On the Privacy-preserving Generalized Eigenvalue Problem

Wei-Hong Chen a34123211@gmail.com
Yu-Feng Huang hideonbush123789.cs13@nycu.edu.tw
Chen-Yu Lee cyli.cs12@nycu.edu.tw
Hung-Yi Chen enderjacky96285.cs13@nycu.edu.tw
Shi-Chun Tsai sctsai@nycu.edu.tw

Editors: Hung-yi Lee and Tongliang Liu

Abstract

Generalized eigenvalues serve as a foundational tool for extracting insights from data and constructing robust statistical learning models, while differential privacy ensures the protection of individual information within these models by minimizing the impact of any single data point. In this work, we propose an (ϵ, δ)-differential privacy algorithm to solve the generalized eigenvalue problem (GEP). Our algorithm gives better classification accuracy over existing methods and has the nearly optimal ℓ_2-norm error bounds in both low and high dimensions. Furthermore, our algorithm guarantees convergence to the solution regardless of the initial vector and this improves a previous method that requires a specific procedure to find a proper starting vector. Our experiments confirm the effectiveness of our algorithm in safeguarding privacy while simultaneously boosting classification accuracy.

Keywords: Differential Privacy, Generalized eigenvalue, Computational Learning Theory

1. Introduction

Generalized eigenvalue problem (GEP) plays significant roles in various scientific disciplines, covering machine learning, statistics, and mathematics. For statistical learning models, sparse GEP is applied in multiple contexts, including principal component analysis (PCA), Fisher's discriminant analysis (FDA), sliced inverse regression (SIR), etc. The formal definition of GEP is given as follows.

Definition 1 (GEP Golub and Van Loan (1996)) *Let* $\mathbf{A}, \mathbf{B} \in \mathbb{R}^{d\times d}$. *The generalized eigenvalues of the symmetric-definite pair* $\{\mathbf{A}, \mathbf{B}\}$ *are denoted by* $\lambda(\mathbf{A}, \mathbf{B}) := \{\lambda | \det(\mathbf{A} - \lambda\mathbf{B}) = 0\}$. *If* $\lambda \in \lambda(\mathbf{A}, \mathbf{B})$ *and* $\mathbf{v}$ *is a nonzero vector that satisfies* $\mathbf{Av} = \lambda\mathbf{Bv}$, *then* $\mathbf{v}$ *is a generalized eigenvector.*

A fundamental approach to solving the GEP is to employ the generalized Rayleigh quotient Pattabhiraman (1974). Specifically, the GEP can be formulated to solve the following optimization problem:

$$\max_{\mathbf{v}\in\mathbb{R}^d} J(\mathbf{v}) = \frac{\mathbf{v}^\top \mathbf{A}\mathbf{v}}{\mathbf{v}^\top \mathbf{B}\mathbf{v}}. \tag{1}$$

Furthermore, when dealing with high-dimensional datasets, it is essential to identify a specific subset of significant features, namely sparse generalized eigenvectors. For instance, certain diseases may be associated with only a tiny fraction of genes in genetic data. However, a significant concern in a data-driven world is using personal information in machine

learning. This information can be inadvertently exposed through training, potentially revealing sensitive details about individuals. Therefore, protecting the privacy of personal data is crucial. Differential privacy (DP) Dwork et al. (2006) is a widely adopted paradigm in related research. Researchers are currently investigating how machine learning models change when differential privacy is incorporated, compared to standard approaches. This work covers theoretical underpinnings, model structures, and practical implementations. Our goal is to solve the (sparse) GEP under the constraint of differential privacy. Despite the availability of established solutions for the (sparse) generalized eigenvalue problem, research on incorporating differential privacy into GEP is largely unexplored, except for the works Hu et al. (2023); Xia et al. (2024). In contrast, the eigenvalue problem with differential privacy, such as DP-PCA, has been extensively studied, producing a wealth of research Chaudhuri et al. (2013); Hardt and Price (2014); Dwork et al. (2014b); Balcan et al. (2016); Jiang et al. (2016); Ge et al. (2018); Wang and Xu (2020); Liu et al. (2022). This disparity suggests a potential gap between these two problems in the context of DP, motivating us to investigate this problem.

We propose a new DP-GEP algorithm by subtly incorporating DP-SGD (differential-private stochastic gradient descent) Bassily et al. (2014); Abadi et al. (2016); Wang et al. (2017) with the Simultaneous Reduction method, which is a classical solver for GEP Martin and Wilkinson (1968); Golub and Van Loan (1996). The reason for using this approach is that the original GEP optimization function in (1) is non-convex. While, this method transforms it into solving two convex optimization functions and eliminates the need for an initial vector sufficiently close to the optimal vector. Moreover, when $\mathbf{B}$ is only positive semidefinite matrix, we incorporate a regularization term into the GEP framework Friedman (1989). The regularization term ensures that the matrices involved in GEP meet the positive definite requirement.

Hu et al. (2023) drew inspiration from the non-private method, *Truncated Rayleigh Flow* proposed by Tan et al. (2018). Their methods (i.e., *DP-Rayleigh Flow* and *DP-Truncated Rayleigh Flow*) involve adding Gaussian noise matrix to the input matrices and utilizing stochastic gradient descent for the optimization problem (1). Hu et al. considered various settings, including low and high dimensions, as well as deterministic and stochastic scenarios. A limitation of their proposed methodology is its susceptibility to local optima, necessitating the provision of an initial vector close to the global optimum. This arises from the non-convex nature of the objective function, which can hinder convergence to the global solution from arbitrary starting points (1). By Theorem 5 of Hu et al. (2023), it can only happen when n is sufficiently large. But this may contradict the high-dimensional assumption $d \gg n$. Our method offers a significant improvement over Hu et al.'s algorithms, as it not only preserves differential privacy but also maintains better error bounds in both low and high dimensions, even in the context of classification tasks. Similar results can be extended to stochastic settings. We summarize our contributions as follows:

- We propose a new (ϵ, δ)-differential privacy algorithm without the assumption on the specific initial vectors and achieve better classification accuracy. This addresses the issue mentioned in Hu et al. (2023) on requiring a near optimal initial vector in high dimensional case.

- Our algorithm achieves error estimation bounds of $\tilde{O}(\frac{d \log \frac{1}{\delta}}{n^2\epsilon^2})$ and $\tilde{O}(\frac{s \log d \log \frac{1}{\delta}}{n^2\epsilon^2})$ for low and high dimensions, which match the best known bounds.

- Our theoretical framework is established upon the regularized GEP approach, ensuring that the matrices involved in GEP satisfy the required positive definiteness.

2. Related Works

A common approach to achieving differential privacy is to add Laplace or Gaussian noise Chaudhuri et al. (2013); Hardt and Price (2014); Balcan et al. (2016). Some methods introduce noise from various distributions into the covariance matrix Dwork et al. (2014b); Jiang et al. (2016). Furthermore, one can employ Gaussian noise in distributed systems Ge et al. (2018), or add Gaussian noise directly to each sample Wang and Xu (2020). Liu et al. (2022) leveraged DP-SGD to address the PCA problem. The additional constraints inherent in DP-GEP prevent the straightforward adaptation of existing DP-PCA approaches. While, there had been some attempts to train deep neural networks by contaminating the gradients during backpropagation. Ghosh and Das Ghosh and Das (2024) applied this technique to make neural networks differentially private.

Hu et al. (2023) proposed two methods for DP-GEP that involve adding Gaussian noise to the input matrices and utilizing stochastic gradient descent for optimizing the objective function of problem (1). But, it requires for a suitable initial vector somehow hinders the applicability of their approach to higher dimensions. Similarly, Xia et al. (2024) proposed a method for DP-SIR. But their method also requires an initial vector sufficiently close to the optimal solution.

3. Preliminaries and Notations

For a vector $\mathbf{v} \in \mathbb{R}^d$, $\|\mathbf{v}\|_2$ denotes as the ℓ_2-norm of the vector $\mathbf{v}$, and $\|\mathbf{v}\|_0$ counts the number of non-zero elements of $\mathbf{v}$. Given a matrix $\mathbf{A}$, let $\lambda_i(\mathbf{A}), \lambda_{\max}(\mathbf{A})$ and $\lambda_{\min}(\mathbf{A})$ denote the i-th, maximum and minimum eigenvalue of $\mathbf{A}$, respectively. The ℓ_2-norm of the matrix $\mathbf{A}$ is defined as $\|\mathbf{A}\|_2 = \max_{\mathbf{v}\in\mathbb{R}^d, \|\mathbf{v}\|_2=1} \|\mathbf{A}\mathbf{v}\|_2$. The Frobenius norm is defined as $\|\mathbf{A}\|_F = \sqrt{\sum_{ij} a_{ij}^2}$. Let $\sigma_i(\mathbf{A}), \sigma_{\max}(\mathbf{A})$ and $\sigma_{\min}(\mathbf{A})$ denote the i-th, maximum and minimum singular value of $\mathbf{A}$, respectively. $\tilde{O}$ notation is used to represent the omission of logarithmic terms and parameters.

Herein, we provide a brief overview of the properties of the fundamental matrix Golub and Van Loan (1996). For a matrix $\mathbf{A} \in \mathbb{R}^{d\times d}$, it is well known that $\|\mathbf{A}\|_2 = \sigma_{\max}(\mathbf{A})$ and $\|\mathbf{A}\|_2 \le \sqrt{d} \cdot \|\mathbf{A}\|_F$. For a symmetric matrix $\mathbf{A} \in \mathbb{R}^{d\times d}$, $\|\mathbf{A}\|_2 = \sigma_{\max}(\mathbf{A}) = \lambda_{\max}(\mathbf{A})$.

Differential privacy (DP) (Dwork et al. (2006)) is a rigorous privacy-preserving technique with well-defined conditions. In essence, DP ensures that the output of an algorithm remains approximately the same, even if a single record in the input dataset is different. We adopt the following definitions and results of DP mostly from Dwork et al. (2014a).

Definition 2 (Differential Privacy) *For any two neighboring datasets $\mathcal{D}, \mathcal{D}' \subseteq \mathcal{X}$, which differ only one sample. A randomized algorithm $\mathcal{M} : \mathcal{X} \to \mathcal{R}$ is (ϵ, δ)-differential privacy if*

for all output distribution $\mathcal{O} \subseteq \mathcal{R}$,

$$\Pr[\mathcal{M}(\mathcal{D}) \in \mathcal{O}] \leq \exp(\epsilon) \cdot \Pr[\mathcal{M}(\mathcal{D}') \in \mathcal{O}] + \delta.$$

Lemma 3 (Post-Processing) *Let* $\mathcal{M} : \mathcal{X} \to \mathcal{R}$ *be a randomized algorithm with* (ϵ, δ)*-differential privacy. Let* $f : \mathcal{R} \to \mathcal{R}'$ *be an arbitrary randomized function. Then* $f \circ \mathcal{M}$ *is* (ϵ, δ)*-differentially private.*

Since we apply the Gaussian mechanism Dwork et al. (2014a) iteratively, the composition theorem and advanced composition theorem are essential for the overall privacy protection.

Lemma 4 (Composition) *Let* $\mathcal{M}_i : \mathcal{X} \to \mathcal{R}_i$ *be an* (ϵ_i, δ_i)*-differential privacy algorithm for* $i \in [k]$*. Then if* $\mathcal{M} : \mathcal{X} \to (\mathcal{R}_1, ..., \mathcal{R}_k)$ *is defined to be* $\mathcal{M} = (\mathcal{M}_1(\mathcal{X}), ..., \mathcal{M}_k(\mathcal{X}))$*, then* $\mathcal{M}$ *is* $(\sum_{i=1}^{k} \epsilon_i, \sum_{i=1}^{k} \delta_i)$*-differential privacy.*

Lemma 5 (Advanced Composition) *For all* $0 < \epsilon, \delta, \delta' < 1$*, the class of* (ϵ, δ)*-differential privacy mechanisms satisfies* $(\epsilon', k\delta + \delta')$*-differential privacy under* k*-fold adaptive composition for* $\epsilon' = \sqrt{2k \ln(1/\delta')}\epsilon + k\epsilon(e^{\epsilon} - 1)$.

Definition 6 (Sensitivity) *The* ℓ_2*-sensitivity of a function* $f : \mathcal{X} \to \mathbb{R}^k$ *is* $\Delta_2(f) = \max_{\mathcal{D}, \mathcal{D}' \subseteq \mathcal{X}} \|f(\mathcal{D}) - f(\mathcal{D}')\|_2$*, where* $\mathcal{D}$ *and* $\mathcal{D}'$ *are neighboring datasets.*

Lemma 7 (Gaussian Mechanism) *Given any function* $f : \mathcal{X} \to \mathbb{R}^k$*, the Gaussian mechanism is defined as* $f(\mathcal{D}) + \boldsymbol{\zeta}$ *where* $\boldsymbol{\zeta} \sim \mathcal{N}(0, \sigma^2 \mathbf{I}_k)$*. If* $\sigma = c\Delta_2(f)/\epsilon$ *where* $c \geq \sqrt{2 \log 1.25/\delta}$ *and* $\Delta_2(f)$ *is the* ℓ_2*-sensitivity of* f*, then the Gaussian mechanism is* (ϵ, δ)*-differential privacy.*

Definition 8 (Zero-Concentrated DP Bun and Steinke (2016)) *For any two neighboring datasets* $\mathcal{D}$ *and* $\mathcal{D}'$*, which differ only in one sample. A randomized algorithm* M *is* ρ*-zero-concentrated DP (zCDP) if for all* $\alpha > 1$*, we have* $D_\alpha(\mathcal{M}(\mathcal{D}) || \mathcal{M}(\mathcal{D}')) \leq \alpha\rho$*, where* $D_\alpha(\mathcal{M}(\mathcal{D}) || \mathcal{M}(\mathcal{D}'))$ *is* α*-Rènyi divergence.*

Lemma 9 *(Bun and Steinke (2016)) For every* $\epsilon > 0$*, if algorithm* $\mathcal{M}$ *is* ϵ*-DP then it will be* $\frac{\epsilon^2}{2}$*-zCDP. If* $\mathcal{M}$ *is* ρ*-zCDP, then it will be* (ϵ, δ)*-DP with* $\epsilon = \rho + 2\sqrt{\rho \log \frac{1}{\delta}}$.

By Lemma 9, if algorithm $\mathcal{M}$ is ρ-zCDP, then it is (ϵ, δ)-DP, where $\rho = (\sqrt{\epsilon + \log \frac{1}{\delta}} - \sqrt{\log \frac{1}{\delta}})^2 \approx \frac{\epsilon^2}{4 \log \frac{1}{\delta}}$. Hence, we can transform all results of ρ-zCDP by replacing ρ with $\frac{\epsilon^2}{4 \log \frac{1}{\delta}}$ when $\log \frac{1}{\delta} \gg \epsilon$. Problem (1) can be reformulated as follows:

$$\mathbf{v} = \arg\max_{\mathbf{v} \in \mathbb{R}^d} \mathbf{v}^\top \mathbf{A} \mathbf{v} \quad \text{subject to} \quad \mathbf{v}^\top \mathbf{B} \mathbf{v} = 1 \tag{2}$$

We apply our DP-GEP algorithm for dimension reduction, specifically for principal component analysis (PCA) and Fisher's discriminant analysis (FDA). PCA is a subproblem of GEP, while the solution to FDA is the GEP.

Principal Component Analysis (PCA). Given n samples dataset $\mathbf{X} \in \mathbb{R}^{d\times n}$ and each sample $\mathbf{x}_i \in \mathbb{R}^d$, PCA aims to identify a projection space that maximizes the covariance matrix of projection data. PCA can be formulated as Problem (2) with

$$\mathbf{A} = \frac{1}{n}\sum_{i=1}^{n}(\mathbf{x}_i - \boldsymbol{\mu}_x)(\mathbf{x}_i - \boldsymbol{\mu}_x)^\top, \quad \boldsymbol{\mu}_x = \frac{1}{n}\sum_{i=1}^{n}\mathbf{x}_i$$

and $\mathbf{B}$ as the identity matrix.

Fisher's Discriminant Analysis (FDA). Given n samples dataset $\mathbf{X} \in \mathbb{R}^{d\times n}$ with K different classes, FDA aims to identify a projection space that maximizes the between-class covariance matrix and minimizes the within-class covariance matrix of projection data. We denote $\boldsymbol{\mu}_x = \frac{1}{n}\sum_{i=1}^{n}\mathbf{x}_i$ and $\boldsymbol{\mu}_k = \frac{1}{n_k}\sum_{i\in\mathcal{C}_k}\mathbf{x}_i$, where $\mathcal{C}_k$ is the index set of the k-th class with n_k samples. Then, FDA can be formulated as Problem (2) with

$$\mathbf{A} = \frac{1}{n}\sum_{k=1}^{K} n_k(\boldsymbol{\mu}_k - \boldsymbol{\mu}_x)(\boldsymbol{\mu}_k - \boldsymbol{\mu}_x)^\top,$$

$$\mathbf{B} = \frac{1}{n}\sum_{k=1}^{K}\sum_{i\in\mathcal{C}_k}(\mathbf{x}_i - \boldsymbol{\mu}_k)(\mathbf{x}_i - \boldsymbol{\mu}_k)^\top.$$

We aim to extract the top k generalized eigenvectors to project sample data onto a lower-dimensional space, enabling efficient classification. On the other hand, we explore four cases of GEP from Problem (2). The first is the low-dimensional case $n \gg d$, representing the basic scenario. The second case is the high-dimensional sparse case $d \gg n$ and we assume that the generalized eigenvector $\mathbf{v}$ will be s-sparse vector for some $s \ll d$, that is, $\|\mathbf{v}\|_0 \leq s$. We can formulate it with the following form:

$$\mathbf{v}_s^* = \arg\max_{\mathbf{v}\in\mathbb{R}^d} \mathbf{v}^\top\mathbf{A}\mathbf{v} \text{ subject to } \mathbf{v}^\top\mathbf{B}\mathbf{v} = 1 \text{ and } \|\mathbf{v}\|_0 \leq s.$$

We can extend our analysis to the stochastic setting, where the sample data is assumed to be drawn from an unknown distribution $\mathcal{N}$. In this case, the problem can be formulated with the following forms:

$$\mathbf{v}^* = \arg\max_{\mathbf{v}\in\mathbb{R}^d} \mathbf{v}^\top\bar{\mathbf{A}}\mathbf{v} \quad \text{subject to} \quad \mathbf{v}^\top\bar{\mathbf{B}}\mathbf{v} = 1, \text{ and}$$

$$\mathbf{v}_s^* = \arg\max_{\mathbf{v}\in\mathbb{R}^d} \mathbf{v}^\top\bar{\mathbf{A}}\mathbf{v} \text{ subject to } \mathbf{v}^\top\bar{\mathbf{B}}\mathbf{v} = 1 \text{ and } \|\mathbf{v}\|_0 \leq s,$$

where $\bar{\mathbf{A}} = \mathbb{E}[\mathbf{A}]$ and $\bar{\mathbf{B}} = \mathbb{E}[\mathbf{B}]$. But, as shown in Hu et al. (2023), in stochastic settings, the ℓ_2-norm of error caused by $\bar{\mathbf{A}} - \mathbf{A}$ or $\bar{\mathbf{B}} - \mathbf{B}$ is $O(\sqrt{\frac{s\log d}{n}})$, which is asymptotically larger than those caused by Gaussian noise $O(\sqrt{\frac{s\log d}{n^2\rho}})$. The final error estimation bounds are similar when Gaussian noise is added in both settings. Given this, we focus our analysis on deterministic settings to investigate the impact of Gaussian noise. We formally state DP-GEP as follows:

Algorithm 1 Simultaneous Reduction Method Martin and Wilkinson (1968); Golub and Van Loan (1996)

Input: Symmetric-definite pair $\{\mathbf{A}, \mathbf{B}\}$
Output: Generalized eigenvectors and eigenvalues.

1: Let $\mathbf{\Phi_B}$ and $\mathbf{\Lambda_B}$ be the eigenvectors and eigenvalues of the matrix $\mathbf{B}$.
2: Denote $\tilde{\mathbf{\Phi}}_\mathbf{B} = \mathbf{\Phi_B}\mathbf{\Lambda_B}^{-\frac{1}{2}}$.
3: Denote $\tilde{\mathbf{A}} = \tilde{\mathbf{\Phi}}_\mathbf{B}^\top \mathbf{A} \tilde{\mathbf{\Phi}}_\mathbf{B}$.
4: Let $\mathbf{\Phi_A}$ and $\mathbf{\Lambda_A}$ be the eigenvectors and eigenvalues of the matrix $\tilde{\mathbf{A}}$.
5: **return** $\tilde{\mathbf{\Phi}}_\mathbf{B}\mathbf{\Phi_A}$ and $\mathbf{\Lambda_A}$.

Definition 10 (DP-GEP) *Given n samples $\mathbf{X} \in \mathbb{R}^{d\times n}$ and each sample $\mathbf{x}_i \in \mathbb{R}^d$ with $\|\mathbf{x}_i\|_2 \leq 1$, the goal of DP-GEP is to privately find generalized eigenvectors based on an (ϵ, δ)-DP algorithm, where matrices $\mathbf{A}$ and $\mathbf{B}$ correspond to the sample data.*

With DP, we aim to achieve two goals: (i) to maximize the cosine similarity between output vectors and optimal vectors, and (ii) to achieve high classification accuracy when applying GEP to dimension reduction methods.

4. Method

We first revisit the well-established gradient descent method for solving Problem (1), which can be solved by minimizing the objective function $-J(\mathbf{v})$. Specifically, in the t-th iteration, vector $\mathbf{v}_t$ is updated with

$$\mathbf{v}_t = \mathbf{v}_{t-1} - \eta \nabla_\mathbf{v}(-J(\mathbf{v}_{t-1})). \tag{3}$$

Intuitively, we can use DP-SGD by adding Gaussian noise to the gradient, as follows:

$$\mathbf{v}_t = \mathbf{v}_{t-1} - \eta(\nabla_\mathbf{v}(-J(\mathbf{v}_{t-1})) + \boldsymbol{\zeta}_t), \tag{4}$$

where each entry of $\boldsymbol{\zeta}_t$ is i.i.d. randomly sampled from $\mathcal{N}(0, \sigma^2)$. But we need to ensure that the sensitivity of the gradient function of Problem (1) is bounded by $O(\frac{1}{n})$. However, as shown in Hu et al. (2023), the sensitivity is not bounded by $O(\frac{1}{n})$. Hence, we explore the Simultaneous Reduction method Martin and Wilkinson (1968); Golub and Van Loan (1996) for addressing DP-GEP. Algorithm 1 outlines the method, which has been shown to outperform direct derivations of generalized eigenvectors Swets and Weng (1996). On the other hand, Step 2 of Algorithm 1 requires the existence of the inverse of $\mathbf{\Lambda_B}$. To address this, we introduce a regularization term to matrix $\mathbf{\Lambda_B}$ to ensure that its inverse always exists Friedman (1989). We consider this a reasonable adjustment, as such situations can arise in practical applications. For instance, in Fisher's discriminant analysis (FDA), matrix $\mathbf{B}$ is a within-class covariance matrix, which can only be guaranteed to be positive semidefinite.

Our approach introduces Gaussian noise in Steps 1 and 4 of Algorithm 1 to achieve the effect of DP. In our first attempt, we also tried adding noise directly to matrices $\mathbf{A}$ and $\mathbf{B}$. But, the resulting classification accuracy is poor. Therefore, we considered using DP-SGD in these two steps. The objective functions of these two steps can be characterized

Algorithm 2 DPSR

Input: Symmetric-definite pair $\{\mathbf{A}, \mathbf{B}\}$, privacy parameters ϵ, δ, step size η_A, η_B, iteration number m, initial vectors $\mathbf{V}_0, \mathbf{V}'_0$ with column unit vectors.
Output: Generalized eigenvectors.

1: **for** $t = 1, ..., m$ **do**
2: $\mathbf{V}_t = \mathbf{V}_{t-1} - \eta_B(\nabla J_1(\mathbf{V}_{t-1}) + \mathbf{Z}_t)$, where $\mathbf{Z}_t \overset{i.i.d.}{\sim} \mathcal{N}(0, \sigma_1^2)$ with $\sigma_1 = \frac{2C_1\sqrt{m\log(1.25m/\delta)}}{n\epsilon}$.
3: $\mathbf{V}_t = \text{Orthonormalize}(\mathbf{V}_t)$.
4: **end for**
5: Denote $\mathbf{\Phi_B} = \mathbf{V}_m$ and $\mathbf{\Lambda_B} = \mathbf{diag}(\mathbf{V}_m^\top \mathbf{B}\mathbf{V}_m)$.
6: Denote $\tilde{\mathbf{\Phi}}_\mathbf{B} = \mathbf{\Phi_B}(\mathbf{\Lambda_B} + \xi\mathbf{I})^{-1/2}$.
7: Denote $\tilde{\mathbf{A}} = \tilde{\mathbf{\Phi}}_\mathbf{B}^\top \mathbf{A} \tilde{\mathbf{\Phi}}_\mathbf{B}$.
8: **for** $t = 1, ..., m$ **do**
9: $\mathbf{V}'_t = \mathbf{V}'_{t-1} - \eta_A(\nabla J_2(\mathbf{V}'_{t-1}) + \mathbf{Z}_t)$, where $\mathbf{Z}_t \overset{i.i.d.}{\sim} \mathcal{N}(0, \sigma_2^2)$ with $\sigma_2 = \frac{2C_2\sqrt{m\log(1.25/\delta)}}{n\epsilon\xi}$.
10: $\mathbf{V}'_t = \text{Orthonormalize}(\mathbf{V}'_t)$.
11: **end for**
12: **return** $\tilde{\mathbf{\Phi}}_\mathbf{B}\mathbf{V}'_m$

as follows:

$$\min J_1(\mathbf{v}) = -\mathbf{v}^\top \mathbf{B}\mathbf{v} \quad \text{subject to} \quad \mathbf{v}^\top\mathbf{v} = 1, \tag{5}$$

$$\min J_2(\mathbf{v}) = -\mathbf{v}^\top \tilde{\mathbf{A}}\mathbf{v} \quad \text{subject to} \quad \mathbf{v}^\top\mathbf{v} = 1. \tag{6}$$

Algorithm 2, denoted as DPSR, shows our method. Steps 1-4 use the DP-SGD method to solve for the eigenvectors and eigenvalues of matrix $\mathbf{B}$. Step 3 ensures that each column vector is orthonormal in each iteration, which can be done with the well known Gram-Schmidt method or Householder reflection Householder (1958). Similarly, Steps 8-11 employ the DP-SGD method to solve for the eigenvectors of the matrix $\tilde{\mathbf{A}}$. In Step 5, the eigenvalues of matrix $\mathbf{B}$ are calculated with the eigenvectors of $\mathbf{B}$. In Step 6, a regularization term is added to matrix $\mathbf{\Lambda_B}$ before performing sign replacement. The remaining steps proceed as outlined in Algorithm 1.

Theorems 11 and 12 establish the (ϵ, δ)-differential privacy of Algorithm 2. Detailed proofs are provided in the Supplementary Material.

Theorem 11 *The sensitivities of ∇J_1 and ∇J_2 are bounded by $\frac{C_1}{n}$ and $\frac{C_2}{n\xi}$, respectively, where C_1, C_2 are constants, n is the sample size, and $\xi > 0$ is the regularization ratio. Moreover, $\nabla J_1 = 2\mathbf{B}\mathbf{v}$, $\nabla J_2 = 2\tilde{\mathbf{A}}\mathbf{v}$ and $\mathbf{v}$ is a unit vector.*

Bounded sensitivities of matrices $\tilde{\mathbf{A}}$ and $\mathbf{B}$ imply bounded gradient sensitivities. The proof for $\tilde{\mathbf{A}}$ yields a stronger result due to including a regularization ratio ξ; otherwise, it would be unbounded when $\mathbf{B}$ is not positive definite. Based on Theorem 11 and Lemma 7, the Gaussian mechanism can be employed iteratively within the gradient descent process.

Algorithm 3 DPSRS

Input: Symmetric-definite pair $\{\mathbf{A}, \mathbf{B}\}$, privacy parameters ϵ, δ, step size η_A, η_B, iteration number m, initial vectors $\mathbf{V}_0, \mathbf{V}_0'$ with column unit s-sparse vectors, and sparsity s.
Output: Generalized eigenvectors.

for $t = 1, ..., m$ **do**
  $\mathbf{V}_t = \mathbf{V}_{t-1} - \eta_B(\nabla J_1(\mathbf{V}_{t-1}) + \mathbf{Z}_t)$, where $\mathbf{Z}_t \overset{i.i.d.}{\sim} \mathcal{N}(0, \sigma_1^2)$ with $\sigma_1 = \frac{2C_1\sqrt{m\log(1.25m/\delta)}}{n\epsilon}$.
  $\mathbf{V}_t = \text{Orthonormalize}(\mathbf{V}_t)$.
  $\mathbf{V}_t = \text{Truncated}(\mathbf{V}_t, s)$
  $\mathbf{V}_t = \text{Orthonormalize}(\mathbf{V}_t)$.
end for
Denote $\mathbf{\Phi_B} = \mathbf{V}_m$ and $\mathbf{\Lambda_B} = \mathbf{diag}(\mathbf{V}_m^\top \mathbf{B} \mathbf{V}_m)$.
Denote $\tilde{\mathbf{\Phi}}_\mathbf{B} = \mathbf{\Phi_B}(\mathbf{\Lambda_B} + \xi \mathbf{I})^{-1/2}$.
Denote $\tilde{\mathbf{A}} = \tilde{\mathbf{\Phi}}_\mathbf{B}^\top \mathbf{A} \tilde{\mathbf{\Phi}}_\mathbf{B}$.
for $t = 1, ..., m$ **do**
  $\mathbf{V}_t' = \mathbf{V}_{t-1}' - \eta_A(\nabla J_2(\mathbf{V}_{t-1}') + \mathbf{Z}_t)$, where $\mathbf{Z}_t \overset{i.i.d.}{\sim} \mathcal{N}(0, \sigma_2^2)$ with $\sigma_2 = \frac{2C_2\sqrt{m\log(1.25/\delta)}}{n\epsilon\xi}$.
  $\mathbf{V}_t' = \text{Orthonormalize}(\mathbf{V}_t')$.
  $\mathbf{V}_t' = \text{Truncated}(\mathbf{V}_t', s)$
  $\mathbf{V}_t' = \text{Orthonormalize}(\mathbf{V}_t')$.
end for
return $\tilde{\mathbf{\Phi}}_\mathbf{B} \mathbf{V}_m'$

Theorem 12 *Let* $\sigma_1 = \frac{2C_1 c\sqrt{m}}{n\epsilon}$ *and* $\sigma_2 = \frac{2C_2 c\sqrt{m}}{n\epsilon\xi}$, *where* $c = \sqrt{2\log(1.25m/\delta)}$ *and* m *is the iteration number. Then, with Theorem 1, Algorithm 2 satisfies* (ϵ, δ)*-differential privacy.*

By the Gaussian mechanism, if we set $\sigma_1 = \frac{2C_1 c\sqrt{m}}{n\epsilon}$ and $\sigma_2 = \frac{2C_2 c\sqrt{m}}{n\epsilon\xi}$, then after each iteration in gradient descent method it preserves $(\frac{\epsilon}{2\sqrt{m}}, \frac{\delta}{2\sqrt{m}})$-DP. According to the advanced composition theorem, the final result of the gradient method is $(\frac{\epsilon}{2}\cdot\frac{\delta}{2})$-DP. Since we use gradient descent methods to train the eigenvectors and eigenvalues of $\mathbf{B}$ and $\tilde{\mathbf{A}}$, respectively, the composition theorem ensures that the final result of Algorithm 2 is (ϵ, δ)-DP.

We extend Algorithm 2 to high-dimensional scenarios. The primary strategy involves applying a truncation operation after each iteration in gradient descent methods to ensure that the resulting vector in each iteration is an s-sparse vector. Specifically, we keep the indices of the top s absolute values for each vector and set the values in other indices to 0. This method is based on the work by Tan et al. (2018), which is invoked in Algorithm 3 in the form $Truncated(\mathbf{V}, s)$. We ensure that the vectors remain orthonormal after each truncated operation by calling $Orthonormalize(\mathbf{V})$. Algorithm 3 shows the detailed steps. Theorems 13 establishes its (ϵ, δ)-differential privacy.

Theorem 13 *With Theorem 11 and Theorem 12, Algorithm 3 satisfies* (ϵ, δ)*-differential privacy.*

Although we performed additional truncated operations and orthonormalization in Algorithm 3, the post-processing property of differential privacy ensures that these operations do not compromise the DP guarantee. Therefore, the result of Algorithm 3 remains (ϵ, δ)-DP.

5. Error Estimation

We analyze the error bound between the optimal generalized eigenvectors and the output of our algorithms. First, we estimate the error caused by the output eigenvectors of matrices $\hat{\mathbf{B}} = \mathbf{B} + \xi\mathbf{I}$ and $\tilde{\mathbf{A}}$ trained by our algorithm, respectively. Finally, we combine both estimates to demonstrate the overall error of the generalized eigenvectors. Since eigenvalue information is not directly accessible in our implementation, we rely on grid search to determine the optimal step size. The step size settings in the following theorems are for theoretical analysis, inspired by the approach in Hu et al. (2023).

Theorem 14 *With Algorithm 2, if we set the step size* $\eta_B = \frac{2}{\lambda_{\max}(\hat{\mathbf{B}})+\lambda_{\min}(\hat{\mathbf{B}})}$, *then with probability at least* $1-\beta$,

$$\|\mathbf{v}_* - \mathbf{v}_m\|_2 \leq O(\frac{\sqrt{d\log(\frac{d}{\beta})\log(\frac{1}{\delta})}}{n\epsilon\xi}), \tag{7}$$

where $\mathbf{v}_*$ *is the optimal eigenvector of* $\hat{\mathbf{B}}$ *and* $\mathbf{v}_m$ *is the vector after Step 4.*

Theorem 15 *With Algorithm 2 and if we set the step size* $\eta_A = \frac{2}{\lambda_{\max}(\tilde{\mathbf{A}})+\lambda_{\min}(\tilde{\mathbf{A}})}$, *then with probability at least* $1-\beta$,

$$\|\mathbf{v}'_* - \mathbf{v}'_m\|_2 \leq O(\frac{\sqrt{d\log(\frac{d}{\beta})\log(\frac{1}{\delta})}}{n\epsilon\xi}), \tag{8}$$

where $\mathbf{v}'_*$ *is the optimal eigenvector of* $\tilde{\mathbf{A}}$ *and* $\mathbf{v}'_m$ *is the vector after Step 11.*

Theorem 16 *With Algorithm 2, Theorem 14 and 15, we have,*

$$1 - \langle\boldsymbol{\phi}^*, \boldsymbol{\phi}\rangle \leq O(\frac{d\log(\frac{d}{\beta})\log(\frac{1}{\delta})}{n^2\epsilon^2\xi^2}), \tag{9}$$

where $\boldsymbol{\phi}^*$ *is the optimal generalized eigenvector of symmetric-definite pair* $\{\mathbf{A}, \mathbf{B}\}$ *and* $\boldsymbol{\phi}$ *is the output vector.*

To compare with the results $\tilde{O}(\frac{d}{n^2\rho})$ of Hu et al. (2023) by ρ-zCDP Bun and Steinke (2016), we can set the privacy parameter $\rho = \frac{\epsilon^2}{\log(1/\delta)}$ and then transform the result $\tilde{O}(\frac{d}{n^2\rho})$ into $\tilde{O}(\frac{d\log\frac{1}{\delta}}{n^2\epsilon^2})$, which matches the result of Theorem 16.

For the error estimation bound of Algorithm 3, since we perform truncated operations, we need to separately analyze their impact on Gaussian noise and the gradient vector. Intuitively, we can assume that among the truncated gradient vectors, only s dimensions are significantly affected by Gaussian noise. Additionally, we have derived that the error caused by the truncated gradient vector is bounded by the error caused by Gaussian noise. Therefore, similar to Theorems 14, 15, and 16, we can derive the error bound of Algorithm 3, as stated in Theorem 17. For detailed derivations, we refer to the Supplementary Material.

Theorem 17 *With Algorithm 3, with probability at least* $1-\beta$*, the output generalized eigenvector* $\boldsymbol{\phi}_s$ *satisfies*

$$1-\langle \boldsymbol{\phi}_s^*, \boldsymbol{\phi}_s \rangle \leq \tilde{O}(\frac{s \log d \log (1/\delta)}{n^2 \epsilon^2}), \tag{10}$$

where $\boldsymbol{\phi}_s^*$ *is the optimal* s*-sparse generalized eigenvector of symmetric-definite pair* $\{\mathbf{A}, \mathbf{B}\}$ *and* s *is the sparsity of eigenvectors.*

To compare with the result $\tilde{O}(\frac{s \log d}{n^2 \rho})$ of Hu et al. (2023) by ρ-zCDP, we can set the privacy parameter $\rho = \frac{\epsilon^2}{\log (1/\delta)}$ and then transform the result $\tilde{O}(\frac{s \log d}{n^2 \rho})$ into $\tilde{O}(\frac{s \log d \log \frac{1}{\delta}}{n^2 \epsilon^2})$, which matches the result of Theorem 17. Furthermore, Theorem 17 is more applicable to the high-dimensional case since Hu et al. (2023) needs n to be sufficiently large, which may contradict the assumption of high-dimensional settings $d \gg n$. Without this restriction, our method can be applied to a broader range of cases.

Table 1: datasets used in experiments

Dataset	MNIST	a9a	Fashion	CIFAR10	Dota2	IoT22
Samples	60000	32561	60000	50000	102944	123117
Features	784	123	784	3072	116	83
Classes	10	2	10	10	2	12

Table 2: Comparison table between DPSR (ours) and DPRF with $\epsilon = 1$.

Method	Metrics	MNIST	a9a	Fashion
		SVM / RBF / RF	SVM / RBF / RF	SVM / RBF / RF
DPRF	Precision	50 / 65 / 63	62 / 78 / 78	60 / 70 / 70
DPSR (ours)		**85** / **90** / **90**	**81** / **82** / **82**	**75** / **77** / **81**
DPRF	Recalls	52 / 65 / 63	77 / 80 / 80	60 / 70 / 70
DPSR (ours)		**85** / **80** / **90**	**82** / **83** / **83**	**75** / **78** / **81**
DPRF	F1-score	50 / 65 / 64	67 / 75 / 78	59 / 69 / 70
DPSR (ours)		**84** / **90** / **90**	**81** / **82** / **82**	**74** / **77** / **81**

Method	Metrics	CIFAR10	Dota2	IoT22
		SVM / RBF / RF	SVM / RBF / RF	SVM / RBF / RF
DPRF	Precision	5 / 26 / 24	29 / 53 / 52	89 / 94 / 98
DPSR (ours)		**15** / **27** / **26**	**39** / **54** / **53**	**93** / **95** / **99**
DPRF	Recalls	12 / 27 / 25	53 / 53 / 52	91 / 94 / 98
DPSR (ours)		**19** / **28** / **27**	**53** / **54** / **53**	**93** / **95** / **99**
DPRF	F1-score	4 / 25 / 24	37 / 45 / 51	89 / 94 / 98
DPSR (ours)		**13** / **26** / **26**	**38** / **46** / **52**	**92** / **95** / **99**

6. Experiments

6.1. Setups

With Algorithm 1, we denote SR+DP as the method by adding Gaussian noise to matrices $\mathbf{A}$ and $\mathbf{B}$, respectively. We evaluate the performance of the following three approaches:

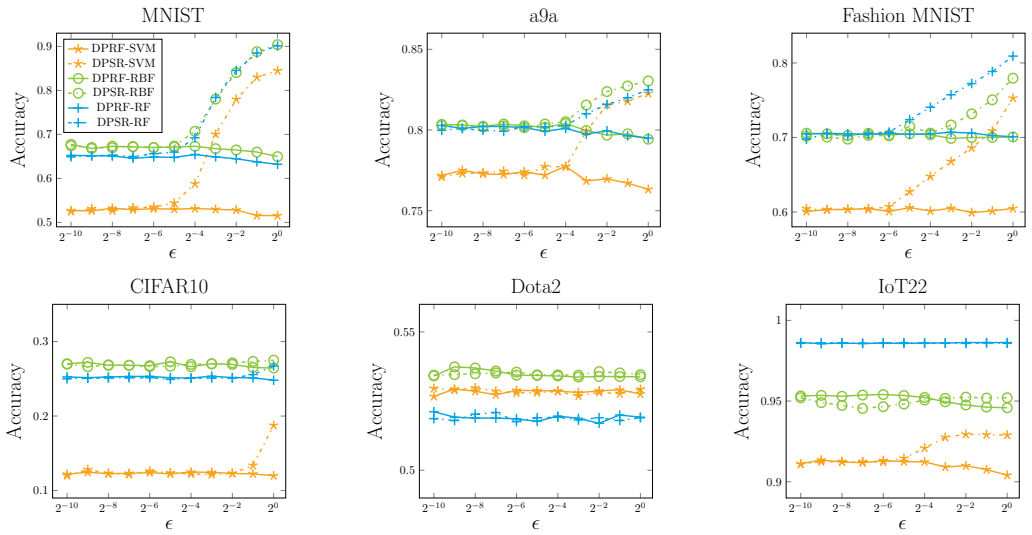

Figure 1: Compare performance between DPSR (ours) and DPRF with $\eta = 1$.

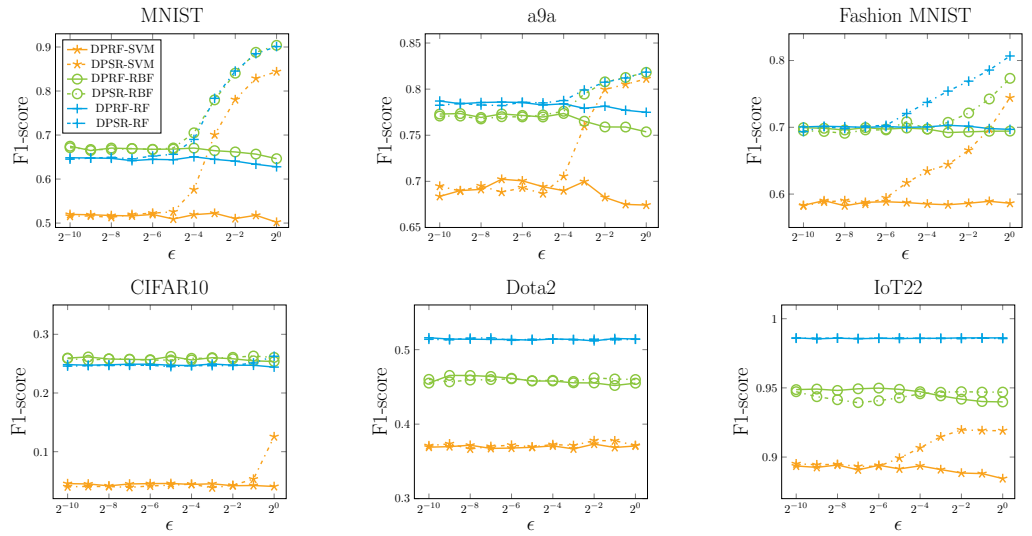

Figure 2: Compare performance between DPSR (ours) and DPRF with $\eta = 1$.

SR+DP, DPSR (Algorithm 2) and DPRF (DP-Rayleigh Flow Hu et al. (2023)) with six datasets. We conduct experiments specifically with FDA, and reduce the dimension to 10.

We normalize each row of data to 1 (i.e. $\|\mathbf{x}_i\|_2 \leq 1$) to ensure the sensitivities of the gradients are bounded by $O(\frac{1}{n})$. For DPRF, we use optimal generalized eigenvectors as initial vectors and then run DPRF to train the generalized eigenvectors. After training the generalized eigenvectors, we project the original data to the low-dimensional space using the obtained generalized eigenvectors of the above three approaches, respectively. Then, we use the *Support Vector Machine* Cortes and Vapnik (1995), including linear SVM, RBF SVM, and *Random Forest* Ho (1995) (with 100 trees) as discriminant classifiers. The evaluation metrics are classification accuracy, F1-score, precision, and recall. In addition, all training

processes use the same parameters, including privacy parameters, step size, sample ratio, iteration number, etc. We set the step size as $\{2^{-2}, 2^{-1}, 1\}$, $\epsilon = \{1, 2^{-1}, \cdots, 2^{-10}\}$, $\delta \approx \frac{1}{n^{1.1}}$, regularization ratio $\xi = 0.01$, iteration number as 15. We do not use a larger iteration number because increasing the iteration number would result in a larger Gaussian noise. Additionally, we conducted a grid search for a suitable regularization ratio. We used six widely adopted datasets in the literature: (1) MNIST LeCun et al. (1998); (2) a9a Chang and Lin (2011); (3) Fashion MNIST Xiao et al. (2017); (4) CIFAR-10 Krizhevsky et al. (2009); and two datasets from the UCI Machine Learning Repository Asuncion and Newman (2007): (5) Dota2; (6) RT-IoT2022. The RT-IoT2022 dataset includes categorical features. We convert categories to numerical values. Table 1 summarizes the dataset information.

6.2. Comparisons

To evaluate DPRF Hu et al. (2023), the optimal generalized eigenvectors were applied directly as the initial vectors. For DPSR, random vectors were used as the initial vectors. The comparisons are shown in Figure 1, 2 and Table 2, 3. The results demonstrate that DPSR consistently outperforms DPRF Hu et al. (2023), even when initialized with random vectors. The better performance of our algorithm demonstrates its effectiveness in addressing this classification task. For convenience, we denote the application of DPSR with SVM, RBF SVM, and random forest as DPSR-SVM, DPSR-RBF, and DPSR-RF, respectively. Similarly, the application of DPRF is denoted as DPRF-SVM, DPRF-RBF, and DPRF-RF, respectively.

We compare the performance of the two algorithms using three metrics, as shown in Table 2. It shows that our algorithm performs better than DPRF. The experiments show that DPSR (ours) outperforms DPRF across all three datasets (MNIST, a9a, and Fashion MNIST), regardless of whether linear SVM, RBF SVM, or random forest is employed as the classifier in Figure 1. For the datasets CIFAR-10 and Dota2, the inherent accuracy is already low without differential privacy, leading to similar performance between DPSR and DPRF after incorporating DP. Interestingly, for the dataset RT-IoT2022, we observed that DPSR surpasses the performance of the Simultaneous Reduction Method (SR), as can be observed in the Supplementary Materials. Additionally, the comparable performance of DPRF and DPSR (ours) across the RT-IoT2022 dataset is less significant, which may be attributed to the inherent classification effectiveness of the classifiers themselves.

To ensure the reliability of our algorithm (DPSR), we conducted Wilcoxon signed-rank tests on the experimental results of DPRF and DPSR, as shown in Table 3. The results clearly demonstrate that DPSR (ours) outperforms DPRF, except for the Dota2 dataset. However, as mentioned above, this can be attributed to the inherent difficulty of predicting on the Dota2 dataset. Additionally, when using random forest as the classifier, the superior performance of random forest itself diminishes the observable differences between DPSR and DPRF, particularly in the RT-IoT2022 dataset. The significance of the results in Table 3 is discussed in the supplementary material.

6.3. Ablation Study

We present a comparison of the effectiveness between directly adding Gaussian noise to Algorithm 1 (SR+DP) and using our DP-SGD method (DPSR). This experiment aims to

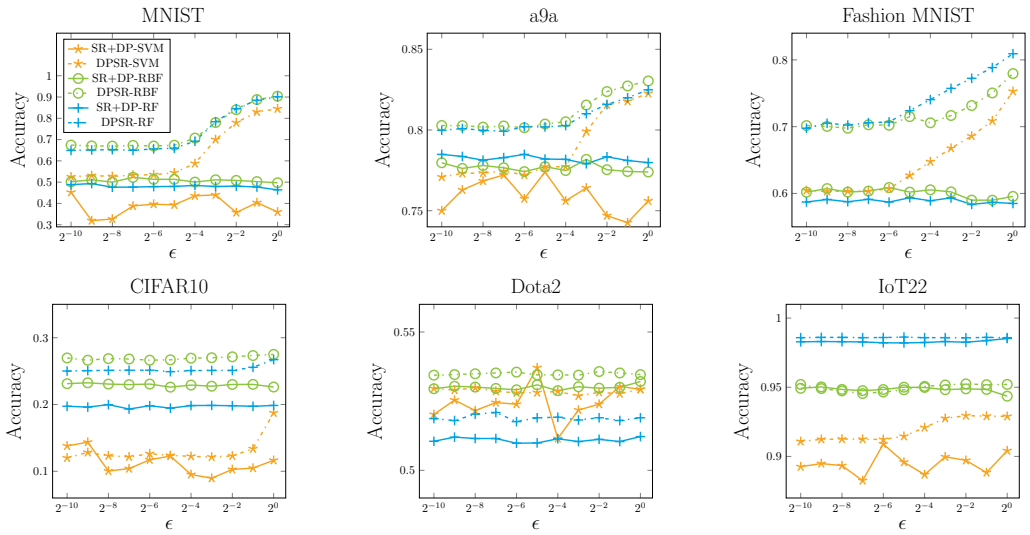

Figure 3: Ablation study: comparison between DPSR and SR+DP.

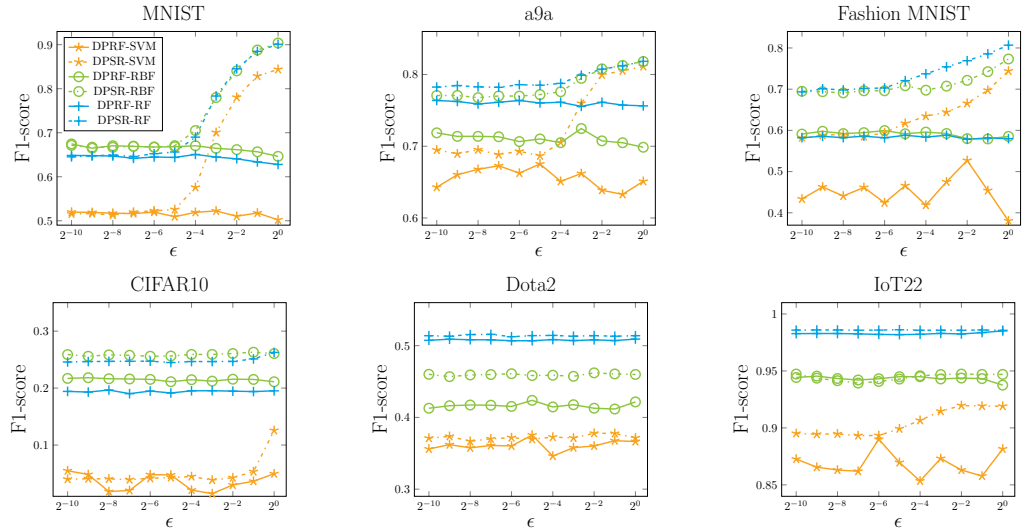

Figure 4: Ablation study: comparison between DPSR and SR+DP.

demonstrate the impact of the different locations where Gaussian noise is added. As shown in Figure 3, 4, adding noise directly to SR results in poor classification performance. This highlights why we chose to use DP-SGD. Similarly to DPSR, we denote the application of SR+DP with SVM, RBF SVM, and random forest as SR+DP-SVM, SR+DP-RBF, and SR+DP-RF, respectively.

Additionally, we conducted a grid search for the regularization ratio, varying from 0.001 to 0.1 in increments of 0.001. The results indicate that the closer the regularization ratio to 0, the better the performance. However, setting it too close to 0 could lead to larger Gaussian noise. This observation can be seen in the experimental results for the a9a and CIFAR-10 datasets. Therefore, we selected 0.01 as the regularization ratio.

Table 3: Wilcoxon signed-rank test for DPSR (ours) and DPRF with $\epsilon = 1$.

Classifier	MNIST	a9a	Fashion	CIFAR10	Dota2	IoT22
linear SVM	0.0000	0.0000	0.0000	0.0000	0.6462	0.0000
RBF SVM	0.0000	0.0000	0.0000	0.0000	0.2069	0.0000
Random Forest	0.0000	0.0000	0.0000	0.0000	0.0539	0.6583

7. Conclusions

In this work, we obtain a similar error bound of Hu et al. (2023) without requiring the initial vector to be sufficiently close to the optimal vector, which poses problems in high-dimensional cases. Our approach allows the initial vector to be any random unit vector. Based on Algorithm 1, we transform the problem into two convex optimization functions, thus eliminating the assumption about the initial vector. Moreover, our algorithm demonstrates better classification accuracy on multiple datasets, especially in the low-dimensional cases.

Both our method and existing approaches suffered performance degradation in high-dimensional settings after applying differential privacy through Gaussian noise and truncation to gradients. There are alternatives for the truncated operations, such as the use of ℓ_0 and ℓ_1 penalties in optimization functions by Journée et al. (2010); Clemmensen et al. (2011) and the greedy algorithm by d'Aspremont et al. (2008). The work by Tan et al. (2018) indicates that the best current approach is to use the truncated operations. However, Cai et al. (2021) pointed out a drawback of the truncation operation: it requires determining the sparsity s of the eigenvectors, which can significantly increase computation time. Consequently, preserving generalized eigenvector sparsity while maintaining classification accuracy in high-dimensional differential privacy remains an open challenge.

References

Martin Abadi, Andy Chu, Ian Goodfellow, H Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. Deep learning with differential privacy. In *Proceedings of the 2016 ACM SIGSAC conference on computer and communications security*, pages 308–318, 2016.

Arthur Asuncion and David Newman. Uci machine learning repository, 2007.

Maria-Florina Balcan, Simon Shaolei Du, Yining Wang, and Adams Wei Yu. An improved gap-dependency analysis of the noisy power method. In *Conference on Learning Theory*, pages 284–309. PMLR, 2016.

Raef Bassily, Adam Smith, and Abhradeep Thakurta. Private empirical risk minimization: Efficient algorithms and tight error bounds. In *2014 IEEE 55th annual symposium on foundations of computer science*, pages 464–473. IEEE, 2014.

Mark Bun and Thomas Steinke. Concentrated differential privacy: Simplifications, extensions, and lower bounds. In *Theory of Cryptography Conference*, pages 635–658. Springer, 2016.

Yunfeng Cai, Guanhua Fang, and Ping Li. A note on sparse generalized eigenvalue problem. In *Advances in Neural Information Processing Systems*, volume 34, pages 23036–23048, 2021.

Chih-Chung Chang and Chih-Jen Lin. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2:27:1–27:27, 2011. Software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm.

Kamalika Chaudhuri, Anand D Sarwate, and Kaushik Sinha. A near-optimal algorithm for differentially-private principal components. *Journal of Machine Learning Research*, 14, 2013.

Line Clemmensen, Trevor Hastie, Daniela Witten, and Bjarne Ersbøll. Sparse discriminant analysis. *Technometrics*, 53(4):406–413, 2011.

Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine learning*, 20: 273–297, 1995.

Alexandre d'Aspremont, Francis Bach, and Laurent El Ghaoui. Optimal solutions for sparse principal component analysis. *Journal of Machine Learning Research*, 9(7), 2008.

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

Cynthia Dwork, Aaron Roth, et al. The algorithmic foundations of differential privacy. *Foundations and Trends® in Theoretical Computer Science*, 9(3–4):211–407, 2014a.

Cynthia Dwork, Kunal Talwar, Abhradeep Thakurta, and Li Zhang. Analyze gauss: optimal bounds for privacy-preserving principal component analysis. In *Proceedings of the forty-sixth annual ACM symposium on Theory of computing*, pages 11–20, 2014b.

Jerome H Friedman. Regularized discriminant analysis. *Journal of the American statistical association*, 84(405):165–175, 1989.

Jason Ge, Zhaoran Wang, Mengdi Wang, and Han Liu. Minimax-optimal privacy-preserving sparse pca in distributed systems. In *International Conference on Artificial Intelligence and Statistics*, pages 1589–1598. PMLR, 2018.

Arghyadeep Ghosh and Mrinal Das. Bottlenecked backpropagation to train differentially private deep neural networks. In *ECAI 2024*, pages 2218–2225. IOS Press, 2024.

Gene H. Golub and Charles F. Van Loan. *Matrix Computations*. Johns Hopkins University Press, 3rd edition, 1996.

Moritz Hardt and Eric Price. The noisy power method: A meta algorithm with applications. *Advances in neural information processing systems*, 27, 2014.

Tin Kam Ho. Random decision forests. In *Proceedings of 3rd international conference on document analysis and recognition*, volume 1, pages 278–282. IEEE, 1995.

Alston S Householder. Unitary triangularization of a nonsymmetric matrix. *Journal of the ACM (JACM)*, 5(4):339–342, 1958.

Lijie Hu, Zihang Xiang, Jiabin Liu, and Di Wang. Nearly optimal rates of privacy-preserving sparse generalized eigenvalue problem. *IEEE Transactions on Knowledge and Data Engineering*, 2023.

Wuxuan Jiang, Cong Xie, and Zhihua Zhang. Wishart mechanism for differentially private principal components analysis. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2016.

Michel Journée, Yurii Nesterov, Peter Richtárik, and Rodolphe Sepulchre. Generalized power method for sparse principal component analysis. *Journal of Machine Learning Research*, 11(2), 2010.

Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

Xiyang Liu, Weihao Kong, Prateek Jain, and Sewoong Oh. Dp-pca: Statistically optimal and differentially private pca. *Advances in neural information processing systems*, 35: 29929–29943, 2022.

RS Martin and James H Wilkinson. Reduction of the symmetric eigenproblem ax= λbx and related problems to standard form. *Numerische Mathematik*, 11:99–110, 1968.

MV Pattabhiraman. The generalized rayleigh quotient. *Canadian Mathematical Bulletin*, 17(2):251–256, 1974.

Daniel L Swets and John Juyang Weng. Using discriminant eigenfeatures for image retrieval. *IEEE Transactions on pattern analysis and machine intelligence*, 18(8):831–836, 1996.

Kean Ming Tan, Zhaoran Wang, Han Liu, and Tong Zhang. Sparse generalized eigenvalue problem: Optimal statistical rates via truncated rayleigh flow. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 80(5):1057–1086, 2018.

Di Wang and Jinhui Xu. Principal component analysis in the local differential privacy model. *Theoretical computer science*, 809:296–312, 2020.

Di Wang, Minwei Ye, and Jinhui Xu. Differentially private empirical risk minimization revisited: Faster and more general. *Advances in Neural Information Processing Systems*, 30, 2017.

Xintao Xia, Linjun Zhang, and Zhanrui Cai. Differentially private sliced inverse regression: Minimax optimality and algorithm, 2024.

Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms, 2017.