Membership Inference Attack Against Principal Component Analysis

Oualid Zari[1], Javier Parra-Arnau[2,3], Ayşe Ünsal[1], Thorsten Strufe[2], and Melek Önen[1]

[1] EURECOM, France
[2] Karlsruhe Institute of Technology, Germany
[3] Universitat Politècnica de Catalunya, Spain

Abstract. This paper studies the performance of membership inference attacks against *principal component analysis* (PCA). In this attack, we assume that the adversary has access to the principal components, and her main goal is to infer whether a given data sample was used to compute these principal components. We show that our attack is successful and achieves high performance when the number of samples used to compute the principal components is small. As a defense strategy, we investigate the use of various differentially private mechanisms. Accordingly, we present experimental results on the performance of Gaussian and Laplace mechanisms under *naive* and *advanced compositions* against MIA as well as the utility of these differentially-private PCA solutions.

Keywords: membership inference attack, principal component analysis, differential privacy, Laplace mechanism, Gaussian mechanism

1 Introduction

Over the past decade, machine learning (ML) algorithms have found application in a vast and rapidly growing number of systems for analyzing and classifying usually privacy-sensitive data.

In order to analyze and interpret such data, PCA [18] is employed as one of the most commonly used unsupervised ML algorithm. PCA is used for summarizing the information content in databases by reducing the dimensionality of the data while preserving as much variability as possible. The output of this statistical tool is a set of *principal components* whose size is usually much smaller than the total number of attributes of the underlying data.

The increasing popularity of ML algorithms, including PCA, opened the door for attackers especially when ML techniques are deployed in critical applications. This work focuses on a particular type of attack named *Membership Inference Attack* (MIA) against PCA, where an adversary is assumed to intercept the principal components computed over some dataset and infer whether a data sample was part of this dataset or not. The membership prediction is yield by comparing the reconstruction error; the distance between the original target sample and its PCA projection against a threshold. In this paper, we study the effectiveness of MIA against PCA and show that it achieves high performance when the number of samples used by PCA is small.

Furthermore, to cope with such attacks that take advantage of the leakage of principal components, we propose to study the use of *differentially private* mechanisms and evaluate the privacy budget affects the success rate of the attack as well as the utility of the PCA under differential privacy (DP).

Our main contributions are summarized as follows.

1. We study, for the first time, the impact of MIA against PCA whereby the adversary has access to the principal components.
2. We propose the use of differentially-private PCA algorithms to cope with MIA and analyze the impact of the privacy budget on both utility and the success rate of MIA for both vector and scalar queries under the *so-called* naive and advanced composition approaches.
3. The experimental results present a comparison between the aforementioned different approaches under Gaussian and Laplace mechanisms for protecting the PCA against MIA.

2 Background

2.1 Principal Component Analysis

Given a set $D = \{x_n \in \mathbb{R}^d : n = 1 : N\}$ of N raw data samples corresponding to N individuals of dimension d, we denote the data matrix where each column is a data sample by $X = [x_1, \ldots, x_N]$. We assume that data X has zero mean, which can be ensured by centering the data. The standard PCA algorithm is to find a $k-$dimensional subspace that approximates each sample x_n. This problem can be formulated as follows:

$$\min_{\Pi_k} \mathcal{L} = \frac{1}{N} \sum_{n=1}^{N} \mathcal{L}_n = \frac{1}{N} \sum_{n=1}^{N} \|x_n - \Pi_k x_n\|_2^2 \tag{1}$$

where $\mathcal{L}$ denotes the average reconstruction error and Π_k is an orthogonal projector which is used for approximating each sample x_n by $\hat{x}_n = \Pi_k x_n$. The solution to this problem can be achieved via singular value decomposition (SVD) of the sample covariance matrix, which is defined by $A = \frac{1}{N} X X^T = \frac{1}{N} \sum_{n=1}^{N} x_n x_n^T$. A is a symmetric positive semi-definite matrix, hence its singular value decomposition is equivalent to its spectral decomposition. SVD of A yields $A = \sum_{i=1}^{d} \lambda_i v_i v_i^T$, where $\lambda_1 \geq \lambda_2 \geq \ldots, \lambda_d \geq 0$ and $v_1, v_2, \ldots, v_d$ denote the eigenvalues and their corresponding eigenvectors of A, respectively. Let us denote the matrix whose columns are the top k eigenvectors by $V_k = [v_1, \ldots, v_k]$. The orthogonal projector $\Pi_k = V_k V_k^T$ is a solution to the problem in (1). PCA uses V_k to project the samples into the low $k-$dimensional subspace $Y = V_k^T X$.

2.2 Membership Inference Attacks

The goal of an MIA is to infer whether or not a target sample is included in the training dataset. When an adversary learns whether or not a target sample was used to release any statistics or to train a machine learning model, this refers to an information leakage. This attack could cause serious problems in terms of privacy if the training dataset contains privacy-sensitive information. An example that highlights the implications of such an attack is [7], which was able to identify individuals contributing their DNA to a health-related project.

3 Related Work

Since the introduction of MIA against deep neural network (DNN) models in [22], this attack has been extensively studied on DNNs and other ML models. The cited work formalized the attack as a binary classification problem and trained neural network (NN) classifiers to distinguish between training members and non-members. The authors demonstrates that the main factor contributing to the success of MIA on DNN models is overfitting. Subsequent works [21, 15, 23, 13, 27] further developed MIAs with different approaches against DNN of different architectures. The work in [23] revealed that by using suitable metrics, metric-based attacks result in similar attack performance when compared with NN-based attacks. Besides DNN, MIAs have also been investigated against logistic regression models [25, 20], k-nearest neighbors [25, 24], and decision tree models[25, 27]. Our work extends the investigations of MIAs against machine learning models to PCA. As we shall elaborate later in Section 4.1, we propose, to this end, a new metric-based MIA against PCA. To the best of our knowledge, there is no previous work trying to perform MIA on PCA.

To mitigate MIAs, DP has been widely applied to various ML models [12, 13, 26, 28]. In [1], the authors show how to train DNNs with DP by adding noise to the gradients or parameters during model training. In [19], the authors empirically evaluate MIAs using the proposal of [1]. They find that DP can partially mitigate the attack with an acceptable level of privacy budget. In our study, we investigate the effectiveness of DP PCA algorithms on mitigating our proposed attack.

4 Membership Inference Attacks against PCA

The first part of this work focuses on the study of the impact of MIA targeting PCA. We aim to investigate how the sample size and the number of the intercepted principle components affect the performance of such attacks. In Section 4.1, we define the threat model and the actual MIA targeting PCA. This is followed by the experimental setup and the corresponding experimental results of Sections 4.2 and 4.3, respectively.

4.1 Threat model and attack methodology

In our setting, the curator computes the principal components V_k using the training dataset D, and sends these to a trusted party. We assume the adversary $\mathcal{A}$ intercepts some or all of those components by eavesdropping the communication channel. With them, the adversary aims to identify whether or not a certain sample z is included in D. In other words, the adversary's goal is to discover members of the training dataset.

Such an attack can of course occur in a distributed setting [2] where several parties may compute the principal components of their individual (and usually smaller [10]) training datasets and send those to an aggregator, which ultimately may compute the global principal components. Analogously to the non-distributed case, here $\mathcal{A}$ would compromise individual privacy by intercepting the principal components conveyed by each party.

To identify whether or not sample z was actually used for the computation of the principal components, $\mathcal{A}$ computes the reconstruction error $\mathcal{L}(z, V_k)$ of the target sample z based on the intercepted V_k, and then compares this error with some tunable decision threshold R. If the reconstruction error of the target sample is lower than the threshold, $\mathcal{A}$ predicts that z is a member of the training dataset D. Otherwise, $\mathcal{A}$ predicts that z is not a member of D. Our intuition is that samples from the training dataset are more likely to incur lower reconstruction error compared to other non-member samples.

4.2 Experimental setup

We proceed with a detailed description of the datasets used in our experiments.

Datasets:[4] We assess the performance of the attack using two groups of datasets: (i) datasets including personal information, namely, UCI Adult [16] (for short, Adult), Census [4], and LFW [8]; and the image dataset MNIST [14], which is typically used in the literature of MIAs. As preprocessing, we standardize the datasets to unit variance before constructing our attack.

- The UCI Adult dataset includes 48,842 records with 14 attributes. It contains both numerical (e.g. age, hours per week, etc.) and categorical (e.g. working class, education, etc.) attributes. We employ the standard one-hot encoding approach to construct the numerical representation of the categorical attributes [9].
- Census: it contains 1080 records with 13 attributes of business statistics.
- Labeled Faces in the Wild (LFW): It includes 13,233 images of 5749 human faces collected from the Web. 1680 of the 5749 people pictured have at least two distinct images in this dataset. The resolution of the images is 25×18. In our evaluation, in order to balance the number of samples for each individual, we only take one picture of each individual in the dataset.

[4] Due to page limit constraint, we report only the results for Adult and LFW datasets. We refer the reader to the full version of this paper [29].

- MNIST: it includes 10 classes of handwritten digits formatted as grayscale 28× 28 pixel images. The dataset is used to predict the class of the digit represented in the image. The total number of samples is 70,000.

Performance metric: As an evaluation metric of the attack's success, we use the area under the receiver operating characteristic(ROC) curve (AUC) metric, which indicates the relationship between true positive and false-negative rates over several decision thresholds R that the adversary can use to construct the attack. In all experiments, we choose equal-sized samples for both members and non-members at random and report the mean of the results over 10 trials.

4.3 Experimental Results

We evaluate the success rate of the attack in terms of the number of principal components intercepted by the adversary, denoted by k. For this, we measure the attacker's performance through the AUC. Figure 1 shows the maximum AUC that the adversary can achieve by observing the top-k principal components. Recall that k may take values from 1 to d, where d is the number of attributes of the dataset. We report results for various number of samples N. The closer the AUC is to 0.5, the less successful the attack is as the adversary cannot distinguish between a member and a non-member.

We observe that the AUC increases with increasing k. This is justified by the fact that the attacker has access to more information and therefore is more likely to succeed in identifying the membership. We also observe that the AUC decreases with increasing N, perhaps, indicating that the sample covariance matrix A converges to the true covariance matrix of the dataset, which renders the reconstruction error of member and non-member samples of D indistinguishable. The same behaviour is observed with NNs when the training dataset is large [22].

The results for the MNIST and LFW datasets indicate that the AUC is always greater than 0.5 and reaches 0.9 when $N = 1,000$. As for the Census and Adult datasets, the corresponding AUC values are much lower (compared to the other datasets). This is mainly justified by the small dimension d of these datasets. We note that MIA against machine learning models trained using the Adult dataset is usually unsuccessful [22, 21].

5 Differentially-private PCA and MIA

In this section, we present PCA(DP-PCA) algorithms introduced in [6, 17], and study their protection against MIAs with various privacy budget values and their utility. Accordingly, we first remind several preliminaries DP in Section 5.1. This is followed by the experimental results in Section 5.3.

5.1 Preliminaries on Differential Privacy

Definition 1 (Neighboring datasets). *Any two datasets that differ in one record are called neighbors. For two neighbor datasets* $\mathbf{x}$ *and* $\mathbf{x}'$*, the following*

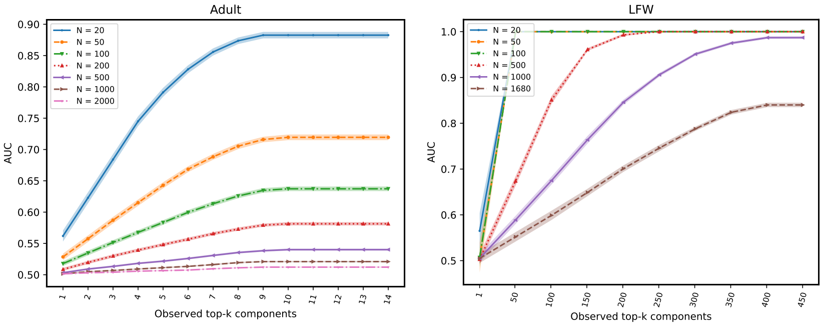


Fig. 1: Impact of the sample size N and the observed top-k components on the attack's performance. Shaded areas show 95% confidence intervals for the mean.

equality holds:

$$d(\mathbf{x}, \mathbf{x}') = 1,$$

where d denotes the Hamming distance.

Definition 2 ((ε, δ)-Differential privacy [5]). *A randomized mechanism $\mathcal{M}$ on a query function f satisfies ε-DP with $\varepsilon, \delta \geqslant 0$ if, for all pairs of neighbor databases $\mathbf{x}, \mathbf{x}'$ and for all $\mathcal{O} \subseteq \mathrm{range}(\mathcal{M})$,*

$$\mathrm{P}\{\mathcal{M}(f(\mathbf{x})) \in \mathcal{O}\} \leqslant e^{\varepsilon}\, \mathrm{P}\{\mathcal{M}(f(\mathbf{x}')) \in \mathcal{O}\} + \delta.$$

We say that $\mathcal{M}$ satisfies *pure* DP if $\delta = 0$, and *approximate* DP otherwise.

Definition 3 (L_p-global sensitivity [5]). *Let $\mathcal{D}$ be the class of possible data sets. The L_p-global sensitivity of a query function $f \colon \mathcal{D} \to \mathbb{R}^d$ is defined as*

$$\varDelta_p(f) = \max_{\forall \mathbf{x}, \mathbf{x}' \in \mathcal{D}} \|f(\mathbf{x}) - f(\mathbf{x}')\|_p,$$

where $\mathbf{x}, \mathbf{x}'$ are any two neighbor datasets.

Definition 4 (Laplace mechanism [5]). *Given any function $f \colon \mathcal{D} \to \mathbb{R}^d$, the Laplace mechanism mechanism is defined as follows:*

$$\mathcal{M}_L(\mathbf{x}, f(\cdot), \varepsilon) = f(\mathbf{x}) + (Y_1, \dots, Y_d),$$

where Y_i are i.i.d. random variables drawn from a Laplace distribution with zero mean and scale $\varDelta_1(f)/\varepsilon$.

Definition 5 (Gaussian mechanism [5]). *Given any function $f \colon \mathcal{D} \to \mathbb{R}^d$, the Gaussian mechanism mechanism is defined as follows:*

$$\mathcal{M}_G(\mathbf{x}, f(\cdot), \varepsilon) = f(\mathbf{x}) + (Y_1, \dots, Y_d),$$

where Y_i are i.i.d. random variables drawn from a Gaussian distribution with zero mean and standard deviation $\varDelta_2(f)\sqrt{2\log(1.25/\delta)}/\varepsilon$.

Theorem 1 **([5]).** *The Laplace mechanism satisfies* $(\varepsilon, 0)$*-DP.*

Theorem 2 **([5]).** *For any* $\varepsilon, \delta \in (0, 1)$*, the Gaussian mechanism satisfies* (ε, δ)*-DP.*

Theorem 3 **([5]).** *If each mechanism* $\mathcal{M}_i$ *in a* k*-fold adaptive composition* $\mathcal{M}_1, \ldots, \mathcal{M}_k$ *satisfies* (ε', δ')*-DP for* $\varepsilon', \delta' \geqslant 0$*, then the entire* k*-fold adaptive composition satisfies* $(\varepsilon, k\delta' + \delta)$*-DP for* $\delta \geqslant 0$ *and*

$$\varepsilon = \sqrt{2k \ln(1/\delta)}\varepsilon' + k\varepsilon'(e^{\varepsilon'} - 1). \tag{2}$$

5.2 Differentially private PCA approaches

As in the previous scenario where no privacy protection was implemented, the first step for the data curator is to compute the principal components of the covariance matrix A, which are to be shared with a trusted entity. However, to protect individual privacy against an adversary who may intercept some or all components of A, the curator now decides adding Laplace noise directly on the coefficients q_{ij} of A. In the context of DP, this approach is called *output perturbation.*

To protect the $\alpha \doteq d(d + 1)/2$ distinct[5] coefficients of A, we consider two strategies: (i) using a *joint* query function that simultaneously queries all such coefficients, and (ii) querying each coefficient *separately.* We shall refer to these procedures as *vector* and *scalar* queries, respectively.

For $i = 1, \ldots, d$, let attribute i take values in the interval $[l_i, u_i]$ after standardization, and denote by Λ_i the absolute difference $|l_i - u_i|$. Recall [17] that $\Delta_1(q_{ij}) = \Lambda_i \Lambda_j / N$, from which we can easily derive an upper bound on $\Delta_1(A)$ just by adding up the sensitivities of all distinct coefficients. Accordingly, the scale of the Laplace noise injected to each coefficient yields $\Delta_1(A)/\varepsilon$ in the vector case, and $\Delta_1(q_{ij})/\varepsilon_{ij}$ in the scalar case, where ε is the total privacy budget and ε_{ij} the fraction thereof assigned to the coefficient q_{ij}.

Using the standard sequential composition property, we can compute the total privacy cost of the scalar strategy by adding up all ε_{ij} for $i \geqslant j$. In our experiments, in order to compare the two approaches for a same total privacy budget, we shall assume $\varepsilon_{ij} = \varepsilon/\alpha$. Note that, in this case, the noise scales will coincide only if $\sum_{i \geqslant j} \Lambda_i \Lambda_j = \alpha \Lambda_i \Lambda_j$.

We shall also consider a variation of the scalar case that relies on the advanced (sequential) composition property. Notice that even though this property is defined in the context of approximate DP, Theorem 3 also applies if the mechanisms being composed satisfy pure ε-DP. With advanced composition, however, the total privacy cost can be estimated more tightly (compared to the standard property) when the number of coefficients is significantly large. Said otherwise, for the same privacy budget ε (and small δ) and for large α, the scale of the noise

[5] Recall that A is a symmetric matrix.

introduced with advanced composition can be reduced notably with respect to that injected with standard sequential composition. The noise scale yields in this case $\Lambda_i \Lambda_j / N\varepsilon'$, where ε' satisfies Eq. (2) for $k = \alpha$ and a given total privacy budget ε, δ.

Finally, the fourth protection approach we shall use in our experimental evaluation guarantees approximate DP through the *Gaussian* mechanism. More specifically, the algorithm in [6] queries all coefficients of A simultaneously and estimates $\Delta_2(A)$ to be $1/N$; the sensitivity bound follows after normalizing D so that each row has at most unit l_2 norm. Accordingly, the scale of the noise added to each coefficient yields $\sqrt{2\log(1.25/\delta)}/N\varepsilon$. Table 1 summarizes the four protection mechanisms we shall evaluate in the next subsection.

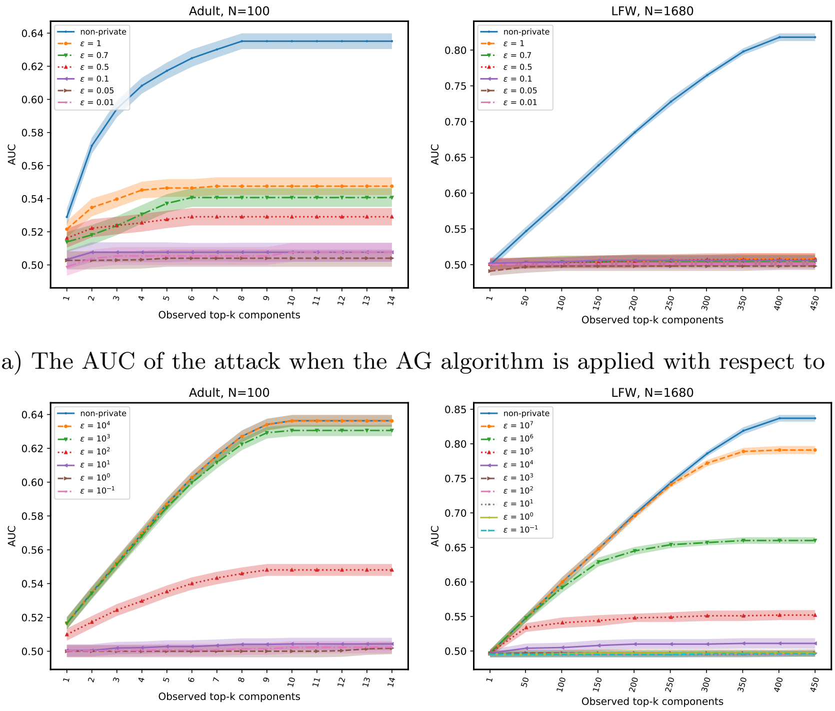
Approach	Privacy notion	noise scale
Laplace scalar query with naive composition	DP	$\alpha \Lambda_i \Lambda_j / N\varepsilon$
Laplace vector query	DP	$\sum_{i \geqslant j} \Lambda_i \Lambda_j / N\varepsilon$
Laplace scalar query with advanced composition	approx. DP	$\Lambda_i \Lambda_j / N\varepsilon'$
Analyze Gauss (AG) Algorithm[6]	approx. DP	$\sqrt{2\log(1.25/\delta)}/N\varepsilon$

Table 1: Overview of the DP mechanisms aimed to protect PCA against MIA. Here, ε denotes the *total* privacy budget and ε' the *fraction* thereof assigned to each coefficient of A.

5.3 Experimental Results

We first study the protection of DP mechanisms against our attack. Therefore, we implement the four aforementioned approaches and evaluate the AUC of the attack with various privacy budgets ε, ranging from 10^{-2} to 10^{8}. We would like to notice that this is not the usual range of values used in the literature. For example, in privacy-preserving data publishing, values of ϵ above 3 progressively seem to lose any meaningful guarantees [3]. However, for us, the fact that we will be using such large values is irrelevant, since we will empirically measure privacy leakage *not* through the ε itself, but through the effectiveness of an MIA. Finally, at the end of this section, we study the utility of the protected data provided by such approaches.

DP mechanisms and AUC Figure 2 shows the performance of the attack with respect to the k observed principal components when AG and Laplace vector query algorithms are used with various values of ε. In the case of the AG algorithm, ε varies from 0.01 to 1 and δ is set to $\delta = \frac{1}{N}$ whereas for the Laplace vector query algorithm, we select larger values of ε from 10^{-1} to 10^{7}. We also present the AUC of the attack in the non-private setting where DP-mechanisms are not adopted. Under AG, we observe that for all values of ε, the AUC of the attack is only marginally above 0.5 (random guess baseline). Hence, the AG algorithm mitigates the effectiveness of MIA. With larger ε values under the Laplace vector query approach, AUC starts to increase and gets closer to the non-private case. We also observe that for the Adult and Census datasets, for $\varepsilon = 10^2$ the Laplacian vector query approach provides roughly the same level of protection than AG for $\varepsilon = 1$. For the LFW dataset, for $\varepsilon = 10^4$ the Laplace vector query approach provides the same protection as AG with $\varepsilon = 1$. Hence, even with a higher privacy budget ε, the Laplace vector query approach limits the success of the attack.



(a) The AUC of the attack when the AG algorithm is applied with respect to k

(b) The AUC of the attack when the Laplace vector query algorithm is applied w.r.t. k.

Fig. 2: The AUC of the attack when the AG algorihtm(a) and the Laplace vector query approach (b) are applied with various values of ε. Shaded areas are the 95% confidence intervals for the mean.

Laplacian approaches Figure 3 compares the protection of the aforementioned Laplacian approaches for various levels of the total privacy budget based on the maximum AUC of the attack. We observe that the advanced composition approach achieves better protection than the naïve one in the low privacy regime (when ε is large). This observation can be explained through the noise scales injected by the two approaches. From Section. 5.2, it is easy to verify that the algorithm based on the advanced composition will introduce less noise than that relying on the naive composition when $\varepsilon' < \varepsilon/\alpha$. In Fig. 4, we plot in the hashed area the set of points (ε, α) where this inequality holds. From the figure, we can see that, for a fixed α, increasing ε will ultimately result in less noise for naive composition. And the other way round, for a fixed ε, increasing the number of coefficients will, at some point, make the advanced mechanism introduce less noise. We thus justify the observation above by assuming that adding more noise leads to stronger protection against the MIA.

Specifically, for the Adult ($d = 14$, $\alpha = 105$) and Census ($d = 13$, $\alpha = 91$) datasets, where the dimension is relatively small, the AUCs corresponding to the two different approaches intersect at $\varepsilon \approx 10^2$. As for LFW ($d = 450$, $\alpha \approx 10^5$) and MNIST ($d = 784$, $\alpha \approx 3 \times 10^5$), where d is large, the intersection occurs at the very low privacy regime at $\varepsilon \approx 10^5$. Furthermore, as depicted in the figure, the vector query and the scalar query with naive composition approaches achieve the same protection, because they consume the same total privacy budget ε.

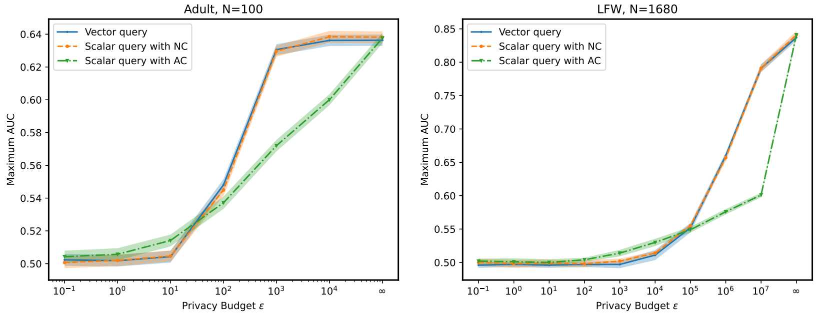


Fig. 3: Attack performance with Laplacian approaches when the adversary intercepts all components ($k = d$). The infinity point represents the non-private case

Trade-off between privacy and utility We use the total privacy budget ε as well as the AUC of an MIA to quantify privacy. As for utility, which refers to the accuracy of the principal components produced by the DP-PCA algorithms of Section 5.2, we adopt the metric introduced in [11]. In particular, we compute the percentage of captured energy of the principal components produced by those algorithms, $\hat{V}_k$, with respect to the principal components of non-private PCA (SVD), V_k. Accordingly, we measure utility as $q = \frac{tr(\hat{V}_k^T A \hat{V}_k)}{tr(V_k^T A V_k)}$, where A is the sample covariance matrix. We note that, for all the datasets, we select the reduced dimension k such that V_k have the captured energy of 90%.

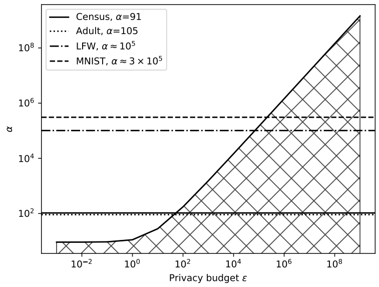


Fig. 4: The hashed shaded area shows where naive composition introduces less noise than advanced composition.

Figure 5 and 6 show the utility of the DP-PCA algorithms as a function of the privacy budget ε, and of the AUC, respectively. We observe that the AG algorithm offers good utility for the Adult and Census datasets. However, AG has a low utility for the other datasets. The Laplacian PCA solutions show lower utility in comparison with AG for $\varepsilon \leq 1$. The vector and scalar query with naive composition approaches show almost the same utility, except for the MNIST and Census datasets, where the scalar query with naive composition achieves better utility than the vector query approach. Advanced composition provides better utility than the naive composition where ε and α are in the blank area of Figure 4. In summary, the utility of the DP-PCA algorithms is influenced by the amount of noise added, as one would expect.

On the other hand, the vector query approach outperforms the scalar query approach if the sensitivity of the coefficients is skewed. In order to enjoy better utility, the scalar query approach with advanced composition should be used rather than with naive composition when the privacy budget ε and the number of queries α are in the blank area of Figure 4.

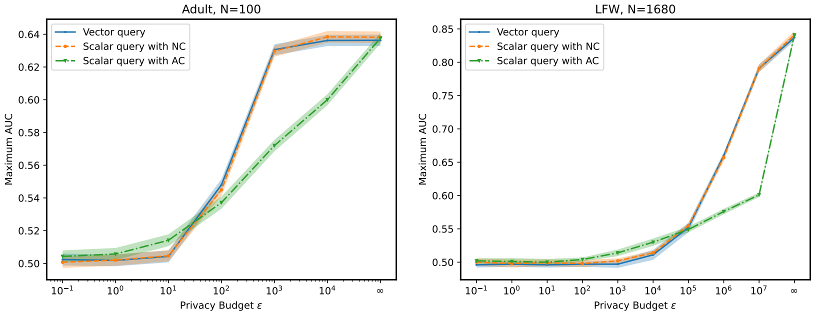


Fig. 5: Trade-off posed by the four DP-PCA algorithms described in Section. 5.2, between the total privacy budget ε and data utility. Utility is measured as the percentage of captured energy w.r.t. SVD.

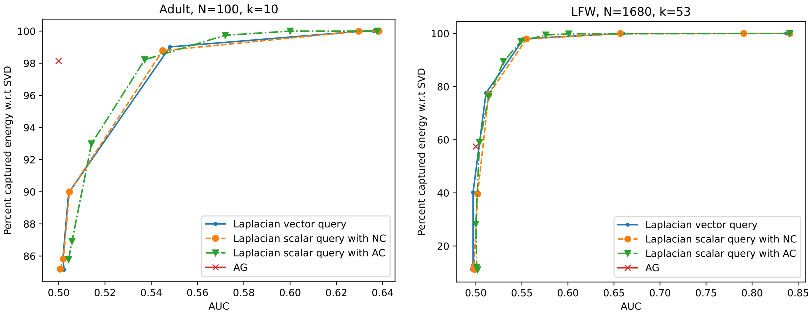


Fig. 6: Trade-off posed by the four DP-PCA algorithms described Section. 5.2, between attack performance and data utility. We measure attack performance through AUC, and utility through the percentage of captured energy w.r.t. SVD.

6 Conclusion

In this paper, we have implemented and evaluated the first membership inference attack against PCA, whereby an adversary has access to some or all principal components. Our attack sheds light on privacy leakage in PCA. Specifically, we have demonstrated that an MIA can be deployed successfully, with high performance, when the number of samples used by PCA is small. We have evaluated the protection of DP-PCA under different protection algorithms, privacy budgets, number of principal components intercepted, and number of covariance coefficients. Our work may be useful to assess the practical value of privacy when DP-PCA algorithms are employed along with the desired utility. For future work, to investigate whether there is a correlation between the vulnerable samples in PCA and the ones in the downstream tasks such as neural network classifiers.

Acknowledgments

This work has been supported by the MESRI-BMBF French-German joint project named PROPOLIS (ANR-20-CYAL-0004-01), the 3IA Côte d'Azur program (ANR19-P3IA-0002). J. Parra-Arnau is an Alexander von Humboldt postdoctoral fellow. The project that gave rise to these results received the support of a fellowship from ”la Caixa” Foundation (ID 100010434) and from the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No 847648. The fellowship code is LCF/BQ/PR20/11770009. This work was also supported by the Spanish Government under research project “Enhancing Communication Protocols with Machine Learning while Protecting Sensitive Data (COMPROMISE)” (PID2020-113795RB-C31/AEI/10.13039/501100011033).

References

[1] Martin Abadi et al. "Deep Learning with Differential Privacy". In: *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*. CCS '16. Vienna, Austria: Association for Computing Machinery, 2016, pp. 308–318. ISBN: 9781450341394. URL: https://doi.org/10.1145/2976749.2978318.

[2] Maria-Florina Balcan et al. "Improved Distributed Principal Component Analysis". In: *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2*. NIPS'14. Montreal, Canada: MIT Press, 2014, pp. 3113–3121.

[3] Alberto Blanco-Justicia et al. *A Critical Review on the Use (and Misuse) of Differential Privacy in Machine Learning*. 2022. URL: https://arxiv.org/abs/2206.04621.

[4] R. Brand, J. Domingo-Ferrer, and J.M. Mateo-Sanz. *Reference data sets to test and compare SDC methods for protection of numerical microdata*. Tech. rep. URL: https://research.cbs.nl/casc/CASCrefmicrodata.pdf.

[5] Cynthia Dwork and Aaron Roth. "The Algorithmic Foundations of Differential Privacy". In: 9.3–4 (Aug. 2014), pp. 211–407. ISSN: 1551-305X.

[6] Cynthia Dwork et al. "Analyze Gauss: Optimal Bounds for Privacy-Preserving Principal Component Analysis". In: *Proceedings of the Forty-Sixth Annual ACM Symposium on Theory of Computing*. STOC '14. New York, New York: Association for Computing Machinery, 2014, pp. 11–20. ISBN: 9781450327107. URL: https://doi.org/10.1145/2591796.2591883.

[7] Nils Homer et al. "Resolving Individuals Contributing Trace Amounts of DNA to Highly Complex Mixtures Using High-Density SNP Genotyping Microarrays". In: *PLoS Genetics* 4 (2008).

[8] Gary B. Huang et al. *Labeled Faces in the Wild: A Database for Studying Face Recognition in Unconstrained Environments*. Tech. rep. 07-49. University of Massachusetts, Amherst, Oct. 2007.

[9] A. Hundepool et al. "Statistical Disclosure Control". In: ed. by S. Fischer-Hübner et al. 2012.

[10] Hafiz Imtiaz and Anand D. Sarwate. "Differentially Private Distributed Principal Component Analysis". In: *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2018, pp. 2206–2210. DOI: 10.1109/ICASSP.2018.8462519.

[11] Hafiz Imtiaz and Anand D. Sarwate. "Symmetric matrix perturbation for differentially-private principal component analysis". In: *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. Mar. 2016, pp. 2339–2343. DOI: 10.1109/ICASSP.2016.7472095.

[12] Bargav Jayaraman and David E. Evans. "Evaluating Differentially Private Machine Learning in Practice". In: *USENIX Security Symposium*. 2019.

[13] Bargav Jayaraman et al. "Revisiting Membership Inference Under Realistic Assumptions". In: *Proceedings on Privacy Enhancing Technologies* 2021 (2021), pp. 348–368.

[14] Yann LeCun and Corinna Cortes. “MNIST handwritten digit database”. In: (2010). URL: http://yann.lecun.com/exdb/mnist/.

[15] Yunhui Long et al. “Understanding Membership Inferences on Well-Generalized Learning Models”. In: *ArXiv* abs/1802.04889 (2018).

[16] C.L. Blake D.J. Newman and C.J. Merz. *UCI Repository of machine learning databases*. 1998. URL: http://www.ics.uci.edu/~mlearn/MLRepository.html.

[17] Javier Parra-Arnau, Josep Domingo-Ferrer, and Jordi Soria-Comas. “Differentially private data publishing via cross-moment microaggregation”. In: *Information Fusion* 53 (2020), pp. 269–288. ISSN: 1566-2535.

[18] K. Pearson. “On Lines and Planes of Closest Fit to Systems of Points in Space”. In: *Philosophical Magazine* 2.11 (1901), pp. 559–572.

[19] Md.Atiqur Rahman et al. “Membership Inference Attack against Differentially Private Deep Learning Model”. In: *Trans. Data Priv.* 11 (2018), pp. 61–79.

[20] Alexandre Sablayrolles et al. “White-box vs Black-box: Bayes Optimal Strategies for Membership Inference”. In: *ICML*. 2019.

[21] Ahmed Salem et al. “ML-Leaks: Model and Data Independent Membership Inference Attacks and Defenses on Machine Learning Models”. In: *CoRR* abs/1806.01246 (2018). arXiv: 1806.01246. URL: http://arxiv.org/abs/1806.01246.

[22] Reza Shokri, Marco Stronati, and Vitaly Shmatikov. “Membership Inference Attacks against Machine Learning Models”. In: *CoRR* abs/1610.05820 (2016). arXiv: 1610.05820. URL: http://arxiv.org/abs/1610.05820.

[23] Liwei Song, R. Shokri, and Prateek Mittal. “Membership Inference Attacks Against Adversarially Robust Deep Learning Models”. In: *2019 IEEE Security and Privacy Workshops (SPW)* (2019), pp. 50–56.

[24] Florian Tramèr et al. “Truth Serum: Poisoning Machine Learning Models to Reveal Their Secrets”. In: *ArXiv* abs/2204.00032 (2022).

[25] Stacey Truex et al. “Demystifying Membership Inference Attacks in Machine Learning as a Service”. In: *IEEE Transactions on Services Computing* 14 (2021), pp. 2073–2089.

[26] Stacey Truex et al. “Effects of Differential Privacy and Data Skewness on Membership Inference Vulnerability”. In: *2019 First IEEE International Conference on Trust, Privacy and Security in Intelligent Systems and Applications (TPS-ISA)* (2019), pp. 82–91.

[27] Samuel Yeom et al. “Privacy Risk in Machine Learning: Analyzing the Connection to Overfitting”. In: *2018 IEEE 31st Computer Security Foundations Symposium (CSF)*. 2018, pp. 268–282.

[28] Zuobin Ying, Yun Zhang, and Ximeng Liu. “Privacy-Preserving in Defending against Membership Inference Attacks”. In: *Proceedings of the 2020 Workshop on Privacy-Preserving Machine Learning in Practice* (2020).

[29] Oualid Zari et al. “Membership inference attack against principal component analysis”. In: (2022). URL: https://www.eurecom.fr/index.php/en/publication/6913.