\Delta})$. We follow a similar approach to obtain a differentially private solution for the P and Q matrices in each iteration of the ALS algorithm. Since the problem formulation in MF is slightly different from the one presented in Chaudhuri et al. (2011), we provide below the sensitivity analysis for the user vector p_u.

Theorem 6 *The L_2-sensitivity of the optimization problem for $p_u(R, Q)$, given in Eq. (13), is $\Delta p_u = \frac{q_{\max}\Delta r}{n_u\lambda}$, where $q_{\max}$ is an upper bound on the L_2-norm of each row q_i in Q.*

Proof Let $R \approx R'$ be two adjacent data sets that are different only in one of the ratings. According to Theorem 5, we denote $f(p_u) = J_Q(p_u, R)$ and $g(p_u) = J_Q(p_u, R) - J_Q(p_u, R')$. In addition, we set $p_1 = \arg\min_{p_u} J_Q(p_u, R)$, and $p_2 = \arg\min_{p_u} J_Q(p_u, R')$. Then the L_2-sensitivity of $p_u(R, Q)$ is given by $\max \|p_1 - p_2\|_2$.

The regularizer function $N(p_u) = \|p_u\|^2$ is 2-strongly convex. Since the loss function $\sum_{R_u}(r_{ui} - p_u q_i^{\mathsf{T}})^2$ for a fixed Q is convex, it follows that $f(p_u) = J_Q(p_u, R)$

and $f(p_u) + g(p_u) = J_Q(p_u, R')$ are $2n_u\lambda$-strongly convex. Finally, $N(p_u)$, $f(p_u)$ and $g(p_u)$ are differentiable at all points. As all the conditions for Theorem 5 are met, we can conclude that

$$\|p_1 - p_2\|_2 \leq \frac{1}{2n_u\lambda} \max_{p_u} \|\nabla g(p_u)\|. \tag{16}$$

Since $g(p_u) = (r_{ui} - p_u q_i^\intercal)^2 - (r'_{ui} - p_u q_i^\intercal)^2$, we have $\nabla g(p_u) = 2q_i^\intercal(r'_{ui} - r_{ui})$. Therefore, we can bound

$$\max_{p_u} \|\nabla g(p_u)\| \leq 2q_{max} \cdot \Delta r, \tag{17}$$

where q_{max} is an upper bound on the L_2-norm of q_i. The theorem then follows from Eqs. (16) and (17). □

Similarly, when fixing the matrix P and optimizing the vectors in Q based on the regularized loss function $J_P(q_i, R) = [\sum_{R_i}(r_{ui} - p_u q_i^\intercal)^2] + n_i\lambda\|q_i\|^2$, the L_2-sensitivity for the evaluation of each row q_i is $\frac{p_{max} \cdot \Delta r}{\lambda n_i}$. For the preprocessed ratings, we have $\Delta r = 2B$, where B is the clamping parameter.

For unbounded differential privacy, where the difference between R and R' is in the inclusion of a rating that user u assigned to item i', we assume without loss of generality that R' includes the additional rating. Then:

$$\begin{aligned} g(p_u) &= \left(r_{ui'} - p_u q_{i'}^\intercal\right)^2 + \lambda\|p_u\|^2 \\ \nabla g(p_u) &= 2q_{i'}^\intercal\left(r_{u,i'} - p_u q_{i'}^\intercal\right) + 2\lambda p_u \\ \max_{p_u} \|\nabla g(p_u)\| &\leq 2q_{max} \cdot (r_{max} + q_{max} \cdot p_{max}) + 2\lambda p_{max} \end{aligned}$$

Table 1 summarizes the global sensitivity of the user-factor and item-factor vectors depending on the variant of privacy required, bounded or unbounded. The noise is calibrated according to the global sensitivity. B is the clamping parameter and we have $B = r_{max} = \Delta r/2$. Since we calculated the L_2-sensitivity of the user-vector p_u and of the item-vector q_i, the noise to be added to these vectors will be taken from the Gamma distribution following Theorem 2.

Following the above analysis, Algorithm 6 outlines a differentially-private ALS algorithm for MF. The algorithm shows the bounded differentially private variant, but

Table 1 The L_2-sensitivity of updates to p_u and q_i depending on privacy variant

Privacy variant →	Bounded	Unbounded
Sensitivity of p_u	$4q_{max} \cdot B$	$2q_{max} \cdot (B + q_{max} \cdot p_{max}) + 2\lambda p_{max}$
Sensitivity of q_i	$4p_{max} \cdot B$	$2p_{max} \cdot (B + q_{max} \cdot p_{max}) + 2\lambda q_{max}$

Input: $R = \{r_{ui}\}$ – (preprocessed) user ratings,
d – number of factors,
λ – regularization parameter,
k – number of ALS iterations,
ϵ – privacy parameter,
$p_{\max}$ – upper bound on $||p_u||_2$,
$q_{\max}$ – upper bound on $||q_i||_2$
Output: Approximate factor matrices $P_{n\times d}$ and $Q_{m\times d}$
Initialize random factor matrices P and Q.
for k iterations **do**
  for each user u, given Q **do**
    Sample noise vector b with pdf $f(b) \propto \exp\left(-\frac{\epsilon\cdot||b||_2}{2k} \cdot \frac{n_u\lambda}{p_{\max}\cdot\Delta r}\right)$
    $p_u \leftarrow \arg\min_{p_u} J_Q(p_u, R_u) + b$
    if $||p_u||_2 > p_{\max}$ **then** $p_u \leftarrow p_u \cdot \frac{p_{\max}}{||p_u||_2}$
  for each item i, given P **do**
    Sample noise vector b with pdf $f(b) \propto \exp\left(-\frac{\epsilon\cdot||b||_2}{2k} \cdot \frac{n_i\lambda}{q_{\max}\cdot\Delta r}\right)$
    $q_i \leftarrow \arg\min_{q_i} J_P(q_i, R_i) + b$
    if $||q_i||_2 > q_{\max}$ **then** $q_i \leftarrow q_i \cdot \frac{q_{\max}}{||q_i||_2}$
return final P and Q.

Algorithm 6: ALS with Output Perturbation

a similar process applies to the unbounded case, with the exception that the noise is calibrated according to the global sensitivity corresponding to the unbounded variant. Similarly to the SGD approach, we calibrate the noise so that each optimization problem is $\epsilon/2k$-differentially private and the overall ALS computation is ϵ-differentially private due to composability.

5 Evaluation

In this section, we evaluate the proposed differentially private MF approaches. Section 5.1 describes the datasets used in the evaluation and outlines the experimental setting. The analysis of the bounded and unbounded differentially-private MF approaches is presented in Sects. 5.2 and 5.3, respectively. The former also includes two sensitivity analyses. Finally, Sect. 5.4 compares the differentially private MF with other differentially private collaborative filtering baselines.

5.1 Experimental setting

We use in the evaluation four datasets: the 100K, 1M and 10M MovieLens datasets collected by the Grouplens group[1] and the Netflix Prize Competition dataset.[2] Table 2 summarizes selected statistical properties of the datasets. It includes the number of users, number of items (in this case, movies), overall number of ratings, density of the dataset, average and variance of ratings in the dataset, average number of movies rated by a user, and average number of ratings assigned to a movie.

It should be noted that all three MovieLens datasets were collected by GroupLens through their experimental MovieLens system (Harper and Konstan 2016).

[1] http://grouplens.org/datasets/movielens/.

[2] http://www.netflixprize.com/.

Table 2 Statistical properties of the datasets

	ML-100K	ML-1M	ML-10M	Netflix
Users	943	6040	71,567	480,189
Movies	1682	3952	65,133	17,770
Ratings	100K	1M	10M	100M
Density	6.3 %	4.19 %	0.21 %	0.21 %
Average rating	3.5299	3.5816	3.5124	3.6043
Variance of ratings	1.2671	1.2479	1.1245	1.1777
Avg. ratings per user	106	165.6	139.7	37.9
Avg. ratings per item	59.4	253	153.5	5654

They contain the ratings assigned by the users of MovieLens to movies, all given on a 1-to-5 star scale. Although collected through the same system, the overlap of the datasets is limited. Specifically, the smallest MovieLens-100K dataset contains ratings assigned during the 7-month period, between September 1997 and April 1998. MovieLens-1M contains only the ratings of users, who joined MovieLens in 2000, but the dataset was created in early 2003, leading to ratings assigned by the users in 2001 and 2002. Hence, MovieLens-100K and MovieLens-1M do not overlap at all. The largest MovieLens-10M dataset contains the ratings of random users across the entire history of MovieLens. As such, there is some overlap between MovieLens-1M and MovieLens-10M, although MovieLens-10M is an order of magnitude larger than MovieLens-1M. Additional user and movie data provided in various MovieLens datasets, e.g., demographic information, movie metadata, rating timestamps, and movie tags, was not used in this work.

We split the available rating data into the training and test sets. For the Movielens datasets, we use tenfold cross validation to train and evaluate the recommender system.[3] For the Netflix dataset, we train the recommender system on the qualifying set and evaluate it on the test set released for the Netflix Challenge. In the experiments reported in the rest of this section, we focus on the predictive accuracy of the recommendations. Hence, we use the Root Mean Square Error (RMSE) metric to measure the accuracy of the predicted ratings. RMSE is computed by

$$\text{RMSE} = \sqrt{\sum_{R} \left(r_{ui} - \hat{r}_{ui}\right)^2 \Big/ |R|}, \tag{18}$$

where r_{ui} and $\hat{r}_{ui}$ is, respectively, the real and predicted rating assigned by user u to movie i, and R denotes the test set of ratings being predicted. Due to the possible discrepancies in the noise introduction, the reported RMSE scores are averaged across 5 runs for the MovieLens-100K and MovieLens-1M datasets, and across 10 runs for the larger MovieLens-10M and Netflix datasets.

[3] We used Matlab and, specifically, *crossvalind*, to split the data.

We compare the performance of the proposed privacy-preserving MF algorithms to the following baselines:

- *Global average (GA)* The average rating is computed over the entire training set, and used as the prediction for all the items in the test set, i.e., $\hat{r}_{ui} = GAvg(R)$. We treat the global average RMSE as the upper bound for error.
- *Item average (IA)* The average rating for each item is computed over all the available item ratings, and used as the prediction for all the ratings for that item in the test set, i.e., $\hat{r}_{ui} = IAvg(i)$. This baseline reflects the RMSE score attainable without personalization.
- *Global effects (GE)* The average ratings IAvg(i) for each item and UAvg(u) for each user are computed over the entire training set. The item and user biases are combined when predicting the test set ratings, i.e., $\hat{r}_{ui} = \text{IAvg}(i) + \text{UAvg}(u)$. We treat this baseline as the most simplistic way to obtain personalization, and we consider RMSE scores below this baseline to represent effective personalization.
- *Clean MF* The ALS algorithm is executed to solve the MF problem without any privacy constraints. The RMSE scores of the clean MF reflect the lower bound for error attainable by the recommender without any privacy constraints in place.

We use the IA and GE baselines to compare the privacy-personalization trade-offs offered by the private algorithms. To this end, we measure the values of ϵ for which the RMSE scores attained by each differentially-private algorithm *cross* the two baseline RMSE scores: IA and GE. It should be highlighted that low values of ϵ are desirable for the baseline crossing, as they indicate that the algorithm can provide the same level of accuracy as the baseline with a low cost in privacy. On the contrary, higher values of ϵ for which a baseline crossing is observed mean that the desired levels of accuracy can be achieved only with low privacy guarantees.

That said, setting specific limits for the acceptable value of privacy parameter ϵ is an open question, which may be influenced, for example, by the data owner's threat analysis or by the users' privacy concerns. In the research literature, privacy settings of $\epsilon = \ln 2$ or $\epsilon = \ln 3$ are typically considered as those providing acceptable levels of privacy (Dwork 2008). Despite that, Dwork et al. (2011) argued that in some cases, e.g., the AOL privacy breach, even values as high as $\epsilon = 12$ could provide meaningful guarantees. In the evaluation, we focus on the observed privacy-personalization trade-offs offered by each of the algorithms, rather than evaluate the performance of the algorithms for certain values of ϵ. Also, we touch upon several factors and considerations that may affect the system performance.

5.2 Bounded differential privacy

In each experiment, given the overall ϵ-differential privacy constraint, we allocated 0.3ϵ for data preprocessing. Out of this, 0.02ϵ was used to compute the global averages, where the MovieLens item averages and user averages were computed with 0.14ϵ each and the Netflix item averages and user averages were computed with 0.06ϵ and 0.22ϵ, respectively. The remaining privacy budget of 0.7ϵ was allocated to the actual factorization of the rating matrix. This allocation of the privacy budget is based

Table 3 Summary of the experimental settings and baseline RMSE

	100K	1M	10M	Netflix
Parameter settings				
Number of factors	3	5	7	20
Regularizer	0.06	0.045	0.03	0.05
Number of iterations	10	10	5	5
Baselines				
Global average (GA)	1.1256	1.1171	1.0604	1.1310
Item average (IA)	1.0278	0.9795	0.9436	1.0537
Global effects (GE)	0.9571	0.9161	0.8738	0.9874
Clean MF—ALS	0.9198	0.8604	0.8013	0.9235

on an offline optimization carried out for the datasets. It is in line with the budget allocations schema of McSherry and Mironov (2009), which allocated a small portion of the budget to the average computations and a larger portion to the recommendation task (specifically, user-to-user covariance matrix). Such a fixed allocation of the privacy budget is not unusual in differential privacy research and has been exploited beyond recommender system applications (Bhaskar et al. 2010; Erlingsson et al. 2014; Friedman and Schuster 2010).

Where applicable, we bounded the L_2-norm of the user vectors to $p_{\max} = 0.4$, and the L_2-norm of the item vectors to $q_{\max} = 0.5$. In both the SGD and the ALS with output perturbation experiments, we set the number of iterations to $k = 5$. Finally, the number of iterations for input perturbation was set to $k = 20$. The upper part of Table 3 details several other dataset specific factorization parameters. The optimization of these parameters was also done offline; some details of the parameterisation process can be found in "Appendix". We note that the selected number of factors and the number of iterations were lower than those usually used in recommender systems implementations of MF, in order to limit the amount of noise introduced by differential privacy. In addition to the parameter values, the bottom part of Table 3 shows the baselines RMSE scores measured for each of the evaluation datasets.

The results of the unbounded differential privacy experiments are summarized in Table 4 and in Fig. 2. Table 4 shows the the values of ϵ, for which each of the algorithms crossed the IA and GE baselines. Figure 2a–c shows the privacy-personalization trade-offs for all the algorithms for the MovieLens-1M, MovieLens-10M and Netflix datasets respectively.[4] In addition to the above-mentioned baselines, the figures show the results of the private global effects approach, where all the privacy budget is used to evaluate the item and user averages; the Input Perturbation approach followed by a non-private Stochastic Gradient Descent algorithm (ISGD); the Private Stochastic Gradient Descent approach (PSGD); and the Private ALS approach (PALS).

[4] Results obtained for the MovieLens-100K dataset exhibit a similar trend and are not shown, but only summarized in Table 4.

Table 4 Summary of the results for bounded differential privacy

	100K	1M	10M	Netflix
Private global effects				
IA crossing	0.5	0.2	0.18	0.18
Input perturbation				
IA crossing	2	0.9	0.7	0.7
GE crossing	5	2.7	2.1	1.8
Stochastic gradient perturbation				
IA crossing	2	0.8	0.6	0.6
GE crossing	20	8	5.5	3.5
ALS with output perturbation				
IA crossing	2	0.8	0.6	0.6
GE crossing	19	8	6	5.5

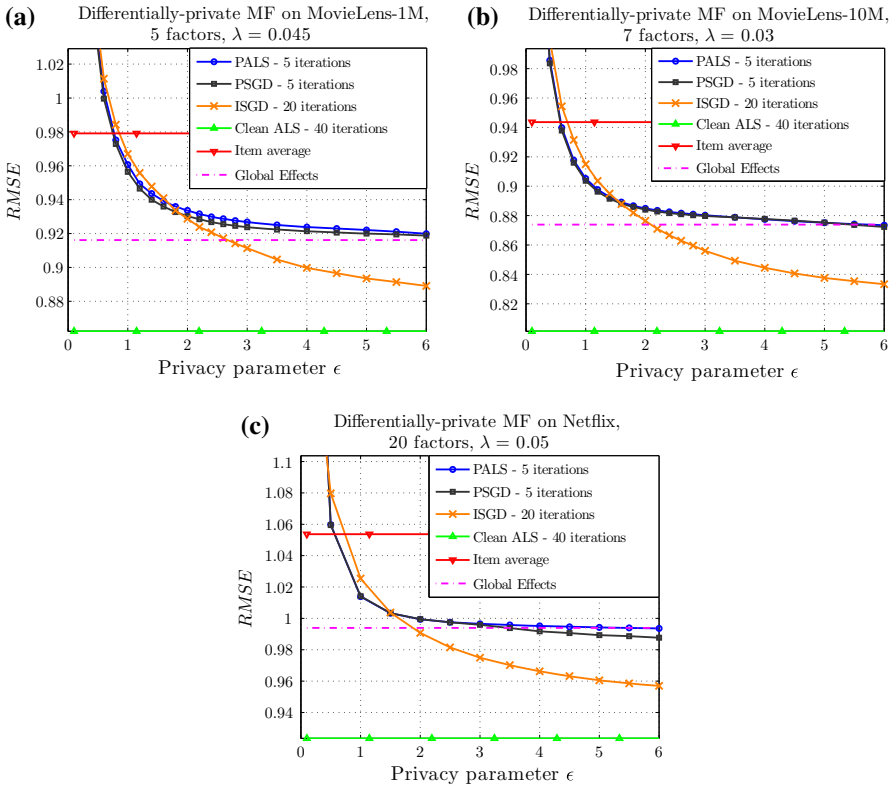

Fig. 2 Bounded differentially-private matrix factorization. **a** MovieLens-1M. **b** MovieLens-10M. **c** Netflix

In general, the performance of all the algorithms improves as the size of the datasets increases. For example, the ISGD approach crosses the IA baseline for MovieLens-100K, MovieLens-1M, and MovieLens-10M at $\epsilon = 2$, $\epsilon = 0.9$, and $\epsilon = 0.7$,

respectively. This result is not surprising; the larger the dataset, the more resilient it is to the noise introduced through differential privacy. Since the noise is calibrated to mask the effect of a single rating, larger datasets provide a higher signal-to-noise ratio, thereby allowing better performance with respect to the baseline for any particular value of ϵ.

We observe, however, that this trend does not apply to the Netflix dataset, despite the fact that it is an order of magnitude larger than the MovieLens-10M dataset and has a similar density of ratings. We attribute this to the lower number of user ratings observed in the Netflix dataset. For example, in the PALS algorithm, the output perturbation noise added to the user vectors is proportional to $\exp(-\epsilon \cdot n_u)$. While a different distribution of the privacy budget between the computation of P and Q allows a slight improvement in RMSE, it is not sufficient to compensate for the lower values of n_u prevalent in the Netflix dataset.

As expected, crossing of the IA baseline is observed for lower values of ϵ than crossing of the GE baseline. This observation is explained by the lower degree of personalization offered by IA, which is achievable with higher levels of noise and, thus, a higher degree of privacy. For all the datasets, the IA crossing ϵ values of the algorithms are similar, but there is a substantial difference between the GE crossing ϵ values. Specifically, the IA crossing ϵ of PSGD and PALS are very close, and both are slightly lower than that of the ISGD approach. However, the GE crossing ϵ of ISGD is much lower than the ϵ of both PSGD and PALS.

For example, consider the MovieLens-10M dataset. The PSGD and PALS approaches cross the IA baseline at $\epsilon = 0.6$, whereas ISGD crosses this baseline at $\epsilon = 0.7$. This stems from the fact that the matrices P and Q in PALS and PSGD are bounded with $L2$-norm bounds p_{max} and q_{max}. Bounding the $L2$-norm of the matrices provides a small improvement to low values of ϵ and gives PALS and PSGD this slightly earlier crossing. However, for higher ϵ, ISGD achieves a better performance, i.e., it crosses the GE baseline at $\epsilon = 2.1$, whereas PSGD and PALS cross it at $\epsilon = 5.5$ and $\epsilon = 6$, respectively. Similar trade-offs are observed also in the experiments involving the other datasets.

5.2.1 Impact of data preprocessing

We briefly demonstrate the observed effect of preprocessing on the performance of the algorithms. Figure 3a shows two variants of the Private Stochastic Gradient Descent (PSGD) algorithm, both evaluated using the MovieLens-1M dataset. The figure shows the RMSE curve of the PSGD algorithm extracted from Fig. 2a and the same PSGD algorithm with the same parameters, but applied without the data preprocessing step. As can be clearly seen, data preprocessing has a significant effect on the RMSE, as it reduces the sensitivity of the introduced noise, in particular for low values of ϵ, when the IA baseline crossing is considered. Similar results were obtained for the PALS and ISGD algorithms and for other datasets. These results highlight the importance of data preprocessing and resemble earlier results obtained for the non-private variants of MF (Koren and Bell 2015).

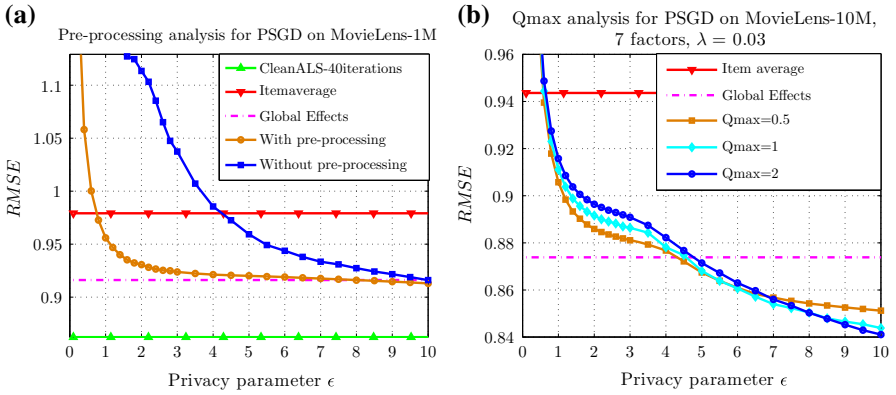

Fig. 3 Impact of preprocessing and bounding. **a** The impact of data preprocessing. **b** The impact of factor vector $L2$-norm bounds

5.2.2 Impact of factor vector L2-norm bounds

We also demonstrate the profound effect of bounding q_{max} in the PSGD algorithm. Specifically, we set p_{max} to 80 % of the q_{max} value, while the regularizer and the number of factors are fixed to the baseline value, i.e., $\lambda = 0.03$ and $d = 7$. We repeat the experiment using three values of q_{max}: $q_{max} = 0.5$, $q_{max} = 1$, and $q_{max} = 2$. Figure 3b shows the results obtained for the MovieLens-10M dataset. While the value of q_{max} does not substantially influence the IA baseline crossing, it does change both the GE crossing ϵ value and the accuracy achieved for higher values of ϵ. In PSGD, the $L2$-norm bounds do not affect the value of noise added to the matrices P and Q and are used only to control the $L2$-norm of the latent user and item vectors. For low values of ϵ, more noise is added and a small bound, e.g., $q_{max} = 0.5$, is preferable over greater bounds, since it eliminates the impact of the noisy elements. However, for higher values of ϵ, using a small bound prevents the factorization from fully realizing the potential of the chosen number of factors, and, therefore, a higher bound allows to achieve a better prediction accuracy when less noise is added.

5.3 Unbounded differential privacy

In this section we evaluate the unbounded variant of the differentially-private MF algorithms. As explained in Sect. 3.2, unbounded differential privacy offers stronger privacy guarantees than the bounded variant, since it protects not only the value of rating but also their existence. For the algorithms to respect the unbounded privacy variant, the preprocessing step has to be performed in a differentially-private manner that will protect the existence of ratings. Then, we can apply the MF algorithms, taking into account the global sensitivity of the computations under the unbounded privacy guarantees, as explained in previous sections.

For the unbounded experiments we set the algorithm parameters (privacy budget, factors, regularizer, and others) to the same values as in the bounded case. The exact

Table 5 Summary of the results for unbounded differential privacy

	100K	1M	10M	Netflix
Private global effects				
IA crossing	1.4	0.55	0.4	0.35
Input perturbation				
IA crossing	4.8	2	–	–
GE crossing	12.5	5.3	–	–
Stochastic gradient perturbation				
IA crossing	5.5	2	1.5	1.2
GE crossing	60	7	5	7.5
ALS with output perturbation				
IA crossing	5.7	2	1.5	1.2
GE crossing	20	8	6	6.5

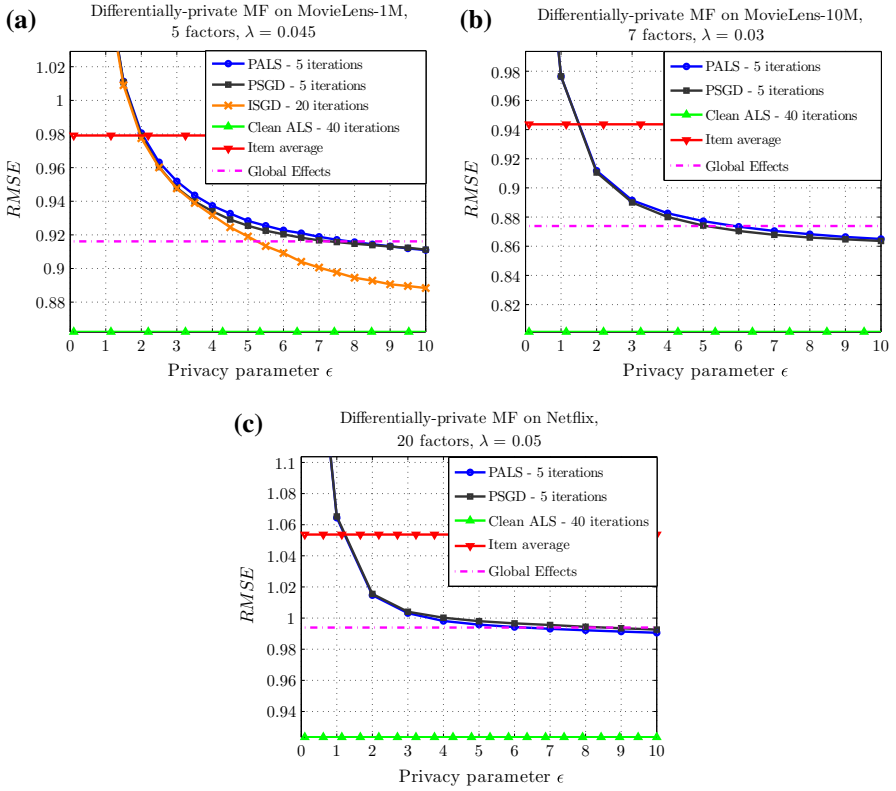

Fig. 4 Unbounded differentially-private matrix factorization. **a** MovieLens-1M. **b** MovieLens-10M. **c** Netflix

details of the parametrization can be found in Table 3. The input perturbation clamping parameter was set in this experiment to $\alpha = 0.6$. Table 5 summarizes the baseline crossing ϵ values in the unbounded differential privacy experiments and Fig. 4a–c shows the

performance of the algorithms observed using the MovieLens-1M, MovieLens-10M, and Netflix datasets.[5] The results generally resemble those obtained in the bounded experiments reported in Sect. 5.2: the baseline crossing ϵ decrease as the datasets get larger; the IA baseline crossing ϵ are comparable; and the GE crossing of ISGD is lower than that of PALS and PSGD, both of which obtain very similar ϵ.

Contrasting the results of unbounded differential privacy shown in Table 5 with the bounded case reported in Table 4, we observe a notable difference in all the baseline crossing ϵ values. Namely, in the unbounded privacy setting, the achieved IA and GE crossing ϵ values are higher than in the bounded case. For example, for the MovieLens-1M dataset, the ISGD algorithm crosses the IA and GE baselines in the unbounded case at $\epsilon = 5.3$ and $\epsilon = 2$, respectively, while in the bounded case the crossing of these baselines was observed at $\epsilon = 2.7$ and $\epsilon = 0.9$. Likewise, the IA crossing of both the PSGD and PALS algorithms is $\epsilon = 2$ in the unbounded case, compared to only $\epsilon = 0.8$ observed for both in the bounded case. These results are not unexpected and can be explained by considering the fact that the privacy guarantees of the unbounded variant are stronger than those of the bounded variant, since the existence of the ratings is also masked. This increases the amount of noise that needs to be applied in order to maintain the privacy guarantees and deteriorates the accuracy of the recommendations, such that the baseline RMSE scores are attainable only for higher values of ϵ.

5.4 Comparison to other collaborative filtering approaches

In this section we compare the results of the privacy-preserving MF approach to two other privacy preserving recommendation approaches: private version of the GE baseline and private k-nearest neighbors (kNN) recommendation algorithm (Ning et al. 2015).

For the private GE algorithm, we used the following allocation for the privacy budget: 0.02ϵ for the computation of global averages, 0.54ϵ (0.19ϵ for Netflix) for the item effects and 0.44ϵ (0.79ϵ for Netflix) for the user effects. For the private kNN recommendation algorithm, we followed the approach described by McSherry and Mironov in McSherry and Mironov (2009).[6] We used a different budget allocation. 0.9ϵ was allocated for data preprocessing, out of which 0.02ϵ was used to compute the global averages, and the item averages and user averages were computed with 0.44ϵ each (0.19ϵ for item averaged and 0.69ϵ for user averages for the Netflix dataset). The remaining privacy budget of 0.1ϵ was allocated to the identification of k-nearest neighbors. It should be highlighted that, unlike in the MF algorithm, the kNN algorithm combines the differentially private movie-to-movie covariance matrix with the private user ratings, giving it an advantage over the presented differentially privacy MF algorithms.

[5] Due to technical limitations (computational time and memory requirements), the experiments with the input perturbation approach could not be conducted on the MovieLens-10M and Netflix datasets. Therefore, ISGD results are not shown in Table 5 and ISGD curves are missing from Fig. 4b, c.

[6] We approached the authors, but unfortunately differentially private implementation of the kNN algorithm outlined in McSherry and Mironov (2009) was not publicly available, such that we were not able to reproduce the exact results reported therein.

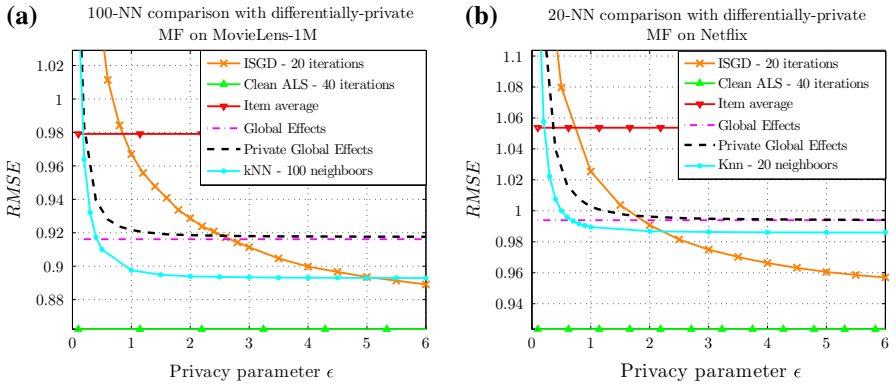

Fig. 5 Private matrix factorization and kNN comparisons. **a** Private kNN on MovieLens-1M. **b** Private kNN on Netflix

Figure 5a, b shows the results for the MovieLens-1M and Netflix datasets respectively.[7] For low values of ϵ, computing only the private GE turned out to produce results superior to those of the MF approaches. For the MovieLens-1M dataset both private kNN and private GE cross the IA baseline at $\epsilon = 0.18$, since they make a smaller number of computations than MF and, therefore, introduce the least amount of noise into the computation. For the Netflix dataset, private kNN crossed the IA baseline at a lower ϵ value than private GE, and both were lower than the IA crossing ϵ of the private MF approaches. However, private GE converged to the non-private one for higher values of ϵ and, therefore, could not take advantage of the weaker privacy constraints, when these were available.

While latent factors models like MF were typically found to outperform neighborhood based approaches like kNN in terms of their predictive accuracy (Koren et al. 2009; Ning et al. 2015; Ricci et al. 2015), our results indicate that this surprisingly is not the case in the presence of privacy constraints. For lower values of ϵ, the improved accuracy offered by MF in the non-private settings does not compensate for the higher noise required to meet the privacy guarantees. However, for a higher ϵ and weaker privacy guarantees, the predictive accuracy advantage of MF becomes apparent, and it outperforms the private kNN algorithm. While it is not impossible that there exist other differentially-private MF approaches that outperform the ones presented in this paper, we posit that neighborhood based approaches are inherently more resilient to the noise introduced by differential privacy than latent factors models.

As apparent in the statistical properties of the databases presented in Tables 2 and 4, we observe that each factor relies, on average, on a few dozens of ratings. Hence, applying even moderate noise deteriorates the signal-to-noise ratio and affects the predictions. In contrast, the private item-to-item covariance matrix relies on thousands of ratings, and is, therefore, more resilient to noise. Our experiments indicate that for lower values of ϵ, the improved accuracy offered by MF in the non-private settings,

[7] Due to memory limitations, kNN implementation for the MovieLens-10M dataset was not feasible.

does not compensate for the higher noise required to meet the privacy guarantees. Due to this, kNN outperforms MF for lower values of ϵ, when the privacy constraints are more stringent. However, for higher values of ϵ and more lenient privacy, the predictive accuracy superiority of the MF algorithms becomes apparent, and they outperform their differentially private kNN counterpart.

6 Discussion

Following our evaluation, we identified the following design choices that should be considered when applying differential privacy in practice to recommender systems.

6.1 Bounded versus unbounded differential privacy

In Sect. 5, we evaluated the privacy/accuracy trade-offs offered by two variants of differential privacy. Masking the influence of a known rating (the bounded variant) is easier than masking the mere existence of a rating in the data (the unbounded variant) (Kifer and Machanavajjhala 2011). This explains the findings that for any value of ϵ, all the algorithms supporting the bounded variant demonstrated a better utility than those supporting the unbounded variant, making them more suitable for accurate recommendations. However, the improved performance comes at a cost, as the bounded variant provides a notably weaker privacy protection. Since the existence of certain ratings could be sufficient for de-anonymization of a user or for mining their private information, the unbounded model may be more adequate from the privacy point of view.

6.2 Data and context considerations

The conducted performance evaluation of the algorithms on the Movielens and the Netflix datasets, has demonstrated that differentially private MF is better suited to datasets in which the average number of ratings per user and per item is large. We posit that this density related property of the datasets may actually be more important than the overall size of the dataset. In general, data (size, density, distribution of ratings among the users and items) and context (temporal, spatial, social) characteristics may affect the obtained privacy/personalization trade-off exhibited by the algorithms (Said et al. 2011). It should be noted that beyond these trade-offs, additional considerations may affect the design choices. For example, the scalability and flexibility of the latent model-based approaches may outweigh the advantage of neighborhood methods in terms of privacy protection, making privacy-preserving MF algorithms a viable option despite their weaker performance. Finally, methods such as input perturbation may be more amenable for data processing in a dynamic setting, since each new rating can be perturbed independently of the others, whereas for the other approaches further work is required to adapt incremental learning models to the privacy preservation setting (Li and Unger 2012).

6.3 Mind your parameters

Typically, MF parameters, such as the number of factors, the regularizer and the learning rate, are tuned to increase the predictive accuracy, while preventing overfitting and ensuring convergence. In the differentially-private setting, these considerations should be augmented to include also the impact of these parameters on the introduced noise (Hay et al. 2016). For example, increasing the number of factors typically results in larger L_2-norms of the factor vectors, and, therefore, requires larger magnitudes of noise applied to obtain the same level of privacy. In turn, this noise abolishes the increased accuracy that might have resulted from the additional factors, and an optimal setting of the parameters should balance these effects. We also found that data preprocessing has a significant impact on the performance of the proposed algorithms. In some cases, the impact of effective preprocessing might be much more important than the choice of the algorithm that obfuscates the data later on.

7 Conclusions

Privacy concerns constitute a growing impediment to the adoption and use of personalized online services. To address such concerns in the context of recommender systems, we investigated in this work the application of differential privacy to MF, the state-of-the-art approach in recommender systems. Differential privacy does not dictate a specific way to conduct a computation, but is rather a property that should be maintained by the system. Therefore, it is possible to design different implementations of algorithms that carry out the same computational task in a differentially private manner, and with a different level of effectiveness. Here, we proposed four differentially-private approaches to MF and evaluated three of them: in input perturbation, the data was protected by introducing noise prior to the data analysis, while in two other investigated approaches, differentially private variants of ALS and gradient descent were employed to guarantee privacy.

The evaluation of the proposed private MF approaches showed that input perturbation performed better than the other two approaches. Unlike the ALS and SGD approaches, in which the privacy budget should be divided and allocated separately to each iteration, the input perturbation approach could perform much better when the privacy parameter ϵ increased. We also compared private MF to other privacy preserving recommendation approaches, namely private GE and kNN. When privacy is a priority (lower ϵ is required), these approaches were found to outperform MF. While our experiments show this only for the proposed implementations of differentially-private MF, we believe this advantage is inherent to recommendations based on sparse datasets, as these approaches are less sensitive to noise than MF. On the other hand, when weaker privacy guarantees are acceptable, MF could prove to be a better alternative. In this case, the prediction accuracy of the private algorithms gets closer to that of the respective non-private variants, where MF was shown to outperform other recommendation approaches.

In summary, the application of any privacy enhancing technology to a practical recommender application requires a system designer to balance and integrate a suite

of considerations ranging beyond the theoretical guarantees offered by each approach. The designer's choices may vary depending on which aspect of the algorithm should take precedence in a particular setting (i.e., emphasis on privacy or personalization) and the properties of the data to be processed. One possible direction of future work includes the evaluation of other approaches to differentially-private MF recommendations, beyond those evaluated in this work but still conforming to the differential privacy constraints.

For example, we single out the potential of *batch gradient descent*, where the gradient is evaluated over the overall error of the ratings for a particular user or item, rather than for each individual rating (Sun et al. 2010). Also, in the private ALS algorithm, the output perturbation approach could be replaced with *objective perturbation* proposed in Chaudhuri et al. (2011), in which a noisy risk function is optimized (rather then adding noise after optimization). Finally, in *output perturbation*, the latent factors are perturbed after the non-private MF algorithm is executed, although the non-convex optimization problem would require introducing large amounts of noise, potentially resulting in poor utility. Various combinations of these MF variants with other ways to apply differential privacy can potentially bridge the apparently conflicting goals of privacy and utility in recommender systems.

Appendix: Parameterisation

Here we briefly describe the parameterization of the differentially private algorithms. We detail the results obtained for the MovieLens-100K dataset and bounded differential privacy case; however, the same methodology was also applied to other datasets.

The goal of the parameterization was to set the most appropriate values of the MF and privacy parameters. To start with, the value of the learning rate parameter was set to $\gamma = 0.01$, as in other MF implementations (Koren and Bell 2015). Next, we optimized the regularization parameter λ and the number of latent factors d. For this, we defined a fixed test set of ratings and repeated the MF predictions for various combinations of values of λ and d. These combinations included the exhaustive set of pairs within the ranges $\lambda \in [0.01, 0.15]$ and $d \in [1, 25]$. For each value of the parameters, the RMSE of the predictions for the same test set was computed. Since a 3D plot of the RMSE is hard to corroborate, Fig. 6a, b shows the 2D projections of the plot obtained for the

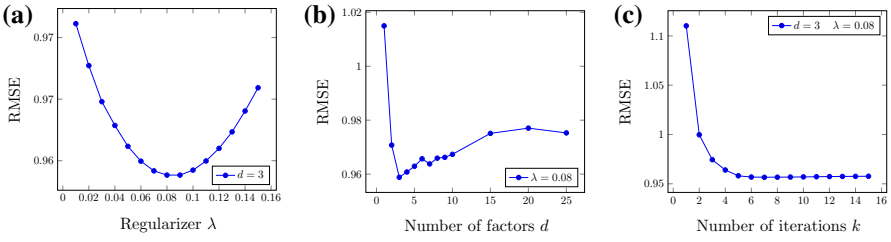

Fig. 6 Parameterization of the differentially-private matrix factorization. **a** Setting λ. **b** Setting d. **c** Setting k

fixed values of λ and d. The best performing combination of $\lambda = 0.08$ and $d = 3$ was used in the unbounded experiments with the MovieLens-100K dataset.

Having set the parameters λ and d, we turned to the number of SGD/ALS iterations, k. For this, we gradually increased the number of iterations from $k = 1$ to $k = 15$, and for each value of k computed the RMSE obtained for the fixed test set. The results of this experiment are shown in Fig. 6c. As expected, the RMSE values stabilise starting from a certain value of k. For example, in this case RMSE is reasonably stable after $k = 7$, such that we parameterize the number of iterations to $k = 10$.

References

Berkovsky, S., Eytani, Y., Kuflik, T., Ricci, F.: Hierarchical neighborhood topology for privacy enhanced collaborative filtering. In: Proceedings of Workshop on Privacy-Enhanced Personalization, PEP 2006, Montreal, Canada, pp. 6–13 (2006)

Berkovsky, S., Kuflik, T., Ricci, F.: The impact of data obfuscation on the accuracy of collaborative filtering. Expert Systems with Applications **39**(5), 5033–5042 (2012)

Berlioz, A., Friedman, A., Kâafar, M.A., Boreli, R., Berkovsky, S.: Applying differential privacy to matrix factorization. In: Proceedings of the 9th ACM Conference on Recommender Systems, RecSys 2015, Vienna, Austria, pp. 107–114 (2015). doi:10.1145/2792838.2800173

Bhaskar, R., Laxman, S., Smith, A.D., Thakurta, A.: Discovering frequent patterns in sensitive data. In: Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD 2010, Washington, DC, USA, pp. 503–512 (2010). doi:10.1145/1835804.1835869

Bilge, A., Gunes, I., Polat, H.: Robustness analysis of privacy-preserving model-based recommendation schemes. Expert Systems With Applications **41**(8), 3671–3681 (2014)

Calandrino, J.A., Kilzer, A., Narayanan, A., Felten, E.W., Shmatikov, V.: "You Might Also Like": Privacy risks of collaborative filtering. In: Proceedings of the 32nd IEEE Symposium on Security and Privacy, S&P 2011, Berkeley, CA, USA, pp. 231–246 (2011). doi:10.1109/SP.2011.40

Canny, J.F.: Collaborative filtering with privacy. In: Proceedings of the 23rd IEEE Symposium on Security and Privacy, S&P 2002, Berkeley, CA, USA, pp. 45–57 (2002). doi:10.1109/SECPRI.2002.1004361

Chaudhuri, K., Monteleoni, C., Sarwate, A.D.: Differentially private empirical risk minimization. Journal of Machine Learning Research **12**, 1069–1109 (2011)

Cheng, Z., Hurley, N.: Trading robustness for privacy in decentralized recommender systems. In: Proceedings of the 21st Conference on Innovative Applications of Artificial Intelligence, IAAI 2009, Pasadena, CA, USA (2009)

Dwork, C.: Differential privacy: A survey of results. In: Proceedings of the 5th International Conference on Theory and Applications of Models of Computation, TAMC 2008, Xi'an, China, pp. 1–19 (2008). doi:10.1007/978-3-540-79228-4_1

Dwork, C., McSherry, F., Nissim, K., Smith, A.: Differential privacy – a primer for the preplexed. In: Joint UNECE/Eurostat work session on statistical data confidentiality. Tarragona, Spain (2011)

Dwork, C., McSherry, F., Nissim, K., Smith, A.D.: Calibrating noise to sensitivity in private data analysis. In: Proceedings of the 3rd Theory of Cryptography Conference, TCC 2006, New York, NY, USA, pp. 265–284 (2006). doi:10.1007/11681878_14

Erlingsson, Ú., Pihur, V., Korolova, A.: RAPPOR: Randomized aggregatable privacy-preserving ordinal response. In: Proceedings of the ACM SIGSAC Conference on Computer and Communications Security, CCS 2014, Scottsdale, AZ, USA, pp. 1054–1067 (2014). doi:10.1145/2660267.2660348

Friedman, A., Knijnenburg, B., Vanhecke, K., Martens, L., Berkovsky, S.: Privacy aspects of recommender systems. In: Ricci, F., Rokach, L., Shapira, B. (eds.) Recommender Systems Handbook, pp. 649–688. Springer, (2015)

Friedman, A., Schuster, A.: Data mining with differential privacy. In: Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD 2010, Washington, DC, USA, pp. 493–502 (2010). doi:10.1145/1835804.1835868

Hardt, M., Talwar, K.: On the geometry of differential privacy. In: Proceedings of the 42nd ACM Symposium on Theory of Computing, STOC 2010, Cambridge, MA, USA, pp. 705–714 (2010). doi:10.1145/1806689.1806786

Harper, F.M., Konstan, J.A.: The movielens datasets: History and context. ACM Transactions on Interactive Intelligent Systems **5**(4), 19 (2016)

Hay, M., Machanavajjhala, A., Miklau, G., Chen, Y., Zhang, D.: Principled evaluation of differentially private algorithms using DPBench. In: Proceedings of the International Conference on Management of Data, SIGMOD 2016, San Francisco, CA, USA, pp. 139–154 (2016). doi:10.1145/2882903.2882931

Jeckmans, A.J., Beye, M., Erkin, Z., Hartel, P., Lagendijk, R.L., Tang, Q.: Privacy in recommender systems. In: Ramzan, N., van Zwol, R., Lee, J.S., Clüver, K., Hua, X.S. (eds.) Social Media Retrieval, pp. 263–281. Springer, (2013)

Kifer, D., Machanavajjhala, A.: No free lunch in data privacy. In: Proceedings of the ACM SIGMOD International Conference on Management of Data, SIGMOD 2011, Athens, Greece, 2011, pp. 193–204 (2011). doi:10.1145/1989323.1989345

Klösgen, W.: Anonymization techniques for knowledge discovery in databases. In: Proceedings of the 1st International Conference on Knowledge Discovery and Data Mining, KDD 1995, Montreal, Canada, pp. 186–191 (1995)

Kobsa, A.: Privacy-enhanced web personalization. In: Brusilovsky, P., Kobsa, A., Nejdl, W. (eds.) The Adaptive Web, pp. 628–670. Springer, (2007)

Koren, Y., Bell, R.: Advances in collaborative filtering. In: Ricci, F., Rokach, L., Shapira, B. (eds.) Recommender Systems Handbook, pp. 77–118. Springer, (2015)

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

Kosinski, M., Stillwell, D., Graepel, T.: Private traits and attributes are predictable from digital records of human behavior. Proceedings of the National Academy of Sciences **110**(15), 5802–5805 (2013)

Lam, S.K., Frankowski, D., Riedl, J.: Do you trust your recommendations? An exploration of security and privacy issues in recommender systems. In: Proceedings of the International Conference on Emerging Trends in Information and Communication Security, ETRICS 2006, Freiburg, Germany, pp. 14–29 (2006). doi:10.1007/11766155_2

Li, T., Unger, T.: Willing to pay for quality personalization? Trade-off between quality and privacy. European Journal of Information Systems **21**(6), 621–642 (2012)

Machanavajjhala, A., Korolova, A., Sarma, A.D.: Personalized social recommendations - accurate or private? Proceedings of the VLDB Endowment **4**(7), 440–450 (2011)

McSherry, F., Mironov, I.: Differentially private recommender systems: Building privacy into the netflix prize contenders. In: Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD 2009, pp. 627–636 (2009). doi:10.1145/1557019.1557090

Narayanan, A., Shmatikov, V.: Robust de-anonymization of large sparse datasets. In: Proceedings of the 29th IEEE Symposium on Security and Privacy, (S&P 2008), Oakland, CA, USA, pp. 111–125 (2008). doi:10.1109/SP.2008.33

Netflix spilled your Brokeback Mountain secret. http://www.wired.com/threatlevel/2009/12/netflix-privacy-lawsuit/. Accessed: July 2016

Nikolaenko, V., Ioannidis, S., Weinsberg, U., Joye, M., Taft, N., Boneh, D.: Privacy-preserving matrix factorization. In: Proceedings of the ACM SIGSAC Conference on Computer and Communications Security, CCS 2013, Berlin, Germany, pp. 801–812 (2013). doi:10.1145/2508859.2516751

Ning, X., Desrosiers, C., Karypis, G.: A comprehensive survey of neighborhood-based recommendation methods. In: Ricci, F., Rokach, L., Shapira, B. (eds.) Recommender Systems Handbook, pp. 37–76. Springer, (2015)

Parameswaran, R., Blough, D.M.: Privacy preserving collaborative filtering using data obfuscation. In: Proceedings of the IEEE International Conference on Granular Computing, GrC 2007, San Jose, CA, USA, pp. 380–386 (2007). doi:10.1109/GRC.2007.129

Polat, H., Du, W.: Achieving private recommendations using randomized response techniques. In: Proceedings of the 10th Pacific-Asia Conference on Knowledge Discovery and Data Mining, PAKDD 2014, Singapore, Singapore, pp. 637–646 (2006). doi:10.1007/11731139_73

Ricci, F., Rokach, L., Shapira, B. (eds.): Recommender Systems Handbook, 2nd edn. Springer, (2015)

Said, A., Berkovsky, S., De Luca, E.W., Hermanns, J.: Challenge on context-aware movie recommendation: Camra2011. In: Proceedings of the ACM Conference on Recommender Systems, RecSys 2011, Chicago, IL, USA, pp. 385–386 (2011). doi:10.1145/2043932.2044015

Sandhu, R.S., Coyne, E.J., Feinstein, H.L., Youman, C.E.: Role-based access control models. IEEE Computers **29**(2), 38–47 (1996)

Sarwar, B.M., Karypis, G., Konstan, J.A., Riedl, J.: Analysis of recommendation algorithms for e-commerce. In: Proceedings of the ACM Conference on Electronic Commerce, Minneapolis, MN, USA, pp. 158–167 (2000). doi:10.1145/352871.352887

Sun, X., Kashima, H., Matsuzaki, T., Ueda, N.: Averaged stochastic gradient descent with feedback: An accurate, robust, and fast training method. In: Proceedings of the 10th IEEE International Conference on Data Mining, ICDM 2010, Sydney, Australia, pp. 1067–1072 (2010). doi:10.1109/ICDM.2010.26

Sweeney, L.: *k*-anonymity: A model for protecting privacy. International Journal on Uncertainty, Fuzziness and Knowledge-based Systems **10**(5), 557–570 (2002)

Vallet, D., Friedman, A., Berkovsky, S.: Matrix factorization without user data retention. In: Proceedings of the 18th Pacific-Asia Conference on Knowledge Discovery and Data Mining, PAKDD 2014, Tainan, Taiwan, pp. 569–580 (2014). doi:10.1007/978-3-319-06608-0_47

Weinsberg, U., Bhagat, S., Ioannidis, S., Taft, N.: BlurMe: Inferring and obfuscating user gender based on ratings. In: Proceedings of the 6th ACM Conference on Recommender Systems, RecSys 2012, Dublin, Ireland, pp. 195–202 (2012). doi:10.1145/2365952.2365989

Zhou, Y., Wilkinson, D.M., Schreiber, R., Pan, R.: Large-scale parallel collaborative filtering for the netflix prize. In: Proceedings of 4th International Conference on Algorithmic Aspects in Information and Management, AAIM 2008, Shanghai, China, pp. 337–348 (2008). doi:10.1007/978-3-540-68880-8_32

Arik Friedman received a Ph.D. in Computer Science from the Technion, Israel Institute of Technology, and M.B.A. with specialization in Technology and Information Systems from Tel-Aviv University. This work was conducted while he worked as a Senior Researcher at NICTA's (National ICT Australia) Mobile Systems Research Group, specializing in privacy-preserving data mining and computer security. He previously held the position of Program Manager at Microsoft R&D between 2007 and 2011, where he worked on privacy and security products.

Shlomo Berkovsky CSIRO, Data61, Australia. Shlomo Berkovsky is a Senior Researcher and leads the Interactive Behavior Analytics team of Data61. He received his B.Sc., M.Sc., and Ph.D. degrees in Computer Science from the University of Haifa. Since 2007 he has been a researcher and then senior researcher at CSIRO, where he works in the areas of personalization, recommender systems, and persuasive technologies. He has authored more than a hundred journal and conference papers, edited several books and special issues, and chaired over a dozen conferences and workshops.

Mohamed Ali Kaafar CSIRO, Data61, Australia. Dr. Mohamed Ali Kaafar is the Group leader of the Networks research Group at CSIRO Data61. He also holds the position of visiting professor at the Chinese Academy of Science (CAS). Dr. Kaafar obtained an Engineering degree, an M.S. and a Ph.D. in Computer Science from Ecole Polytechnique Nice Sophia Antipolis. He has worked in several areas of networked systems including performance analysis and modeling, network security, and online privacy. He has co-authored over 200 scientific peer-reviewed papers, repeatedly publishing in prestigious venues.