Research Article

Differentially Private Singular Value Decomposition for Training Support Vector Machines

Zhenlong Sun, Jing Yang, and Xiaoye Li

1College of Computer Science and Technology, Harbin Engineering University, Harbin 150001, China
2College of Computer and Control Engineering, Qiqihar University, Qiqihar 161006, China

Correspondence should be addressed to Jing Yang; yangjing@hrbeu.edu.cn

Received 25 January 2022; Revised 16 February 2022; Accepted 21 February 2022; Published 26 March 2022

1. Introduction

In the past decade, more and more personal information has been stored in electronic databases for machine learning and personalized recommendation. The data sharing and analysis bring lots of convenience to people’s lives, but pose a great threat to personal privacy. Support vector machine (SVM) [1] is a popular classification method that searches for the best hyperplane that separates two class instances by solving a quadratic optimization problem. It has been applied in pattern recognition such as image recognition and text classification. In the classification model of SVM, the most serious privacy issue is that the support vectors (SVs) are directly obtained from the training datasets [2]. Therefore, the classification model should be privately published to avoid disclosing personal sensitive information.

Differential privacy (DP) [3–6] has a strict mathematical definition and the level of privacy protection can be quantified by a small parameter \( \varepsilon \) named privacy budget. DP has been becoming an accept standard. It guarantees that the result of an analysis is virtually independent of the addition or removal of one record. DP has attracted a growing research attention [7]. The common mechanisms for implementing DP include Laplace mechanism [8], Gaussian mechanism [9], and exponential mechanism [10].

Principal component analysis (PCA) [9] solves a low-rank subspace, which completely captures the variance of matrix \( A \). The main advantages of working with the low-rank approximation of \( A \) include higher time and space efficiency, less noise, and removal of correlation between features. Through PCA, the original instances are projected into a low-dimensional subspace and the features become linearly independent. Eigenvalue decomposition (EVD) and singular value decomposition (SVD) are two common algorithms to perform PCA. They are related to the familiar theory of matrix diagonalization. The EVD is used for a symmetric matrix and SVD for an arbitrary matrix. Furthermore, SVD does not need to compute the matrix of covariance compared with EVD [11].
This study researches the privacy leakage problem of SVM classifier. To overcome some shortcomings in the existing private SVMs, a differentially private singular value decomposition (DPSVD) algorithm is proposed to keep SVs private in the classification model of SVM. This study makes the following innovations:

(i) The authors proposed an idea that projecting the training instances into the low-dimensional singular subspace and the SVM can train the classification model on it while not violating the privacy requirements for the training data.

(ii) The projection process of the DPSVD satisfies DP, and the generated singular vectors are also private which can be provided directly to users for classification testing.

(iii) In the DPSVD, the projection process is implemented by SVD. The main advantage is that SVD does not need to calculate the matrix of covariance compared with EVD. It takes up a lot of memory space for high-dimensional data.

(iv) Our method protects privacy of the training instances before training the classification model; many optimization methods of SVMs can be applied directly to the training progress.

(v) After proving that the DPSVD satisfies differential privacy in theory, several experiments were carried out. The experimental results confirm that our method achieved higher accuracy and better stability on different real datasets, compared with other existing private PCA algorithms used to train SVM.

2. Related Work

From a privacy perspective, SVMs have serious privacy issues, because SVs tend to be directly obtained from the training datasets. There is a lot of work to solve this privacy problem based on DP. Chaudhuri et al. [12, 13] proposed two perturbation-based methods for problems like linear SVM classification. For nonlinear kernel SVM, they derived the kernel function through random projection and linearized the function. However, it is hard to analyze the sensitivity of the output perturbation, and the differentiability criteria are required in the loss function of objective perturbation. To learn SVM privately, Rubinstein et al. [14] developed two feature mapping methods by adding noise to the output classifier. But their methods only apply to the kernels that do not change with translation. Li et al. [15] designed a mixed SVM, which alleviate much of the noise through Fourier transform based on a few open-consented information. Zhang et al. [16] proposed DPSVMDVP by adding Laplace noise to the dual variables based on the error rate. Liu et al. [17] presented an innovative private classifier called LabSam by random sampling under the exponential mechanism. Sun et al. [18] proposed the DPWSS, which introduces randomness into SVM training; they also proposed another private SVM algorithm DPKSVMEL based on exponential and Laplace hybrid mechanism [19] for the kernel SVM to prevent privacy leakage of the SVs.

PCA constructs a set of new features to describe the instances in a low-dimensional subspace. When the generated projection vectors are private, the new instances in the low-dimensional subspace are private as well, and they can be used directly to train SVMs without compromising the privacy of the instances. There are several researches on private PCA. Blum et al. [20] developed SuLQ by disturbing the matrix of covariance with Gaussian noise. However, the greatest eigenvalue might be not real, due to the asymmetry of the noise matrix. Chaudhuri et al. [21] modified SuLQ framework with a symmetric noise matrix and used it for data publishing. Dwork et al. [9] disturbed the matrix of covariance with Gaussian noise. Mitizaz and Sarwate [22, 23] and Jiang et al. [24] disturbed the matrix of covariance with Wishart noise and it guarantees that the perturbed matrix of covariance is positive semidefinite. Xu et al. [25] and Huang et al. [26] added symmetric Laplace noise to the matrix of covariance. Those methods above all generate the perturbed matrix of covariance by adding a noise matrix and then perform EVD to implement PCA. And only [26] measured the availability of the private PCA by SVM, but not to research private PCA from the privacy perspective of SVM. Recently, SVM has been widely used in collaborative filtering [27], deep learning [28], data compression [29, 30], and images watermarking [31]. There are few researches on privacy-preserving data mining based on SVD. Keyvanpour et al. [32] defined a method that combined SVD and feature selection to benefit from the advantages of both domains. Li et al. [33] gave a new algorithm for protecting privacy based on nonnegative matrix factorization and SVD. Kousika et al. [34] proposed a methodology based on SVD and 3D rotation data perturbation for preserving privacy of data.

3. Background

Table 1 summarizes the symbols used in this study.

3.1. Support Vector Machines. Given training instances \( x_i \in \mathbb{R}^d \) and labels \( y_i \in \{-1, 1\} \) the classification model of SVM can be obtained by solving the following optimization problem [35]:

\[
\min_{\alpha} f(\alpha) = \frac{1}{2} \alpha^T Q \alpha - e^T \alpha. 
\]

Subject to \( 0 \leq \alpha_i \leq C, i = 1, ..., l \)

\[ y^T \alpha = 0, \]

where \( \alpha \) is a dual vector; \( Q \) is a symmetric matrix, \( Q_{ij} = y_i y_j K(x_i, x_j) \), and \( K \) is the kernel function.

Let \( x \) be a new instance. The label of \( x \) can be predicted by the decision function as follows:

\[
f(x) = \sum_{i=1}^{n} \alpha_i y_i K(x_i, x) + b. \]
In the classification model, only the SVs determine the maximal margin and correspond to the nonzero $\alpha_i$s, and others equal zero. From a privacy perspective, the classification model has serious privacy issues as the SVs are intact instances.

### 3.2. Principal Component Analysis

PCA computes a low-rank subspace and achieves the dimensionality reduction for high-dimensional data, shedding light on the use of private SVM in high-dimensional data classification. For a given data matrix $D \in \mathbb{R}^{n \times d}$ with $d$ features of $n$ instances, the $i$-th row of $D$ is denoted by $x_i$ and assumes that its $\ell_2$ norm satisfies $\|x_i\|_2 \leq 1$. After the matrix is centralized by column, the matrix of covariance can be obtained as

$$A = \frac{1}{n} D^T D = \frac{1}{n} \sum_{i=1}^{n} x_i^T x_i.$$  

(4)

The matrix of covariance is a real symmetric matrix; therefore its eigenvalues and corresponding eigenvectors can be obtained by EVD:

$$A v_i = \lambda_i v_i,$$  

(5)

where $\lambda_i$ is one of the eigenvalues and $v_i$ is its corresponding eigenvector. The $\lambda_i$ could be treated as variance of the $i$-th principal component to denote its importance and is sorted in descending order. Generally, the threshold of the accumulative contribution rate of principal components $\gamma (0 \leq \gamma \leq 1)$ is set to decide the target dimension $k$ by

$$\sum_{i=1}^{k} \frac{\lambda_i}{\lambda_i} \geq \gamma.$$  

(6)

According to the diagonalization theory of matrix and (5), it obtains another representation of EVD as follows:

$$A = V \sum \lambda_i v_i,$$  

(7)

where $V$ is an orthogonal matrix consisting of eigenvectors in columns and $\sum$ is a diagonal matrix taking eigenvalues as diagonal entries. Compared with EVD, SVD can be applied to an arbitrary real matrix and it does not need to calculate the matrix of covariance. The representation of SVD is shown as follows:

$$D = USV^T,$$  

(8)

where $U$ and $V$ are the left and right singular matrices, which consist of left and right singular vectors, respectively; $S$ is a diagonal matrix taking singular values as diagonal entries. The singular value $\sigma_i$ is also sorted in descending order. The relationship between EVD and SVD is as follows:
where $U^T U = I$ and $V^T V = I$, because $U$ and $V$ are both made up of unit orthogonal vectors; they also are called orthonormal basis matrices. The coefficient $1/n$ has nothing to do with the eigenvectors and the proportionality of eigenvalues. We generally use $D^T D$ to approximate the matrix of covariance. From (9) and (10), we can conclude that the SVD of an arbitrary real matrix yields a similar result to the EVD of its matrix of covariance. In the SVD of $D$, the right singular vectors serve as the eigenvectors of $D^T D$, and the left ones serve as those of $DD^T$. The singular values equal the square roots of the nonzero eigenvalues of $D^T D$ and $DD^T$.

3.3. Differential Privacy

Definition 1 (differential privacy (see [3])). A stochastic mechanism $M$ satisfies $(\varepsilon, \delta)$-differential privacy, provided that, for every two adjacent matrices $D$ and $D'$ differing in exactly one row, and for all subsets of probable outcomes $O \subseteq \text{Range}(M)$,

$$\Pr [M(D) \in O] \leq \exp(\varepsilon) \times \Pr [M(D') \in O] + \delta. \quad (11)$$

When $\delta$ equals zero, $M$ satisfies $\varepsilon$-differential privacy.

Definition 2 (sensitivity (see [3])). For a given function $q : D \rightarrow \mathbb{R}^d$, and adjacent matrices $D$ and $D'$, the sensitivities $S_1$ and $S_2$ of function $q$ can be, respectively, expressed as

$$S_1 = \max_{D,D'} \| q(D) - q(D') \|_1 \quad \text{and} \quad S_2 = \max_{D,D'} \| q(D) - q(D') \|_2. \quad (12)$$

$S_1$ corresponding to $\ell_1$ norm is usually used in Laplace mechanism, while $S_2$ corresponding to $\ell_2$ norm is used in Gaussian mechanism.

Definition 3 (Laplace mechanism (see [8])). For a numeric function $q : D \rightarrow \mathbb{R}^d$, with scale factor $b = S_1/\varepsilon$. The Laplace mechanism, which adds independent random noise distributed as $\text{Laplace}(b)$ to each output of $q(D)$, ensures $\varepsilon$-differential privacy.

Definition 4 (Gaussian mechanism (see [9])). For a numeric function $q : D \rightarrow \mathbb{R}^d$, let $b = S_2 \sqrt{2 \ln(1.25/\delta)}/\varepsilon$. The Gaussian mechanism, which adds independent random noise distributed as $\mathcal{N}(0, b^2)$ to each output of $q(D)$, ensures $(\varepsilon, \delta)$-differential privacy.

4. Materials and Methods

To overcome the shortcomings in the existing private SVMs, we proposed the DPSVD. The DPSVD privately projects the original instances to a low-dimension singular subspace and trains a SVM classification model in it to protect the privacy of training instances.

4.1. Algorithm Description: Algorithm 1 Is the Pseudocode of the DPSVD

The algorithm 1 describes the implementation process of the DPSVD for training a private classification model of SVM. Firstly, it generates a noise matrix sampled from Gaussian distribution, and this step does not need to symmetrize the noise matrix as the existing private PCA algorithms. Secondly, it adds the noise matrix to the raw data matrix rather than the matrix of covariance of the raw data. When features far outnumber instances, the matrix of covariance will take up a lot of memory space, especially for high-dimensional data. Meanwhile, the matrix of covariance will magnify errors in the raw data to some extent. Thirdly, the DPVSD algorithm computes the singular values and singular matrices by SVD, while the existing private PCA algorithms use EVD. Generally, SVD can be considered a black box and has higher execution efficiency compared with EVD, although the two decomposition methods generate the same projection subspace by singular vectors or eigenvectors under the non-private situation. There are similar computing processes with the EVD in the next three steps. Lastly, the DPSVD distributes the private classification model to predict the new instances, prior to idea that it projects them to the same singular subspace by the private singular vectors. In brief, the DPSVD trains a private SVM classifier for predicting the new instances in future.

4.2. Privacy Analysis

Firstly, the sensitivity of the function $q(D)$ is analyzed, and then the DPSVD is demonstrated to satisfy $(\varepsilon, \delta)$-differential privacy. In the DPSVD algorithm, the noise matrix is added to the data matrix $D$; therefore $q(D) = D$. Given that two adjacent data matrices $D$ and $D'$ differ by exactly one row corresponding to an instance, we set $D'$ obtained from $D$ by deleting the last row, $D = [x_1, ..., x_n]^T \in \mathbb{R}^{n \times d}$ and $D' = [x_1, ..., x_{n-1}, 1]^T \in \mathbb{R}^{(n-1) \times d}$, and assume each row has unit $\ell_2$ norm $\| x \| \leq 1$ at the most.

Lemma 1. The sensitivity of the function $q(D)$ $S_2$ equals one.

Proof. According to Definition 2, it obtains $S_2$ by the following inequation:

$$S_2 = \max_{D,D'} \| q(D) - q(D') \|_2 = \max_{D,D'} \| D - D' \|_2 = \max_{D,D'} \| x_n \|_2 \leq 1 \quad (13)$$

Therefore, the sensitivity of the function $q(D)$ equals one.
generates the private singular vectors \( V_k \); they also satisfy DP. Step (5) selects the target dimension \( k \) according to \( \sum_{i=1}^{k} \sigma_i^2 / \sum_{i=1}^{d} \sigma_i^2 \geq \gamma \); select first \( k \) singular vectors \( V_k \) to project the original training instances to the low-dimensional singular subspace \( Y = DV_k \); compute the classification model \( f(x) \) in the singular subspace; use \( f(x) \) and \( V_k \) to predict the new instances.

\textbf{Algorithm 1: DPSVD.}

\begin{algorithm}
\textbf{Input}: Raw data matrix \( D \in \mathbb{R}^{n \times d} \), instances \( n \), features \( d \), privacy parameters \( \varepsilon \), \( \delta \) and \( \beta \), accumulative contribution rate of principal components \( \gamma \);
\textbf{Output}: Classification model \( f(x) = \sum_{i=1}^{n} \alpha_i y_i K(x_i, x) + b \), private singular vectors \( V_k \);
\textbf{Begin}
Generate a noise matrix \( E \in \mathbb{R}^{n \times d} \), every entry is i.i.d. and sampled from \( \mathcal{N}(0, \beta^2) \);
Add the noise matrix to the raw data matrix \( D' = D + E \);
Compute the singular values \( \sigma \) and singular matrices \( U, V \) of \( D' \) by SVD, \( D' = USV^T \);
Select the target dimension \( k \) according to \( \sum_{i=1}^{k} \sigma_i^2 / \sum_{i=1}^{d} \sigma_i^2 \geq \gamma \);
Select first \( k \) singular vectors \( V_k \) to project the original training instances to the low-dimensional singular subspace \( Y = DV_k \);
Compute the classification model \( f(x) \) in the singular subspace;
Use \( f(x) \) and \( V_k \) to predict the new instances.
\textbf{End}
\end{algorithm}

\textbf{Table 2: The comparison between the three algorithms.}

| Algorithm   | PCA    | Adding mode | Noise form  | Noise scale   | Mechanism  | Privacy level |
|-------------|--------|-------------|-------------|---------------|------------|---------------|
| DPSVD       | SVD    | \( D \)     | Asymmetric  | \( O(\sqrt{d/n}) \) | Gaussian   | \((\varepsilon, \delta)\) |
| AG          | EVD    | \( D^2D/n \) | Symmetric   | \( O(\sqrt{d/n}) \) | Gaussian   | \((\varepsilon, \delta)\) |
| DPPCA-SVM   | EVD    | \( D^2D/n \) | Symmetric   | \( O(d/n) \)   | Laplace    | \((\varepsilon, 0)\)   |

\textbf{Table 3: Test datasets.}

| Indices | Datasets | Instances | Features | Ranges   |
|---------|----------|-----------|----------|----------|
| 1       | A1a      | 1605      | 119      | [0, 1]   |
| 2       | Mushrooms| 8124      | 112      | [-1, 1]  |
| 3       | Musk     | 6598      | 166      | [-1, 1]  |
| 4       | Splice   | 1000      | 60       | [-1, 1]  |

**Theorem 1.** The DPSVD satisfies \((\varepsilon, \delta)\)-differential privacy.

\textbf{Proof.} To demonstrate that the DPSVD satisfies \((\varepsilon, \delta)\)-differential privacy, it is necessary to demonstrate every step in the algorithm satisfies it. According to Lemma 1, it obtains that \( S_2 \) equals one. Let \( \beta = S_2 \sqrt{2 \ln(1.25/\delta)}/\varepsilon = \sqrt{2 \ln(1.25/\delta)}/\varepsilon \); then Step (1) and Step (2) satisfy DP according to Definition 4. Step (3) and Step (4) postprocess the private data matrix \( D' \); they also satisfy DP. Step (5) generates the private singular vectors \( V_k \); the projected instances \( Y \) in the low-dimensional singular subspace are private as well. Meanwhile, \( Y \) does not need to be distributed to users. Step (6) and Step (7) compute the classification model based on private projection instances and distribute it together with private singular vector to predict the new instances. The last three steps do not violate the privacy requirement of DP. Therefore, the DPSVD satisfies \((\varepsilon, \delta)\)-differential privacy.

4.3. \textbf{Algorithm Comparison.} The three algorithms were compared theoretically between DPSVD, AG [9], and DPPCA-SVM [28] summarized in Table 2. Other ones have been compared by the DPPCA-SVM algorithm. Our algorithm uses SVD to perform PCA; it does not need to compute matrix of covariance and symmetrize the noise matrix as the description above. It obtains the same noise scale as AG algorithm, because they use identical mechanism of DP to generate the noise matrix.

Therefore, the classification model and the singular vectors for projection are both private; they can be used to predict the new instances in the same singular subspace. The main advantage of the DPSVD compared with other private SVMs is that our algorithm trains the classification model in the private low-dimensional singular subspace generated by SVD. In this way, the features of the instances in the singular subspace become linearly independent and low-dimensional and therefore have higher time and space efficiency for training the classification model. The difference between our algorithm and other private PCA algorithms is that it does not need to calculate the matrix of covariance or symmetrize the noise matrix. Meanwhile, the DPSVD protects privacy of the training instances before training the classification model; many optimization methods of SVMs can be applied directly to the training progress.

\section{Results}

5.1. \textbf{Datasets.} Our experiments were carried out on four popular datasets for testing SVM performance. Table 3
Table 4: Performance comparison of algorithms on different datasets.

| Datasets   | ε   | Algorithm   | Accuracy | SV  |
|------------|-----|-------------|----------|-----|
|            |     |             | Mean     | Std  | Max   | Min   | Mean | Std  | Max   | Min   |
| A1a        | --  | SVM         | 83.49    | 754  | --    | --    | --   | --   | --    | --    |
|            | 0.1 | DPSVD       | 83.35    | 0.15  | 83.61 | 763   | 6    | 771  | 756   | --    |
|            |     | AG          | 83.08    | 0.24  | 83.30 | 743   | 4    | 747  | 738   | --    |
|            | 0.5 | DPSVD       | 83.50    | 0.22  | 83.86 | 755   | 9    | 768  | 743   | --    |
|            |     | AG          | 83.09    | 0.16  | 83.18 | 686   | 3    | 689  | 682   | --    |
|            | 1   | DPSVD       | 83.59    | 0.13  | 83.74 | 753   | 5    | 759  | 748   | --    |
|            |     | AG          | 83.45    | 0.24  | 83.80 | 697   | 4    | 702  | 693   | --    |
|            |     | DPPCA-SVM   | 83.46    | 0.41  | 84.11 | 756   | 14   | 778  | 742   | --    |
| Mushrooms  | --  | SVM         | 99.90    | 617   | --    | --    | --   | --   | --    | --    |
|            | 0.1 | DPSVD       | 99.89    | 0.01  | 99.90 | 639   | 39   | 700  | 607   | --    |
|            |     | AG          | 99.18    | 0.02  | 99.21 | 518   | 3    | 521  | 514   | --    |
|            | 0.5 | DPSVD       | 99.90    | 0.42  | 99.89 | 633   | 25   | 674  | 604   | --    |
|            |     | AG          | 99.19    | 0.05  | 99.26 | 524   | 5    | 531  | 517   | --    |
|            | 1   | DPSVD       | 99.90    | 0.38  | 99.90 | 633   | 25   | 674  | 604   | --    |
|            |     | AG          | 99.79    | 0.04  | 99.83 | 445   | 22   | 469  | 417   | --    |
|            |     | DPPCA-SVM   | 99.83    | 0.08  | 99.98 | 651   | 81   | 779  | 559   | --    |
| Musk       | --  | SVM         | 93.95    | 1351  | --    | --    | --   | --   | --    | --    |
|            | 0.1 | DPSVD       | 94.08    | 0.11  | 94.23 | 1351  | 15   | 1369 | 1330  | --    |
|            |     | AG          | 88.96    | 0.08  | 89.09 | 1865  | 10   | 1876 | 1855  | --    |
|            | 0.5 | DPSVD       | 94.14    | 0.11  | 94.27 | 1359  | 14   | 1379 | 1341  | --    |
|            |     | AG          | 88.94    | 0.01  | 88.95 | 1866  | 6    | 1874 | 1858  | --    |
|            | 1   | DPSVD       | 94.14    | 0.15  | 94.35 | 1345  | 10   | 1358 | 1333  | --    |
|            |     | AG          | 88.93    | 0.02  | 88.95 | 1872  | 10   | 1887 | 1860  | --    |
|            |     | DPPCA-SVM   | 94.19    | 0.17  | 94.35 | 1318  | 40   | 1384 | 1275  | --    |
| Splice     | --  | SVM         | 94.30    | 607   | --    | --    | --   | --   | --    | --    |
|            | 0.1 | DPSVD       | 91.08    | 0.75  | 92.40 | 635   | 17   | 662  | 616   | --    |
|            |     | AG          | 90.56    | 0.83  | 91.30 | 591   | 16   | 605  | 568   | --    |
|            | 0.5 | DPSVD       | 92.00    | 0.80  | 92.80 | 610   | 10   | 625  | 600   | --    |
|            |     | AG          | 93.58    | 0.38  | 94.00 | 588   | 5    | 595  | 582   | --    |
|            | 1   | DPSVD       | 87.22    | 0.73  | 88.40 | 659   | 34   | 706  | 615   | --    |
|            |     | AG          | 93.56    | 0.56  | 93.10 | 618   | 19   | 645  | 594   | --    |
|            |     | DPPCA-SVM   | 87.36    | 0.94  | 88.30 | 641   | 21   | 667  | 614   | --    |

Figure 1: Accuracy at various ε on dataset A1a.
describes their basic information, including the number of the instances, the number of the features, and the ranges of values in the data. They are accessible at https://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/ and http://archive.ics.uci.edu/ml/datasets.php. It trains the SVM to compare the performance of algorithms based on LIBSVM (version 3.25) [36] with the radial basis function as the kernel function and default parameters.

5.2. Algorithm Performance Experiments. The performance of the DPSVD was compared with AG, DPPCA-SVM, and the nonprivate SVM on the four real datasets. In the
experiments, it designed two metrics of algorithm performance: **Accuracy** and **SV**. **Accuracy** denotes how accurate the classification is, and **SV** denotes how many SVs are contained in the classifier. The higher the **Accuracy**, the greater the usability of the classifier. The closer **SV** to that of nonprivate SVM, the better the stability of the algorithm. The privacy budget $\varepsilon$ was set at 0.1, 0.5, and 1, $\delta$ at $1/n^2$, and the accumulative contribution rate of principal components $\gamma$ at 90%. Three private algorithms were implemented five times under every privacy budget parameter. The mean values, standard deviation, and maximum and minimum values of the two metrics are given in Table 4.

From the experiments results in Table 4, the DPSVD was the most accurate in classification than the other two private classifiers under different privacy budget for most of the datasets. Sometimes, our algorithm even outperformed the

![Graph showing Accuracy at various $\varepsilon$ on dataset Splice.](attachment:image1.png)

**Figure 4:** Accuracy at various $\varepsilon$ on dataset Splice.

![Graph showing SV at various $\varepsilon$ on dataset A1a.](attachment:image2.png)

**Figure 5:** SV at various $\varepsilon$ on dataset A1a.
nonprivate SVM. This is mainly because our algorithm removes the linear dependence between features and unimportant features by SVD. Meanwhile, our algorithm has the better stability of the algorithm as its SV is much closer to the nonprivate SVM. To compare algorithm performance more intuitively, the mean values of the two metrics for the four algorithms are shown in Figures 1–8.

In Figure 1 to Figure 3, the DPSVD achieved the highest classification accuracy in the three private algorithms and closer to the nonprivate SVM than the other two algorithms. In Figure 4, the AG achieved the higher classification accuracy than the DPSVD as the privacy budget increases. In Figure 5 to Figure 8, the DPSVD contained the closer number of SVs in the classifier to the nonprivate SVM than the other two algorithms. Therefore, the PDSVD achieved higher classification accuracy and better algorithm stability on most of the datasets and approximated the performance of the nonprivate SVM. The AG algorithm on dataset Musk in Figure 3 and the DPPCA-SVM algorithm on dataset Splice in Figure 4 have relatively lower classification accuracy. It also shows that the DPSVD has better algorithm stability.

6. Conclusions

To solve the privacy leak of SVM classifiers, especially on high-dimensional data, the DPSVD algorithm was proposed to project the training instances into the low-dimensional singular subspace and train a private SVM classifier on it while not violating the privacy requirements for the training data. The DPSVD is proved to satisfy DP. The main advantages of the DPSVD include three aspects. Firstly, it trains the classification model in the private low-dimensional singular subspace; therefore it has higher time and space efficiency compared with other private SVMs. Secondly, it does not need to calculate the matrix of covariance or symmetrize the noise matrix and has higher classification accuracy and better stability of the algorithm than other existing private PCA algorithms through the comparison experiments. Thirdly, it protects privacy of the training instances before training the classification model, and many optimization methods of SVMs can be applied directly to the training progress. Meanwhile, its algorithmic ideas can be applied to other machine learning areas to solve data privacy problems. However, the DPSVD can only solve the linear dependence between the data features. In future work, we will consider the nonlinear dependence to train a private classification model. In addition, the problem of data instances compression through SVD is another research direction.
**Data Availability**

The raw data of the four datasets are available at https://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/ and http://archive.ics.uci.edu/ml/datasets.php.

**Conflicts of Interest**

The authors declare that they have no conflicts of interest.

**Acknowledgments**

This work was supported by the National Natural Science Foundation of China under Grants 61672179, 61370083, 61402126, and 61501275, by the National Science Foundation of Heilongjiang Province under Grant QG2016083, by the Postdoctoral Fellowship of Heilongjiang Province under Grant F2015030, by the Natural Science Foundation of China under Grants 61672179, 61370083, and 61402126, and by the Fundamental Research Funds in Heilongjiang Provincial Universities under Grant 135509312.

**References**

[1] C. J. C. Burges, “A tutorial on support vector machines for pattern recognition,” *Data Mining and Knowledge Discovery*, vol. 2, no. 2, pp. 121–167, 1998.

[2] K.-P. Lin and M.-S. Chen, “On the design and analysis of the privacy-preserving SVM classifier,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 23, no. 11, pp. 1704–1717, 2011.

[3] C. Dwork, “Differential privacy,” in *Automata, Languages and Programming*, Proceedings of the 33th Colloquium on Automata, Languages and Programming, pp. 1–12, Venice, Italy, July 2006.

[4] C. Liu, J. Yang, W. Zhao et al., “Differential privacy protection of face images based on region growing,” * Traitement du Signal*, vol. 38, no. 5, pp. 1385–1401, 2021.

[5] J. Yu, “Design of a privacy-preserving algorithm for peer-to-peer network based on differential privacy,” *Ingénierie des Systèmes d’Information*, vol. 24, no. 4, pp. 433–437, 2019.

[6] C. Liu, J. Yang, Y. Zhang et al., “Non-global privacy protection facing sensitive areas in face images,” *Traitement du Signal*, vol. 38, no. 6, pp. 1677–1687, 2021.

[7] 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.

[8] C. Dwork, F. McSherry, K. Nissim, and A. Smith, “Calibrating noise to sensitivity in private data analysis,” *Theory of Cryptography*, in *Proceedings of the 3rd Conference on theory of Cryptography*, pp. 265–284, New York, NY, USA, 4-7 March 2006.

[9] C. Dwork, K. Talwar, A. Tzurka, 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, May 31 - June 3, 2014.

[10] F. McSherry and K. Talwar, “Mechanism design via differential privacy,” in *Proceedings of the 48th Annual IEEE Symposium on Foundations of Computer Science (FOCS’07)*, Providence, RI, USA, 21-23 Oct. 2007.

[11] D. Kalman, “A singularly valuable decomposition: the SVD of a matrix,” *The College Mathematics Journal*, vol. 27, no. 1, pp. 2–23, 1996.

[12] K. Chaudhuri and C. Monteleoni, “Privacy-preserving logistic regression,” in *Proceedings of the 2008 21st ACM Conference on Advances in Neural Information Processing Systems*, pp. 289–296, Austin, TX, USA, November 2009.

[13] K. Chaudhuri, C. Monteleoni, and A. D. Sarwate, “Differentially private empirical risk minimization,” *Journal of Machine Learning Research: JMLR*, vol. 12, no. 3, pp. 1069–1109, 2011.

[14] B. Ip Rubinstein, P. L. Bartlett, L. Huang, and N. Taft, “Learning in a large function space: privacy-preserving mechanisms for SVM learning,” 2009, https://arxiv.org/abs/0911.5708.

[15] H. Li, L. Xiong, L. M. Ohno, and X. Jiang, “Privacy preserving RBF kernel support vector machines,” *BioMed Research International*, vol. 2014, p. 10, Article ID 827371, 2014.

[16] Y. L. Zhang, Z. F. Hao, and S. P. Wang, “A differential privacy support vector machine classifier based on dual variable perturbation,” *IEEE Access*, vol. 7, p. 98251, 2019.

[17] X. Liu, Q. Li, and T. Li, “Private classification with limited labeled data,” *Knowledge-Based Systems*, vol. 133, no. 1, pp. 197–207, 2017.

[18] Z. Sun, J. Yang, X. Li, and J. Zhang, “DPWSS: differentially private set selection for training support vector machines,” *PeerJ Computer Science*, vol. 7, p. e799, 2021.

[19] Z. Sun, J. Yang, X. Li, and J. Zhang. “Differentially private kernel support vector machines based on the exponential and Laplace hybrid mechanism,” *Security and Communication Networks*, vol. 2021, p. 19, Article ID 9506907, 2021.

[20] 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, New York, NY, USA, June 2005.

[21] K. Chaudhuri, A. D. Sarwate, and S. Kaushik, “A near-optimal algorithm for differentially-private principal components,” *Journal of Machine Learning Research*, vol. 14, no. 1, pp. 2905–2943, 2005.

[22] 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)*, 2339–2343, ICASSP, Shanghai, China, March 2016.

[23] H. Imtiaz and A. D. Sarwate, “Differentially private distributed principal component analysis,” in *Proceedings of the 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2206–2210, 15-20 April 2018.

[24] W. Jiang, C. Xie, and Z. Zhang, “Wishart mechanism for differentially private principal components analysis,” in *Proceedings of the Birtieth AAAI Conference on Artificial Intelligence*, AAAI Press, Phoenix, AZ, USA, February 2016.

[25] Y. H. Xu, G. Yang, and S. J. Bai, ”Laplace input and output perturbation for differentially private principal components analysis,” *Security and Communication Networks*, vol. 2019, p. 10, Article ID 9169802, 2019.

[26] Y. X. Huang, G. Yang, Y. H. Xu, and H. Zhou, “Differential privacy principal component analysis for support vector machines,” *Security and Communication Networks*, vol. 2021, p. 12, Article ID 5542283, 2021.

[27] T. Huang, R. Zhao, L. Bi, D. Zhang, and C. Lu, “Neural embedding singular value decomposition for collaborative filtering,” *IEEE Transactions on Neural Networks and Learning Systems*, pp. 1–9, 2021.
[28] S. N. Qasem and A. Mohammadzadeh, "A deep learned type-2 fuzzy neural network: singular value decomposition approach," *Applied Soft Computing*, vol. 105, p. 107244, 2021.

[29] S. Hashemipour, J. Aghaei, A. Kavousi-fard et al., "Optimal singular value decomposition based big data compression approach in smart grids," *IEEE Transactions on Industry Applications*, vol. 57, no. 4, pp. 3296–3305, 2021.

[30] R. Pourramezan, R. Hassani, H. Karimi, M. Paolone, and J. Mahseredjian, "A real-time synchrophasor data compression method using singular value decomposition," *IEEE Transactions on Smart Grid*, vol. 13, no. 1, pp. 564–575, Jan. 2022.

[31] Y. Sun, Q. Su, H. Wang, and G. Wang, "A blind dual color images watermarking based on quaternion singular value decomposition," *Multimedia Tools and Applications*, vol. 81, no. 5, pp. 6091–6113, 2022.

[32] M. R. Keyvanpour and S. S. Moradi, "A perturbation method based on singular value decomposition and feature selection for privacy preserving data mining," *International Journal of Data Warehousing and Mining*, vol. 10, no. 1, pp. 55–76, 2014.

[33] G. Li and R. Xue, "A new privacy-preserving data mining method using non-negative matrix factorization and singular value decomposition," *Wireless Personal Communications*, vol. 102, no. 2, pp. 1799–1808, 2018.

[34] N. Kousika and K. Premalatha, "An improved privacy-preserving data mining technique using singular value decomposition with three-dimensional rotation data perturbation," *The Journal of Supercomputing*, vol. 77, no. 9, pp. 10003–10011, 2021.

[35] R. E. Fan, P. H. Chen, and C. J. Lin, "Working set selection using second order information for training support vector machines," *Journal of Machine Learning Research*, vol. 6, no. 12, pp. 1889–1918, 2005.

[36] C.-C. Chang and C.-J. Lin, "Libsvm," *ACM Transactions on Intelligent Systems and Technology*, vol. 2, no. 3, pp. 1–27, 2011.