Privacy-Preserving Non-Negative Matrix Factorization with Outliers

SWAPNIL SAHA and HAFIZ IMTIAZ, Department of Electrical and Electronic Engineering, Bangladesh University of Engineering and Technology, Bangladesh

Non-negative matrix factorization is a popular unsupervised machine learning algorithm for extracting meaningful features from inherently non-negative data. Such data often contain privacy-sensitive user information. Additionally, the dataset can contain outliers, which may lead to extracting sub-optimal features from the data. It is, therefore, necessary to address these two issues while analyzing privacy-sensitive data that may contain outliers. In this work, we develop a non-negative matrix factorization algorithm in the privacy-preserving framework that (i) considers the presence of outliers in the data, and (ii) can achieve results comparable to those of the non-private algorithm. We design our method in such a way that one has the control to select the degree of privacy grantee based on the required utility gap. We show the effectiveness of our proposed algorithm’s performance on six real and diverse datasets. The experimental results show that our proposed method can achieve a performance that closely approximates the performance of the non-private algorithm under some parameter choices, while ensuring strict privacy guarantees.

CCS Concepts: • Security and privacy → Privacy-preserving protocols;

Additional Key Words and Phrases: Differential privacy, non-negative matrix factorization, Gaussian mechanism, privacy accounting, Rényi differential privacy, topic modelling, facial features

ACM Reference format:
Swapnil Saha and Hafiz Imtiaz. 2024. Privacy-Preserving Non-Negative Matrix Factorization with Outliers. ACM Trans. Knowl. Discov. Data. 18, 3, Article 64 (January 2024), 26 pages.
https://doi.org/10.1145/3632961

1 INTRODUCTION

Non-negative matrix factorisation (NMF) is an unsupervised machine learning (ML) technique for discovering the part-based representation of inherently non-negative data [29]. Consider a data matrix \( V \in \mathbb{R}^{D \times N} \) with entries \( v_{ij} \geq 0 \) \( \forall i \in \{1, 2, \ldots, D\} \) and \( \forall j \in \{1, 2, \ldots, N\} \), where \( N \) is the number of the data samples, and \( D \) is the data dimension. The NMF objective is to (approximately) decompose the data matrix \( V \) into two non-negative matrices as follows:

\[
V \approx WH,
\]

S. Saha research performed as a part of undergraduate thesis at the Department of Electrical and Electronic Engineering, Bangladesh University of Engineering and Technology.

Authors’ address: S. Saha and H. Imtiaz (Corresponding author), Department of Electrical and Electronic Engineering, Bangladesh University of Engineering and Technology, Dhaka-1205, Bangladesh; e-mails: swapnilsahabd@gmail.com, hafizimtiaz@eee.buet.ac.bd.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

© 2024 Copyright held by the owner/author(s). Publication rights licensed to ACM.
1556-4681/2024/01-ART64 $15.00
https://doi.org/10.1145/3632961

ACM Transactions on Knowledge Discovery from Data, Vol. 18, No. 3, Article 64. Publication date: January 2024.
where \( W \in \mathbb{R}^{D \times K} \) is the basis matrix, \( H \in \mathbb{R}^{K \times N} \) is the coefficient matrix, and \( K \) is the latent dimension. The decomposition is performed, usually through minimizing some divergence between the data matrix and the factor matrices, such that each entry of \( W \) satisfies \( w_{ik} \geq 0 \), and each entry of \( H \) satisfies \( h_{kj} \geq 0, \forall k \in \{1, 2, \ldots, K\} \). In short, NMF performs dimension reduction by mapping the ambient data dimension \( D \) into latent dimension \( K \) for \( N \) data samples. Unlike other dimension reduction methods, such as Principal Component Analysis (PCA) or Independent Component Analysis (ICA), NMF is well-suited for inherently non-negative data because it finds non-subtractive and interpretable basis and coefficients.

As mentioned before, NMF approximates the \( j \)th column of \( V \) as \( v_j \approx W h_j \), where \( h_j \) is the \( j \)th column of \( H \). Essentially, the \( j \)th column of \( V \) is represented as a linear combination of the columns of \( W \), with the coefficients being the corresponding entries from \( h_j \). Therefore, the dictionary matrix \( W \) can be interpreted as storing the “parts” of the data matrix \( V \). This part-based decomposition, and considering only the non-negative values, make NMF popular in many practical applications including, but not limited to:

- dimension reduction
- topic modeling in text mining
- representation learning
- extracting local facial features from human faces
- unsupervised image segmentation
- speech denoising
- community detection in social networks.

However, the acquired data samples may be contaminated by localized outliers [66], such as salt and pepper noise, occlusions, shadows, moving objects in videos, and so on. The presence of such outliers can affect the accurate estimation of basis and thus, the latent low-rank structure of the data [70, 71].

**NMF and Privacy.** Many services of modern era operate on sensitive user data to provide better suggestions and experiences. These are often done via ML algorithms, such as the ones mentioned above, and the ML algorithms are trained on users’ sensitive data. As shown in theoretical and applied works, the users are rightfully concerned regarding their privacy being compromised by the outputs of the ML algorithms. For example, the seminal work of Homer et al. [28] showed that the presence of an individual in a genome dataset can be identified from simple summary statistics about the dataset. In the ML setting, typically the model parameters (as a matrix \( W \) or vector \( w \)) are learned by training on the users’ data. To that end, *Membership Inference Attacks* [22] are discussed in detail by Hu et al. [30], and Shokri et al. [60]. They showed that given the learned parameters \( W \), an adversary can identify users in the training set. Basically, the trained weights of the model can be used to extract sensitive information. The tendency of the weights to memorize the training examples can be used to regenerate the training example. Several other works [37, 47, 62] also showed how personal data leakage occurred from modern ML and signal processing tasks. Additionally, it has been shown that simple anonymization of data does not provide any privacy in the presence of auxiliary information available from other/public sources [62]. One prime example is the Netflix Challenge, where an anonymized dataset was released in 2007, which contained anonymized movie ratings from Netflix subscribers. Nevertheless, researchers found a way to recover 99% of the removed personal data [47] by using the publicly available IMDb datasets. In addition, privacy leakage can happen through gradient sharing [72] as well. In distributed ML systems, several nodes exchange gradient values. The authors in [72] showed how one can extract private training information from such gradient sharing. These discoveries led to widespread concern over using private data for ML algorithms. Evidently, personal information leakage is the
main hindrance to collecting and analyzing sensor data for training ML and signal processing algorithms. It is, therefore, necessary to employ a framework where one can share private data without disclosing their participation or identity. **Differential privacy (DP)** is a rigorous mathematical framework that can provide protection against such information leakage \[18\]. The definition of differential privacy is motivated by the cryptography work, which has gained significant attention in ML and data mining communities \[17\]. DP introduces randomness into the computation pipeline for preserving privacy. This, however, leads to degradation of the computation accuracy, and the user needs to quantitatively choose the optimum privacy budget considering the required privacy-utility tradeoff.

**Our Contributions.** In this work, we consider applications of non-negative matrix factorization involving sensitive data, such as dimension reduction/feature extraction, topic modeling/text mining, image segmentation, denoising, and community detection. We intend to perform NMF decomposition on inherently non-negative and privacy-sensitive data (that may contain sparse outliers) to discover novel structure/features of the dataset. That is, (i) the estimated basis matrix \(W\) should closely approximate the “true” basis matrix in the presence of the outliers, and (ii) since the data matrix \(V\) contains user-specific privacy-sensitive information, the estimated \(W\) should ensure formal privacy guarantee such that an adversary cannot extract sensitive information regarding the users. To address this, we propose a differentially-private \[18\] non-negative matrix factorization algorithm to compute the basis matrix \(W\) considering the presence of outliers in the dataset. We adopt the outlier model from Reference \[71\]. To the best of our knowledge, our proposed method is the first differentially-private NMF algorithm considering outliers. Employing the mathematically rigorous privacy guarantee differential privacy allows us to compute \(W\) such that it reflects very little about any particular user’s data, and is relatively unaffected by the presence of outliers. Our algorithm design makes sure that the learned \(W\) closely approximates the “true” dictionary matrix, while ensuring strict privacy guarantees. Our major contributions are summarized below:

- We develop a novel privacy-preserving non-negative matrix factorization algorithm capable of operating on sensitive data, while closely approximating the results of the non-private algorithm.
- We consider the effect of outliers by specifically modeling them as in Reference \[71\], such that the presence of outliers has very little effect on our estimated differentially-private basis matrix \(W\).
- Considering the multi-round nature of the proposed algorithm, we analyze our algorithm using Rényi Differential Privacy (RDP) \[46\] to provide a tighter characterization of privacy under composition. We obtain a much better accounting of the overall privacy loss, compared with the conventional strong composition theorem \[18\].
- We perform extensive experimentation on six diverse real datasets to show the effectiveness of our proposed algorithm. We compare the results with those of an existing algorithm \[57\] and the non-private algorithm. We observe that the basis matrix \(W\), which our proposed algorithm estimates, provides a close approximation to the non-private results for certain parameter choices, easily outperforming the existing algorithm.
- We present the results in a way to make the privacy-utility tradeoff more comprehensive, and that the user can choose between the overall privacy budget and the required “closeness” to non-private results.

**Related Works.** The non-negative matrix factorization is attained in the literature by minimizing the following objective function:

\[
\min_{W \in C, h_{k,j} \geq 0, \forall k,j} \| V - WH \|_F^2 ,
\]
where $C \subseteq \mathbb{R}^{D \times K}$ is the constraint set for $W$. Several algorithms have been proposed to obtain the optimal point of this objective function, such as the multiplicative updates [38], alternating direction method of multipliers (ADMM) [67], block principal pivoting [35], active set method [33], and projected gradient descent (PGD) [43]. Most of these algorithms are based on alternatively updating $W$ and $H$. This essentially divides the optimization problem into two sub-problems, each of which can be optimized using the standard optimization techniques, such as the projected gradient or the interior point method. A detailed survey of these optimization techniques can be found in References [34, 64].

Additionally, some algorithms have been proposed to solve the NMF problem considering the outliers [59, 66, 70, 71]. Zhao and Tan [71] proposed an online NMF algorithm with two different solvers: a PGD-based solver and an ADMM-based solver. However, they proposed to use a fixed step size for the iterative updates and their stopping condition does not necessarily indicate that the solution is close to a stationary point. Moreover, due to the online nature of the algorithm, it can be very slow for datasets with high-dimensions and large numbers of samples. Zhang et al. [70] proposed another algorithm for robust NMF, which uses multiplicative updates for updating the basis and the coefficient matrix and a soft-thresholding function for updating the outliers matrix. However, it has been shown in multiple literature (see e.g., Gonzalez and Zhang [26] and Lin [43]) that the multiplicative update method lacks convergence guarantee and optimization properties. The robust NMF proposed by Shen et al. [59] also uses a multiplicative update rule for the basis matrix. Our work is based on the robust NMF algorithm using PGD [71].

Extensive works and surveys exist in the literature on differential privacy. In particular, Dwork and Smith’s survey [19] contains the initial theoretical work. We refer the reader to References [1, 5, 12, 13, 32, 40, 41, 61, 63] for the most relevant works in differentially private machine learning, deep learning, optimization, gradient descent, and empirical risk minimization. Adding randomness in the gradient calculation is one of the most common approaches for implementing differential privacy [5, 61]. Other common approaches are employing the output [13] and objective perturbations [51], the exponential mechanism [45], the Laplace mechanism [17], and Smooth sensitivity [50].

Several researchers have attempted to employ differential privacy in the context of general matrix factorization, focusing on recommendation systems [44] and not explicitly handling the non-negativity constraint. For example, Nikolaenko et al. [49] proposed a privacy preserving matrix factorization for recommendation systems using partially homomorphic encryption with Yao’s garbled circuits. Zhang et al. [69], Hua et al. [31], and Ran et al. [56] used the objective perturbation [13] (or its modification) for matrix factorization. Berlioz et al. [7] proposed differentially private matrix factorization into three different ways: input perturbation, perturbation of the gradients and output perturbation. In distributed-data setting, Ermis et al. [20] applied local differential privacy in collective matrix factorization problem. The relevant works with our work are the work of Alsulaimawi [3], which introduced a privacy filter with federated learning for NMF factorization; the work of Fu et al. [23], which proposed privacy-preserving NMF for dimension reduction using the Paillier Cryptosystem, and the work on privacy preserving data mining [2] using combined NMF and Singular Value Decomposition (SVD) methods. Last but not the least, the authors in Reference [57] proposed differentially private NMF for the recommender system, and we showed the comparison analyses in Section 4.5. However, to the best of our knowledge, no work has introduced differential privacy in the universal NMF decomposition framework considering outliers, and accounted for the overall privacy spent for multi-stage implementation to achieve the optimal privacy budget.
2 PROBLEM FORMULATION

2.1 Notations

We use bold lower-case letters (v), bold upper-case letters (V) and unbolded letters (M) to denote vectors, matrices and scalars, respectively. To indicate the iteration instant, we use subscript t. For example, W_t denotes the dictionary matrix after t iterations. The superscript ‘+’ indicates a single update. We denote the indices with lowercase unbolded letter. For example, the n-th column of matrix V is denoted as v_n, v_i,j indicates the entry of the i-th row and j-th column of the matrix V. Inequality x ≥ 0 or X ≥ 0 apply entry-wise. For element-wise matrix multiplication, we use the notation ⊙. We denote the L_2 norm (Euclidean norm) with ||.||_2, the L_1,1 norm with ||.||_1,1, and the Frobenius norm with ||.||_F. R, and R_+ denote the set of real numbers and set of positive real numbers, respectively. P_R denotes the Euclidean projector that projects onto the non-negative orthant, and P_R_+ denotes normalization projection that bounds the norm of column units. We denote the inner product of two matrices A and B as ⟨A, B⟩ = tr(A^T B). Finally, the probability density function of zero mean unit variance Gaussian random variable is given as follows: f(x) = \frac{1}{\sqrt{2\pi}} \exp(-\frac{x^2}{2}).

2.2 Definitions and Preliminaries

In this section, we review some definitions and propositions that are necessary for our problem formulation.

Definition 2.1 ((ε, δ)-DP [17]). Let D be the domain of the datasets consisting of N records, and D, D’ be neighbouring datasets that differ by only one record. Then, an algorithm f : D → T provides (ε, δ)-differential privacy ((ε, δ)-DP) if Pr(f(D) ∈ S) ≤ δ + ε Pr(f(D’) ∈ S) for all measurable S ⊆ T and for all neighbouring datasets D, D’ ∈ D.

Here, ε, δ ≥ 0 are the privacy parameters and determine how the algorithm will perform in providing utility and preserving privacy. The parameter ε indicates how much the algorithm’s output deviates in probability when we replace one single person’s data with another. The parameter δ indicates the probability that the privacy mechanism fails to give the guarantee of ε. Intuitively, higher privacy results in poor utility. That is, smaller ε and δ guarantee more privacy, but lower utility. There are several mechanisms to implement differential privacy: Gaussian [17], Laplace mechanism [18], random sampling, and exponential mechanism [45] are well-known. Among the additive noise mechanisms, the noise standard deviation is scaled by the privacy budget and the sensitivity of the function.

Definition 2.2 (L_2 Sensitivity [17]). If D and D’ are neighbouring datasets differing by only one record, then the L_2 sensitivity of function f(D) is

\[ Δ := \max_{D, D'} ||f(D) - f(D')||_2. \]

The L_2 sensitivity of a function gives the upper bound of how much the function can change if one sample at the input is changed. Consequently, it dictates the amount of randomness/perturbation needed at the function’s output to guarantee differential privacy. In other words, it captures the maximum change in the output by changing any one user in the worst-case scenario.

Definition 2.3 (Gaussian Mechanism [18]). Let f : D → R^D be an arbitrary function with L_2 sensitivity Δ. The Gaussian mechanism with parameter τ adds noise from N(0, τ^2) to each of the D entries of the output and satisfies (ε, δ) differential privacy for ε ∈ (0, 1) and δ ∈ (0, 1), if

\[ τ ≥ \frac{Δ}{ε} \sqrt{2 \log \frac{1.25}{δ}}. \]

Here, (ε, δ)-differential privacy is guaranteed by adding noise drawn from N(0, τ^2) distribution. Note that, there are an infinite number of combinations of (ε, δ) for a given τ^2.
Definition 2.4 (RDP [46]). A randomized algorithm $f : \mathcal{D} \mapsto \mathcal{T}$ is $(\alpha, \epsilon_r)$-Rényi differentially private if, for any adjacent $\mathcal{D}, \mathcal{D}' \in \mathcal{D}$, the following holds: $D_\alpha(\mathcal{A}(\mathcal{D}) \mid \mid \mathcal{A}(\mathcal{D}')) \leq \epsilon_r$. Here, $D_\alpha(P(x)||Q(x)) = \frac{1}{\alpha-1} \log \mathbb{E}_x - Q(x)^\alpha$ and $P(x)$ and $Q(x)$ are probability density functions defined on $\mathcal{T}$.

We use RDP for calculating the total privacy budget spent in our multi-stage algorithm. RDP provides a much simpler rule for calculating overall privacy risk $\epsilon$ that is shown to be tight [71].

Proposition 1 (From RDP to DP [46]). If $f$ is an $(\alpha, \epsilon_r)$-RDP mechanism, it also satisfies $(\epsilon_r + \frac{\log 1/\delta}{\alpha-1}, \delta)$-differential privacy for any $0 < \delta < 1$.

Proposition 2 (Composition of RDP [46]). Let $f_1 : \mathcal{D} \rightarrow \mathcal{R}_1$ be $(\alpha, \epsilon_1)$-RDP and $f_2 : \mathcal{R}_1 \times \mathcal{D} \rightarrow \mathcal{R}_2$ be $(\alpha, \epsilon_2)$-RDP, then the mechanism defined as $(X_1, X_2)$, where $X_1 \sim f_1(\mathcal{D})$ and $X_2 \sim f_2(X_1, \mathcal{D})$ satisfies $(\alpha, \epsilon_1 + \epsilon_2)$-RDP.

Proposition 3 (RDP and Gaussian Mechanism [46]). If $f$ has $L_2$ sensitivity 1, then the Gaussian mechanism $\mathcal{G}_\sigma f(\mathcal{D}) = f(\mathcal{D}) + \epsilon$ where $\epsilon \sim \mathcal{N}(0, \sigma^2)$ satisfies $(\alpha, \frac{\sigma}{\epsilon^{1/2}})$-RDP. Additionally, a composition of $T$ such Gaussian mechanisms satisfies $(\alpha, \frac{T \sigma}{\epsilon^{1/2}})$-RDP.

2.3 NMF Problem Formulation

As mentioned before, we adopt the outlier model presented in Reference [71]. We first reformulate the robust NMF model [71] for an offline setting. More specifically, our problem formulation and optimization technique involve updating the coefficient matrix $H$, the outlier matrix $R$, and the basis/dictionary matrix $W$ based on the entire data matrix $V$, rather than the single user data entry. Since we want to guarantee differential privacy for the estimated basis matrix $W$, which involves addition of Gaussian noise scaled to the sensitivity of some function, it is customary to have a small sensitivity to attain better “utility” or closeness to the true basis matrix. As shown in Section 3.2, operating on the entire dataset provides a significant advantage in this regard.

Most of the existing algorithms discussed in Section 1 follow the two-block coordinate descent framework shown in Algorithm 1. First, the dictionary matrix $W$ is updated, while keeping the coefficient $H$ fixed. Then, the updated dictionary matrix $W$ is used to update the coefficient $H$. The process continues until some convergence criteria are met. As mentioned before, we follow the robust NMF approach based on PGD that considers the presence of outliers [71]. We intend to decompose the data matrix $V$ as $V \approx WH + R$, and release the differentially-privacy basis matrix $W$, where $W$ and $H$ are defined as before. The matrix $R = [r_1, r_2, \ldots, r_N] \in \mathbb{R}^{D \times N}$ is a matrix containing the outliers of the data. Thus, the NMF optimization problem is reformulated as

$$
\min_{W \in C, H \geq 0, R \in Q} \frac{1}{N} \left( \frac{1}{2} \| V - WH - R \|_F^2 + \lambda \| R \|_{1,1} \right),
$$

where $C \subseteq \mathbb{R}^{D \times K}$ is the constraint set for updating $W$, $Q$ is the feasible set for $R$ and $\lambda > 0$ is the regularization parameter. Intuitively, the outlier matrix $R$ is sparse in nature and contains a smaller values compared to the noise-free original data entries. The sparsity of the outlier $R$ is enforced by the choice of $L_{1,1}$-norm regularization [71]. The level of sparsity is controlled by the hyper-parameter $\lambda$. Evidently, a grid search can be performed to select optimal hyper-parameters for our proposed differentially-privacy NMF. In Section 4, we discuss hyper-parameter tuning and matrix initialization methods for better optimization. We note that several robust NMF algorithms exist in the literature that consider the effect of outliers in the dataset, and utilize different norm on the objective function (for example, $L_2$, $L_{2,1}$, $L_{1,2}$ norms instead of the conventional $L_2$ norm) to reduce the effect of the outliers on $W$ and $H$ [24, 36, 59]. In this article, we adopted the robust NMF model as in Reference [71]. Evidently, exploring other robust NMF settings in the context.

ACM Transactions on Knowledge Discovery from Data, Vol. 18, No. 3, Article 64. Publication date: January 2024.
ALGORITHM 1: General NMF Algorithm: Two-Block Coordinate Descent Method [54]

Require: Data matrix $V$, number of iteration $n$
Ensure: Dictionary matrix $W$

1: Initialize: $W_0, H_0$
2: for $t = 0$ to $n - 1$ do
3: $W_{t+1} \leftarrow$ update($V, W_t, H_t$)
4: $H_{t+1} \leftarrow$ update($V, W_{t+1}, H_t$)
5: end for
6: return Optimized dictionary matrix $W$

of differential privacy, and comparing their performances would be an interesting direction of research.

Note that the robust NMF algorithm cannot guarantee exact recovery of the original data matrix $V$, as the loss function \((1)\) is not convex in nature [71]. However, it can be shown empirically that the estimated basis matrix can be meaningful, and the difference between the data matrix $V$ and the reconstructed matrix is negligible and sparse [21, 59, 71]. Now, following the two-block coordinate descent method mentioned in Algorithm 1, we reformulate our optimization steps as follows:

— Update the coefficient matrix $H_t$ and outlier matrix $R_t$ based on the previous dictionary matrix $W_{t-1}$. Here, the optimization can be done as
\[
(H_t, R_t) = \arg\min_{H \geq 0, R \in Q} L(V, W_{t-1}, H, R),
\]
where the loss function $L$ is
\[
L(V, W, H, R) \triangleq \frac{1}{N} \left( \frac{1}{2} \|V - WH - R\|_F^2 + \lambda \|R\|_{1,1} \right).
\]

Here, the constraint set $Q \triangleq \{r \in \mathbb{R}^D : \|r\|_\infty \leq M\}$ keeps the entries of the outlier matrix $R$ uniformly bounded. The value of $M$ depends on the dataset and noise distribution. For example, in the gray scale image data with $2^b - 1$ levels in each pixel, we can set $M = 2^b - 1$ where $b$ is the number of bits to represent the pixel value.

— After optimization with respect to $H_t$ and $R_t$, compute $W_t$ by minimizing the same loss function (3):
\[
W_t = \arg\min_{W \in C} L(V, W_t, H_t, R_t).
\]

Here, the set $C$ constrains the columns of dictionary matrix $W$ into a unit (non-negative) $L_2$ ball to keep the matrix entries bounded [42, 43].

PGD Solver for (2). The optimization problem in (2) can be solved by following the two steps alternatively for a given $W$ [71]:
\[
H^+ := \arg\min_{H \geq 0} Q(H'|H),
\]
\[
R^+ = \arg\min_{R' \in Q} \frac{1}{N} \left( \frac{1}{2} \|V - WH - R'\|_F^2 + \lambda \|R'\|_{1,1} \right),
\]
where
\[
Q(H'|H) \triangleq q(H) + \langle \nabla q(H), H' - H \rangle + \frac{1}{2\eta N} \|H' - H\|_2^2,
\]
\[
q(H) \triangleq \frac{1}{2N} \|V - WH - R\|_F^2.
\]
Here, $\eta$ is the fixed step size. Minimizing both (5) and (6) have closed-form solutions [71]. For (5), the solution can be expressed as the following:

$$H^+ := \mathcal{P}_s(H - \eta H \nabla q(H)).$$

For (6), the solution is straightforward [71]:

$$R^+ = S_{\lambda,M}(V - WH^+).$$

For (6), the solution is straightforward [71]:

$$R^+ = S_{\lambda,M}(V - WH^+).$$

Here, $S_{\lambda,M}(X)$ performs element-wise thresholding as

$$S_{\lambda,M}(X)_{ij} := \begin{cases} 0, & \|x_{ij}\| < \lambda \\ x_{ij} - \text{sgn}(x_{ij})\lambda, & \lambda \leq \|x_{ij}\| \leq \lambda + M \\ \text{sgn}(x_{ij})M, & \|x_{ij}\| > \lambda + M \end{cases}$$

In the $t$th iteration, we update the matrices $H$ and $R$ according to (8) and (10); until some stopping criteria is met [71].

**PGD Solver for (4).** We can rewrite (4) as follows

$$W_t = \arg \min_{W \in C} \frac{1}{2} \text{tr}(W^TWA_t) - \text{tr}(W^TB_t),$$

where $A_t \triangleq \frac{1}{N}H_tH_t^T$, and $B_t \triangleq \frac{1}{N}(V - R_t)H_t^T$. To calculate the gradient value, we define new function $f_W(W)$.

$$f_W(W) = \frac{1}{2} \text{tr}(W^TWA_t) - \text{tr}(W^TB_t).$$

Taking partial derivative of (11) with respect to $W$, we find the following expression.

$$\nabla f_W(W) = WA_t - B_t.$$  

We use the some matrix properties and lemmas [53] to derive the expressions in (9) and (12). We write the update equation ensuring the constraints of $W$.

$$W^+ = \mathcal{P}_C(W - \eta_W \nabla f_W(W)).$$

Here, $\eta_W$ is the step size to update $W$. As in (8), we use a fixed step size. The constraint projection function keeps the columns of $W$ in the unit $L_2$ ball. In (13), each column is being updated as following:

$$w_k^+ := \mathcal{P}_+(w_k - \eta_W \nabla f_W(w_k)) = \max \left(1, \frac{\|\mathcal{P}_+(w_k - \eta_W \nabla f_W(w_k))\|_2}{\|w_k\|_2}\right), \forall k \in [K].$$

Similarly, as the PGD Solver for (2), we use (13) to update the dictionary matrix $W$. The steps stated above are followed until we reach the optimum loss point of (1). Evidently, estimating $W$ depends on the potentially sensitive data. As we intend to release the estimated $W$, while guaranteeing formal privacy, we need to modify the aforementioned steps. In the following, we discuss our proposed differentially-private NMF algorithm in detail to show how to preserve and control privacy leakage for NMF.
In this section, we describe our proposed algorithm to estimate \( W_{\text{private}} \) in detail. We first specify our privacy model containing a data curator and a data analyst. We then analytically show how to incorporate differential privacy for estimating the basis matrix \( W_{\text{private}} \), such that it provides close approximation to the true basis matrix \( W \). Next, we analyze our multi-round gradient descent algorithm with RDP \([46]\) to attain a tight characterization of the privacy loss encountered over the course of the gradient descent. And finally, we discuss regarding the convergence of the algorithm.

### 3.1 Separate Private and Non-private Training Nodes

As mentioned earlier, we are interested in estimating the dictionary matrix \( W \), which captures the meaningful population features, and then releasing it publicly. More specifically, our goal is to release a differentially-private estimate of the true dictionary matrix \( W \), considering the presence of the outliers. Intuitively, we need to process the user-sensitive data and population feature data separately. Figure 1 shows the basic system diagram, which serves the purposes. Here, we have two data processing centers: one is the "Data Curator", which holds the sensitive data: data matrix \( V \) and the coefficient matrix \( H \) and as well as the outlier matrix \( R \). It updates the matrices \( H \) and \( R \) as mentioned in the Section 2.3. It also calculates the gradient \( \nabla f_W(W) \) of the loss function \( f_W(W) \) and then follows the DP protocol to add noise to the gradient before sending it to the other data processing center. Note that, the variance of the noise depends on the privacy budget and \( L_2 \) sensitivity of the gradient \( \nabla f_W(W) \).

The other data processing center is the "Data Analyst", where the noisy gradient is sent to. This center then updates the dictionary matrix \( W \) with the received differentially-private estimate of the gradient and sends the updated \( W \) to the Data Curator. This cycle continues until some stopping criteria is met. At the end, we get a differentially-private dictionary matrix \( W_{\text{private}} \) at the Data Analyst.

Note that, the Data Curator can choose from several mechanisms to ensure DP: Laplace mechanism, Gaussian mechanism, Exponential mechanism. Although the Laplace mechanism offers pure \( \epsilon \)-DP, it adds too much noise compared to Gaussian mechanism because the \( L_1 \) sensitivity depends on the data dimension. Additionally, the Exponential mechanism is more suitable for discrete-valued algorithms, and often times, it could involve a probability density that is very difficult to sample from Reference [14]. Therefore, we employ the Gaussian mechanism for this work. Now, the Data Curator has several options to employ the Gaussian mechanism to ensure the privacy of the gradient \( \nabla f_W(W) \): objective perturbation, input perturbation, output perturbation to name a few [13]. Objective perturbation involves adding certain noise to the objective function.
However, it requires the objective function to satisfy some stringent conditions that are difficult to ensure in our NMF problem. The output perturbation involves adding certain noise to the output of the algorithm (the basis matrix $W$ in our case). However, in this approach, analyzing the effect of changing one data point on $W$ is non-trivial due to the intricate relation of $W$, $H$, and $R$ with the loss function. Therefore, we choose a variant of the input perturbation—the noisy gradient descent, where we estimate the gradient of the loss function satisfying DP [5, 61].

### 3.2 Estimating Differentially-Private $W_{\text{private}}$

In this section, we show the necessary proofs and derivations related to estimating $W_{\text{private}}$. According to Gaussian Mechanism [18], we need to calculate the $\ell_2$ sensitivity of the gradient $\nabla f_W(W)$ of the loss function $f_W(W)$ with respect to $W$ for estimating the dictionary matrix $W$ satisfying DP. We have, $\nabla f_W(W) = WA - B$. As $\nabla f_W(W)$ depends on the statistics matrices $A$ and $B$, we can calculate their $\ell_2$ sensitivities separately.

In the following, we first calculate the $\ell_2$ sensitivity of matrix $A = \frac{1}{N}HH^T$. Consider two neighboring data matrices $V$ and $V'$ and their corresponding coefficient matrices $H$ and $H'$. By definition, they differ by only one user data (e.g., the $N$th column). We calculate the $\ell_2$ sensitivity $\Delta_A$ as

$$
\Delta_A = \max \frac{1}{N} \left\| HH^T - H'H'^T \right\|_F
= \max \frac{1}{N} \left\| h_N h_N^T - h'_N h'_N^T \right\|_F
\leq \max \frac{1}{N} \left( \left\| h_N h_N^T \right\|_F + \left\| h'_N h'_N^T \right\|_F \right)
\leq \max \frac{1}{N} \left( \left\| h_N \right\|_2 \left\| h_N \right\|_2 + \left\| h'_N \right\|_2 \left\| h'_N \right\|_2 \right)
= \max \frac{1}{N} \left( \left\| h_N \right\|_2^2 + \left\| h'_N \right\|_2^2 \right)
= \frac{2}{N} \times (\text{max } \ell_2 \text{ norm of columns of } H)^2,
$$

where we have used the triangle inequality and $\|ab\| \leq \|a\| \|b\|$. To get a bounded value in (14), we need the maximum $\ell_2$ norm of the columns of $H$. One way to do that is by normalizing each column of $H$ during the updates. We can have $\Delta_A = \frac{2}{N}$. Next, we calculate the $\ell_2$ sensitivity $\Delta_B$ of $B = \frac{1}{N}(V - R)H^T$ as follows:

$$
\Delta_B = \max \frac{1}{N} \left\| (V - R)H^T - (V' - R)H'^T \right\|_F
= \max \frac{1}{N} \left\| (v_N - r_N)h_N^T - (v'_N - r_N)h'_N^T \right\|_F
\leq \max \frac{1}{N} \left( \left\| (v_N - r_N)h_N^T \right\|_F + \left\| (v'_N - r_N)h'_N^T \right\|_F \right)
\leq \max \frac{1}{N} \left( \left\| (v_N - r_N) \right\|_2 \left\| h_N \right\|_2 + \left\| (v'_N - r_N) \right\|_2 \left\| h'_N \right\|_2 \right)
= \max \frac{1}{N} \left( \left\| (v_N - r_N) \right\|_2 + \left\| (v'_N - r_N) \right\|_2 \right)
= \max \frac{2}{N} \left\| (v_N - r_N) \right\|_2
\leq \max \frac{2}{N} (\|v_N\| + \|r_N\|),
$$

(15)
where the second-last equality follows from $\max_{\mathbf{h}_n} \| \mathbf{h}_n \|_2 = 1$, and the last inequality follows from $\| \mathbf{a} - \mathbf{b} \| \leq \| \mathbf{a} \| + \| \mathbf{b} \|$. To get a constant $L_2$ sensitivity value in (15), we can normalize the columns of $\mathbf{V}$ and $\mathbf{R}$ to have unit $L_2$-norm. Thus, we have $\Delta_B = \frac{4}{N}$. Note that, if we do not model the outliers explicitly, we would have $\Delta_B = \frac{2}{N}$.

Now, as we have computed the $L_2$ sensitivities $\Delta_A$ and $\Delta_B$, we can generate noise perturbed statistics $\tilde{\mathbf{A}}, \tilde{\mathbf{B}}$ following the Gaussian mechanism [18]. Using these values, we can compute the differentially-private estimate of the true gradient as $\nabla f_{\mathbf{W}}(\mathbf{W}) = \mathbf{W}\tilde{\mathbf{A}} - \tilde{\mathbf{B}}$ and update our dictionary matrix $\mathbf{W}$. At the end of optimization, we will obtain the differentially private dictionary matrix $\mathbf{W}_{\text{private}}$. The detailed step-by-step description of our proposed method is summarized in Algorithm 2.

Note that, one can avoid analytically computing the sensitivities of $\mathbf{A}_t$ and $\mathbf{B}_t$, and compute $\nabla f_{\mathbf{W}}(\mathbf{W})$ by employing some form of norm clipping [1, 4]. Norm clipping can be performed after noise addition to achieve a bounded gradient, as well. However, limiting the gradient norm has two opposing effects [1]: clipping destroys the unbiasedness of the gradient estimate, and the average clipped gradient may point in a sub-optimal direction. On the other hand, increasing the norm clipping bound results in addition of more noise to the gradients. To remedy this, the authors in Reference [4] proposed an adaptive norm clipping approach in the context of user-level DP guarantee in the federated learning setting for neural network training. We refer the reader to Reference [4] for further details.

**Theorem 3.1 (Privacy of Algorithm 2).** Consider Algorithm 2 in the setting of Section 2.3. Then Algorithm 2 releases $(\frac{T_{\alpha_{\text{opt}}}}{2}(\frac{\Delta_A^2}{\tau_A^2} + \frac{\Delta_B^2}{\tau_B^2}) + \log \frac{2}{\alpha_{\text{opt}}-1}, \delta)$ differentially-private basis matrix $\mathbf{W}_{\text{private}}$ for any $0 < \delta < 1$ after $T$ iterations, where $\alpha_{\text{opt}} = 1 + \sqrt{\frac{2}{T(\frac{\Delta_A^2}{\tau_A^2} + \frac{\Delta_B^2}{\tau_B^2})}}$, $\tau_A = \frac{\Delta_A}{\epsilon} \sqrt{2 \log \frac{25}{\delta}}$ and $\tau_B = \frac{\Delta_B}{\epsilon} \sqrt{2 \log \frac{25}{\delta}}$.

**Proof.** For the proof of Theorem 3.1, we analyze Algorithm 2 using the Gaussian Mechanism [18] and the RDP [46]. Recall that, at each iteration $t$, we compute the DP estimate of the gradient $\nabla f_{\mathbf{W}}(\mathbf{W})$, using two differentially-private matrices $\mathbf{A}_t$ and $\mathbf{B}_t$. According to Proposition 3, computation of these matrices satisfy $(\alpha, \frac{\alpha}{2(\frac{\Delta_A}{\tau_A})^2})$-RDP and $(\alpha, \frac{\alpha}{2(\frac{\Delta_B}{\tau_B})^2})$-RDP, respectively.

According to Proposition 2, each step of Algorithm 2 is $(\alpha, \frac{\alpha}{2(\frac{\Delta_A}{\tau_A})^2})$-RDP. If the number of required iterations for reaching convergence is $T$, then under $T$-fold composition of RDP, the overall algorithm is $(\alpha, \frac{T\alpha}{2}(\frac{\Delta_A^2}{\tau_A^2} + \frac{\Delta_B^2}{\tau_B^2}))$-RDP. From Proposition 1, we have that the algorithm satisfies $(\frac{T\alpha}{2}(\frac{\Delta_A^2}{\tau_A^2} + \frac{\Delta_B^2}{\tau_B^2}) + \log \frac{1}{\alpha_{\text{opt}}-1}, \delta)$-DP for any $0 < \delta < 1$. For a given $\delta$, we can compute the optimal (that provides the smallest overall $\epsilon$) $\alpha$ as $\alpha_{\text{opt}} = 1 + \sqrt{\frac{2}{T(\frac{\Delta_A^2}{\tau_A^2} + \frac{\Delta_B^2}{\tau_B^2})}}$, $\log \frac{1}{\alpha_{\text{opt}}-1}$. This $\alpha_{\text{opt}}$ provides the lowest privacy risk. Therefore, Algorithm 2 releases a $(\frac{T\alpha_{\text{opt}}}{2}(\frac{\Delta_A^2}{\tau_A^2} + \frac{\Delta_B^2}{\tau_B^2}) + \log \frac{1}{\alpha_{\text{opt}}-1}, \delta)$ differentially-private basis matrix $\mathbf{W}_{\text{private}}$ for any $0 < \delta < 1$. \qed

**Convergence of Algorithm 2.** We note that the objective function is non-increasing under the two update steps (i.e., steps 3 and 6 of Algorithm 2), and the objective function is bounded below. Additionally, the noisy gradient estimate $\nabla f_{\mathbf{W}}(\mathbf{W})$ essentially contains zero mean noise. Although this does not provide guarantees on the excess error, the estimate of the gradient converges in
ALGORITHM 2: Privacy Preserving NMF with Outliers

Require: Data matrix $V$, step size $\eta_H$ and $\eta_W$, maximum number of iterations $T$, sensitivities $\Delta_A$ and $\Delta_B$, privacy parameters $\epsilon$, $\delta$, regularization parameter $\lambda$

Ensure: Differentially private dictionary matrix $W_{\text{private}}$

1: Initialize: $W_0$, $H_0$, $R_0$.
2: for $t = 1$ to $T$ do
3: /* At Data Curator */ Learn $H_t$ and $R_t$ based on $W_{t-1}$
   $(H_t, R_t) := \operatorname*{argmin}_{H \geq 0, R \in Q} L(V, W_{t-1}, H, R),$
   where matrices update as follows
   $H^+ := P_+(H - \eta_H \nabla q(H)).$
   $R^+ := S_{\lambda, M}(V - WH^+).$
4: Calculate the DP statistics matrices
   $A_t := \frac{1}{N} H_t H_t^\top + N(0, \tau_A^2)^{K \times K},$
   $B_t := \frac{1}{N} (V - R_t) H_t^\top + N(0, \tau_B^2)^{D \times K},$
   where
   $\tau_A = \frac{\Delta_A}{\epsilon} \sqrt{\frac{2 \log \frac{1.25}{\delta}}},$
   $\tau_B = \frac{\Delta_B}{\epsilon} \sqrt{\frac{2 \log \frac{1.25}{\delta}}}$.
5: Calculate DP gradient: $\nabla f_W(W) = W_{t-1} A_t - B_t$
6: /* At Data Analyst */ Estimate the basis matrix with the noisy gradient $\nabla f_W$: $W_t := \operatorname*{argmin}_{W \in C} \frac{1}{2} \text{tr}(W^T W A_t) - \text{tr}(W^T B_t).$
   where dictionary matrix update is:
   $W^+ := P_C(W - \eta_W \nabla f_W(W)).$
7: end for
8: return $W_{\text{private}} = W_T$

We defer a theoretical analysis of the intricate relation between the excess error and the privacy parameters for future work. We refer to the reader Bassily et al. [5] for further details.

4 EXPERIMENTAL RESULTS

In this section, we show the effectiveness of our proposed algorithm by comparing the utilities of existing, and non-private algorithms. We define the Objective Value in (16) as the performance index, as it quantifies how well the algorithm can decompose the data matrix. The Objective Value is calculated using the following formula:

$$\text{Objective Value} = \frac{1}{2N} \|V^0 - WH\|_F^2.$$  

(16)

Here, $V^0$ is the noise-free clean dataset. The reason for choosing this performance index is that some NMF algorithms do not model the outliers explicitly like we do. We intend to evaluate how well the estimated basis and coefficient matrices explain the clean data $V^0$. Another reason is that, we cannot use Mean Absolute Error (MAE) or Root-mean Squared Error (RMSE) with the
Table 1. Summary of Datasets

| Dataset               | N    | D    | K   |
|-----------------------|------|------|-----|
| Guardian News Articles| 4551 | 10285| 8   |
| UCI-news-aggregator   | 3000 | 93   | 7   |
| RCV1                  | 9625 | 2999 | 4   |
| TDT2                  | 9394 | 3677 | 30  |
| YaleB                 | 2414 | 1024 | 38  |
| CBCL                  | 2429 | 361  | 50  |

Table 2. Comparison Average Coherence Scores

| ε     | Guardian News Articles | UCI-news aggregator |
|-------|------------------------|---------------------|
| 0.5   | 0.4511                 | 0.9996975           |
| 0.6   | 0.4529                 | 0.9996895           |
| 0.7   | 0.4535                 | 0.9996849           |
| 0.8   | 0.4547                 | 0.9996836           |
| 0.9   | 0.4543                 | 0.9996832           |
| 0.999 | 0.4540                 | 0.9996839           |
| Non-Private        | 0.4658                 | 0.9996816           |

estimated basis matrix. This is because, often times we do not know the true basis matrix (although we can assume the basis matrix estimated from a non-private NMF algorithm is the true basis matrix). But more importantly, the order of the columns of the basis matrix is undefined. Therefore, we choose to use the Objective Value as defined above as the performance index.

To evaluate our proposed method, we use six real datasets. Among which, four are text datasets, and two are face image datasets. The size of the datasets and the corresponding latent dimensions \( K \) are mentioned in the Table 1. For selection of the latent dimension \( K \), we mentioned one procedure in Appendix A, where we calculate the topic coherence score of the Guardian News Articles datasets to select the optimum topic number. For the rest of the datasets, similar procedures can be followed. For all of the experiments,\(^1\) we used a fixed \( \delta = 10^{-5} \).

4.1 Hyper-Parameter Selection and Initialization

We followed the hyper-parameter settings as Reference [71], except for the learning rate. The authors in Reference [71] suggested using the same learning rate for updating both the dictionary matrix \( W \) and the coefficient matrix \( H \) for ease of parameter tuning. The optimization process requires a different configuration in our privacy-preserving implementation. As discussed in Section 3, we add Gaussian noise in the gradient calculation \( \nabla f_W(W) \) for updating \( W \), whereas matrix \( H \) is updated with its unperturbed gradient calculation \( \nabla q(H) \). This procedure for updating \( W \) and \( H \) warrants for different learning rates. Note that to get faster convergence, we need to choose proper learning rates for updating \( W \) and \( H \). To that end, we can employ a grid search to find the optimum learning rates for \( H \) and \( W \). The time required for such a search depends on the dataset and the search space. With sub-optimal learning rates, the convergence may be delayed, but the excess error of the proposed algorithm should be approximately the same given that sufficient time is allowed, and small enough learning rate is used. In our experiments, we performed a grid search and found that the learning rate for updating \( W \) should be much lower (about 1/10000 to 1/20000 times depending on the dataset) compared to that of updating \( H \). To initialize \( W_0 \) and \( H_0 \), we followed the Non negative Double Singular Value Decomposition (NNDSVD) [10] approach, which performs much better compared with random initializations in minimizing the objective value in (16). For sparseness, we initialized \( R_0 \) with all zeros. Lastly, we normalized each data sample of data matrix \( V \) so that it has unit maximum entry.

4.2 Text Datasets

With the text datasets, we consider the topic modeling problem. Topic modeling identifies the abstract topics present in the set of documents. A document generally contains multiple topics

\(^1\)Codes are available at https://github.com/swapnil-saha/NMF_DP.git
with different proportions. Suppose a document is “80% about religion and 20% percent about politics”—it implies that about 80 percent of words are related to religion, and the rest 20 percent of words are related to politics. Topic modeling tries to find the unique “cluster of words” that indicates one single topic from the document. We discuss more about it and its implementation in Appendix A.

We show the performance of our proposed method with the learning curve plotted against variable privacy budget $\epsilon$ in each iteration. We demonstrate the topic word distribution for different $\epsilon$. We also calculate the overall privacy budget, $\epsilon_{\text{overall}}$, using RDP and show the comparison of objective value with that of the non-private mechanism to select the best $\epsilon$ in each iteration. For two datasets, we compared the topic word distribution and average coherence score between the non-private NMF algorithm and our proposed algorithm. Short descriptions of each of the text datasets are below:

**Guardian News Articles.** This dataset consists of 4551 news articles collected from Guardian News API in 2006. The detailed mechanism of collecting the articles is described in this article [52]. Here, we extract eight distinguished topics ($K = 8$) from the dataset and show the high-scoring word distributions corresponding to the topics.

**UCI News Aggregator Dataset.** This dataset [16] is formed by collecting news from a web aggregator from 10-March-2014 to 10-August-2014. There is a total of 422,937 news articles in the dataset. The topics covered in the news articles are entertainment, science and technology, business, and health. We take 750 news articles from each category and apply the NMF algorithm.

**RCV1.** Reuters Corpus Volume I (RCV1) [39] archive consists of over 800,000 manually categorized news wires. For our experiment, we randomly select approximately $\frac{1}{10}$th of the features that contain 9,625 documents.

**TDT2.** The TDT2 [15] text database contains 9,394 documents of size $9394 \times 36771$-dimensional matrix. Here, we randomly select $\frac{1}{10}$th of the features.

**Utility Comparison on Text Datasets.** In Figure 2, we show the utility gap between private and non-private mechanism’s output for the text datasets. Except for the Guardian News Articles dataset, there exists a very small utility gap, and this gap decreases further for a higher privacy budget $\epsilon$. Comparing the convergence speed, our proposed algorithm needs more epochs to reach the optimal point. This is because we have to keep the learning rate lower in private learning. Additionally, the noisy gradient descent somewhat affects the true gradient structure, which results in a comparatively slower convergence.

**Average Topic Coherence Score.** In Table 2 we show the average topic coherence score comparison for Guardian News Articles and UCI-news-aggregator datasets. Topic coherence score measures quantitatively how the topic modeling algorithm performs. In short, topic coherence attempts to represent human intractability through a mathematical framework by measuring the semantic link between high-scoring words. The greater the coherence score, the more human-interpretable the “cluster of words” is. It is also used to tune the hyper-parameter $K$. We discuss about topic coherence in more detail in Appendix A.

As Table 2 shows, topic coherence score increases with increasing the privacy budget $\epsilon$ for Guardian News Articles dataset. Recall that bigger $\epsilon$ means less noise and bigger privacy risk. For UCI-news-aggregator dataset, all the scores are very close to each other in respect to privacy budget. This is also evident from Figure 2(b), as the objective values of non-private and our proposed algorithm are very close.

$\epsilon_{\text{overall}}$ on Text Datasets. In Figure 3, we show the $\epsilon_{\text{overall}}$ and the utility gap between private and non-private mechanisms after reaching the optimum solution. These plots can be read as follows: if the user has a specific privacy budget $\epsilon_{\text{overall}}$, they can use this plot to see what’s the best utility level (objective value) attainable for a particular dataset, and then choose the corresponding $\epsilon$ from...
the x-axis to use in our Algorithm 2. On the other hand, if the user has to reach a particular utility level (objective value), they can use this plot to see what’s the overall privacy budget they need to spend, and the required $\epsilon$ for each stage of iteration of Algorithm 2 from the x-axis.

**Topic Word Comparison.** In Figure 4, we show the topic word comparison for the non-private algorithm and our proposed algorithm. We choose two datasets to show this comparison: the UCI-news-aggregator and the Guardian News Articles. The topic word distribution looks almost similar for our proposed algorithm and the non-private algorithm, which justifies the high similarity of coherence scores (Table 2).

### 4.3 Face Image Datasets

With the face image dataset, we generate the fundamental facial features that explain all the face images of the dataset. The details of the implementation are discussed in B. As the text dataset, we also show the $\epsilon_{\text{overall}}$ and utility comparison to select $\epsilon$ for each iteration. Additionally, as the effect of outliers is more visible and common in practice for image data, we conduct our experiments with additional noise outlier dataset. A short description of each text dataset is given below:

**YaleB.** There are 2,414 face images in Yaleb [68] of size 32 × 32. The sample images are captured in different light conditions. There are 38 subjects (male and female) in the dataset.

**CBCL.** The CBCL [65] database contains 2,429 face images of size 19×19. The facial photos consist of 19 × 19 hand-aligned frontal shots. Each face image is processed beforehand. The grey scale intensities of each image are first linearly adjusted so that the pixel mean and standard deviation are equal to 0.25, and then clipped to the range [0, 1].
Utility Comparison on Face Image Datasets. In Figure 5, we show the learning curves of the non-private algorithm and our proposed algorithm for the face image datasets. The characteristics of the result plots are quite similar to those of the text datasets. We note that the utility gaps are especially small and even smaller for higher privacy budget ($\epsilon$).
Fig. 5. Utility comparison on face image dataset.

Fig. 6. $\epsilon_{\text{overall}}$ and objective value on face image datasets.

$\epsilon_{\text{overall}}$ on Face Image Datasets. Figure 6 shows the $\epsilon_{\text{overall}}$ and utility gap after reaching the optimum loss point. Based on the criteria of preserving privacy as well as the tolerance of utility gap, one can select how much privacy budget $\epsilon$ one needs to introduce in each iteration.

Basic Representation Comparison. Figure 7 shows how an algorithm learns the fundamental representation of face image under DP and non-private conditions. In the case of $(\epsilon = 0.5, \delta = 10^{-5})$-DP, the facial features are quite noisy compared to the non-private method. However, they can still generate the interpretable human facial features. Additionally, increasing the $\epsilon$ budget would certainly produce representative images much closer to the ones from the non-private algorithm.

Datasets with Outliers. We also performed experiments to demonstrate the effect of outliers. We contaminated the Yaleb dataset with outliers following [71]: we randomly chose 70% of user data from the dataset, and then we contaminated 10% of the pixel with uniform noise distribution $\mathcal{U}[−1, 1]$. The simulation results are shown in Figure 8. In Section 3, it has been mathematically shown that the $L_2$ sensitivity of $B$ is double when we allow for updating the outlier matrix $R$. Higher $L_2$ sensitivity results in more noise to ensure $(\epsilon, \delta)$-DP. Thus, the basic representation of Figure 8(d) is noisier compared to the Figure 7(b).

4.4 Effect of Outliers on Estimating $W$

In this section, we investigate the effect of outliers present in the data on the estimated basis $W$. This also provides us with justification of our choice of modeling the outliers explicitly in $R$. We show two experiments: (i) comparing the topic coherence scores (see Appendix A.4) on the
Guardian News Articles dataset; and (ii) comparing the objective value (16) on a noisy synthetic dataset. In both cases, it showed that modelling R improves the performance of the optimization.

**Guardian News Articles Dataset with Outliers.** We followed a similar procedure as creating the noisy Yaleb dataset for adding the outliers. The only difference is that this time the noise was added to the normalized word embeddings. We then used our proposed DP NMF algorithm on the noisy text dataset and compared the topic coherence scores (17). Note that the topic coherence scores are measured only based on the basis matrix $W$. The better the quality of $W$ in representing the
data, the higher the topic coherence scores are. The experimental results are shown in Table 3. Our proposed DP NMF algorithm was employed with privacy parameters ($\epsilon = 0.5$, $\delta = 10^{-5}$) on three experimental setups. We observe that when we do not explicitly model the outliers, the coherence score drops (3.17%) for the dataset with outliers. We can achieve a better topic coherence score (1.51%) if we model the outlier matrix.

**Synthetic Dataset with Outliers.** We followed the steps mentioned in Reference [71] to create a synthetic dataset with outliers. More explicitly, we first created a clean synthetic dataset, and added outliers and observation noise to it. The steps to create a clean dataset are as follows: we generate the matrices $W^o$ and $H^o$ with each entry of $W^o$ and $H^o$ is drawn from the half normal distribution $\mathcal{H}\mathcal{N}(0, \frac{1}{\sqrt{W}})$. Then, we set $V^o = \mathcal{P}_U(W^oH^o)$. The normalization projection matrix keeps the norm of columns of $V^o$ bounded. To create an outlier matrix, $R \in \mathbb{R}^{D \times N}$, we choose 70% columns randomly from $R$, set 10% elements of the selected columns with values drawn from the uniform distribution $U[-1, 1]$ and set the remaining entries 0. With that, the noisy synthetic dataset becomes $V = \mathcal{P}_U(V^o + R^o + N)$, where $N \in \mathbb{R}^{D \times N}$ is defined as observation noise with entries drawn from the standard normal distribution $\mathcal{N}(0, 1)$. We applied our proposed DP NMF algorithm with privacy parameters ($\epsilon = 0.8$, $\delta = 10^{-5}$). We show the results of this experiment in Table 4. We observe that there is 5.67% improvement of the Objective Value (16) when we explicitly model the outlier matrix $R$ compared with no outlier modeling.

### 4.5 Comparison with Existing Differentially-Private NMF Algorithm [57]

The only existing differently private NMF (DPNMF) algorithm [57] is proposed for recommender systems using the Laplace mechanism [18], and does not consider the presence of outliers. On the other hand, our proposed algorithm works for any part-based NMF learning task. Another crucial distinction of our proposed method with the DPNMF [57] is that DPNMF employs the Laplace mechanism, which involves addition of noise from the Laplace distribution scaled to the $L_1$ sensitivity of the associated function. For any $D$-dimensional vector $x$, the following always holds: $||x||_2 \leq ||x||_1 \leq \sqrt{D} ||x||_2$ (using Cauchy–Schwarz inequality). Therefore, the $L_1$ sensitivity can be much larger than the $L_2$ sensitivity of the same function, for most practical scenarios and datasets with large ambient dimensions. Since the Gaussian mechanism [18] involves addition of noise from the Gaussian distribution scaled to the $L_2$ sensitivity of the associated function, the Gaussian mechanism typically performs much better than the Laplace mechanism in terms of utility [12, 13, 48, 61]. As mentioned before, we use the Gaussian mechanism in our proposed algorithm, and show that our proposed algorithm performs very well on several real datasets. Additionally, the authors in Reference [57] did not analytically calculate the $L_1$ sensitivity of the desired objective function, whereas we analytically showed how to calculate the $L_2$ sensitivity of the gradient for estimating the gradient satisfying differential privacy. This paves the way for further analysis and performance enhancement through adaptive norm-clipping [4]. Finally, the existing DPNMF [57] used the alternating non-negative least square (ANLS) algorithm [25], which has no mechanism to consider the effect of outliers. Nevertheless, we performed experiments to compare the performance of DPNMF [57] with our proposed algorithm on the MovieLens 1M dataset.
Table 5. RMSE Comparison on MovieLens 1M Dataset

| Privacy Budget | Proposed Method | DPNMF [57] |
|----------------|-----------------|------------|
| Non-private    | 0.0624          | 0.8568     |
| \( \epsilon = 0.3 \) | 0.0675          | 1.1953     |
| \( \epsilon = 0.5 \) | 0.0648          | 1.0785     |
| \( \epsilon = 0.7 \) | 0.0640          | 1.0155     |

Dataset [27]. For this experiment, we used RMSE as the performance measure as in Reference [57]:

\[
\text{RMSE} = \frac{1}{\sqrt{N}} \| \hat{V} - \hat{V} \odot X \|_2,
\]

where \( V \in \mathbb{R}_+^{U \times I} \) is the user-item matrix, \( \hat{V} \) and \( X \), same shape as \( V \), are the predicted user-item matrix and the observation mask, respectively, and \( N \) is the number of user-item pairs. Each entry \( v_{ui} \) in user-item matrix denotes how much a user \( u \in U \) gives rating to an item \( i \in I \). Each entry \( x_{ui} \) of observation mask matrix is set to 1 if user \( u \) has rated the item \( i \), and 0 otherwise. The simulation results are provided in Table 5. We observe from the table that our proposed method comfortably outperforms the existing DPNMF. We believe the primary reason is employing the Gaussian mechanism (and tight privacy characterization using the RDP analysis) as opposed to the Laplace mechanism. Essentially, we are proposing to use a relaxed definition of privacy (that is, \((\epsilon, \delta)-DP\)) instead of the pure \( \epsilon-\)DP in exchange for much better utility.

5 CONCLUSION AND FUTURE WORKS

We proposed a novel privacy-preserving NMF algorithm that can learn the differentially-private basis matrix \( W_{\text{private}} \) from the data matrix \( V \) while explicitly modeling the outliers in the data and offers utility close to the non-private setting. For ensuring differential privacy, we used the Gaussian mechanism: we introduced randomness into the computation pipeline in the form of carefully tuned noise. To this end, we presented a novel analysis of calculating the \( L_2 \) sensitivity of the associated functions for estimating the gradient satisfying DP. Recognizing the multi-shot nature of the proposed algorithm, we analyzed our algorithm with RDP to achieve a tight characterization of the privacy spent. The proposed algorithm offers a excellent results, closely approximating that of the non-private algorithm. The overall privacy budget \( \epsilon_{\text{overall}} \) along with the utility plots can provide means to select privacy budget per iteration, given the required utility level or privacy constraints. We empirically showed the effectiveness of our proposed method and compared the results on six real datasets. All the results show that our proposed algorithm can provide a utility close to that of the non-private algorithm for some parameter choices. For the text datasets, we quantitatively measured the performance using the Topic Coherence score. For the face image datasets, we compared facial feature construction. We observed that the features are similar in showing the fundamental facial parts of the human face as with the non-private method. An interesting future work will be further reducing the noise effect on the face image datasets. Another possible future work could be along the lines of federated learning and decentralized learning, where we need to perform NMF on private and decentralized data matrix.

APPENDICES

A TOPIC MODELING AND ITS IMPLEMENTATION

A.1 Topic Modeling

Topic modeling is a statistical model used in statistics and natural language processing to discover the abstract “topics” that occur in a collection of documents. Topic modeling is a common...
Privacy-Preserving Non-Negative Matrix Factorization with Outliers

64:21
text-mining technique to uncover hidden semantic structures within a text body. Given that a document is about a specific topic, one would expect certain words to appear more or less frequently: “dog” and “bone” will appear more frequently in documents about dogs, “cat” and “meow” will appear more frequently in documents about cats, and “the” and “is” will appear roughly equally in both. A document typically addresses multiple topics in varying proportions; therefore, a document that is 10% about cats and 90% about dogs would likely contain nine times as many dog words as cat words. The “topics” generated by topic modeling techniques are word clusters. A topic model encapsulates this intuition in a mathematical framework, enabling the examination of a set of documents and the identification of their potential topics and balance of topics based on the statistics of their words.

Topic models are also known as probabilistic topic models, which refer to statistical algorithms for identifying the latent semantic structures of a large text body. In this information age, the amount of written material we encounter daily exceeds our capacity to process it. Extensive collections of unstructured text bodies can be organized and comprehended better with topic models. Originally developed as a tool for text mining, topic models have been used to detect instructive structures in data, including genetic information, images, and networks. They have applications in fields such as computer vision \[11\] and bioinformatics \[8\].

A text document consists of one or more topics. In the mathematical context, we can say that the linear combination of topics forms each text document. Each topic reflects its semantic meaning by some representative “cluster of words”. In topic modeling, we find these representative clusters of words from the corpus and the coefficient weights, which say how much a single topic is more present than others in a single document. In the context of NMF decomposition, data matrix \(V\) contains the text documents, dictionary matrix \(W\) contains topic words and coefficient matrix \(H\) contains the coefficient weight.

A.2 Text Pre-Processing

The first step before applying any topic modeling algorithm is to do text preprocessing. Raw documents contain textual words which need to convert into numerical form. To do so, we split each word from the document and give a unique token to each of them.

Let us say we have five documents in our corpus of documents, and there is a total of 100 unique words present in all documents. Then, after tokenizing the corpus of documents, we will form a matrix \(A\) of size 100 \(\times\) 5. Column entry indicates the document number and row number indicates the specific term word. If \(a_{ij} = 3\) in \(A\) where \(i = 50, j = 4\), it means that 50 “no-term word” is used 3 times in the 4th document.

However, we need further preprocessing to do actual topic modeling. Intuitively all the words in a document do not contribute equal contributions to determine the topic category of this document. Besides, some high-frequency words (like articles and auxiliary verbs) and low-frequency words do not indicate a specific topic. We remove these unnecessary words and add weight to the important topic words. The first one is done easily by simple text preprocessing like maximum frequency filtering, minimum frequency filtering, and stop-word (which stores predefined high-frequency English words) filtering. To give extra weight to important topic words, we need to introduce a new mathematical framework: term frequency-inverse document frequency (TF-IDF) \[6, 58\].

TF-IDF wants to calculate quantitatively how a term word is “important” to determine the nature of the specific document’s topic category. The calculation involves two steps. First, it computes the frequency of word terms in that specific document. Then, it computes the frequency in all the documents. The second calculation wants to penalize if the term word is common in all documents.
The equation of TF-IDF is as follows \[ w_{d} = f_{w,d} \times \log\left(\frac{|D|}{f_{w,D}}\right). \] (17)

In our implementation, we use scikit-learn default function TfidfVectorizer() to produce TF-IDF normalized document-term matrix. According to our notation, TfidfVectorizer() produces matrix of size \( D \times N \) where \( D \) is the number of word term after processing the text data and \( N \) is the total number of documents present in the corpus. Now, the corpus of raw documents is ready to use the NMF algorithm.

A.3 Implementation through NMF

If we apply the NMF algorithm on the TF-IDF normalized document-term, we get two matrices: one is dictionary matrix \( W (D \times K) \), and the other is coefficient matrix \( H (K \times N) \) where \( K \) is the topic number presented in the corpus. The \( W \) matrix shows the topic distribution word. We can tell about the topic category by observing the highest entry values.

Let us revisit the experimental implementation of the Guardian News Articles dataset discussed in the Section 4. We get the following topic word distribution in Figure 9. Here, the eight distinguish topic word distribution indicates eight distinguished topics. Applying NMF in the topic modeling algorithm requires one important hyper-parameter selection: topic number \( K \). Though here, we assume the topic number \( K = 8 \) before applying NMF, there is a systemic way to tune this hyper-parameter. This is done by measuring the Topic Coherence score.

A.4 Topic Coherence

Topic Coherence measures the semantic similarity between high-scoring words. These measurements help to differentiate between semantically interpretable topics and those that are statistical artifacts of inference. There are numerous methods for measuring coherence, such as NPMI, UMass, TC-W2V, and so on \[52\]. Our study uses the TC-W2V method to measure the coherence score.

Figure 10 shows the comparison of mean coherence scores with respect to the number of topics. This figure suggests selecting \( K = 8 \) as the topic number to get the optimum human interpretability from the topic word distribution.

B EXTRACTING LOCAL FACIAL FEATURE BY NMF

B.1 Interpret the Decomposition of Face Image

The extraction of local facial features is one of the most beautiful and practical applications of NMF. The basic idea behind this decomposition is to extract fundamental local facial features
so that one can reconstruct any image of the dataset using appropriate weight. To extract these fundamental features, one needs first to construct the data matrix $V$ where each column of $V$ represents the pixel information of the individual image. If we now apply the NMF algorithm on matrix $V$, we generate the two matrices: matrix $W$ stores the facial feature, and $H$ stores the coefficient. Figure 11 shows the visual representation of the result.

If we want to reconstruct an image of dataset: let’s example, we want to reconstruct the 100th column image in matrix $V$. Then, we will take all the facial features from matrix $W$ and 100th column vector from matrix $H$ as coefficients. Then, we will multiply the features with the coefficients and add them linearly. This will reconstruct the 100th column image of matrix $V$ with little loss.

ACKNOWLEDGMENTS

The authors would like to express their sincere gratitude towards the authorities of the Department of Electrical and Electronic Engineering and Bangladesh University of Engineering and Technology (BUET) for providing constant support throughout this research.

REFERENCES

[1] Martin Abadi, Andy Chu, Ian Goodfellow, H. Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. 2016. Deep learning with differential privacy. In Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security. 308–318.

[2] Afsana Afrin, Mahit Kumar Paul, and A. H. M. Sarowar Sattar. 2019. Privacy preserving data mining using non-negative matrix factorization and singular value decomposition. In Proceedings of the 2019 4th International Conference on Electrical Information and Communication Technology (EICT). IEEE, 1–6.

[3] Zahir Alsulaimawi. 2020. A non-negative matrix factorization framework for privacy-preserving and federated learning. In Proceedings of the 2020 IEEE 22nd International Workshop on Multimedia Signal Processing (MMSP). IEEE, 1–6.

[4] Galen Andrew, Om Thakkar, Brendan McMahan, and Swaroop Ramaswamy. 2021. Differentially private learning with adaptive clipping. In Advances in Neural Information Processing Systems, M. Ranzato, A. Beygelzimer, Y. Dauphin, P. S. Liang, and J. Wortman Vaughan (Eds.). Vol. 34, 17455–17466. https://proceedings.neurips.cc/paper_files/paper/2021/file/91c0f01af640a24e7f97a5ab407889f-Paper.pdf

[5] Raef Bassily, Adam Smith, and Abhradeep Thakurta. 2014. Private empirical risk minimization: Efficient algorithms and tight error bounds. In Proceedings of the 2014 IEEE 55th Annual Symposium on Foundations of Computer Science. IEEE, 464–473.

[6] Adam Berger, Rich Caruana, David Cohn, Dayne Freitag, and Vibhu Mittal. 2000. Bridging the lexical chasm: Statistical approaches to answer-finding. In Proceedings of the 23rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. 192–199.
[7] Arnaud Berlioz, Arik Friedman, Mohamed Ali Kaafar, Roksana Boreli, and Shlomo Berkovsky. 2015. Applying differential privacy to matrix factorization. In Proceedings of the 9th ACM Conference on Recommender Systems. 107–114.

[8] David M. Blei. 2012. Probabilistic topic models. Communications of the ACM 55, 4 (2012), 77–84.

[9] Léon Bottou. 1998. On-line learning in neural networks. In On-line Learning in Neural Networks, David Saad (Ed.). Cambridge University Press, New York, NY, USA, Chapter On-line Learning and Stochastic Approximations, 9–42. Retrieved from http://dl.acm.org/citation.cfm?id=304710.304720

[10] Christos Boutsidis and Efstratios Gallopoulos. 2008. SVD based initialization: A head start for nonnegative matrix factorization. Pattern Recognition 41, 4 (2008), 1350–1362.

[11] Liangliang Cao and Li Fei-Fei. 2007. Spatially coherent latent topic model for concurrent segmentation and classification of objects and scenes. In Proceedings of the 2007 IEEE 11th International Conference on Computer Vision. IEEE, 1–8.

[12] Kamalika Chaudhuri and Claire Monteleoni. 2008. Privacy-preserving logistic regression. In Advances in Neural Information Processing Systems, D. Koller, D. Schuurmans, Y. Bengio, and L. Bottou (Eds.). Vol. 21, Curran Associates, Inc. https://proceedings.neurips.cc/paper_files/paper/2008/file/8065d07da4a77621450aa84fece5656d9-Paper.pdf

[13] Kamalika Chaudhuri, Claire Monteleoni, and Anand D. Sarwate. 2011. Differentially private empirical risk minimization. Journal of Machine Learning Research 12, 3 (2011), 1069–1109.

[14] Kamalika Chaudhuri, Anand D. Sarwate, and Kaushik Sinha. 2013. A near-optimal algorithm for differentially-private principal components. Journal of Machine Learning Research 14, 1 (2013), 2905–2943.

[15] David Graff, Chris Cieri, Stephanie Strassel, and Nii Martey. 1999. The TDT-2 text and speech corpus. In Proceedings of the DARPA Broadcast News Workshop. 57–60.

[16] Dheeru Dua and Casey Graff. 2017. UCI Machine Learning Repository. Retrieved from http://archive.ics.uci.edu/ml

[17] Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. 2006. Calibrating noise to sensitivity in private data analysis. In Proceedings of the Theory of Cryptography Conference. Springer, 265–284.

[18] Cynthia Dwork and Aaron Roth. 2014. The algorithmic foundations of differential privacy. Foundations and Trends in Theoretical Computer Science 9, 3–4 (2014), 211–407.

[19] Cynthia Dwork and Adam Smith. 2010. Differential privacy for statistics: What we know and what we want to learn. Journal of Privacy and Confidentiality 1, 2 (2010), 135–154.

[20] Beysa Ermiş and A. Taylan Cemgil. 2020. Data sharing via differentially private coupled matrix factorization. ACM Transactions on Knowledge Discovery from Data (TKDD) 14, 3 (2020), 1–27.

[21] Cédric Févotte and Nicolas Dobigeon. 2015. Nonlinear hyperspectral unmixing with robust nonnegative matrix factorization. IEEE Transactions on Image Processing 24, 12 (2015), 4810–4819.

[22] Matt Fredrikson, Somesh Jha, and Thomas Ristenpart. 2015. Model inversion attacks that exploit confidence information and basic countermeasures. In The 22nd ACM SIGSAC Conference on Computer and Communications Security. 1322–1333.

[23] Anmin Fu, Zhenzhu Chen, Yi Mu, Willy Susilo, Yinxia Sun, and Jie Wu. 2019. Cloud-based outsourcing for enabling privacy-preserving large-scale non-negative matrix factorization. IEEE Transactions on Services Computing 15, 1 (2019), 266–278.

[24] Hongchang Gao, Feiping Nie, Weidong Cai, and Heng Huang. 2015. Robust capped norm nonnegative matrix factorization: Capped norm nmf. In Proceedings of the 24th ACM International Conference on Information and Knowledge Management. 871–880.

[25] Nicolas Gillis. 2011. Nonnegative matrix factorization: Complexity, algorithms and applications. Unpublished Doctoral Dissertation, Université Catholique de Louvain. Louvain-La-Neuve: CORE (2011).

[26] Edward F. Gonzalez and Yin Zhang. 2005. Accelerating the lee-seung algorithm for non-negative matrix factorization. IEEE Transactions on Services Computing 15, 1 (2015), 266–278.

[27] Nils Homer, Szabolcs Szellinger, Margot Redman, David Duggan, Waibhav Tembe, Jill Muehling, John V. Pearson, Dietrich A. Stephan, Frank McSherry, and Adam Smith. 2006. Calibrating noise to sensitivity in private data analysis. In Proceedings of the Theory of Cryptography Conference. Springer, 265–284.

[28] Edward F. Gonzalez and Yin Zhang. 2005. Accelerating the lee-seung algorithm for non-negative matrix factorization. IEEE Transactions on Services Computing 15, 1 (2015), 266–278.

[29] Patrik O. Hoyer. 2004. Non-negative matrix factorization with sparseness constraints. Journal of Machine Learning Research 5, 9 (2004), 1457–1469.

[30] Hongsheng Hu, Zoran Salcic, Lichao Sun, Gillian Dobbie, Philip S. Yu, and Xuyun Zhang. 2022. Membership inference attacks on machine learning: A survey. ACM Computing Surveys (CSUR) 54, 11s (2022), 1–37.

[31] Jingyu Hua, Chang Xia, and Sheng Zhong. 2015. Differentially private matrix factorization. In Proceedings of the International Joint Conference on Artificial Intelligence.

[32] Zhanglong Ji, Zachary C. Lipton, and Charles Elkan. 2014. Differential privacy and machine learning: A survey and review. arXiv:1412.7584. Retrieved from https://arxiv.org/abs/1412.7584

ACM Transactions on Knowledge Discovery from Data, Vol. 18, No. 3, Article 64. Publication date: January 2024.
[33] Hyunsoo Kim and Haesun Park. 2008. Nonnegative matrix factorization based on alternating nonnegativity constrained least squares and active set method. *SIAM Journal on Matrix Analysis and Applications* 30, 2 (2008), 713–730.

[34] Jingu Kim, Yunlong He, and Haesun Park. 2014. Algorithms for nonnegative matrix and tensor factorizations: A unified view based on block coordinate descent framework. *Journal of Global Optimization* 58, 2 (2014), 285–319.

[35] Jingu Kim and Haesun Park. 2008. Toward faster nonnegative matrix factorization: A new algorithm and comparisons. In *Proceedings of the 2008 8th IEEE International Conference on Data Mining*. IEEE, 353–362.

[36] Deguang Kong, Chris Ding, and Heng Huang. 2011. Robust nonnegative matrix factorization using l21-norm. In *Proceedings of the 20th ACM International Conference on Information and Knowledge Management*. 673–682.

[37] Jerome Le Ny and George J. Pappas. 2013. Differentially private filtering. *IEEE Transactions on Automatic Control* 59, 2 (2013), 341–354.

[38] Daniel Lee and H. Sebastian Seung. 2001. Algorithms for non-negative matrix factorization. In *Advances in Neural Information Processing Systems*. T. Leen, T. Dietterich, and V. Tresp (Eds.), Vol. 13. MIT Press. Retrieved from https://proceedings.neurips.cc/paper/2000/file/9d1152547c0bde01830b7e8bd60024c-Paper.pdf

[39] David D. Lewis, Yiming Yang, Tony G. Rose, and Fan Li. 2004. RCV1: A new benchmark collection for text categorization research. *Journal of Machine Learning Research* 5, Apr. (Dec. 2004), 361–397. Retrieved from http://dl.acm.org/citation.cfm?id=1005332.1005345

[40] Katrina Ligett, Seth Neel, Aaron Roth, Bo Waggoner, and Steven Z. Wu. 2017. Accuracy first: Selecting a differential privacy level for accuracy constrained ERM. In *Advances in Neural Information Processing Systems*. I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (Eds.), Vol. 30, Curran Associates, Inc. https://proceedings.neurips.cc/paper_files/paper/2017/file/86df7dc6966ca2674f757a2463eb9-Paper.pdf

[41] Arvind Narayanan and Vitaly Shmatikov. 2006. How to break anonymity of the Netflix prize dataset. arXiv:cs/0610105. Retrieved from https://arxiv.org/abs/cs/0610105

[42] Chih-Jen Lin. 2007. On the convergence of multiplicative update algorithms for nonnegative matrix factorization. *IEEE Transactions on Neural Networks* 18, 6 (2007), 1589–1596.

[43] Chih-Jen Lin. 2007. Projected gradient methods for nonnegative matrix factorization. *Neural Computation* 19, 10 (2007), 2756–2779.

[44] Ziqi Liu, Yu-Xiang Wang, and Alexander Smola. 2015. Fast differentially private matrix factorization. In *Proceedings of the 9th ACM Conference on Recommender Systems*. 171–178.

[45] Frank McSherry and Kunal Talwar. 2007. Mechanism design via differential privacy. In *Proceedings of the 48th Annual IEEE Symposium on Foundations of Computer Science (FOCS’07)*. IEEE, 94–103.

[46] Derek O’callaghan, Derek Greene, Joe Carthy, and Pádraig Cunningham. 2015. An analysis of the coherence of descriptors in topic modeling. *Expert Systems with Applications* 42, 13 (2015), 5645–5657.

[47] Gerard Salton and Christopher Buckley. 1988. Term-weighting approaches in automatic text retrieval. *Information Processing & Management* 24, 5 (1988), 513–523.

ACM Transactions on Knowledge Discovery from Data, Vol. 18, No. 3, Article 64. Publication date: January 2024.
[59] Bin Shen, Bao-Di Liu, Qifan Wang, and Rongrong Ji. 2014. Robust nonnegative matrix factorization via L 1 norm regularization by multiplicative updating rules. In Proceedings of the 2014 IEEE International Conference on Image Processing (ICIP). IEEE, 5282–5286.

[60] Reza Shokri, Marco Stronati, Congzheng Song, and Vitaly Shmatikov. 2017. Membership inference attacks against machine learning models. In Proceedings of the 2017 IEEE Symposium on Security and Privacy (SP). IEEE, 3–18.

[61] Shuang Song, Kamalika Chaudhuri, and Anand D. Sarwate. 2013. Stochastic gradient descent with differentially private updates. In Proceedings of the 2013 IEEE Global Conference on Signal and Information Processing. IEEE, 245–248.

[62] Latanya Sweeney. 2015. Only you, your doctor, and many others may know. Technology Science 2015092903, 9 (2015), 29.

[63] Di Wang, Minwei Ye, and Jinhui Xu. 2017. Differentially private empirical risk minimization revisited: Faster and more general. In Advances in Neural Information Processing Systems. I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (Eds.). Vol. 30, Curran Associates, Inc. https://proceedings.neurips.cc/paper_files/paper/2017/file/f337d999d9ad116a7b4f3d409fccc6480-Paper.pdf

[64] Yu-Xiong Wang and Yu-Jin Zhang. 2012. Nonnegative matrix factorization: A comprehensive review. IEEE Transactions on Knowledge and Data Engineering 25, 6 (2012), 1336–1353.

[65] B. Weyrauch, B. Heisele, J. Huang, and V. Blanz. 2004. Component-based face recognition with 3D morphable models. In Proceedings of the 2004 Conference on Computer Vision and Pattern Recognition Workshop (CVPRW’04) Volume 5 - Volume 05 (CVPRW ’04). IEEE Computer Society, 85–. Retrieved from http://dl.acm.org/citation.cfm?id=1032636.1032976

[66] Hyenkyun Woo and Haesun Park. 2014. Robust asymmetric nonnegative matrix factorization. Computational and Applied Mathematics Reports, University of California, USA 2 (2014).

[67] Yangyang Xu, Wotao Yin, Zaiwen Wen, and Yin Zhang. 2012. An alternating direction algorithm for matrix completion with nonnegative factors. Frontiers of Mathematics in China 7, 2 (2012), 365–384.

[68] Yale. 2001. The extended yale face database B. (2001). Retrieved from http://vision.ucsd.edu/~iskwak/ExtYaleDatabase/ExtYaleB.html

[69] Feng Zhang, Victor E. Lee, and Kim-Kwang Raymond Choo. 2018. Jo-DPMF: Differentially private matrix factorization learning through joint optimization. Information Sciences 467 (2018), 271–281.

[70] Lijun Zhang, Zhengguang Chen, Miao Zheng, and Xiaofei He. 2011. Robust non-negativity matrix factorization. Frontiers of Electrical and Electronic Engineering in China 6, 2 (2011), 192–200. DOI: https://doi.org/10.1007/s11460-011-0128-0

[71] Renbo Zhao and Vincent Y. F. Tan. 2016. Online nonnegative matrix factorization with outliers. IEEE Transactions on Signal Processing 65, 3 (2016), 555–570.

[72] Ligeng Zhu, Zhijian Liu, and Song Han. 2019. Deep leakage from gradients. In Advances in Neural Information Processing Systems, H. Wallach, H. Larochelle, A. Beygelzimer, F. d’Alché-Buc, E. Fox, and R. Garnett, Vol. 32, Curran Associates, Inc. https://proceedings.neurips.cc/paper_files/paper/2019/file/60abc4002cc7b29142def8871531281a-Paper.pdf

Received 19 March 2023; revised 11 July 2023; accepted 2 November 2023

ACM Transactions on Knowledge Discovery from Data, Vol. 18, No. 3, Article 64. Publication date: January 2024.