Hindawi
Security and Communication Networks
Volume 2019, Article ID 9169802, 10 pages
https://doi.org/10.1155/2019/9169802

Research Article

Laplace Input and Output Perturbation for Differentially Private Principal Components Analysis

Yahong Xu,[1] **Geng Yang**,[1,2,3] **and Shuangjie Bai**[1]

[1]*College of Computer Science and Software, Nanjing University of Posts and Telecommunications, Nanjing, Jiangsu 210023, China*
[2]*Key Laboratory of Broadband Wireless Communication & Sensor Networks Technology of Ministry of Education, Nanjing University of Posts and Telecommunications, Nanjing 210003, China*
[3]*Jiangsu Key Laboratory of Big Data Security & Intelligent Processing, Nanjing, Jiangsu 210023, China*

Correspondence should be addressed to Geng Yang; yangg@njupt.edu.cn

Received 27 January 2019; Revised 10 July 2019; Accepted 13 August 2019; Published 3 November 2019

Academic Editor: Clemente Galdi

With the widespread application of big data, privacy-preserving data analysis has become a topic of increasing significance. The current research studies mainly focus on privacy-preserving classification and regression. However, principal component analysis (PCA) is also an effective data analysis method which can be used to reduce the data dimensionality, commonly used in data processing, machine learning, and data mining. In order to implement approximate PCA while preserving data privacy, we apply the Laplace mechanism to propose two differential privacy principal component analysis algorithms: Laplace input perturbation (LIP) and Laplace output perturbation (LOP). We evaluate the performance of LIP and LOP in terms of noise magnitude and approximation error theoretically and experimentally. In addition, we explore the variation of performance of the two algorithms with different parameters such as number of samples, target dimension, and privacy parameter. Theoretical and experimental results show that algorithm LIP adds less noise and has lower approximation error than LOP. To verify the effectiveness of algorithm LIP, we compare our LIP with other algorithms. The experimental results show that algorithm LIP can provide strong privacy guarantee and good data utility.

1. Introduction

In many modern information systems, the amount of data is very large. Massive data increase the difficulty of data analysis and processing. Principal component analysis (PCA) is a standard data analysis method, which can be used to reduce the data dimensionality. More specifically, it projects the original high-dimensional data to the space of principal components composed by the eigenvectors of the covariance matrix of the data to get low-dimensional data, which can represent most of information of the original data. PCA simplifies the data, making data easier to use while saving on the computational complexity of the algorithm. For example, face recognition is much faster when first projecting the data into lower dimension.

Financial and medical data often deal with private or sensitive information. If machine learning tasks or data mining algorithms work directly on the original data, the outputs of these algorithms will leak private information, which may pose potential threats to individuals. Therefore, privacy preservation has become an urgent problem that needs to be solved. Differential privacy (DP) [1] is an effective and provable privacy protection model. It attends to hide private information while ensuring basic statistics of the original data. The notion of differential privacy has two types: $(\varepsilon, 0)$-DP and (ε, δ)-DP [2]. $(\varepsilon, 0)$-DP is usually called pure differential privacy, while (ε, δ)-DP with $\delta > 0$ is called approximate differential privacy. (ε, δ)-DP is a weaker version of $(\varepsilon, 0)$-DP as the former provides freedom to violate strict differential privacy for some low probability events.

There are several approaches to making approximate PCA while satisfying differential privacy. Input perturbation adds noise to the data before computing the PCA, while output perturbation adds noise to the output of PCA. We

can add Laplace noise to implement input perturbation and output perturbation. Both approaches can effectively simplify data and preserve the data privacy; however, there are few studies on their performance. At the same privacy protection level, better performance (less noise and lower error) mean better data utility. In this paper, we propose two differential privacy principal component analysis algorithms and evaluate their performance.

Our main contributions are as follows:

(1) We apply Laplace mechanism to propose two differential privacy principle component analysis algorithms, Laplace input perturbation (LIP) and Laplace output perturbation (LOP), and give proof for its $(\varepsilon, 0)$-DP.

(2) We offer two criteria, i.e., noise magnitude and approximation error, to evaluate the performance of two algorithms. Less noise and lower approximation error result in better performance. Through theoretical verification, we ensure that LIP has better performance than LOP.

(3) We conduct the experiments to verify the performance of LIP and LOP in terms of noise magnitude and approximation error on five real datasets. We further explore the variation of performance of the two algorithms with different parameters such as number of samples, target dimension, and privacy parameter. The experimental results show that at the different parameters, algorithm LIP always adds less noise and has lower approximation error than LOP. Compared with other algorithms, LIP can also provide good data utility.

The rest of the paper is organized as follows. Section 3 introduces principle component analysis, differential privacy, and Laplace mechanism. Section 4 first describes the two differential privacy principle component analysis algorithms and then analyzes the privacy and utility. Section 5 shows the performance of two algorithms on five real datasets. Section 6 concludes the paper.

2. Related Work

Since Dwork proposed the concept of differential privacy, data preservation in the field of data mining and machine learning has received considerable attention. The current research studies mainly focus on privacy-preserving classification, regression, and frequent itemset mining.

Classification technology plays an important role in data prediction, which aims to build models that can describe and distinguish data. The typical privacy protection classification algorithms are SuLQ-Based ID3, DiffP-C4.5, and DiffGen. The basic idea of SuLQ-Based ID3 [3] is to add noise to true count value before calculating the information gain of the attributes and finally generate the corresponding decision tree. Although this method can satisfy differential privacy, the added noise is too large. To overcome the disadvantages of SuLQ-Based ID3, DiffP-C4.5 [4] first selects and splits attributes by exponential mechanism. However, this method can only support few analyses and queries. The classification accuracy of DiffGen [5] is higher than SuLQ-Based ID3 and DiffP-C4.5 from the perspective of theory and practical application; unfortunately, when the dimension of the classification attribute is very large, the selection method based on the exponential mechanism is inefficient and may exhaust the privacy budget. Frequent itemset mining is an effective data analysis method; it aims to discover itemsets that frequently appear in the dataset. Bhaskar et al. proposed algorithm truncated frequency (TF) [4]; it reduces the number of candidate itemsets depending on their own frequency. However, when the number of target itemsets is large, this method will fail. Considering this weakness, Li et al. proposed algorithm PrivBasis [5] according to the idea of θ-base (θ is a threshold) to generate candidate itemsets. However, generating θ-base is not very easy. Inspired by Zeng and Li, Wang et al. proposed algorithm PrivSuper [6] that randomly truncates transactions in a dataset, which will cause large truncation error. Regression is a common data analysis method in machine learning; it is a quantitative relationship that determines the interdependence of two or more attributes. The typical regression algorithms based on differential privacy are logistic regression and linear regression. In algorithms LPLog [7] and ObjectivePerb [8], the noise magnitude is decided by the sensitivity of the weight vector and the cost of computing sensitivity is high. Considering the disadvantages of the two algorithms, algorithm functional mechanism (FM) [9] controls the noise magnitude by the sensitivity of function itself instead of the weight vector.

However, there are few studies on differential privacy principal component analysis. Blum et al. [10] first proposed the early input perturbation framework SULQ, but not for data publishing. Chaudhuri et al. [11] proposed a privacy-preserving PCA algorithm MOD-SULQ based on the exponential mechanism, which can be used for data publishing. Kapralov and Talwar [12] argued that the algorithm (Chaudhuri et al.) lacks convergence time guarantee, and they also designed a complex algorithm using the exponential mechanism, but it is complicated to implement for high-dimensional data. Dwork et al. [13] provided the algorithms for (ε, δ)-DP, adding Gaussian noise to the original sample covariance matrix. Inspired by Dwork, Imtiaz et al [14, 15] and Jiang et al. [2] designed their algorithms for $(\varepsilon, 0)$-DP. Both of them added Wishart noise with parameters chosen to have a better utility bound.

3. Preliminaries

Given a dataset $X = [x_1, x_2, \ldots, x_n]^T$ where $x_i \in R^d$ is the i-th record. The matrix $X \in R^{n \times d}$ contains information about d attributes of n individuals (generally $d < n$). Following previous work on privacy-preserving PCA, we also assume $\|x_i\|_2 \leq 1$, $\|\cdot\|_2$ denotes the l_2 norm. For a vector $a \in R^d$, $\|a\|_2 = \sqrt{\sum_{i=1}^{d} a_i^2}$.

The covariance matrix of the original data is

$$A = \frac{1}{n} X^T X = \frac{1}{n} \sum_{i=1}^{n} x_i^T x_i, \tag{1}$$

where A is a $d \times d$ symmetric matrix.

The principal components are obtained by computing the eigenvalues and corresponding eigenvectors of the covariance matrix A:

$$Av_i = \lambda_i v_i, \tag{2}$$

where $\lambda_i (1 \le i \le d)$ is the eigenvalue, denoting the proportion of information that corresponding component includes. Larger λ_i means the component is more important. We assume λ_i are ordered decreasingly, i.e., $\lambda_1 \ge \lambda_2 \ge \cdots \ge \lambda_d \ge 0$. v_i is the corresponding eigenvector.

In order to reduce the data to low dimension, a target dimension k is needed. We want to select first k eigenvectors which correspond to the top k eigenvalues. Given a threshold $\alpha (0 \le \alpha \le 1)$, α denotes accumulative contribution rate of the principal components [16]. Target dimension k can be decided by

$$\frac{\sum_{i=1}^{k} \lambda_i}{\sum_{i=1}^{d} \lambda_i} \ge \alpha. \tag{3}$$

Suppose $V_k = (v_1, v_2, \ldots, v_k)$ is the first k eigenvectors of A, v_i and v_j are orthonormal. We project the original data X to the V_k to get low-dimensional data:

$$Y = XV_k, \tag{4}$$

where $Y \in R^{n \times k}$; we can also get the rank-k approximation [17] of X:

$$Z = XV_k V_k^T. \tag{5}$$

Our algorithms want to keep the statistics of X as much as possible, and the approximation error between Z and X can be measured by

$$\text{MSE} = \|Z - X\|_F. \tag{6}$$

Lower MSE provides better data utility. $\|\cdot\|_F$ denotes the Frobenius norm. For a matrix $C \in R^{m \times n}$, $\|C\|_F = \sqrt{\sum_{i=1}^{m} \sum_{j=1}^{n} c_{ij}^2}$.

Now, we introduce the definition of differential privacy.

Definition 1 (differential privacy) [18]. A randomized mechanism M is differentially private if for any neighbouring datasets D and D' (with at most one different record) and for all output $O (O \subseteq \text{Range}(M))$,

$$\Pr[M(D) \in O] \le e^{\varepsilon} \times \Pr[M(D') \in O], \tag{7}$$

where ε is the privacy budget controlling the strength of privacy guarantee; lower ε ensures more privacy.

Sensitivity is the key parameter that determines how much noise is required.

Definition 2 (sensitivity) [19]. For a function $f : D \longrightarrow R^d$ and any neighbouring datasets D and D', the sensitivity of f is defined as

$$\Delta f = \max_{D,D'} \|f(D) - f(D')\|_1. \tag{8}$$

The sensitivity describes the largest change due to an data entry replacement. Sensitivity Δf is only related to the function f.

The Laplace mechanism adds independent noise to the data; we use $\text{Lap}(b)$ to represent the noise sampled from Laplace distribution with a scaling of b.

Definition 3 (Laplace mechanism) [19]. Given a dataset D, for a function $f : D \longrightarrow R^d$, with sensitivity Δf, the mechanism M provides ε-DP satisfying

$$M(D) = f(D) + \text{Lap}\left(\frac{\Delta f}{\varepsilon}\right). \tag{9}$$

Here, $\text{Lap}(\cdot)$ is a random variable. Its probability density function is

$$p(x) = \frac{1}{2b} e^{-(|x|/b)}. \tag{10}$$

4. Proposed Algorithms and Analysis

In this section, we describe two differential privacy principal component analysis algorithms: LIP and LOP. Through theoretical analysis, we prove the two algorithms satisfy $(\varepsilon, 0)$-DP. Meanwhile, we investigate the utility of proposed algorithms.

4.1. Algorithm Description. In algorithm LIP, we use Laplace distribution to generate symmetric noise matrix and then add it to the data covariance matrix. After computing the eigenvalues and corresponding eigenvectors of the noised covariance matrix, we select first k eigenvectors to make up principal components space. In the end, we obtain low-dimensional data by projecting the original high-dimensional data to the principal components space. Algorithm LIP is described in Algorithm 1.

V_k' are the first k eigenvectors of the noised covariance matrix A', which is close to the true first k eigenvectors V_k of covariance matrix A [13].

Besides adding noise prior to computing PCA, we also add noise to the output of PCA. According to differential privacy parallel composition [20], the whole dataset is private as long as each record is private; a simple idea is adding noise to each record to protect private information. However, if this privacy preservation method is directly applied to big data, the introduced noise will significantly increase so that data utility dramatically drops. In order to reduce noise without decreasing the level of privacy preservation, we can add noise to fewer but most important parts of data. Algorithm LOP projects the original high-dimensional data to the principal component space to get low-dimensional data. The low-dimensional data are important data, so we add noise to them to protect data privacy. Algorithm LOP is described in Algorithm 2.

4.2. Privacy Analysis. Before proving that LIP and LOP satisfy $(\varepsilon, 0)$-DP, we should analyze the sensitivities of these two algorithms. Suppose there are two neighbouring datasets $X = [x_1, \ldots, x_i, \ldots, x_n]^T \in R^{n \times d}$ and $X' = [x_1, \ldots,$

Input: matrix $X \in R^{n\times d}$, number of samples n, attributes d, privacy parameter ε;
Output: Z_1: the rank-k approximation matrix
(1) Compute covariance matrix $A = (1/n)X^T X$;
(2) Noise matrix $E_1 \in R^{d\times d}$ is a symmetric matrix where the upper triangle is $(d^2 + d)/2$ i.i.d. sample from $\mathrm{Lap}(2d/n\varepsilon)$, and each lower triangle entry is copied from the opposite position;
(3) Add noise $A' = A + E_1$;
(4) Compute eigenvalues and corresponding eigenvectors of the noised covariance matrix $A'v_i' = \lambda_i' v_i'$;
(5) Given a threshold α, select top k eigenvectors V_k' of A', low-dimensional data $Y_1 = XV_k'$;
(6) The rank-k approximation matrix $Z_1 = Y_1 V_k'^T$;

Algorithm 1: Laplace input perturbation (LIP).

Input: matrix $X \in R^{n\times d}$, number of samples n, attributes d, privacy parameter ε;
Output: Z_2: the rank-k approximation matrix
(1) Compute covariance matrix $A = (1/n)X^T X$;
(2) Compute eigenvalues and corresponding eigenvectors $Av_i = \lambda_i v_i$;
(3) Given a threshold α, select top k eigenvectors V_k of A, low-dimensional data $Y_2 = XV_k$;
(4) Noise matrix E_2 is a $n \times k$ matrix where the whole elements are i.i.d. samples from $\mathrm{Lap}(2d/\varepsilon)$
(5) Add noise $Y_2' = Y_2 + E_2$;
(6) The rank-k approximation matrix $Z_2 = Y_2' V_k^T$;

Algorithm 2: Laplace output perturbation (LOP).

$x_i', \ldots, x_n]^T \in R^{n\times d}$ where $x_i \neq x_i'$, we assume the normalized data vector $\|x_i\|_2 \leq 1$.

Lemma 1. *In algorithm LIP, for all the input data, denote $f(X) = (1/n)X^T X$; then, the sensitivity of the function $f(X)$ equals $2d/n$.*

Proof. Suppose that A_1 and A_2 are the covariance matrices of X and X', respectively. □

$$\begin{cases} A_1 = \dfrac{1}{n}X^T X, \\ \\ A_2 = \dfrac{1}{n}X'^T X'. \end{cases} \tag{11}$$

According to Definition 2, the sensitivity of function $f(X)$ is $\max A_1 - A_{21}$. Then, we have

$$\|A_1 - A_2\|_1 = \left\|\frac{1}{n}X^T X - \frac{1}{n}X'^T X'\right\|_1 = \frac{1}{n}\left\|x_i^T x_i - x_i'^T x_i'\right\|_1, \tag{12}$$

where $\|\cdot\|_1$ denotes the l_1 norm, for a matrix $C \in R^{m\times n}$, $\|C\|_1 = \max_j \sum_{i=1}^m |c_{ij}| \, (1 \leq j \leq n)$. For the normalized $\|x_i\|_2 \leq 1$, we have

$$\begin{aligned} \|A_1 - A_2\|_1 &= \frac{1}{n}\|x_i^T x_i - x_i'^T x_i'\|_1 \\ &\leq \frac{1}{n}\left(\|x_i^T x_i\|_1 + \|x_i'^T x_i'\|_1\right) \\ &\leq \frac{2d}{n}. \end{aligned} \tag{13}$$

Theorem 1. *Algorithm LIP satisfies $(\varepsilon, 0)$-DP.*

Proof. For A' derived from algorithm LIP on X and X', we obtain $A' = A_1 + N_1$ and $A' = A_2 + N_2$ where N_1 and N_2 are the corresponding noise matrices. □

$$\frac{f(A' \mid X)}{f(A' \mid X')} = \frac{p(N_1)}{p(N_2)} = e^{(n\varepsilon/2d)(\|N_2\|_1 - \|N_1\|_1)}, \tag{14}$$

where $p(N_1)$ and $p(N_2)$ are the density functions of the output functions at neighbouring datasets X and X'. According to Lemma 1, we have

$$\|N_2\|_1 - \|N_1\|_1 \leq \|N_2 - N_1\|_1 = \|A_1 - A_2\|_1 \leq \frac{2d}{n}. \tag{15}$$

Combining equations (14) and (15), we can obtain

$$\frac{P(A' \mid X)}{P(A' \mid X')} \leq e^{\varepsilon}. \tag{16}$$

Therefore, algorithm LIP satisfies $(\varepsilon, 0)$-DP.

Lemma 2. *In algorithm LOP, given V_k, denote $g(X) = XV_k$; then, the sensitivity of the function $g(X)$ equals $2d$.*

Proof. Suppose that Y_1, Y_2 and V_k, V_k' are the low-dimensional data and first k orthogonal eigenvectors of X and X', respectively. □

$$\begin{cases} Y_1 = XV_k, \\ Y_2 = X'V_k' \end{cases} \tag{17}$$

According to Definition 2, the sensitivity of function $g(X)$ is $\max \|Y_1 - Y_2\|_1$. Then, we have

$$\|Y_1 - Y_2\|_1 = \|XV_k - X'V'_k\|_1. \tag{18}$$

Since $V_k = (v_1, \ldots, v_k)$ and $V'_k = (v'_1, \ldots, V'_k)$ are both composed of k unit orthogonal eigenvectors,

$$\|Y_1 - Y_2\|_1 = \|XV_k - X'V'_k\|_1 \le \|X - X'\|_1 \le \|x_i - x'_i\|_1. \tag{19}$$

For the normalized $\|x_i\|_2 \le 1$, we have

$$\begin{aligned}\|Y_1 - Y_2\|_1 &\le \|x_i - x'_i\|_1 \le \|x_i\|_1 + \|x'_i\|_1 \\ &\le 2d.\end{aligned} \tag{20}$$

Theorem 2. *Algorithm LOP satisfies* $(\varepsilon, 0)$*-DP.*

Proof. For Y' *derived* from algorithm LOP on X and X', we obtain $Y' = Y_1 + N_1$ and $Y' = Y_2 + N_2$, where N_1 and N_2 are the corresponding noise matrices. □

$$\frac{g(Y' \mid X)}{g(Y' \mid X')} = \frac{p(N_1)}{p(N_2)} = e^{(\varepsilon/2d)(\|N_2\|_1 - \|N_1\|_1)}, \tag{21}$$

where $p(N_1)$ and $p(N_2)$ are the density functions of the output functions at neighbouring datasets X and X'. According to Lemma 2, we have

$$\|N_2\|_1 - \|N_1\|_1 \le N_2 - \|N_1\|_1 = \|Y_1 - Y_2\|_1 \le 2d. \tag{22}$$

Combining equations (22) and (23), we can obtain

$$\frac{P(Y' \mid X)}{P(Y' \mid X')} \le e^{\varepsilon}. \tag{23}$$

Therefore, algorithm LOP satisfies $(\varepsilon, 0)$-DP.

4.3. Utility Analysis. In Section 4.2, we proved that algorithms LIP and LOP both satisfy $(\varepsilon, 0)$-DP. Next, we evaluate the performance of the two algorithms. In order to protect data privacy, we add noise to covariance matrix and low-dimensional matrix in LIP and LOP, respectively. Adding noise may have effect on the performance of algorithms, and noise magnitude directly determines the magnitude of effect. In addition, approximation error also describes the performance of algorithms. Better data utility means less noise and lower approximation error, so we evaluate algorithms LIP and LOP in terms of noise magnitude and approximation error.

Theorem 3. *For a given privacy parameter ε, algorithm LIP adds less noise than LOP. The larger the samples n and target dimension k are, the less noise the algorithm LIP adds than LOP.*

Proof. In algorithm LIP, noise matrix E_1 has d^2 elements, each element adds noise $\mathrm{Lap}(2d/n\varepsilon)$, and the variance of noise is about $N_1 = O(d^4/n^2\varepsilon^2)$. □

In algorithm LOP, noise matrix E_2 has $n \cdot k$ elements, each element adds noise $\mathrm{Lap}(2d/\varepsilon)$, and the variance of noise is about $N_2 = O(nkd^2/\varepsilon^2)$.

Now, we compare N_1 and N_2 to measure the noise magnitude of two algorithms:

$$\begin{aligned}N_1 = O\left(\frac{d^4}{n^2\varepsilon^2}\right) &= O\left(d^2 \cdot \frac{d^2}{n^2\varepsilon^2}\right) < O\left(d^2 \cdot \frac{n^2}{n^2\varepsilon^2}\right) < O\left(\frac{d^2}{\varepsilon^2}\right) \\ &< O\left(\frac{nkd^2}{\varepsilon^2}\right) = N_2,\end{aligned} \tag{24}$$

where $d < n$. From formula (24), we observe $N_1 < N_2$, that is, algorithm LIP adds less noise than LOP.

Let $\theta = N_1/N_2 = O(d^4/n^2\varepsilon^2)/O(nkd^2/\varepsilon^2) = O(d^2/n^3k) < O(1)$ be the noise ratio. We observe that $\theta < 1$. Furthermore, θ and n, k show strong negative correlation. That is, if we take a larger sample n and target dimension k, LIP will add less noise than LOP.

Theorem 4. *For a given privacy parameter ε, algorithm LIP has lower error than LOP in the rank-k approximation of raw data.*

Proof. In algorithm LIP, the rank-k approximation of X is

$$Z_1 = XV'_kV'^T_k. \tag{25}$$

□

In algorithm LOP, the rank-k approximation of X is

$$Z_2 = (XV_k + E_2)V_k^T. \tag{26}$$

Let

$$\mathrm{MSE1} = \|Z_1 - X\|_F = \|XV'_kV'^T_k - X\|_F, \tag{27}$$

$$\mathrm{MSE2} = \|Z_2 - X\|_F = \|(XV_k + E_2)V_k^T - X\|_F. \tag{28}$$

MSE1 and MSE2 denote approximation errors in X and Z_1, X and Z_2, respectively. Now, we compare MSE1 and MSE2 to measure the approximation errors of two algorithms. Based on linear algebra, we have

$$\|XV_kV_k^T - X\|_F^2 = \frac{\lambda_{k+1} + \cdots + \lambda_d}{\lambda_1 + \cdots + \lambda_d}\|X\|_F^2. \tag{29}$$

In equation (29), V_k and $\lambda_1, \ldots, \lambda_d$ are computed with the accurate matrix A while V'_k is computed based on the matrix A' with noise in equation (27). Theorem 6 in Dwork et al. [13] provides the closeness between V_k and V'_k. V'_k not only captures large amount of variance, but is also close to the V_k of A. Theorem 6 in Dwork et al. [13] also gives the upper bound between V_k and V'_k; when $\sigma_k - \sigma_k^2 = \omega(\sqrt{n}\,\Delta_{\varepsilon,\delta})$, there is

$$\|V_kV_k^T - V'_kV'^T_k\|_F = O\left(\frac{\sqrt{kn}\,\Delta_{\varepsilon,\delta}}{\sigma_k^2 - \sigma_{k+1}^2}\right), \tag{30}$$

where $\Delta_{\varepsilon,\delta} = \sqrt{2\ln(1.25/\delta)}/\varepsilon$ is the noise parameter in Gaussian distribution. In Gaussian mechanism [13], noise matrices are

samples from $N(0, \Delta^2_{\varepsilon,\delta})$, equalling Lap$(0, \ (2d/n\varepsilon))$ in our Algorithm 1. σ_k is a singular value in SVD; according to the relationship between PCA and SVD, we have $\sigma_k^2 = \lambda_k$.

From equation (30), we know that V_k and V_k' are very close but still have little difference. Under the effect of difference and noise, we have

$$\left\| XV_k'V_k'^T - X \right\|_F > \left\| XV_kV_k^T - X \right\|_F. \tag{31}$$

Combining equations (29) and (31), we have

$$\text{MSE1} = \left\| XV_k'V_k'^T - X \right\|_F > \sqrt{\frac{\lambda_{k+1} + \cdots + \lambda_d}{\lambda_1 + \cdots + \lambda_d}} \|X\|_F. \tag{32}$$

From equation (28), V_k is computed based on the accurate matrix A.

$$\text{MSE2} = \left\| (XV_k + E_2)V_k^T - X \right\|_F = \left\| XV_kV_k^T - X + E_2V_k^T \right\|_F. \tag{33}$$

According to $\|A + B\|_F \le \|A\|_F + \|B\|_F$, we have

$$\begin{aligned} \text{MSE2} &\le \left\| XV_kV_k^T - X \right\|_F + \left\| E_2V_k^T \right\|_F \\ &\le \sqrt{\frac{\lambda_{k+1} + \cdots + \lambda_d}{\lambda_1 + \cdots + \lambda_d}} \|X\|_F + \left\| E_2V_k^T \right\|_F. \end{aligned} \tag{34}$$

Let $\eta = \text{MSE1/MSE2}$; $\eta < 1$ indicates algorithm LIP has lower approximation error than LOP:

$$\begin{aligned} \eta &= \frac{\text{MSE1}}{\text{MSE2}} \\ &> \frac{\sqrt{(\lambda_{k+1} + \cdots + \lambda_d)/(\lambda_1 + \cdots + \lambda_d)} \|X\|_F}{\sqrt{(\lambda_{k+1} + \cdots + \lambda_d)/(\lambda_1 + \cdots + \lambda_d)} \|X\|_F + \left\| E_2V_k^T \right\|_F} \\ &> \frac{1}{1 + \left(\left\| E_2V_k^T \right\|_F / \sqrt{(\lambda_{k+1} + \cdots + \lambda_d)/(\lambda_1 + \cdots + \lambda_d)} \, \|X\|_F \right)}. \end{aligned} \tag{35}$$

E_2 is a $n \times k$ matrix. With the increase of target dimension k, the value of $\|E_2V_k^T\|_F$ will increase, while $\sqrt{(\lambda_{k+1} + \cdots + \lambda_d)/(\lambda_1 + \cdots + \lambda_d)}$ will decrease. Thus, approximation error ratio η decreases and $\max \eta < 1$. Since $\eta < 1$, LIP has lower error than LOP in the rank-k approximation of raw data.

Theorem 3 and Theorem 4 prove that algorithm LIP adds less noise and has lower approximation error than LOP, that is, algorithm LIP outperforms LOP in data utility.

5. Experimental Results and Analysis

In this section, we will give some experimental results to verify that algorithm LIP outperforms LOP in data utility. We compare algorithms LIP and LOP in terms of noise magnitude and approximation error. In addition, we investigate the variation of performance of the two algorithms with different key parameters such as number of samples n, target dimension k, and privacy parameter ε. Five UCI datasets are used in our experiments: Secom [21], Covtype [22], Musk [23], Handwritten [24] and Waveform [25]. We preprocess the data by subtracting the mean and normalizing data to meet the condition $x_{i2} \le 1$. We select target dimension k so that the accumulative contribution rate of the principal components α is at least 85%. In all cases, we show the average performance over 100 runs of each algorithm.

5.1. Experiments for Noise Magnitude. In this section, we evaluate the performance of algorithms LIP and LOP by comparing the magnitude of introduced noise. In Theorem 3, the ratio of introduced noise $\theta = N_1/N_2 = O(d^2/n^3k)$ indicates that θ and n, k have strong negative correlation and $\theta < 1$. In the experiment, we verify that above conclusions are correct (we use the Frobenius norm of noise matrices E_1 and E_2 to represent N_1 and N_2).

In order to better present the experimental results of all datasets in one figure, we unify the noise ratio θ, and all the objects are scaled to same size (θ in datasets Secom, Covtype, Musk, Handwritten, and Waveform are 1, 10^5, 10^2, 10^2, and 10^3 times the original value, respectively). For this experiment, we keep number of samples n fixed to investigate the relationship between the ratio of introduced noise θ and target dimension k. In Figure 1 we observe that even expanding the value of θ, θ is always less than 1; with the increase of value k, the ratio of introduced noise θ on five datasets continuously decreases. That is, θ and k are negatively correlated; the larger the target dimension k is, the less noise the LIP adds than LOP. The result is consistent with Theorem 3.

Then, we explore the effect of samples n on the ratio of introduced noise θ. In case of fixing target dimension k, Figure 2 shows θ is always less than 1, and θ decreases as the value of n increases in five datasets. That is, θ and n are negatively correlated; the larger the samples n are, the less noise the LIP adds than LOP. The result is consistent with Theorem 3.

5.2. Experiments for Approximation Error. In this section, we evaluate the performance of algorithms LIP and LOP by comparing the approximation error. In Theorem 4, the ratio of approximation error $\eta = \text{MSE1/MSE2}$ indicates that $\eta < 1$. Thus, in the experiment, we verify (1) η is less than 1 and (2) η and k are negatively correlated while η and ε are positively correlated.

In Theorem 3 and Section 5.1, we observe that the larger the target dimension k is, the less noise the algorithm LIP adds than LOP. Less noise results in lower error. In addition, we further explore mathematical expression of η in Theorem 4 and find that $\eta < 1$, and η and k are negatively correlated. Similarly, we unify the approximation error ratio η, and all the objects are scaled to same size (η in each dataset is 10^3, 10^2, 10^3, 10^2, and 10^2 times the original value). For this experiment, we keep privacy parameter ε fixed to investigate the relationship between the ratio of approximation error η and target dimension k. As shown in Figure 3, when the

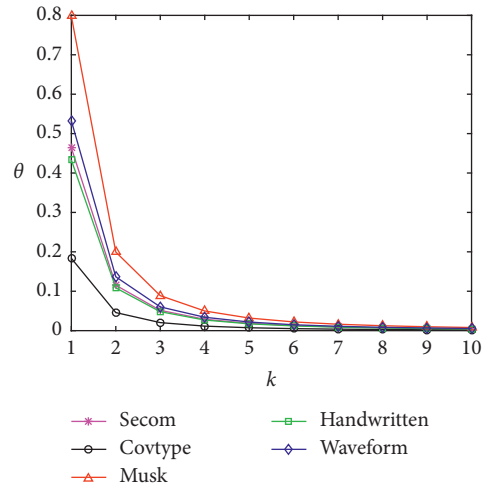

Figure 1: Variation of θ with different values of target dimension k.

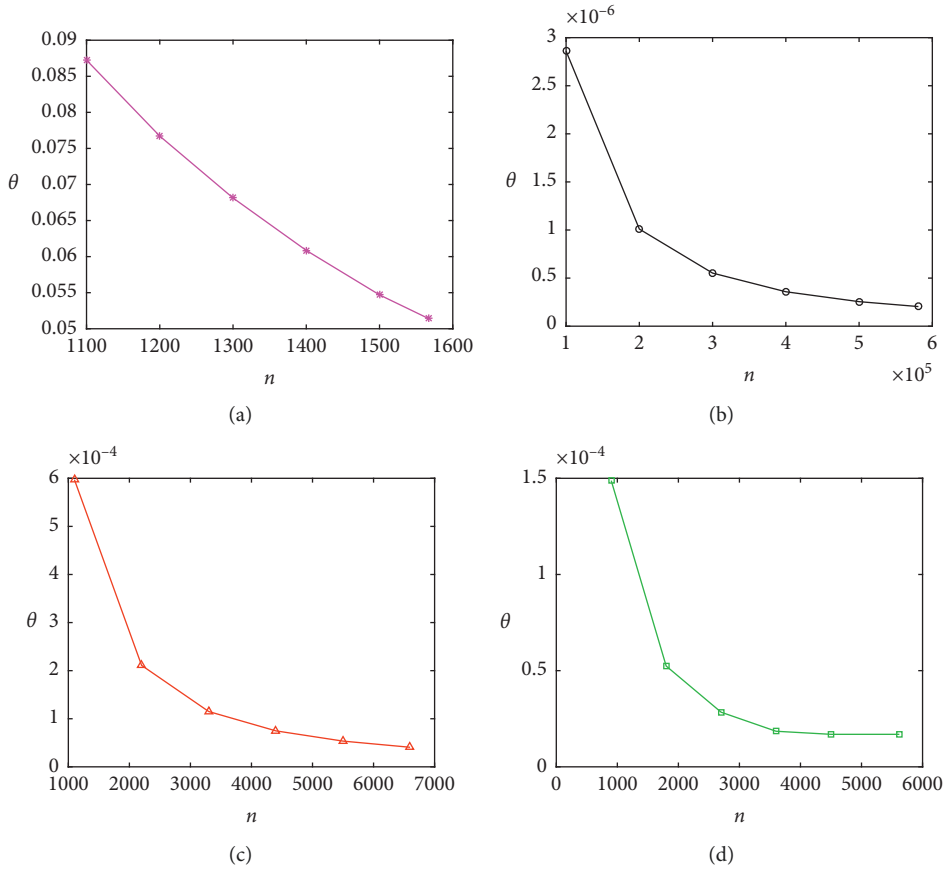

Figure 2: Continued.

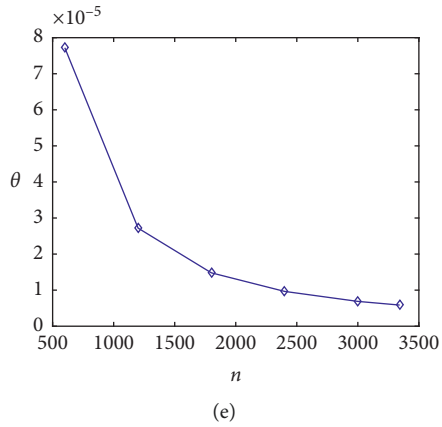

Figure 2: Variation of θ with different values of samples n. (a) Secom. (b) Covtype. (c) Musk. (d) Handwritten. (e) Waveform.

value of k increases, the ratio of approximation error η decreases and $\max \eta < 1$. In other words, η and k are negatively correlated, which means when k is larger, LIP has lower error than LOP in the rank-k approximation of raw data. The experimental result is consistent with Theorem 4.

Finally, we explore the variation of η with privacy parameter ε. Similarly, we unify the approximation error ratio η, and all the objects are scaled to same size (η in each dataset is 10, 10, 10^3, 10^3, and 10^2 times the original value). In Figure 4, for all the datasets, we observe that as ε increases, the ratio of approximation error η increases. Furthermore, η and ε are positively correlated, even in the case of no privacy preserving, i.e., $\varepsilon = 10$, η is still less than 1. It can be explained as follows: Dwork et al. pointed out that V_k' is close to the true top k eigenvectors eigenvectors V_k in input perturbation [13], that is, algorithm LIP is not very sensitive to privacy parameter ε. Output perturbation due to directly adding noise to the output and privacy parameter ε plays a negative role in data utility. Lower ε means more noise and higher approximation error. Thus, when ε increases, MSE1 decreases slightly while MSE2 decreases greatly and η = MSE1/MSE2 increases. Therefore, at the same privacy protection level, algorithm LIP has lower error than LOP in the rank-k approximation of raw data.

5.3. Experiments for Accuracy. In Sections 5.1 and 5.2, we verify that algorithm LIP outperforms LOP in data utility. To verify the effectiveness of algorithm LIP compared with the existing algorithms AG [13] and PPM [26], we evaluate the classification accuracy on Handwritten and Waveform datasets. The classifier used in the experiment is linear support vector machine (SVM). In SVM, there are many parameters that can affect accuracy; we mainly consider the influence of privacy parameter ε on the accuracy.

In Figure 5, we show the variation of accuracy with different values of ε. For all the datasets, we observe that as ε increases (higher privacy risk), the accuracy increases significantly, which indicates that the value of ε has an important effect on accuracy. On the other hand, the accuracy

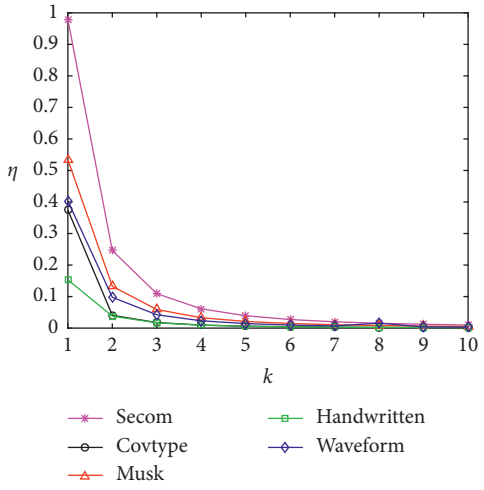

Figure 3: Variation of η with different values of target dimension k.

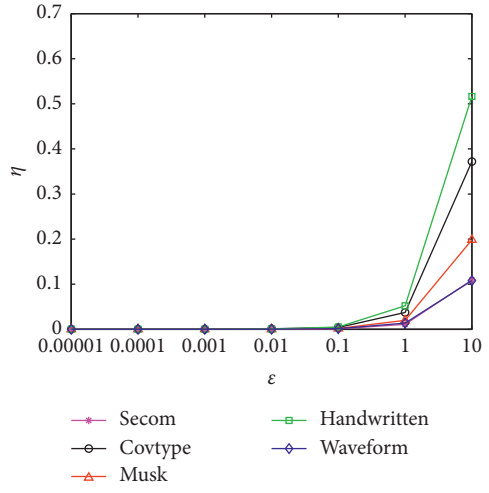

Figure 4: Variation of η with different values of privacy parameter ε.

of algorithms AG and LIP are higher than that of PPM on the two datasets. In addition, algorithm AG outperforms LIP in accuracy; it can be explained as the utility gap between

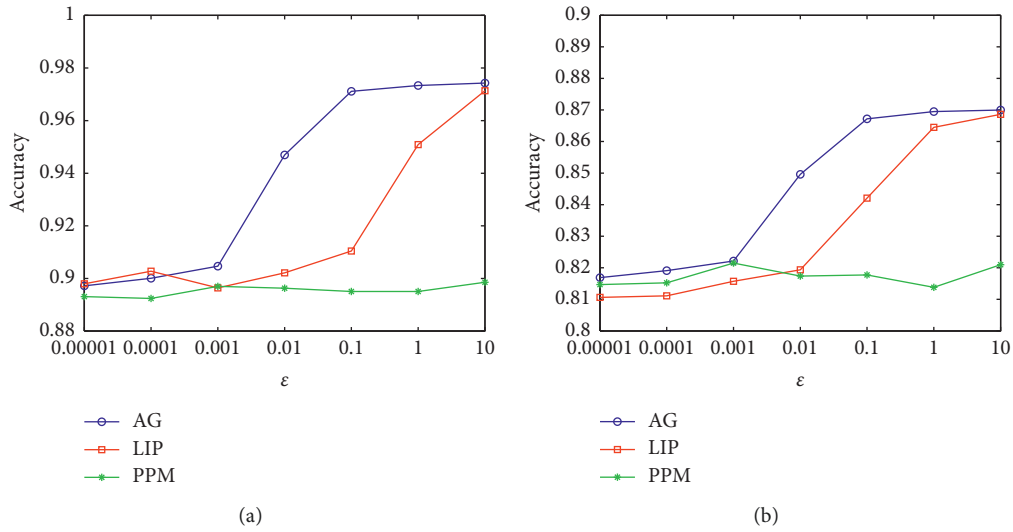

Figure 5: Variation of accuracy with different values of privacy parameter ε. (a) Handwritten. (b) Waveform.

(ε, δ)-DP and $(\varepsilon, 0)$-DP (AG satisfies (ε, δ)-DP and LIP satisfies $(\varepsilon, 0)$-DP). $(\varepsilon, 0)$-DP provides stronger privacy guarantee and weaker data utility than (ε, δ)-DP. For large enough ε, our algorithm LIP can match the performance of AG. More important, it can provide a stronger privacy guarantee than AG. In conclusion, algorithm LIP achieves both strong privacy guarantee and good data utility.

6. Conclusions

In this paper, we propose two algorithms Laplace input perturbation (LIP) and Laplace output perturbation (LOP) for differential privacy principal component analysis. We compare the performance of LIP and LOP in terms of noise magnitude and approximation error via theoretical analysis. Then we conduct many experiments to verify the performance of two algorithms on five data sets. In the experiments, we show the variation of performance of the two algorithms with different parameters such as privacy parameter, target dimension and samples. Our theoretical and experimental results indicate that algorithm Laplace input perturbation (LIP) adds less noise and has lower approximation error than Laplace output perturbation (LOP). Last, to verify the effectiveness of algorithm LIP, we compare our LIP with other recent algorithms AG and PPM, the experimental results show that algorithm LIP can provide strong privacy guarantee and good data utility.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This study was supported by the National Natural Science Foundation of China (61572263, 61502251, 61602263, and 61872197), the Postgraduate Research & Practice Innovation Program of Jiangsu Province (KYCX18_0891), the Natural Science Foundation of Jiangsu Province (BK20161516 and BK20160916), the Postdoctoral Science Foundation Project of China (2016M601859), and the Natural Research Foundation of Nanjing University of Posts and Telecommunications (NY217119).

References

[1] C. Dwork, F. McSherry, K. Nissim, and A. Smith, "Calibrating noise to sensitivity in private data analysis," in *Proceedings of the Theory of Cryptography Conference*, pp. 265–284, Springer, New York, NY, USA, March 2006.

[2] W. Jiang, C. Xie, and Z. Zhang, "Wishart mechanism for differentially private principal components analysis," in *Proceedings of the Association for the Advancement of Artificial Intelligence*, pp. 1730–1736, AAAI, Phoenix, AZ, USA, February 2016.

[3] N. Mohammed, R. Chen, B. Fung, and P. S. Yu, "Differentially private data release for data mining," in *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 493–501, ACM, San Diego, CA, USA, August 2011.

[4] R. Bhaskar, S. Laxman, A. Smith, and A. Thakurta, "Discovering frequent patterns in sensitive data," in *Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 503–512, ACM, Washington, DC, USA, July 2010.

[5] N. Li, W. Qardaji, D. Su, and J. Cao, "PrivBasis," *Proceedings of the VLDB Endowment*, vol. 5, no. 11, pp. 1340–1351, 2012.

[6] N. Wang, X. Xiao, Y. Yang, Z. Zhang, Y. Gu, and G. Yu, "Privsuper: a superset-first approach to frequent itemset mining under differential privacy," in *Proceedings of the IEEE 33rd International Conference on Data Engineering (ICDE)*, pp. 809–820, IEEE, San Diego, CA, USA, April 2017.

[7] J. Zhang, Z. Zhang, X. Xiao, Y. Yang, and M. Winslett, "Functional mechanism," *Proceedings of the VLDB Endowment*, vol. 5, no. 11, pp. 1364–1375, 2012.
[8] X. Cheng, S. Su, S. Xu, L. Xiong, K. Xiao, and M. Zhao, "A two-phase algorithm for differentially private frequent subgraph mining," *IEEE Transactions on Knowledge and Data Engineering*, vol. 30, no. 8, pp. 1411–1425, 2018.
[9] B. Anandan and C. Clifton, "Differentially private feature selection for data mining," in *Proceedings of the Fourth ACM International Workshop on Security and Privacy Analytics*, pp. 43–53, ACM, Tempe, AZ, USA, March 2018.
[10] A. Blum, C. Dwork, F. McSherry, and K. Nissim, "Practical privacy: the sulq framework," in *Proceedings of the Twenty-Fourth ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems*, pp. 128–138, ACM, Baltimore, MD, USA, June 2005.
[11] K. Chaudhuri, A. Sarwate, and K. Sinha, "Near-optimal differentially private principal components," in *Proceedings of the Advances in Neural Information Processing Systems*, pp. 989–997, Lake Tahoe, NV, USA, December 2012.
[12] M. Kapralov and K. Talwar, "On differentially private low rank approximation," in *Proceedings of the Twenty-Fourth Annual Acm-Siam Symposium on Discrete Algorithms*, pp. 1395–1414, SIAM, New Orleans, LA, USA, January 2013.
[13] C. Dwork, K. Talwar, A. Thakurta, and L. Zhang, "Analyze gauss: optimal bounds for privacy-preserving principal component analysis," in *Proceedings of the Forty-Sixth Annual ACM Symposium on Theory of Computing*, pp. 11–20, ACM, New York, NY, USA, June 2014.
[14] H. Imtiaz and A. D. Sarwate, "Symmetric matrix perturbation for differentially-private principal component analysis," in *Proceedings of the 2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 2339–2343, IEEE, Shanghai, China, March 2016.
[15] H. Imtiaz and A. D. Sarwate, "Differentially private distributed principal component analysis," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 2206–2210, IEEE, Calgary, AB, Canada, April 2018.
[16] W. Feng, Y. Zhao, and J. Deng, "Application of svm based on principal component analysis to credit risk assessment in commercial banks," in *Proceedings of the 2009 GCIS'09. WRI Global Congress on Intelligent Systems*, pp. 49–52, IEEE, Xiamen, China, May 2009.
[17] X. Jiang, Z. Ji, S. Wang, N. Mohammed, S. Cheng, and L. Ohno-Machado, "Differential-private data publishing through component analysis," *Transactions on Data Privacy*, vol. 6, no. 1, p. 19, 2013.
[18] C. Dwork, "Differential privacy," in *Automata, Languages and Programming*, pp. 1–12, Springer, Berlin, Germany, 2006.
[19] C. Dwork, "A firm foundation for private data analysis," *Communications of the ACM*, vol. 54, no. 1, pp. 86–95, 2011.
[20] T. Zhu, G. Li, W. Zhou, and P. S. Yu, "Differentially private data publishing and analysis: a survey," *IEEE Transactions on Knowledge and Data Engineering*, vol. 29, no. 8, pp. 1619–1638, 2017.
[21] http://archive.ics.uci.edu/ml/machine-learning-databases/secom/.
[22] https://archive.ics.uci.edu/ml/machine-learning-databases/covtype/.
[23] http://archive.ics.uci.edu/ml/machine-learning-databases/musk/.
[24] http://archive.ics.uci.edu/ml/machine-learning-databases/semeion/.
[25] http://archive.ics.uci.edu/ml/machine-learning-databases/waveform/.
[26] M. Hardt and E. Price, "The noisy power method: a meta algorithm with applications," in *Proceedings of the Advances in Neural Information Processing Systems*, pp. 2861–2869, Montreal, Canada, December 2014.