Incorporating Auxiliary Information in Collaborative Filtering Data Update with Privacy Preservation

Xiwei Wang, Jun Zhang,
Pengpeng Lin, Nirmal Thapa
Dept. of Computer Science
University of Kentucky
Lexington, Kentucky 40506-0633
Email: {xiwei, jzhang}@cs.uky.edu,
{m.lin, nirmalthapa}@uky.edu

Yin Wang
Dept. of Math and Computer Science
Lawrence Technological University
Southfield, Michigan, 48075
Email: ywang12@ltu.edu

Jie Wang
Dept. of Computer Information Systems
Indiana University Northwest
Gary, Indiana, 46408
Email: wangjie@iun.edu

***Abstract*—Online shopping has become increasingly popular in recent years. More and more people are willing to buy products through Internet instead of physical stores. For promotional purposes, almost all online merchants provide product recommendations to their returning customers. Some of them ask professional recommendation service providers to help develop and maintain recommender systems while others need to share their data with similar shops for better product recommendations. There are two issues, (1) how to protect customers' privacy while retaining data utility before they release the data to the third parties; (2) based on (1), how to handle data growth efficiently.**

In this paper, we propose a NMF (Nonnegative Matrix Factorization)-based data update approach in collaborative filtering (CF) that solves the problems. The proposed approach utilizes the intrinsic property of NMF to distort the data for protecting user's privacy. In addition, the user and item auxiliary information is taken into account in incremental nonnegative matrix tri-factorization to help improve the data utility. Experiments on three different datasets (MovieLens, Sushi and LibimSeTi) are conducted to examine the proposed approach. The results show that our approach can quickly update the new data and provide both high level privacy protection and good data utility.

***Keywords*—auxiliary information; collaborative filtering; data growth; nonnegative matrix factorization; privacy**

I. Introduction

The emergence of E-commerce not only helps sellers save resources and time but also facilitates online transactions. Different kinds of promotions have been adopted by merchants to advertise their products. Conventional stores usually present popular products, e.g., batteries, gift cards, and magazines at the checkout line besides offering discounts, which is a typical way of product recommendations. For returning customers, online stores are far superior with respect to product recommendation since they use users'[1] purchase history in recommender system to achieve accurate recommendation. The so called recommender system is a program that utilizes algorithms to predict users' purchase interests by profiling their shopping patterns. Most popular recommender systems utilize CF techniques, e.g., item/user correlation based CF [22], SVD (Singular Value Decomposition) based latent factor CF [24], and NMF (Nonnegative Matrix Factorization) based CF [34], [4].

In many online recommender systems, it is inevitable for data owners to expose their data to other parties. For instance, due to the lack of easy-to-use technology, some online merchants buy services from professional recommendation service providers to help build their recommender systems. In addition, many shops share their real time data with partners for better product recommendations. Such examples include two or more online book stores that sell similar books, and online movie rental websites that have similar movies in their systems. In these scenarios, exposed data can cause privacy leakage of user information if no preprocessing is done. Typical privacy information includes the ratings of a user left on particular items and on which items that this user has rated. People would not like others (except the website where they purchased the products because they have no choice) to know what they are interested in and to what extent they like or dislike the items. This is the most fundamental privacy problem in collaborative filtering. Thus privacy preserving collaborative filtering algorithms [3], [21], [19] were proposed to tackle the problem.

Most CF algorithms work on user-item rating matrices to make recommendations. These numerical matrices store user's ratings on particular items, typically with users corresponding to the rows and items corresponding to the columns. In general, the rating matrices are very sparse, meaning that there are lots of missing values. Therefore, two tasks need to be done before a data owner (merchant) releases the data to a third party: missing value imputation and data perturbation[2].

Furthermore, data owners are responsible for efficiently handling the fast growth of data. Once new data arrives, data owners need to perform incremental data update and send the imputed and perturbed data to the third parties. To this end, Wang and Zhang[30] proposed an SVD-based privacy preserving data update scheme to handle data growth efficiently and preserve privacy as well. Nevertheless, their SVD-based update scheme has a few deficiencies: (1) The SVD algorithm

[1] The terms "customer" and "user" will be used interchangeably as they refer to the same thing in this context. Same convention applies to "product" and "item".

[2] Data perturbation is a form of privacy-preserving data mining technique. It falsifies the data before publication by introducing error to elements purposely for confidentiality reasons [8]. Data perturbation is widely used in collaborative filtering for privacy preservation.

cannot be applied to incomplete matrix so missing values imputation is required. Choosing a good imputation method is not quite straightforward and it is dependant on different datasets. (2) The update scheme only utilizes the rating data while ignores other auxiliary information. It is known that in some datasets, e.g., MovieLens dataset [24], Sushi preference dataset [12] and LibimSeTi Dating Agency (LibimSeTi for short) dataset [2], auxiliary information of users or items, e.g., user's demographic data, item's categorical data, is also provided. This information, if properly used, can improve the recommendation accuracy especially when the original rating matrix is extremely sparse. (3) The time complexity of their method contains a cubic term with respect to the number of new rows or columns. It is a potentially expensive factor in the update process, especially when a large amount of new data comes in.

In this paper, we propose a NMF-based data update approach that solves the issues. The approach, named iAux-NMF is based on the incremental nonnegative matrix tri-factorization algorithms [7]. We start with computing the weighted and constrained nonnegative matrix tri-factorization for the original sparse rating matrix (with a lot of missing values), utilizing both the rating matrix itself and the auxiliary information. The factor matrices of NMF are then used to approximate the original rating matrix with missing values imputed. Meanwhile, the data is automatically perturbed due to the intrinsic properties of NMF [29]. For new data, iAux-NMF is performed to produce imputed and perturbed data. This process can conceal which items the users have rated as there is no more missing entries and disguise the true rating values since the processed ratings and the original ones are different. By doing so, even though the third party has this data in its hand, it does not know which ratings it can trust or to what extent it can trust. Therefore, user's privacy is protected.

We examine our approach in several aspects: (1) correctness of the approximated rating matrix, (2) clustering analysis on the approximated rating matrix for investigating user rating distribution, (3) privacy level of the approximated rating matrix, (4) time cost of the algorithms, and (5) parameter study. The results demonstrate that our approach imputes and perturbs the new data in a timely manner with satisfying privacy level and high data utility (less compromised data accuracy). The processed data is also reasonable from the clustering point of view.

The contributions of this paper are threefold:

1) No particular missing value imputation methods required during the data update;
2) Incorporating auxiliary information into the update process to improve data utility;
3) Higher data update efficiency.

The remainder of this paper is organized as follows. Section II gives the related work. Section III defines the problem and related notations. Section IV describes the main idea of the proposed approach. Section V presents the experiments and discusses the results. Some concluding remarks and future work are given in VI.

II. Related Work

Privacy preserving data update was first studied by Wang et al.[28] who presented a data value hiding method for clustering algorithms based on incremental SVD technique [26]. Their method can produce a significant increase in speed for the SVD-based data value hiding model, better scalability, and better real-time performance of the model. Motivated by their work, Wang and Zhang[30] incorporated the missing value imputation and randomization-based perturbation as well as a post-processing procedure into the incremental SVD to update the new data with privacy preservation in collaborative filtering.

Besides SVD, NMF has also been studied in collaborative filtering. Zhang et al.[34] applied NMF to collaborative filtering to learn the missing values in the rating matrix. They compared an expectation maximization (EM) based procedure (using NMF as its solution) with the weighted nonnegative matrix factorization (WNMF) based method which was previously applied to missing value imputation in matrix of network distances [18]. By integrating the advantages of both algorithms, they presented a hybrid method and demonstrated its effectiveness on real datasets. Chen et al.[4] proposed an orthogonal nonnegative matrix tri-factorization (ONMTF) [7] based collaborative filtering algorithm. Their algorithm also took into account the user similarity and item similarity. Our approach is generally based on the nonnegative matrix tri-factorization (NMTF) but we add further constraints to the objective function.

NMF with additional constraints has been applied to different fields. Li et al.[16] proposed nonnegative matrix factorization with orthogonality constraints for detection of a target spectrum in a given set of Raman spectra data. Hoyer et al.[10] extended NMF by adding a sparsity-inducing penalty to the objective function to include the option for explicit sparseness control. Ferdowsi et al.[9] proposed a constrained NMF algorithm for separation of active area in the brain from fMRI. In their work, prior knowledge of the sensory stimulus is incorporated into standard NMF to find new update rules for the decomposition process.

Thapa et al.[25] proposed explicit incorporation of the additional constraint, called "clustering constraint", into NMF in order to suppress the data patterns in the process of performing the matrix factorization. Their work is based on the idea that one of the factor matrices in NMF contains cluster membership indicators. The clustering constraint is another indicator matrix with altered class membership in it. This constraint then guides NMF in updating factor matrices. Enlightened by that paper, we convert users' and items' auxiliary information into cluster membership indicator matrices and apply them to NMTF as additional constraints. We do not hide data pattern, but update factor matrices in a more reasonable way for better missing value imputation.

III. Problem Description

Assume the data owner has three matrices: a sparse user-item rating matrix (denoted by $R \in \mathbb{R}^{m \times n}$), a user feature matrix (denoted by $F_U \in \mathbb{R}^{m \times k_U}$), and an item feature matrix (denoted by $F_I \in \mathbb{R}^{n \times k_I}$), where there are m users, n items, k_U user features, and k_I item features. An entry r_{ij} in R

represents the rating left on item j by user i. The valid range of rating value varies from website to website. Some use the $1 \sim 5$ scale with 1 as the lowest rating (most disliked) and 5 as the highest rating (most favorated) while some others use the $-10 \sim 10$ scale with -10 as the lowest rating, 0 as neutral rating, and 10 as the highest rating.

The original rating matrix contains the real rating values left by users on items, which means it can be used to identify the shopping patterns of users. These patterns can reveal some user privacy, so releasing the original rating data without any privacy protection will cause the privacy breach. One possible way to protect the user privacy before releasing the rating matrix is to impute the matrix and then perturb it. In this procedure, imputation estimates the missing ratings as well as conceals the user preference on particular items (no missing value means there is no way to tell which items have been rated by users since all items are marked as rated.) while the perturbation distorts the ratings so that user's preferences on particular items are blurred.

As for the user feature matrix F_U and item feature matrix F_I, they contain users' and items' information, respectively. They are taken into account to help impute the missing entries in rating matrix for better accuracy. The processed (imputed and perturbed) matrix, denoted by $R_r \in \mathbb{R}^{m\times n}$ is the one that will be handed over to the third party.

When new users' transactions arrive, the new rows (each row contains the ratings left on items by the corresponding user), denoted by $T \in \mathbb{R}^{p\times n}$, should be appended to the original matrix R. Meanwhile, this new users' auxiliary information is also available, and thus the feature matrix is updated as well, i.e.,

$$\begin{bmatrix} R \\ T \end{bmatrix} \rightarrow R' \qquad \begin{bmatrix} F_U \\ \Delta F_U \end{bmatrix} \rightarrow F'_U \tag{1}$$

where $\Delta F_U \in \mathbb{R}^{p\times k_U}$.

Similarly, when new items arrive, the new columns (each column contains the ratings left by users on the corresponding item), denoted by $G \in \mathbb{R}^{m\times q}$, should be appended to the original matrix R, so should the item feature matrix, i.e.,

$$[\ R \quad G\ ] \rightarrow R'', \qquad \begin{bmatrix} F_I \\ \Delta F_I \end{bmatrix} \rightarrow F'_I \tag{2}$$

where $\Delta F_I \in \mathbb{R}^{q\times k_I}$.

To protect users' privacy, the new rating data must be processed before it is released. We use $T_r \in \mathbb{R}^{p\times n}$ to denote the processed new rows and $G_r \in \mathbb{R}^{m\times q}$ for processed new columns.

IV. Using iAux-NMF for Privacy Preserving Data Update

In this section, we will introduce the iAux-NMF (incremental auxiliary nonnegative matrix factorization) algorithm and its application in incremental data update with privacy preservation.

A. Aux-NMF

While iAux-NMF deals with the incremental data update, we want to present the non-incremental version, named Aux-NMF beforehand. This section is organized as follows: developing the objective function, deriving the update formula, convergence analysis, and the detailed algorithm.

1) Objective Function: Nonnegative matrix factorization (NMF)[15] is a widely used dimension reduction method in many applications such as clustering [7], [13], text mining [31], [20], image processing and analysis [33], [23], data distortion based privacy preservation [11], [25], etc. NMF is also applied in collaborative filtering to make product recommendations [34], [4].

A conventional NMF is defined as follows [15],

$$R_{m\times n} \approx U_{m\times k} \cdot V^T_{n\times k} \tag{3}$$

The goal is to find a pair of orthogonal nonnegative matrices U and V (i.e., $U^TU = I, V^TV = I$) that minimize the Frobenius norm (or Euclidean norm) $\|R - UV^T\|_F$. It comes up with the objective function

$$min_{U\geq 0,V\geq 0} f(R,U,V) = \|R - UV^T\|^2_F \tag{4}$$

In this paper, we want to develop a NMF-based matrix factorization technique which takes into account the weight and constraint. It is expected to preserve the data privacy by imputing and perturbing the values during its update process.

It is worth noting that one of the significant distinctions between collaborative filtering data and other data is the missing value issue. One user may have rated only a few items and one item may receive only a small number of ratings. It results in a very sparse rating matrix which cannot be simply fed to the matrix factorization algorithms, such as SVD and NMF. Those missing values should be imputed properly during the pre-processing step. Existing imputation methods include random value imputation, mean value imputation [24], EM (Expectation Maximization) imputation [5], [32], and linear regression imputation [27], etc. Nevertheless, all of them require extra time to compute the missing values. In contrast, weighted NMF (WNMF) [34] can work with sparse matrix without separate imputation.

Given a weight matrix $W \in \mathbb{R}^{m\times n}$ that indicates the value existence in the rating matrix R (see Eq. (6)), the objective function of WNMF is

$$min_{U\geq 0,V\geq 0} f(R,W,U,V) = \|W \circ (R - UV^T)\|^2_F \tag{5}$$

where $\circ$ denotes the element-wise multiplication.

$$w_{ij} = \begin{cases} 1 & if \quad r_{ij} \neq 0 \\ 0 & if \quad r_{ij} = 0 \end{cases} \quad (w_{ij} \in W, r_{ij} \in R) \tag{6}$$

When WNMF converges, $\tilde{R} = UV^T$ is the matrix with all missing entries filled. Since the residual exists, $\tilde{R}$ is different from R, making it a perturbed version of R. As we stated in Section I, users do not want their privacy, i.e., their ratings left on particular items and on which items they have rated, to be released to other people. In WNMF, both of them are protected.

In [6], Ding et al. showed the equivalency between NMF and K-Means clustering. When given a matrix R with objects as rows and attributes as columns, the two matrices U and V produced by NMF on R describe the clustering information of the objects: each column vector of U, u_i, can be regarded as a basis and each data point r_i is approximated by a linear combination of these k bases, weighted by the components of V [17], where k is the rank of factor matrices. Thus the objects are grouped into clusters in terms of matrix U.

However, in some cases, the data matrix R can represent relationships between two sorts of objects, e.g., a user-item rating matrix in collaborating filtering applications and a term-document matrix in text mining applications. It is expected that both row (user/term) clusters and column (item/document) clusters can be obtained by performing NMF on R. Due to the intrinsic property of NMF, it is very difficult to find two matrices U and V that represent user clusters and item clusters respectively at the same time. Hence, an extra factor matrix is needed to absorb the different scales of R, U, V for simultaneous row clustering and column clustering [7]. Eq. (7) gives the objective function of NMTF(Nonnegative Matrix Tri-Factorization).

$$min_{U\geq 0,S\geq 0,V\geq 0} f(R,U,S,V) = \|R - USV^T\|_F^2 \tag{7}$$

where $U \in \mathbb{R}^{m\times k}$, $S \in \mathbb{R}^{k\times l}$, and $V \in \mathbb{R}^{n\times l}$ (U and V are orthogonal matrices).

The use of S brings in a large scale of freedom for U and V so that they can focus on row and column clustering and preserve more privacy during the factorization process. In this scheme, both U and V are cluster membership indicator matrices while S plays the role of coefficient matrix. Note that objects corresponding to rows in R are clustered into k groups and objects corresponding to columns are clustered into l groups.

With auxiliary information of users and items, we can convert the NMTF to a supervised learning process by applying cluster constraints to the objective function (7), i.e.,

$$\begin{aligned} &min_{U\geq 0,S\geq 0,V\geq 0} f(R,U,S,V,C_U,C_I) = \\ &\alpha\cdot\|R - USV^T\|_F^2 + \beta\cdot\|U - C_U\|_F^2 \\ &+\gamma\cdot\|V - C_I\|_F^2 \end{aligned} \tag{8}$$

where α, β, and γ are coefficients that control the weight of each part. C_U and C_I are user cluster matrix and item cluster matrix. They are obtained by running K-Means clustering algorithm on user feature matrix F_U and item feature matrix F_I as mentioned in Section III.

Combining (5) and (8), we develop the objective function for weighted and constrained nonnegative matrix tri-factorization, as

$$\begin{aligned} &min_{U\geq 0,S\geq 0,V\geq 0} f(R,W,U,S,V,C_U,C_I) = \\ &\alpha\cdot\|W\circ(R - USV^T)\|_F^2 + \beta\cdot\|U - C_U\|_F^2 \\ &+\gamma\cdot\|V - C_I\|_F^2. \end{aligned} \tag{9}$$

We name this matrix factorization Aux-NMF, indicating that it incorporates the user/item auxiliary information into the factorization.

2) Update Formula: In this section, we illustrate the derivation of update formulae for Aux-NMF.

Let $L = f(R,W,U,S,V,C_U,C_I)$, $X = \|W\circ(R - USV^T)\|_F^2$, $Y = \|U - C_U\|_F^2$, and $Z = \|V - C_I\|_F^2$. Take derivative of X with respect to U, S, and V:

$$\frac{\partial X}{\partial U} = -2(W\circ R)VS^T + 2W\circ(USV^T)VS^T \tag{10}$$

$$\frac{\partial X}{\partial S} = -2U^T(W\circ R)V + 2U^T[W\circ(USV^T)]V \tag{11}$$

$$\frac{\partial X}{\partial V} = -2(W\circ R)^T US + 2[W\circ(USV^T)]^T US \tag{12}$$

Take derivative of Y with respect to U, S, and V:

$$\frac{\partial Y}{\partial U} = 2U - 2C_U,\quad \frac{\partial Y}{\partial S} = \frac{\partial Y}{\partial V} = 0 \tag{13}$$

Take derivative of Z with respect to U, S, and V:

$$\frac{\partial Z}{\partial U} = \frac{\partial Z}{\partial S} = 0,\quad \frac{\partial Z}{\partial V} = 2V - 2C_I \tag{14}$$

Using (10) to (14), we get the derivatives of L:

$$\begin{aligned} \frac{\partial L}{\partial U} &= 2\alpha[W\circ(USV^T)]VS^T + 2\beta U \\ &- 2\alpha(W\circ R)VS^T - 2\beta C_U \end{aligned} \tag{15}$$

$$\begin{aligned} \frac{\partial L}{\partial V} &= 2\alpha[W\circ(USV^T)]^T US + 2\gamma V \\ &- 2\alpha(W\circ R)^T US - 2\gamma C_I \end{aligned} \tag{16}$$

$$\begin{aligned} \frac{\partial L}{\partial S} &= 2\alpha U^T[W\circ(USV^T)]V \\ &- 2\alpha U^T(W\circ R)V \end{aligned} \tag{17}$$

To obtain update formula, we use the Karush-Kuhn-Tucker (KKT) complementary condition [14] for the nonnegativity of U, S, and V. We have

$$\begin{aligned} &\{2\alpha[W\circ(USV^T)]VS^T + 2\beta U \\ &\quad - 2\alpha(W\circ R)VS^T - 2\beta C_U\}_{ij}U_{ij} = 0 \end{aligned} \tag{18}$$

$$\begin{aligned} &\{2\alpha[W\circ(USV^T)]^T US + 2\gamma V \\ &\quad - 2\alpha(W\circ R)^T US - 2\gamma C_I\}_{ij}V_{ij} = 0 \end{aligned} \tag{19}$$

$$\{2\alpha U^T[W\circ(USV^T)]V - 2\alpha U^T(W\circ R)V\}_{ij}S_{ij} = 0 \tag{20}$$

They give rise to the corresponding update formulae:

$$U_{ij} = U_{ij}\cdot\frac{[\alpha(W\circ R)VS^T + \beta C_U]_{ij}}{\{\alpha[W\circ(USV^T)]VS^T + \beta U\}_{ij}} \tag{21}$$

$$V_{ij} = V_{ij}\cdot\frac{[\alpha(W\circ R)^T US + \gamma C_I]_{ij}}{\{\alpha[W\circ(USV^T)]^T US + \gamma V\}_{ij}} \tag{22}$$

$$S_{ij} = S_{ij}\cdot\frac{[U^T(W\circ R)V]_{ij}}{\{U^T[W\circ(USV^T)]V\}_{ij}} \tag{23}$$

Assume $k, l \ll \min(m,n)$, the time complexities of updating U, V, and S in each iteration are all $O(mn(k+l))$. Therefore, the time complexity of Aux-NMF in each iteration is $O(mn(k+l))$.

3) Convergence Analysis: We follow [15] to prove that the objective function L is nonincreasing under the update formulas (21), (22), and (23).

Definition 1: $H(u, u')$ is an auxiliary function for $F(u)$ if the conditions

$$H(u, u') \geq F(u), \quad H(u, u) = F(u) \tag{24}$$

are satisfied.

Lemma 1: If H is an auxiliary function for F, then F is nonincreasing under the update

$$u^{t+1} = \underset{u}{\operatorname{argmin}}\, H(u, u^t) \tag{25}$$

Lemma 1 can be easily proved since we have $F(u^{t+1}) = H(u^{t+1}, u^{t+1}) \leq H(u^{t+1}, u^t) \leq H(u^t, u^t) = F(u^t)$.

We will prove the convergences of the update formulas (21), (22), and (23) by showing that they are equivalent to (25), with proper auxiliary functions defined.

Let us rewrite the objective function L,

$$\begin{aligned} L = {} & tr[\alpha(W \circ R)^T \cdot (W \circ R)] \\ & + tr\{-2\alpha(W \circ R)^T \cdot [W \circ (USV^T)]\} \\ & + tr\{\alpha[W \circ (USV^T)]^T \cdot [W \circ (USV^T)]\} \\ & + tr(\beta U^T U) + tr(-2\beta U^T C_U) + tr(\beta C_U^T C_U) \\ & + tr(\gamma V^T V) + tr(-2\gamma V^T C_I) + tr(\gamma C_I^T C_I) \end{aligned} \tag{26}$$

where $tr(*)$ is the trace of a matrix.

Eliminating the irrelevant terms, we define the following functions that are only related to U, V, and S, respectively.

$$\begin{aligned} L(U) = {} & tr\{-2\alpha(W \circ R)^T \cdot [W \circ (USV^T)] \\ & + \alpha[W \circ (USV^T)]^T \cdot [W \circ (USV^T)] \\ & + \beta U^T U - 2\beta U^T C_U\} \\ = {} & tr\{[-2[\alpha(W \circ R)VS^T + \beta C_U]U^T \\ & + U^T[\alpha W \circ (USV^T)VS^T] + U^T(\beta U)\} \end{aligned} \tag{27}$$

$$\begin{aligned} L(V) = {} & tr\{-2\alpha(W \circ R)^T \cdot [W \circ (USV^T)] \\ & + \alpha[W \circ (USV^T)]^T \cdot [W \circ (USV^T)] \\ & + \gamma V^T V - 2\gamma V^T C_I \\ = {} & tr\{[-2[\alpha(W \circ R)^T US + \gamma C_I]V^T \\ & + V^T[\alpha(W \circ (USV^T))^T US] + V^T(\gamma V)\} \end{aligned} \tag{28}$$

$$\begin{aligned} L(S) = {} & tr\{-2\alpha(W \circ R)^T \cdot [W \circ (USV^T)] \\ & + \alpha[W \circ (USV^T)]^T \cdot [W \circ (USV^T)] \\ = {} & tr\{[-2\alpha U^T(W \circ R)V]S^T \\ & + [\alpha U^T(W \circ (USV^T))V]S^T\} \end{aligned} \tag{29}$$

Lemma 2: For any matrices $X \in \mathbb{R}_+^{n \times n}$, $Y \in \mathbb{R}_+^{k \times k}$, $F \in \mathbb{R}_+^{n \times k}$, $F' \in \mathbb{R}_+^{n \times k}$, and X, Y are symmetric, the following inequality holds

$$\sum_{i=1}^{n} \sum_{j=1}^{k} \frac{(XF'Y)_{ij} F_{ij}^2}{F'_{ij}} \geq tr(F^T XFY) \tag{30}$$

The proof of Lemma 2 is presented in [7]. We will use this lemma to build an auxiliary function for $L(U)$ (since it is similar to $L(V)$ and $L(S)$, we will not discuss the convergences for them).

Lemma 3:

$$\begin{aligned} H(U, U') = {} & -2 \sum_{ij} \{[\alpha(W \circ R)VS^T + \beta C_U]U^T\}_{ij} \\ & + \sum_{ij} \frac{(\alpha W \circ (U'SV^T)VS^T + \beta U')_{ij} U_{ij}^2}{U'_{ij}} \end{aligned} \tag{31}$$

is an auxiliary function of $L(U)$ and the global minimum of $H(U, U')$ can be achieved by

$$U_{ij} = U'_{ij} \cdot \frac{[\alpha(W \circ R)VS^T + \beta C_U]_{ij}}{\{\alpha[W \circ (U'SV^T)]VS^T + \beta U'\}_{ij}} \tag{32}$$

Proof: We need to prove two conditions as specified in Definition 1. It is apparent that $H(U, U) = L(U)$. According to Lemma 2, we have

$$\begin{aligned} & \sum_{ij} \frac{(\alpha W \circ (U'SV^T)VS^T + \beta U')_{ij} U_{ij}^2}{U'_{ij}} \\ & = \sum_{ij} \frac{(\alpha W \circ (U'SV^T)VS^T)_{ij} U_{ij}^2}{U'_{ij}} + \sum_{ij} \frac{(\beta U')_{ij} U_{ij}^2}{U'_{ij}} \\ & \geq tr\{U^T[\alpha W \circ (USV^T)VS^T]\} + tr[U^T(\beta U)] \end{aligned} \tag{33}$$

I.e., $H(U, U') \geq L(U)$. Thus $H(U, U')$ is an auxiliary function of $L(U)$.

To find the global minimum of $H(U, U')$ with U' fixed, we take derivative of $H(U, U')$ with respect to U_{ij} and let it be zero:

$$\begin{aligned} \frac{\partial H(U, U')}{\partial U_{ij}} = {} & \{-2[\alpha(W \circ R)VS^T + \beta C_U]\}_{ij} \\ & + 2 \frac{(\alpha W \circ (U'SV^T)VS^T + \beta U')_{ij} U_{ij}}{U'_{ij}} = 0 \end{aligned} \tag{34}$$

Solving for U_{ij}, we have

$$U_{ij} = U'_{ij} \cdot \frac{[\alpha(W \circ R)VS^T + \beta C_U]_{ij}}{\{\alpha[W \circ (U'SV^T)]VS^T + \beta U'\}_{ij}} \tag{35}$$

Since the Hessian matrix $\partial^2 H(U, U')/\partial U_{ij} \partial U_{kl}$ is positive definite, $H(U, U')$ is a convex function and the minimum obtained by Eq. (35) is also the global minimum. ■

Similarly, the convergences of update formulas (23) and (22) can be proved as well.

4) Detailed Algorithm: In this section, we present the specific algorithm for Aux-NMF in collaborating filtering which is the basis of incremental Aux-NMF.

Algorithm 1 depicts the whole process of performing Aux-NMF on a rating matrix.

Though Aux-NMF will eventually converge to a local minimum, it may take hundreds or even thousands of iterations. In our algorithm, we set an extra stop criterion - the maximum iteration counts. In collaborative filtering, this value varies from 10 ∼ 100 and can generally produce good results.

Algorithm 1 Aux-NMF

```
Require:
    User-Item rating matrix: $R \in \mathbb{R}^{m \times n}$;
    User feature matrix: $F_U \in \mathbb{R}^{m \times k_U}$;
    Item feature matrix: $F_I \in \mathbb{R}^{n \times k_I}$;
    Column dimension of U: $k$;
    Column dimension of V: $l$;
    Coefficients in objective function: $\alpha$, $\beta$, and $\gamma$;
    Number of maximum iterations: $MaxIter$.
Ensure:
    Factor matrices: $U \in \mathbb{R}^{m \times k}$, $S \in \mathbb{R}^{k \times l}$, $V \in \mathbb{R}^{n \times l}$;
    User cluster membership indicator matrix: $C_U \in \mathbb{R}^{m \times k}$;
    Item cluster membership indicator matrix: $C_I \in \mathbb{R}^{n \times l}$;
    User cluster centroids: $Centroids_U$;
    Item cluster centroids: $Centroids_I$;

Cluster users into $k$ groups based on $F_U$ by K-Means algorithm $\rightarrow C_U$, $Centroids_U$;
Cluster items into $l$ groups based on $F_I$ by K-Means algorithm $\rightarrow C_I$, $Centroids_I$;
Initialize $U$, $S$, and $V$ with random values;
Build weight matrix $W$ by Eq. (6);
Set $iteration = 1$ and $stop = false$;
while $(iteration < MaxIter)$ $and$ $(stop == false)$ do
    $U_{ij} \leftarrow U_{ij} \cdot \frac{[\alpha(W \circ R)VS^T + \beta C_U]_{ij}}{\{\alpha[W \circ (USV^T)]VS^T + \beta U\}_{ij}}$;
    $V_{ij} \leftarrow V_{ij} \cdot \frac{[\alpha(W \circ R)^T US + \gamma C_I]_{ij}}{\{\alpha[W \circ (USV^T)]^T US + \gamma V\}_{ij}}$;
    $S_{ij} \leftarrow S_{ij} \cdot \frac{[U^T(W \circ R)V]_{ij}}{\{U^T[W \circ (USV^T)]V\}_{ij}}$;
    $L \leftarrow \alpha \cdot \|W \circ (R - USV^T)\|_F^2 + \beta \cdot \|U - C_U\|_F^2 + \gamma \cdot \|V - C_I\|_F^2$;
    if ($L$ increases in this iteration) then
        $stop = true$;
        Restore $U$, $S$, and $V$ to their values in last iteration.
    end if
end while
Return $U, S, V, C_U, C_I, Centroids_U$, and $Centroids_I$.
```

B. iAux-NMF

As discussed in Section III, new data can be regarded as new rows or new columns in the matrix. They are imputed and perturbed by iAux-NMF (incremental Aux-NMF) with the aid of $U, S, V, C_U, C_I, Centroids_U$, and $Centroids_I$ generated by Algorithm 1.

iAux-NMF is technically the same as Aux-NMF, but focuses on a series of new rows or new columns. Hence, in this section we will describe the incremental case of Aux-NMF by row update and column update separately.

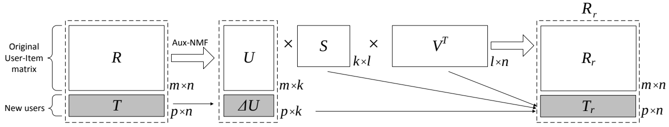


Fig. 1: Updating New Rows in iAux-NMF

1) Row/User Update: In Eq. (1), we see that $T \in \mathbb{R}^{p \times n}$ is added to R as a few rows. This process is illustrated in Fig. 1. T should be imputed and perturbed before being released. As we did in Section IV-A1, the objective function is developed here, i.e.,

$$min_{\Delta U \geq 0} f(T, W_T, \Delta U, S, V, \Delta C_U) = \alpha \cdot \|W_T \circ (T - \Delta USV^T)\|_F^2 + \beta \cdot \|\Delta U - \Delta C_U\|_F^2 \tag{36}$$

As in Section IV-A2, we obtain the update formula for this objective function, as

$$\Delta U_{ij} = \Delta U_{ij} \cdot \frac{[\alpha(W_T \circ T)VS^T + \beta \Delta C_U]_{ij}}{\{\alpha[W_T \circ (\Delta USV^T)]VS^T + \beta \Delta U\}_{ij}} \tag{37}$$

Convergence of (37) can be proved similarly as in Section IV-A3. Since row update only works on new rows, the time complexity of the algorithm in each iteration is $O(pn(l+k) + pkl)$. Assume $k, l \ll \min(p, n)$, the time complexity is then simplified to $O(pn(l+k)$.

Algorithm 2 illustrates the row update in iAux-NMF.

Algorithm 2 iAux-NMF for Row Update

```
Require:
    New rating data: $T \in \mathbb{R}^{p \times n}$;
    New user feature matrix: $\Delta F_U \in \mathbb{R}^{p \times k_U}$;
    Coefficients in objective function: $\alpha$, $\beta$, and $\gamma$;
    Factor matrices: $U \in \mathbb{R}^{m \times k}$, $S \in \mathbb{R}^{k \times l}$, $V \in \mathbb{R}^{n \times l}$;
    User cluster membership indicator matrix: $C_U \in \mathbb{R}^{m \times k}$;
    User cluster centroids: $Centroids_U$;
    Number of maximum iterations: $MaxIter$.
Ensure:
    Updated factor matrix: $U' \in \mathbb{R}^{(m+p) \times k}$;
    Updated user cluster membership indicator matrix: $C'_U \in \mathbb{R}^{(m+p) \times k}$;
    Updated user cluster centroids: $Centroids'_U$;
    Imputed and perturbed new data: $T_r \in \mathbb{R}^{p \times n}$;

Cluster new users into $k$ groups based on $\Delta F_U$ and $Centroids_U$ by K-Means algorithm $\rightarrow \Delta C_U$, $Centroids'_U$;
Initialize $\Delta U \in \mathbb{R}^{p \times k}$ with random values;
Build weight matrix $W_T$ by Eq. (6);
Set $iteration = 1$ and $stop = false$;
while $(iteration < MaxIter)$ $and$ $(stop == false)$ do
    $\Delta U_{ij} \leftarrow \Delta U_{ij} \cdot \frac{[\alpha(W_T \circ T)VS^T + \beta \Delta C_U]_{ij}}{\{\alpha[W_T \circ (\Delta USV^T)]VS^T + \beta \Delta U\}_{ij}}$
    $L \leftarrow \alpha \cdot \|W_T \circ (T - \Delta USV^T)\|_F^2 + \beta \cdot \|\Delta U - \Delta C_U\|_F^2$;
    if ($L$ increases in this iteration) then
        $stop = true$;
        Restore $U'$ to its value in last iteration.
    end if
end while
Append $\Delta C_U$ to $C_U \rightarrow C'_U$;
Append $\Delta U$ to $U \rightarrow U'$;
Calculate $\Delta USV^T \rightarrow T_r$;
Return $U', C'_U, Centroids'_U$, and $T_r$.
```

2) Column/Item Update: Column update is almost identical to row update. When new data $G \in \mathbb{R}^{m \times q}$ arrives, they are updated by Algorithm 3. The time complexity for column update is $O(qm(l+k)$.

Algorithm 3 iAux-NMF for Column Update

```
Require:
    New rating data: $G \in \mathbb{R}^{m \times q}$;
    New item feature matrix: $\Delta F_I \in \mathbb{R}^{q \times k_I}$;
    Coefficients in objective function: $\alpha$, $\beta$, and $\gamma$;
    Factor matrices: $U \in \mathbb{R}^{m \times k}$, $S \in \mathbb{R}^{k \times l}$, $V \in \mathbb{R}^{n \times l}$;
    Item cluster indicator membership matrix: $C_I \in \mathbb{R}^{n \times l}$;
    Item cluster centroids: $Centroids_I$;
    Number of maximum iterations: $MaxIter$.
Ensure:
    Updated factor matrix: $V' \in \mathbb{R}^{(n+q) \times l}$;
    Updated item cluster membership indicator matrix: $C'_I \in \mathbb{R}^{(n+q) \times l}$;
    Updated item cluster centroids: $Centroids'_I$;
    Imputed and perturbed new data: $G_r \in \mathbb{R}^{m \times q}$;

Cluster new items into $l$ groups based on $\Delta F_I$ and $Centroids_I$ by K-Means algorithm $\rightarrow \Delta C_I, Centroids'_I$;
Initialize $\Delta V \in \mathbb{R}^{q \times l}$ with random values;
Build weight matrix $W_G$ by Eq. (6);
Set $iteration = 1$ and $stop = false$;
while ($iteration < MaxIter$) $and$ ($stop == false$) do
    $\Delta V_{ij} \leftarrow \Delta V_{ij} \cdot \frac{[\alpha(W_G \circ G)^T US + \gamma \Delta C_I]_{ij}}{\{\alpha[W_G \circ (US\Delta V^T)]^T US + \gamma \Delta V\}_{ij}}$
    $L \leftarrow \alpha \cdot \|W_G \circ (G - US\Delta V^T)\|_F^2 + \gamma \cdot \|\Delta V - \Delta C_I\|_F^2$;
    if ($L$ increases in this iteration) then
        $stop = true$;
        Restore $V'$ to its value in last iteration.
    end if
end while
Append $\Delta C_I$ to $C_I \rightarrow C'_I$;
Append $\Delta V$ to $V \rightarrow V'$;
Calculate $US\Delta V^T \rightarrow G_r$;
Return $V', C'_I, Centroids'_V$, and $G_r$.
```

Data owner should hold the updated factor matrices (U', S, and V') and the cluster information (user/item cluster membership indicator matrices and centroids) for future update. Note that we leave the matrices S and V (S and U) unchanged in row update (column update), which does not indicate they will never change. We will show when Aux-NMF should be recomputed to ensure the data utility and privacy in the experimental study section.

V. Experimental Study

In this section, we discuss the test datasets, data preprocessing, evaluation strategy, and experimental results.

A. Data Description

In the experiments, we adopt MovieLens [24], Sushi [12] preference, and LibimSeTi [2] dating datasets as the test data. Table I collects the statistics of the datasets.

TABLE I: Statistics of the data

Dataset	#users	#items	#ratings	Sparsity
MovieLens	943	1,682	100,000	93.7%
Sushi	5,000	100	50,000	90%
LibimSeTi	2,000	5,625	129,281	98.85%

The public MovieLens dataset that we use has 943 users and 1,682 items. The 100,000 ratings, ranging from 1 to 5, were divided into two parts: the training set (80,000 ratings) and the test set (20,000 ratings). In addition to rating data, users' demographic information and items' genre information are also available.

The Sushi dataset describes users' preferences on different kinds of sushi. There are 5,000 users and 100 sushi items. Each user has rated 10 items, with a rating ranging from 1 to 5. That is to say, there are 50,000 ratings in this dataset. To build the test set and training set, for every user, we randomly select 2 out of 10 ratings and put them into the test set (10,000 ratings) while the rest of ratings are used as training set (40,000 ratings). Similar to MovieLens, the Sushi dataset comes with user's demographic information as well as item's group information and some attributes (e.g., the heaviness/oiliness in taste, how frequently the user eats the sushi etc.).

The LibimSeTi dating dataset is gathered by LibimSeTi.cz, an online dating website. It contains 17,359,346 anonymous ratings of 168,791 profiles made by 135,359 LibimSeTi users as dumped on April 4, 2006. However, only user's gender is provided with the data. We will show how to deal with this problem (lack of item information) in later section. Confined to the memory limitation of the test computer, we pick up 2,000 users and 5,625 items (profiles are considered as items for this dataset) with 108,281 ratings in training set and 21,000 ratings in test set. Ratings are on a 1 $\sim$ 10 scale where 10 is best.

B. Data Preprocessing

The proposed algorithms require user and item feature matrices as the input. To build such feature matrices, we pre-process the auxiliary information of users and items. In MovieLens dataset, user's demographic information includes user id, age, gender, occupation, and zip code. Amongst them, we utilize age, gender, and occupation as features. For age, the numbers are categorized into 7 groups: 1-17, 18-24, 25-34, 35-44, 45-49, 50-55, >=56. For gender, there are two possible values: male and female. As per statistics, there are 21 occupations: administrator, artist, doctor, and so on. Based on these possible values, we build a user feature matrix F_U with 30 features ($k_U = 30$), i.e., each user is represented as a row vector with 30 elements. An element will be set to 1 if the corresponding feature value is true for this user and 0 otherwise. An example is, for a 48 years old female user, who is an artist, the elements in the columns corresponding to female, 45-49, and artist should be set to 1. All other elements should be 0. Similar with user feature matrix, item feature matrix is built according to their genres. Movies in this dataset are attributed to 19 genres and hence the item feature matrix F_I has 19 features ($k_I = 19$) in it.

In Sushi dataset, we use some of the user's demographic information, i.e., gender and age. In this case, user's age has been divided into 6 groups by the data provider: 15-19, 20-29, 30-39, 40-49, 50-59, >=60. User gender consists of male and female, which is same as MovieLens data. Thus, the user feature matrix for this dataset has 5,000 rows and 8 columns. The item feature matrix, on the other hand, has 100 rows and 16 columns. The 16 features include 2 styles (maki and other), 2 major groups (seafood and other), and 12 minor groups (aomono (blue-skinned fish), akami (red meat fish), shiromi (white-meat fish), tare (something like baste; for eel or sea eel), clam or shell, squid or octopus, shrimp or crab , roe, other seafood, egg, meat other than fish, vegetables).

Different from MoiveLens and Sushi datasets, LibimSeTi dataset only provides user's gender as its auxiliary information so we directly use it as user's cluster indicator matrix C_U. It is worth noting that in this dataset, there are three possible gender values: male, female, and unknown. To be consistent, the number of user clusters is set to 3.

C. Evaluation Strategy

For comparison purposes, we run the proposed approach and the SVD-based data update approach [30] on the datasets to measure the error of unknown value imputation and the privacy level of the perturbed data, as well as their time cost. The SVD-based data update approach first uses the column mean to impute missing values in the new data and then performs the incremental SVD update on the imputed data. The machine we use is equipped with Intel® Core™ i5-2405S processor, 8GB RAM and is installed with UNIX operating system. The code was written and run in MATLAB.

We start with the partial training matrix R (also referred to as the original data, which is built by removing ratings left on some items or left by some users from the complete training matrix[3]), and then add the rest of data (also referred to as the new data) to R in several rounds.

When building R, we use the split ratio to decide how many ratings will be removed from the complete training data. For example, there are 1000 users and 500 items with their companion ratings in the training data. If the split ratio is 40% and we will do a row update, we use the first 400 rows as the original data, i.e., R ($\in \mathbb{R}^{400\times 500}$). The remaining 600 rows of the training matrix will be added to R in several rounds. Similarly, if we are going to perform a column update, we use the first 200 columns as the original data ($R \in \mathbb{R}^{1000\times 200}$) while the remaining 300 columns will be added to R in several rounds.

In each round, we add 100 rows/columns to the original data. If the number of the rows/columns of new data is not divisible by 100, the last round will update the rest. Therefore, in this example, the remaining 600 rows will be added to R in 6 rounds with 100 rows each. Note that Sushi data only has 100 items in total but we still want to test the column update on it so we add 10 items instead of 100 in each round.

The basic procedure of the experiments is as follows:

1) Perform Aux-NMF and SVD on R, producing the approximated matrix R_r (see Fig. 1);
2) Append the new data to R_r by iAux-NMF and SVD-based data update algorithm (SVDU for short) [30], yielding the updated rating matrix $\tilde{R}_r$;
3) Measure imputation error[4] and privacy of the updated rating matrix $\tilde{R}_r$;
4) Compare and study the results.

The imputation error is obtained by calculating the difference between the actual ratings in the test data and the imputed ratings in the released data. A common and popular criterion is the MAE (Mean Absolute Error), which can be calculated as follows:

$$MAE = \frac{1}{|TestSet|} \sum_{r_{ij}\in TestSet} |r_{ij} - p_{ij}| \tag{38}$$

where r_{ij} is the actual value while p_{ij} is the predicted value.

When measuring the privacy, we define the privacy level in Definition 2

Definition 2: **Privacy level** $\Pi(Y|X)$ is a metric that indicates to what extent a random variable Y could be estimated if given random variable X.

$$\Pi(Y|X) = 2^{h(Y|X)} \tag{39}$$

where $h(Y|X)$ is the differential entropy of Y given X.

This privacy measure was proposed by Agrawal et al.[1] and was applied to measure the privacy in collaborative filtering by Polat et al.[21], and Wang et al.[30]. In our experiment, we take $\Pi(Y|X)$ (the higher the better) as privacy measure to quantify the privacy, where random variable Y corresponds to the values in training set and X corresponds to the perturbed values (at same position as those in training set) in released data.

D. Results and Discussion

In this section, we present and discuss our experimental results in two stages. We first run Aux-NMF and SVD on the complete training data to evaluate the performance of the non-incremental algorithm. Then we follow the steps as specified in the previous section to evaluate the incremental algorithms.

1) Test on complete Training Data: Some parameters of the proposed algorithms need to be determined in advance. Table II gives the parameter setup in Aux-NMF (see Algorithm 1).

TABLE II: Parameter Setup in Aux-NMF

Dataset	α	β	γ	k	l	$MaxIter$
MovieLens	0.2	0	0.8	7	7	10
Sushi	0.4	0.6	0	7	5	10
LibimSeTi	1	0	0	3	10	10

For MovieLens dataset, we set $\alpha = 0.2$, $\beta = 0$, and $\gamma = 0.8$, which means that we rely mostly on the item cluster matrix, and then the rating matrix, whereas eliminate the user cluster matrix. This combination was selected after probing many possible cases. We will discuss how we choose the parameters in Section V-D3. We believe there still exist better combinations. Both k and l are set to 7. We set these values because K-Means was prone to generate empty clusters with

[3] Here, "complete" means all the ratings from the dataset are in the matrix. It is still a sparse matrix.

[4] We use the term "imputation error" because all missing values are imputed and will be compared with the real values, though no specific imputation technique is used in Aux-NMF and iAux-NMF.

greater k and l, especially on the data with very few users or items. Note that if β or γ is a non-zero value, the user or item cluster matrix will be used and k or l is equal to the number of user clusters or item clusters. As long as β or γ is zero, the algorithm will eliminate the corresponding cluster matrix and k or l will have nothing to do with the number of user clusters or item clusters.

For Sushi dataset, we set $\alpha = 0.4$, $\beta = 0.6$, and $\gamma = 0$. The parameters indicate that the user cluster matrix plays the most critical role during the update process. In contrast, rating matrix is the second important factor as it indicates the user preference on items. The item cluster matrix seems trivial so it does not participate the computation. We set k to 7 and l to 5 based on the same reason as mentioned in previous paragraph.

For LibimSeTi dataset, we give the full weight to the rating matrix. Zero weight is received for user and item cluster matrices since they do not contribute anything to the good results. As mentioned in data description, user's auxiliary information only includes the gender with three possible values. So we set k to 3. In this case, l only denotes the column rank of V and is set to 10.

In SVD, since it cannot run on an incomplete matrix, we use item mean to impute the missing values (see [30]). The rank is set to 13 for MovieLens, 7 for Sushi, and 10 for LibimSeTi. Table III presents the results on three datasets.

TABLE III: Results on MovieLens dataset

Dataset	Method	MAE	$\Pi(Y\|X)$	Time Cost
MovieLens	Aux-NMF	0.7481	1.2948	0.9902s
	SVD	0.7769	1.2899	34.1341s
Sushi	Aux-NMF	0.9016	1.4588	0.5350s
	SVD	0.9492	1.4420	5.4175s
LibimSeTi	Aux-NMF	1.2311	1.0715	5.7962s
	SVD	1.2154	1.0537	390.2246s

In this table, the time cost of SVD includes the imputation time while the time cost of Aux-NMF includes the clustering time. For instance, on MovieLens dataset, the imputation took 32.2918 seconds and SVD itself took 1.8423 seconds, as 34.1341 seconds in total; the clustering time took 0.0212 seconds and Aux-NMF itself took 0.9690 seconds, as 0.9902 seconds in total. As can be seen, Aux-NMF outperformed SVD in all aspects on all three datasets. We notice that the former ran much faster than the latter (saves 97% time on MovieLens, 90% time on Sushi, and 98% time on LibimSeTi). This is mainly because SVD-based algorithm needs imputation, which is time consuming, but for Aux-NMF, it can directly work on sparse matrix though it needs to cluster beforehand (it is very fast in general).

It is interesting to take a look at the results of running K-Means on the final matrix generated by Aux-NMF and the matrix generated by SVD. As shown in Fig. 2(a), MovieLens users with ratings produced by Aux-NMF were clustered into 7 groups with clear boundaries. However, the result is different for SVD - most users were grouped together and thus the clusters cannot be distinguished from each others. Note that the axes in both figures are ratings left by users on items. The results indicate the more normally distributed ratings in the matrix generated by Aux-NMF than SVD. Remember that our goal is to provide good imputation accuracy as well as high privacy level. In addition, the data should look as if it is the real data. To this end, we should make the ratings distribute normally, i.e., people may leave more 3 stars on a $1 \sim 5$ scale than 1 star and 5 stars. In this regard, Aux-NMF generated more reasonable data than SVD did.

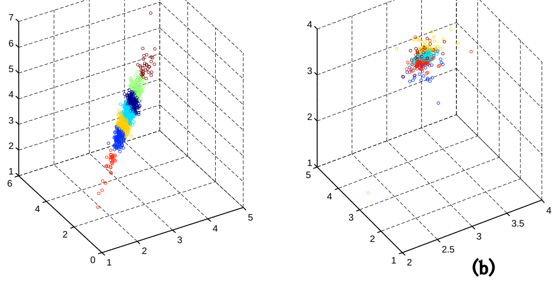
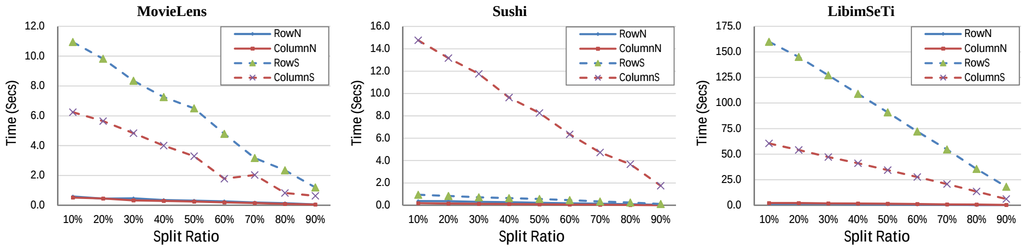


Fig. 2: Clustering results on ratings predicted by Aux-NMF (a) and SVD (b) on MovieLens dataset

2) The Incremental Case: In previous section, we examined the Aux-NMF on three datasets in terms of MAE, privacy level, as well as time cost. Now we measure the same metrics on iAux-NMF (incremental Aux-NMF).

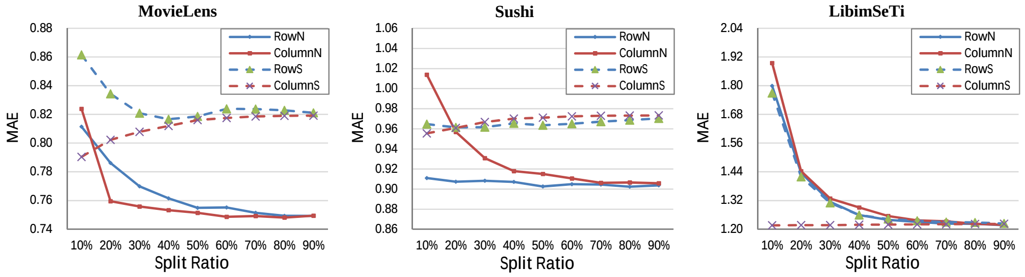
Fig. 3 shows the time cost for updating new rows and columns by iAux-NMF and SVDU (SVD-based data update algorithm). We use "RowN" and "ColumnN" to represent row and column updates in iAux-NMF. Similarly, "RowS" and "ColumnS" are for row and column updates in SVDU. We use the same parameter setup in Table II.

It can be seen that iAux-NMF outperformed SVDU in both row and column updates. As pointed out in Section IV-B, the time complexity of row update in iAux-NMF is $O(pn(l+k)$ and column update has a time complexity of $O(qm(l+k)$. As a reference, the time complexities of row and column updates in SVDU are $O(k^3 + (m+n)k^2 + (m+n)kp + p^3)$ and $O(k^3 + (m+n)k^2 + (m+n)kq + q^3)$, respectively. When the rating matrix has high dimensions, the time cost difference can be huge. For example, the LibimSeTi dataset has both more users and more items than MovieLens so the improvement of iAux-NMF over SVDU plotted in Fig. 3(c) was greater than Fig. 3(a). However, the Sushi data is a bit special as the time difference between two methods in row update was very small, though iAux-NMF still ran faster. In Section V-D1, we broke the time cost of both methods into two pieces: for SVDU, the time consists of imputation time and SVD computation time; for Aux-NMF, the time consists of clustering time and Aux-NMF computation time (Before running the algorithms, the parameters need to be determined. We will discuss the time cost for this part in Section V-D3.). By tracking the time cost of each stage, we found that the imputation in SVDU took considerably shorter time in row update than column update on this dataset but the time cost of Aux-NMF in row update and column update did not differ a lot. Essentially, the faster imputation in row update can be attributed to the small number of items. Since SVDU uses the column mean to impute the



(a) (b) (c)

Fig. 3: Time cost variation with split ratio



(a) (b) (c)

Fig. 4: MAE variation with split ratio

missing values, if there are only a few items, the mean value calculation can be fast.

However, with the substantial improvement in time cost, iAux-NMF should not produce a significantly higher imputation error than SVDU.

Fig. 4 shows the mean absolute errors of the prediction. When the split ratio was greater than 20%, iAux-NMF achieved lower errors than SVUD on MovieLens and Sushi datasets. The average improvement on MovieLens was 9.79% for row update and 9.76% for column update. The Sushi dataset had a little less average improvement than MoiveLens but it was still noticeable. Nevertheless, both of them had large errors by iAux-NMF than by SVD when the split ratio was less than 20%. This is because the centroids picked up by K-Means algorithm did not distribute over the data that was not large enough to reflect the global picture. With badly selected centroids, K-Means cannot produce a good clustering result which further affects the Aux-NMF and iAux-NMF so the errors would be large. Unlike MovieLens and Sushi, the LibimSeTi dataset got different results. In this case, iAux-NMF still performed better than SVDU but the gap tended to be smaller as the split ratio increased. The results imply that auxiliary information is important to iAux-NMF as it is used as constraint in the update process. On the contrary, SVDU does not need it. This can explain why SVDU performed better than iAux-NMF on LibimSeTi (no auxiliary information is used).

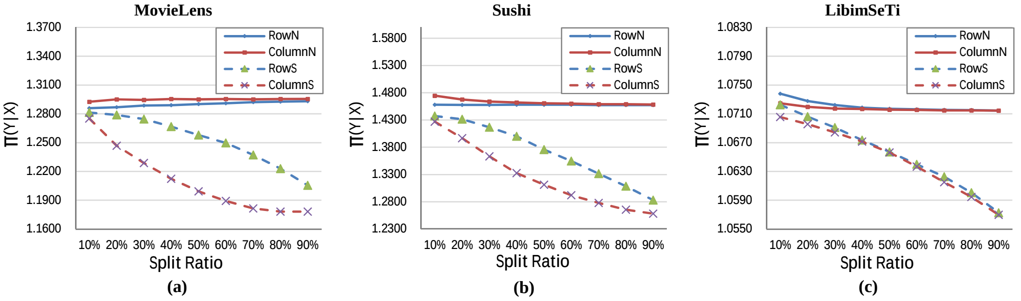
In Section IV-B2, we mentioned the issue of Aux-NMF re-computation. As presented in Fig. 4, the MAE's of both row and column updates on MovieLens dataset dropped more slowly at 70% and nearly kept the same after this point. Similarly but more interestingly, the MAE of row update on Sushi dataset began to increase at 70%. Therefore, a re-computation can be performed at 70% for these two datasets. For LibimSeTi dataset, the MAE's did not seem to stop decreasing so the re-computation is not immediately necessary.

In addition to MAE, we want to investigate the privacy metrics presented in Section V-C. The privacy level with varying split ratio is plotted in Fig. 5. The curve shows that the privacy level of the data produced by iAux-NMF were higher and more stable than SVDU while the latter had decreasing trend with greater split ratios. The results are encouraging.

As a summary, the iAux-NMF data update algorithm ran much faster than SVDU while maintaining nearly the same data utility and privacy as SVDU, if not better.

3) Parameter Study: In iAux-NMF, three parameters, i.e., α, β, and γ need to be set. In this section, we do some comparisons over several parameter combinations and discuss the results. Note that we keep the split ratio at 40% and pregenerate the initial random matrices in Algorithms 2 and 3 to eliminate the effect of randomness in the experiments. We adopt the parameter setup in Table II because it is the best combination obtained by probing many possible cases. The pseudocode in Algorithm 4 shows the procedure to find out the parameters that produce the lowest MAE's. The step is set to 0.1 when we increment the parameters. Since there is a constraint $\alpha + \beta + \gamma = 1$, the total number of parameter combinations is 66. It took 806.28 seconds to run a full test on MovieLens dataset, 1116.9 seconds on Sushi, and 11517.87 seconds on LibimSeTi. The times are relatively long when compared with the times of running the incremental algorithms. However, the parameters only need to be determined offline once so it will not affect the online performance.

Table IV lists some representative combinations with their



(a) (b) (c)

Fig. 5: Privacy level variation with split ratio

Algorithm 4 Pseudocode for Parameter Probing

```
for $\alpha$ = 0 : 0.1 : 1 do
    for $\beta$ = 0 : 0.1 : 1 - $\alpha$ do
        $\gamma = 1 - \alpha - \beta$.
        Run Aux-NMF and iAux-NMF on a dataset with
        parameter $\alpha$, $\beta$, and $\gamma$, saving the MAE's as well
        as $\alpha$, $\beta$, and $\gamma$ to the corresponding variables.
    end for
end for
Find out the lowest MAE and obtain the associated parameters.
```

results on MovieLens dataset. The best combinations are in bold font. We notice that if the updates simply relied on the rating matrix, the results were only a little worse than taking into account the auxiliary information. In contrast, if only the auxiliary information was utilized, the MAE was unacceptable, though the privacy level was the highest. It is clear that between user features and item features, the latter made good contribution to the results while the former seems trivial. Nevertheless, the weight of rating matrix can be lowered but should not be removed. The Sushi dataset (Table V) had a similar conclusion but it is the user features that played a more dominant role.

TABLE IV: Parameter Probe on MovieLens dataset

Parameters	Update	MAE	$\Pi(Y\|X)$
$\alpha = 1, \beta = 0, \gamma = 0$	Row	0.7643	1.2913
	Column	0.7538	1.2964
$\alpha = 0.5, \beta = 0.5, \gamma = 0$	Row	0.7643	1.2913
	Column	0.7539	1.2963
$\alpha = 0.5, \beta = 0, \gamma = 0.5$	Row	0.7624	1.2909
	Column	0.7534	1.2958
$\alpha = 0, \beta = 0.5, \gamma = 0.5$	Row	0.9235	1.3149
	Column	0.9164	1.3150
$\boldsymbol{\alpha = 0.2, \beta = 0, \gamma = 0.8}$	Row	0.7616	1.2890
$\boldsymbol{\alpha = 0.4, \beta = 0, \gamma = 0.6}$	Column	0.7533	1.2955

TABLE V: Parameter Probe on Sushi dataset

Parameters	Update	MAE	$\Pi(Y\|X)$
$\alpha = 1, \beta = 0, \gamma = 0$	Row	0.9083	1.4578
	Column	0.9221	1.4613
$\alpha = 0.5, \beta = 0.5, \gamma = 0$	Row	0.9073	1.4580
	Column	0.9201	1.4614
$\alpha = 0.5, \beta = 0, \gamma = 0.5$	Row	0.9085	1.4580
	Column	0.9221	1.4614
$\alpha = 0, \beta = 0.5, \gamma = 0.5$	Row	1.0468	1.4851
	Column	1.0371	1.4849
$\boldsymbol{\alpha = 0.4, \beta = 0.6, \gamma = 0}$	Row	0.9071	1.4580
$\boldsymbol{\alpha = 0.2, \beta = 0.8, \gamma = 0}$	Column	0.9180	1.4620

As shown in Table VI, the rating matrix of LibimSeTi dataset was the only information used in the computation. This indicates that even the dataset comes with users' genders, they did not help in our model. This is reasonable as the gender is not a necessary factor for people to determine their ratings (A female can rate another female with a fairly high rating.). Note that since there is no item features coming with this dataset, γ was always set to zero.

Therefore, we can conclude that, the rating matrix should always be utilized while the auxiliary information makes contributions to the improved results as well.

TABLE VI: Parameter Probe on LibimSeTi dataset

Parameters	Update	MAE	$\Pi(Y\|X)$
$\boldsymbol{\alpha = 1, \beta = 0, \gamma = 0}$	Row	1.2589	1.0719
	Column	1.2911	1.0717
$\alpha = 0.5, \beta = 0.5, \gamma = 0$	Row	1.3378	1.0713
	Column	1.3926	1.0709
$\alpha = 0, \beta = 1, \gamma = 0$	Row	5.4017	1.0782
	Column	5.4017	1.0782

VI. Conclusion and Future Work

In this paper, we proposed a NMF-based privacy preserving data update approach for collaborative filtering purpose. This approach utilizes the auxiliary information to build the cluster membership indicator matrices for users and items. These matrices are regarded as additional constraints in updating the weighted nonnegative matrix tri-factorization. The proposed approach, named iAux-NMF, can incorporate the incremental data into existing data quite efficiently while maintaining the high data utility and privacy. Furthermore, the inevitable missing value imputation issues in collaborative filtering is solved in a subtle manner by this approach without using any particular imputation methods. Experiments conducted on three different datasets demonstrate the superiority of iAux-NMF over the existing privacy-preserving SVD-based data update method in the situation of incremental data update.

In future work, we will consider the automated clustering update when new data comes in. This new feature will decide the number of clusters by itself and recompute the NMF when needed. We believe it can provide better data utility and privacy. We will also investigate the distributed data update in collaborative filtering and attempt to propose the corresponding distributed algorithms.

VII. Acknowledgments

We thank Oldrich Neuberger for providing the LibimSeTi dating data and Lukas Brozovsky for cleaning up and generating the dataset.

References

[1] D. Agrawal and C. C. Aggarwal. On the design and quantification of privacy preserving data mining algorithms. In *Proceedings of the Twentieth ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems*, PODS '01, pages 247–255. ACM, 2001.

[2] L. Brozovsky and V. Petricek. Recommender system for online dating service. In *Proceedings of Znalosti 2007 Conference*. VSB, 2007.

[3] J. Canny. Collaborative filtering with privacy. In *Proceedings of the 2002 IEEE Symposium on Security and Privacy*, pages 45–57. IEEE Computer Society, 2002.

[4] G. Chen, F. Wang, and C. Zhang. Collaborative filtering using orthogonal nonnegative matrix tri-factorization. In *Proceedings of the Seventh IEEE International Conference on Data Mining Workshops*, ICDMW '07, pages 303–308. IEEE, 2007.

[5] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)*, 39(1):1–38, 1977.

[6] C. Ding, X. He, and H. D. Simon. On the equivalence of nonnegative matrix factorization and spectral clustering. In *Proceedings of the 2005 SIAM International Conference on Data Mining*, SDM '05, pages 606–610. SIAM, 2005.

[7] C. Ding, T. Li, W. Peng, and H. Park. Orthogonal nonnegative matrix tri-factorizations for clustering. In *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, SIGKDD '06, pages 126–135. ACM, 2006.

[8] Eurostat. *Manual on Disclosure Control Methods*. Office for Official Publications of the European Communities, 1996.

[9] S. Ferdowsi, V. Abolghasemi, and S. Sanei. A constrained nmf algorithm for bold detection in fmri. In *Proceedings of the 2010 IEEE International Workshop on Machine Learning for Signal Processing*, pages 77–82, 2010.

[10] P. O. Hoyer and P. Dayan. Non-negative matrix factorization with sparseness constraints. *Journal of Machine Learning Research*, 5:14571469, 2004.

[11] S. M. A. Kabir, A. M. Youssef, and A. K. Elhakeem. On data distortion for privacy preserving data mining. In *Proceedings of Canadian Conference on Electrical and Computer Engineering*, CCECE 2007, pages 308 – 311. IEEE, 2007.

[12] T. Kamishima and S. Akaho. Efficient clustering for orders. In *Proceedings of the 2nd International Workshop on Mining Complex Data*, pages 274–278, 2006.

[13] J. Kim and H. Park. Sparse nonnegative matrix factorization for clustering. Technical report, , Georgia Institute of Technology, 2008.

[14] H. Kuhn and A. Tucker. Nonlinear programming. In *Proceedings of the Second Berkeley Symposium on Mathematical Statistics and Probability*, pages 481–492, 1951.

[15] D. D. Lee and H. S. Seung. Algorithms for non-negative matrix factorization. *Advances in Neural Information Processing Systems*, 13:556–562, 2001.

[16] H. Li, T. Adali, W. Wang, D. Emge, and A. Cichocki. Non-negative matrix factorization with orthogonality constraints and its application to raman spectroscopy. *Journal of VLSI Signal Processing Systems*, 48(1-2):83–97, 2007.

[17] H. Liu and Z. Wu. Non-negative matrix factorization with constraints. In *Proceedings of the Twenty-Fourth AAAI Conference on Artificial Intelligence*, AAAI '10, pages 506–511. AAAI, 2010.

[18] Y. Mao and L. K. Saul. Modeling distances in large-scale networks by matrix factorization. In *Proceedings of the 4th ACM SIGCOMM Conference on Internet Measurement*, IMC '04, pages 278–287. ACM, 2004.

[19] F. McSherry and I. Mironov. 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 '09, pages 627–636. ACM, 2009.

[20] V. P. Pauca, F. Shahnaz, M. W. Berry, and R. J. Plemmons. Text mining using nonnegative matrix factorizations. In *Proceedings of 2004 SIAM Interational Conference on Data Mining*, volume 54 of *SDM '09*, pages 452–456. SIAM, 2004.

[21] H. Polat and W. Du. Privacy-preserving collaborative filtering. *International Journal of Electronic Commerce*, 9(4):9–35, 2005.

[22] P. Resnick, N. Iacovou, M. Suchak, P. Bergstrom, and J. Riedl. Grouplens: An open architecture for collaborative filtering of netnews. In *Proceedings of ACM 1994 Conference on Computer Supported Cooperative Work*, pages 175–186. ACM, 1994.

[23] R. Sandler and M. Lindenbaum. Nonnegative matrix factorization with earth mover's distance metric for image analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33:1590–1602, 2011.

[24] B. M. Sarwar, G. Karypis, J. A. Konstan, and J. T. Riedl. Application of dimensionality reduction in recommender systems – a case study. In *Proceedings of ACM WebKDD Workshop*. ACM, 2000.

[25] N. Thapa, L. Liu, P. Lin, J. Wang, and J. Zhang. Constrained nonnegative matrix factorization for data privacy. In *Proceedings of the 7th International Conference on Data Mining*, DMIN '11, pages 88–93, 2011.

[26] J. Tougas and R. J. Spiteri. Updating the partial singular value decomposition in latent semantic indexing. *Computational Statistics & Data Analysis*, 52:174–183, 2007.

[27] S. Vucetic and Z. Obradovic. Collaborative filtering using a regression-based approach. *Knowledge and Information Systems*, 7:1–22, 2005.

[28] J. Wang, J. Zhan, and J. Zhang. Towards real-time performance of data value hiding for frequent data updates. In *Proceedings of the IEEE International Conference on Granular Computing*, pages 606–611. IEEE Computer Society, 2008.

[29] J. Wang, W. Zhong, and C. Zhang. Nnmf-based factorization techniques for high-accuracy privacy protection on non-negative-valued datasets. In *Proceedings of the sixth IEEE International Conference on Data Mining Workshops*, ICDM Workshops 2006, pages 513–517. IEEE, 2006.

[30] X. Wang and J. Zhang. SVD-based privacy preserving data updating in collaborative filtering. In *Proceedings of the World Congress on Engineering 2012*, WCE 2012, pages 377–284. IAENG, 2012.

[31] W. Xu, X. Liu, and Y. Gong. Document clustering based on non-negative matrix factorization. In *Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Informaion Retrieval*, SIGIR '03, pages 267–273. ACM, 2003.

[32] K. Yu, S. Zhu, J. Lafferty, and Y. Gong. Fast nonparametric matrix factorization for large-scale collaborative filtering. In *Proceedings of the 32nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '09, pages 211–218. ACM, 2009.

[33] J. Zhang. Image fusion based on nonnegative matrix factorization. In *Proceedings of 2004 International Conference on Image Processing*, volume 2 of *ICIP '04*, pages 973–976. IEEE, 2004.

[34] S. Zhang, W. Wang, J. Ford, and F. Makedon. Learning from incomplete ratings using non-negative matrix factorization. In *Proceedings of the Sixth SIAM International Conference on Data Mining*, pages 548–552. SIAM, 2006.