Published in *Privacy Preserving Machine Learning*, NeurIPS 2018 Workshop, Montreal, Canada, December 8, 2018.

How to Profile Privacy-Conscious Users in Recommender Systems

Fabrice Benhamouda
IBM Research
Yorktown Heights, NY, USA

Marc Joye
OneSpan
Brussels, Belgium

Abstract

Matrix factorization is a popular method to build a recommender system. In such a system, existing users and items are associated to a low-dimension vector called a *profile*. The profiles of a user and of an item can be combined (via inner product) to predict the rating that the user would get on the item. One important issue of such a system is the so-called cold-start problem: how to allow a user to learn her profile, so that she can then get accurate recommendations?

While a profile can be computed if the user is willing to rate well-chosen items and/or provide supplemental attributes or demographics (such as gender), revealing this additional information is known to allow the analyst of the recommender system to infer many more personal sensitive information. We design a protocol to allow privacy-conscious users to benefit from matrix-factorization-based recommender systems while preserving their privacy. More precisely, our protocol enables a user to learn her profile, and from that to predict ratings without the user revealing any personal information. The protocol is secure in the standard model against semi-honest adversaries.

1 Introduction

Matrix factorization [4, 8] is a popular method to build a recommender system. As exemplified by the *Netflix Prize* competition [14], it has become a dominant technology within collaborative-filtering recommenders. Matrix factorization provides a better predictive accuracy compared to classical neighborhood methods while at the same time is scalable and offers much flexibility for modeling a variety of real-life situations [10].

The cold-start problem. A major problem facing collaborative-filtering recommender systems is how to provide recommendations when rating data is too sparse for a subset of users or items. As a special case, the so-called *cold-start problem* [20] is how to make recommendations to new users who have not yet rated any item or to deal with new items that have not yet been rated by users.

The cold-start problem is usually addressed by incorporating additional input sources to compensate for the lack of rating data. In addition to ratings, the analyst may for example collect certain user attributes, such as gender, age, or other demographic information [1, 9].

Another approach for dealing with the cold-start problem is to ask users to rate a minimum number of (well chosen) items [18].

Inference attacks. Relying on additional input sources to address the cold-start problem may be difficult to deploy in practice as privacy-conscious users may be reluctant to supply some of their attributes. The second approach does not require to collect extra information beyond the ratings and is very efficient. Unfortunately, the additional received ratings may reveal a lot of information about a user to the analyst. Recent research has indeed demonstrated that this can be used by the analyst to infer private user attributes such as political affiliation [11, 19], sexual orientation [11], age [22], gender [19, 22], and even drug use [11]. Further privacy threats are reported in [2, 3, 13].

A natural question therefore raised in [7] is how a privacy-conscious user can benefit from recommender systems while preventing the inference of her private information.

Contributions. In this paper, we show how a privacy-conscious user can learn her profile without revealing any information to the analyst. The protocol is practical and proven secure against semi-honest adversaries. The communication complexity of the protocol only grows with the square-root of the number of items.

Once the privacy-conscious user learns her profile $\boldsymbol{u}$, she can run a straightforward protocol to learn the predicted rating of any item j in the database. This indeed only requires to compute the inner product between the user profile $\boldsymbol{u}$ (known to the user) and the item profile $\boldsymbol{v_j}$ (known to the analyst) [15, Sect. 4.1].

Related work. In [7], Ioannidis et al. propose a learning protocol which enables the user to prevent the analyst from learning some (previously defined) private user attributes. This protocol perfectly hides these chosen attributes to the analyst, in an information-theoretic way. The authors also prove that no such protocol can be more accurate, when the analyst ends up knowing the resulting profile, nor can disclose less information for the same accuracy.

Unfortunately, this protocol has also several drawbacks, most of them inherent to the fact it is information-theoretically secure and does not rely on computational assumptions. First, this protocol still needs to disclose some information about the analyst database to everybody. Second, this protocol is not as accurate as a non-privacy-preserving protocol would be. This is inherent to the fact that Ioannidis et al. restricted themselves to protocols where the analyst learns an approximate profile of the user at the end, so that the resulting user profile shall not contain any information about the private attribute. Third, it can only hide a small fixed set of attributes: all attributes which are not explicitly hidden may be recovered by the analyst. And it may be hard for a user to decide which attributes are really important to her, due to the wide range of possible attributes. Finally, the analyst needs to ask users[1] to reveal which attributes they deem private. This may not only bother a lot these users, but also brings up the question of the reliability of these data. No user will be likely admitting she is a drug addict, for example, even if she is ensured that this data will not be disclosed.[2]

2 Preliminaries

$\mathbb{R}$ is the field of real numbers. For any integer n, $\mathbb{Z}_n$ is the ring of integers modulo n, while $\mathbb{Z}_n^*$ is its multiplicative group. Vectors are always column vectors and are denoted as $\boldsymbol{u}$ or $\boldsymbol{v_j}$. Matrices are denoted with capital letters.

2.1 Cryptographic tools

Public-key encryption. A *public-key encryption scheme* is defined by three algorithms: KeyGen, Enc, and Dec. $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KeyGen}(1^\kappa)$ generates a matching pair of public key pk and secret key sk, given a security parameter 1^κ (unary notation). The public key pk is used to encrypt a message $x \in \mathcal{M}$ into a ciphertext c: $c \leftarrow \mathsf{Enc}(\mathsf{pk}, x)$. The secret key sk is used to decrypt a ciphertext c: $x \leftarrow \mathsf{Dec}(\mathsf{sk}, c)$. We assume that the encryption scheme is perfectly correct and semantically secure (i.e., IND-CPA) [6].

Homomorphic encryption. An *additively homomorphic encryption scheme* is such that the message set $\mathcal{M}$ is an additive group, and there exists a randomized operation $\boxplus$ such that

[1] In the simplest scenario, we have to restrict to non-privacy-conscious users. But it would also be possible to compute item profiles using privacy-preserving matrix factorization [15].

[2] Notice in particular that, in the privacy-preserving matrix factorization protocol in [15], in case of collusion between the CSP (Crypto Service Provider) and the analyst, it is possible to recover all data sent by the user. This means that governmental agencies may force the recommendation systems to disclose these private user attributes.

$\mathsf{Enc}(\mathsf{pk}, x) \boxplus \mathsf{Enc}(\mathsf{pk}, y)$ is distributed identically to a fresh ciphertext of $x + y$. This operation can be extended to a scalar multiplication by an integer k: $k \boxdot \mathsf{Enc}(\mathsf{pk}, x)$ is a fresh ciphertext of $k \cdot x$; that is, $x + x + \cdots + x$ (k times).

To simplify the notation, we will sometimes use $[\![x]\!]_{\mathsf{pk}}$ for $\mathsf{Enc}(\mathsf{pk}, x)$ and omit pk when clear from the context. We so have $[\![x + y]\!] = [\![x]\!] \boxplus [\![y]\!]$ and $[\![k \cdot x]\!] = k \boxdot [\![x]\!]$.

Example 1 (Paillier encryption scheme). We recall the Paillier encryption scheme [17], which is an homomorphic encryption scheme that is semantically secure under the Decisional Composite Residuosity (DCR) assumption. $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KeyGen}(1^\kappa)$ generates two large equal-length primes p and q, computes $n = pq$, and sets $\lambda = \operatorname{lcm}(p - 1, q - 1)$ and $\mu = \lambda^{-1} \bmod n$. The public key is $\mathsf{pk} = n$ while the secret key is $\mathsf{sk} = (n, \lambda, \mu)$. $c \leftarrow \mathsf{Enc}(\mathsf{pk}, x)$ picks a uniformly random integer $\rho \leftarrow [1, n)$ and returns $c = (1 + x\, n)\rho^n \bmod n^2$. $x \leftarrow \mathsf{Dec}(\mathsf{sk}, c)$ returns $x = L(c^\lambda \bmod n) \cdot \mu \bmod n$ where $L(a) = (a - 1)/n$. The scheme is additively homomorphic: given $c_x \leftarrow \mathsf{Enc}(\mathsf{pk}, x)$ and $c_y \leftarrow \mathsf{Enc}(\mathsf{pk}, y)$, $c_x \boxplus c_y = c_x \cdot c_y \cdot \theta^n \bmod n^2$ with $\theta \leftarrow [1, n)$.

Oblivious transfer. A 1-*out-of-M oblivious transfer (OT) protocol* is a cryptographic protocol between two parties: a sender and a receiver. The receiver has an index $j^* \in \{1, \ldots, M\}$ as input. The sender knows a database $\{x_j\}_{j=1,\ldots,M}$. At the end of the protocol, the receiver learns x_{j^*}, while the sender learns nothing.

As in our protocol M is the number of items in the database, we need to use practical OT protocols with communication complexity sublinear in M. We propose to use as 1-out-of-M OT the basic PIR (Private Information Retrieval) protocol in [16, Sect. 2.2] using the Paillier homomorphic encryption scheme, together with a classical 1-out-of-$\sqrt{M}$ OT [12] which is used to mask the PIR database. The resulting OT has two rounds (one message from the receiver to the sender followed by one message from the sender to the receiver) and its communication complexity is proportional to $\sqrt{M}$.

2.2 Matrix factorization

The goal of matrix factorization is to predict *unobserved* ratings $r_{i,j}$ for some user i and some item j, given access to a set $\mathcal{D}$ of user/item pairs (i, j) for which a rating $r_{i,j} \in \mathbb{R}$ has been generated. Matrix factorization provides d-dimensional vectors $\boldsymbol{u_i}, \boldsymbol{v_j} \in \mathbb{R}^d$ such that

$$r_{i,j} \approx \hat{r}_{i,j} := \langle \boldsymbol{u_i}, \boldsymbol{v_j} \rangle = \sum_{k=1}^{d} u_{i,k}\, v_{j,k} \quad \text{for } (i, j) \in \mathcal{D} \ . \tag{1}$$

This allows the analyst to predict missing ratings (i.e., those with $(i, j) \notin \mathcal{D}$). Vector $\boldsymbol{u_i}$ is referred to as the *profile of user* i while vector $\boldsymbol{v_j}$ as the *profile of item* j.

2.3 Learning the profile of a user

Specifically, when a new user wishes to use the service, she submits a batch of s ratings $\{r_j\}_{j\in\mathcal{S}}$ for a subset $\mathcal{S}$ of $s \geq d$ items. Upon receiving these ratings, the analyst can estimate her profile $\boldsymbol{u}$ through the following least-squares estimation,[3]

$$\operatorname*{arg\,min}_{\boldsymbol{u}\in\mathbb{R}^d} \sum_{j\in\mathcal{S}} \bigl(r_j - \langle \boldsymbol{u}, \boldsymbol{v_j} \rangle\bigr)^2 \,, \tag{2}$$

and subsequently predict ratings for items $j \notin \mathcal{S}$, using Eq. (1).

Defining matrix $V_{\mathcal{S}} = (\ \boldsymbol{v_{j_1}} \mid \ldots \mid \boldsymbol{v_{j_s}}\ ) \in \mathbb{R}^{d\times s}$ and column vector $\boldsymbol{r} = (r_{j_1}, \ldots, r_{j_s})^\intercal \in \mathbb{R}^s$, the profile $\boldsymbol{u}$ of a user can be computed as follows:

$$\boldsymbol{u} = (V_S \cdot V_S^\intercal)^{-1} \cdot V_S \cdot \boldsymbol{r} = \left(\sum_{k=1}^{s} \boldsymbol{v_{j_k}} \cdot \boldsymbol{v_{j_k}^\intercal} \right)^{-1} \cdot \left(\sum_{k=1}^{s} r_{j_k} \cdot \boldsymbol{v_{j_k}} \right) \ . \tag{3}$$

[3] To ease the presentation, linear regression is considered but the proposed techniques readily apply to the more general setting of ridge regression.

3 Our learning protocol

We design a two-round learning protocol between a privacy-conscious user i and an analyst, allowing the user to learn her profile $\boldsymbol{u}$ from her (private) ratings $\{r_j\}_{j \in \mathcal{S}}$, where $\mathcal{S} = \{j_1, \ldots, j_s\}$. At the end of the protocol, the analyst will learn nothing (except the size s of $\mathcal{S}$), while the user will only learn her profile $\boldsymbol{u}$ and nothing else about the analyst database (except the dimension d, the database of items and its size M, and bounds B_r and B_v on entries of ratings r_j and of profiles of items v_j, respectively).

We insist that our protocol hides the set of actual items $\mathcal{S}$ that the user is rating as they might already leak significant information about her. If an upper bound S on $|\mathcal{S}|$ is known, the exact size s of $\mathcal{S}$ can trivially be masked by adding fake items (with profile $\mathbf{0}$ and fake rating 0) so that the protocol always uses a set $\mathcal{S}$ of size S.

3.1 Protocol

Consider the ring $\mathbb{Z}_n$. We assume that n is either a prime or is hard to factor, so that for all intents and purposes $\mathbb{Z}_n$ behaves as a field (since a non-zero non-invertible element of $\mathbb{Z}_n$ would yield a factor of n). Up to using fixed point arithmetic (e.g., by multiplying values by some integer 2^ℓ), we suppose that the entries of V_S and $\boldsymbol{r}$ are integers, and so can be considered as elements of $\mathbb{Z}_n$.

Round 1. The user generates a key pair $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KeyGen}(1^\kappa)$ for the homomorphic encryption scheme and encrypts her ratings r_{j_k}: $c_k \leftarrow \llbracket r_{j_k} \rrbracket$ for $1 \leq k \leq s$. She also initiates s independent OT protocols as a receiver with respective selection indexes $j_1, \ldots, j_s$.

Round 2. The analyst generates and computes the following matrices (over $\mathbb{Z}_n$ and over the ciphertext space respectively):[4]

$$A_{k,j} = R_0 \cdot \boldsymbol{v_j} \cdot \boldsymbol{v_j}^\intercal + R_k \,, \qquad\qquad \llbracket \boldsymbol{\alpha_{k,j}} \rrbracket = (R_0 \cdot \boldsymbol{v_j}) \boxdot c_k \boxplus \llbracket \boldsymbol{\rho_k} \rrbracket \,,$$

where $\{R_k\}_{1 \leq k \leq s}$ and $\{\boldsymbol{\rho_k}\}_{1 \leq k \leq s}$ are uniformly random matrices and vectors summing up to zero in $\mathbb{Z}_n^{d \times d}$ and $\mathbb{Z}_n^d$ respectively, and R_0 is a uniform matrix in $\mathsf{GL}(d, \mathbb{Z}_n)$ (the group of invertible matrices in $\mathbb{Z}_n^{d \times d}$).

The analyst then answers the k-th OT message from the user, as an OT sender with database $\{x_{k,j} = (A_{k,j}, \llbracket \boldsymbol{\alpha_{k,j}} \rrbracket)\}_{1 \leq j \leq M}$.

Final step. The user receives $x_{k,j_k} = (A_{k,j_k}, \llbracket \boldsymbol{\alpha_{k,j_k}} \rrbracket)$ through the OT protocols. We write $A_k = A_{k,j_k}$ and $\boldsymbol{\alpha_k} = \mathsf{Dec}(\mathsf{sk}, \llbracket \boldsymbol{\alpha_{k,j_k}} \rrbracket)$. We then remark that, in $\mathbb{Z}_n$:

$$(V_S \cdot V_S^\intercal)^{-1} \cdot V_S \cdot \boldsymbol{r} = \left(\sum_{k=1}^{s} A_k \right)^{-1} \cdot \left(\sum_{k=1}^{s} \boldsymbol{\alpha_k} \right) \,.$$

So if n is large enough, the user can compute back $\boldsymbol{u}$ using rational reconstruction [21, 5] (we recall that $\boldsymbol{u}$ satisfies Eq. (3) over the rationals, and that $\boldsymbol{u}$ is not necessarily an integer).

Bounds for correctness. The scheme is correct when the above rational reconstruction succeeds. From [21, 5] and Hadamard's inequality, we can show correctness when $n > 2d^{d+1/2}s^{2d+1}B_V^{4d+1}B_r$, where B_V and B_r are upper bounds on the absolute values of the coefficients of the item profiles $\boldsymbol{v_j}$ and of the ratings r_j, respectively. For example, if $B_V = 2^{20}$, $B_r = 4$, $d = 8$, $s = 10$, this is already satisfied for an integer n of 806 bits.

Security. Security against semi-honest adversaries follows from the security of the OT protocol, the IND-CPA property of the homomorphic encryption scheme, and from the following fact: since $V_S \cdot V_S^\intercal$ is invertible and $\mathsf{GL}(d, \mathbb{Z}_n)$ is a group, $\{A_k\}_k$ and $\{\boldsymbol{\alpha_k}\}_k$ only reveal $\boldsymbol{u}$.

[4] We slightly abuse notation here. For vectors, the bracket notation and $\boxplus$ and $\boxdot$ operators are applied component-wise.

3.2 Instantiation using Paillier homomorphic encryption scheme

The scheme can be instantiated using the Paillier encryption scheme and the OT described in Section 2. We can use the internal construction of the OT, to avoid sending ciphertexts of r_{j_k}. Concretely, in the OT construction, the user encrypts a vector $(0, \dots, 1, \dots, 0)$ used to "select" the correct value to be received. If we use two OT protocols for each k, one for A_k and one for $\boldsymbol{\alpha_k}$ (instead of a single one for the pair $(A_k, \boldsymbol{\alpha_k})$), then for the second OT, the user just encrypts r_{j_k} instead of 1, she will receive r_{j_k} times the value to be received.

The resulting protocol for $M = 100$ items, dimension $d = 8$, and $s = 10$ ratings from the user (modulus n of size 1024 bits for Paillier encryption scheme and an elliptic curve over a 256-bit prime field for the base OT [12]), has the following performance on a non-optimized single-thread implementation (on a laptop, CPU Intel® i7-7567U, 3.5GHz, turbo 4GHz): less than 0.4s to generate the first round by the user, less than 150s to generate the second round by the analyst, less than 1.4s to finalize the protocol by the user. The user requires less than 2s of computation (excluding communication). The analyst time is mostly spent in the exponentiations required in the OT protocol (modulo n^2): there are $M \cdot (d^2 + d) \cdot s$ of them. These exponentiations can be trivially parallelized. The communication complexity is less than 2MB and essentially grows linearly with $\sqrt{M}$.

References

[1] Gediminas Adomavicius and Alexander Tuzhilin. Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE Transactions on Knowledge and Data Engineering*, 17(6):734–749, June 2005.

[2] Smriti Bhagat, Udi Weinsberg, Stratis Ioannidis, and Nina Taft. Recommending with an agenda: Active learning of private attributes using matrix factorization. In Alfred Kobsa et al., editors, *8th ACM Conference on Recommender Systems (RecSys 2014)*, pages 65–72. ACM Press, October 2014.

[3] Joseph A. Calandrino, Ann Kilzer, Arvind Narayanan, Edward W. Felten, and Vitaly Shmatikov. "You Might Also Like:" Privacy risks of collaborative filtering. In *2011 IEEE Symposium on Security and Privacy (S&P 2011)*, pages 231–246. IEEE Press, May 2011.

[4] Emmanuel J. Candès and Benjamin Recht. Exact matrix completion via convex optimization. *Foundations of Computational Mathematics*, 9(6):717–772, 2009.

[5] Pierre-Alain Fouque, Jacques Stern, and Jan-Geert Wackers. Cryptocomputing with rationals. In Matt Blaze, editor, *6th International Conference on Financial Cryptography (FC 2002)*, volume 2357 of *LNCS*, pages 136–146. Springer, March 2003.

[6] Shafi Goldwasser and Silvio Micali. Probabilistic encryption. *Journal of Computer and System Sciences*, 28(2):270–299, 1984.

[7] Stratis Ioannidis, Andrea Montanari, Udi Weinsberg, Smriti Bhagat, Nadia Fawaz, and Nina Taft. Privacy tradeoffs in predictive analytics. In Sujay Sanghavi et al., editors, *2014 International Conference on Measurement and Modeling of Computer Systems (SIGMETRICS 2014)*, pages 57–69. ACM Press, June 2014.

[8] Raghunandan H. Keshavan, Andrea Montanari, and Sewoong Oh. Learning low rank matrices from $O(n)$ entries. In *46th Annual Allerton Conference on Communication, Control, and Computing*, pages 1365–1372. IEEE Press, September 2008.

[9] Yehuda Koren. Factorization meets the neighborhood: A multifaceted collaborative filtering model. In Ying Li, Bing Liu, and Sunita Sarawagi, editors, *14th ACM International Conference on Knowledge Discovery and Data Mining (KDD 2008)*, pages 426–434. ACM Press, August 2008.

[10] Yehuda Koren, Robert Bell, and Chris Volinsky. Matrix factorization techniques for recommender systems. *Computer*, 42(8):30–37, August 2009.

[11] Michal Kosinskia, David Stillwella, and Thore Graepel. Private traits and attributes are predictable from digital records of human behavior. *Proceedings of the National Academy of Sciences of the United States of America*, 110(15):5802–5805, April 2013.

[12] Moni Naor and Benny Pinkas. Efficient oblivious transfer protocols. In S. Rao Kosaraju, editor, *12th Annual Symposium on Discrete Algorithms (SODA 2001)*, pages 448–457. ACM-SIAM, January 2001.

[13] Arvind Narayanan and Vitaly Shmatikov. Robust de-anonymization of large sparse datasets. In *2008 IEEE Symposium on Security and Privacy (S&P 2008)*, pages 111–125. IEEE Press, May 2008.

[14] Netflix Prize. http://www.netflixprize.com/.

[15] Valeria Nikolaenko, Stratis Ioannidis, Udi Weinsberg, Marc Joye, Nina Taft, and Dan Boneh. Privacy-preserving matrix factorization. In Ahmad-Reza Sadeghi, Virgil D. Gligor, and Moti Yung, editors, *20th ACM Conference on Computer and Communications Security (ACM-CCS 2013)*, pages 801–812. ACM Press, November 2013.

[16] Rafail Ostrovsky and William E. Skeith III. A survey of single-database private information retrieval: Techniques and applications (invited talk). In Tatsuaki Okamoto and Xiaoyun Wang, editors, *10th International Conference on Practice and Theory in Public-Key Cryptography (PKC 2007)*, volume 4450 of *LNCS*, pages 393–411. Springer, April 2007.

[17] Pascal Paillier. Public-key cryptosystems based on composite degree residuosity classes. In Jacques Stern, editor, *18th Annual International Conference on the Theory and Applications of Cryptographic Techniques (EUROCRYPT '99)*, volume 1592 of *LNCS*, pages 223–238. Springer, May 1999.

[18] Al Mamunur Rashid, Istvan Albert, Dan Cosley, Shyong K. Lam, Sean M. McNee, Joseph A. Konstan, and John Riedl. Getting to know you: Learning new user preferences in recommender systems. In *7th International Conference on Intelligent User Interfaces*, pages 127–134. ACM Press, January 2002.

[19] Salman Salamatian, Amy Zhang, Flávio du Pin Calmon, Sandilya Bhamidipati, Nadia Fawaz, Branislav Kveton, Pedro Oliveira, and Nina Taft. How to hide the elephant –or the donkey– in the room: Practical privacy against statistical inference for large data. In *IEEE Global Conference on Signal and Information Processing (GlobalSIP 2013)*, pages 269–272. IEEE Press, December 2013.

[20] Andrew I. Schein, Alexandrin Popescul, Lyle H. Ungar, and David M. Pennock. Methods and metrics for cold-start recommendations. In *25th Annual International ACM Conference on Research and Development in Information Retrieval*, pages 253–260. ACM Press, August 2002.

[21] Paul S. Wang, M. J. T. Guy, and James H. Davenport. P-adic reconstruction of rational numbers. *SIGSAM Bull.*, 16(2):2–3, May 1982.

[22] Udi Weinsberg, Smriti Bhagat, Stratis Ioannidis, and Nina Taft. BlurMe: Inferring and obfuscating user gender based on ratings. In Padraig Cunningham et al., editors, *6th ACM Conference on Recommender Systems (RecSys 2012)*, pages 195–202. ACM Press, September 2012.