Project Report
"Differentially private covariance estimation"

Jean-Baptiste Astruc, Sami Kader - Yettefti César Denost
Université Paris Dauphine - PSL

Abstract

In this report, we analyze the work of Amin et al. (2019) in their article called "Differentially Private Covariance Estimate". In the latter, the authors propose a data release approach in order to produce a private covariance matrix. They do so through a composition of a Laplace mechanism for the estimation of the eigenvalues, and an Exponential mechanism for the estimation of eigenvectors. Their method outperforms pre-existing methods, especially in high privacy setups, which could be extremely useful in context such as social sciences or medical research.

1. Introduction

Privacy could be defined as "the state or condition of being alone, undisturbed, or free from public attention, as a matter of choice or right" (Oxford English Dictionary). As we advance into what has been described numerous times as the Digital Age, our physical identity is more and more duplicated in a virtual world. Not through form, but rather through information. As massive amounts of data are collected, and are used to train algorithms to better themselves for different use cases, there is an increasing need for the protection of such data. Differential Privacy (DP) is a standard framework for protecting privacy in data analysis (without being bullet-proof to any vector of attacks). One of its main characteristics concerns post-processing. In a formal way, we can see it as follows:

If $F(X)$ satisfies $(\epsilon) - DP$, then for any (deterministic or randomized) function g, we have that $g(F(X))$ satisfies $(\epsilon) - DP$

Put in simpler terms, it is impossible to reverse the privacy protection that has been provided by a DP mechanism simply by applying a form of post-processing of the data. It guarantees that no individual information could be leaked, even after preprocessing. In

general, the pre-existing algorithms that have been modified to be differentially private have been trained using these two main approaches:

- Adding noise directly to machine learning algorithms.
- Preprocessing data privately before analysis (data release).

This second approach enables the release of protected datasets while assuring individual confidentiality. This method is more flexible and also requires less access to the raw data (that might be sensitive). In this paper, this approach is chosen. Indeed, the authors propose a differentially private algorithm for estimating a covariance matrix while guaranteeing the confidentiality of the individuals' data.

Rather than using the raw data directly, the authors aim to build a private version of the matrix $C = XX^T$, which can be used for models such as linear regression, Principal Component Analysis (PCA), or clustering.

This approach falls within the framework of pure ϵ-differential privacy, which offers stricter guarantees than (ϵ, δ)-DP methods but is often more complex to implement and less efficient in practice.

The algorithm is based on the eigenvalue decomposition of C, treating separately:

- The eigenvalues (λ) estimated via the Laplace mechanism.
- The eigenvectors (v) selected using the Exponential mechanism, ensuring that they remain orthogonal to the previous ones.

This construction ensures the overall confidentiality of the algorithm through the composition of the mechanisms. The proposed algorithm reduces the reconstruction error compared to other methods, especially when privacy is strong (low ϵ). However, this algorithm is not always more performant when confidentiality is more relaxed (higher ϵ and δ), and the number of observations is large. In such cases, the Gaussian mechanism becomes more efficient, as the added noise is proportionally smaller and does not significantly impact the accuracy of the results. The authors therefore recommend a *hybrid* approach, utilizing simpler methods when confidentiality is less critical and more detailed algorithms when the data is sensitive or limited.

2. Literature Review

The authors operate a sort of literature review of the techniques used for the computation of DP-estimates covariance matrices.

They first talk about a simple naive approach consisting of adding Laplace noise to every element of the raw and original covariance matrix. While this approach seems conceptually simple to implement, it is impractical because the amount of noise increases linearly with the number of dimensions.

Another approach that seems to be better is to add some Gaussian noise, like it is done in Dwork et al (2014). The problem with this approach is that it is not $(\epsilon) - DP$, but only $(\epsilon, \delta) - DP$. This is of course a limit because, as said before, the $(\epsilon) - DP$ framework offers stricter privacy guarantees (even if the the former is generally used as a more practical solution).

The authors also touch upon a proposition made by Upadhyay (2018), in which the author conducts a sort of dimensionality reduction combined with DP-mechanism. More precisely, he exposes a private way of generating low dimensional representations of X. Nevertheless, this does not correspond to the same task that the authors are trying to achieve. Furthermore, Upadhyay (2018) only achieves $(\epsilon, \delta) - DP$.

Moreover, Chaudhuri et al (2012) also pro-

pose a way to compute a private version of the PCA. But this approach has a number of limitations leading to complex negative consequences as well as a possible breach of privacy. To be more precise, their approach only works for computing top eigenvectors, complexifying the process. Moreover, the sampling strategy is complex and requires a Gibbs sampler (which corresponds to a Markov Chain Monte Carlo algorithm used for sampling from probability distributions). Because we do not know when the sampler converges, adding noise can lead to a privacy breach.

Finally the method to which the authors proposed algorithm resembles the most is the one provided by Kapralov and Talwar (2013). In a nutshell, the latter computes a DP low-rank matrix approximation, in which they estimate the Singular Value Decomposition (SVD). The algorithm proposed by Kapralov and Talwar constructs a differentially private rank-1 approximation of a matrix C, subtracts this approximation from C, and then iterates the process on the remaining residual matrix. Similarly, the approach presented by the authors estimates the eigenvectors of C iteratively. However, instead of subtracting a rank-1 component at each step, their method repeatedly projects C onto the subspace orthogonal to the previously estimated eigenvectors. This projective update improves both theoretical error bounds and empirical performance. Furthermore, this approach enables the use of a simple rejection sampling technique, as introduced by Kent et al. (2018), to efficiently select eigenvectors while preserving differential privacy.

3. Setup and Tools

The authors show precisely the setup in which they are working as well as the tools they are using. They have a matrix $\mathbf{X}$ where each column represents a data point with d-dimensions. Their main goal is to produce a private estimate of the unnormalized and uncentered covariance matrix $C = \mathbf{X}\tilde{\mathbf{X}}^T$.

They start by defining $(\epsilon) - DP$, also known as pure differential privacy. Hence, given $\mathbf{X}$ and $\tilde{\mathbf{X}}$ two neighboring data matrices (meaning that they differ by at most one column), an algorithm $\mathcal{A}$ is $(\epsilon) - DP$ if for any set of possible outcomes S, we have:

$$Pr(\mathcal{A}(\mathbf{X}) \in S) \leq e^{\epsilon} Pr(\mathcal{A}(\tilde{\mathbf{X}}) \in S)$$

From there, the authors derive the composability characteristic (which is, in our case, adaptative), which can be defined as follows:

Given $\mathcal{A}_1 : \mathbb{R}^{d\times n} \to \mathcal{Y}_1$ is $(\epsilon_1) - DP$ and $\mathcal{A}_2 : \mathbb{R}^{d\times n} \times \mathcal{Y}_1 \to \mathcal{Y}_2$ is $(\epsilon_2 - DP)$. Then the composition $\mathcal{A}(\mathbf{X}) = \mathcal{A}_2(\mathbf{X}, \mathcal{A}_1(\mathbf{X}))$ is $(\epsilon_1 + \epsilon_2) - DP$.

This property will allow them to then use the two following mechanisms in their algorithm. First, for the estimation of the eigenvalues, they will use the Laplace mechanism. The latter for f can be defined as follows:

Given the L^1-sensitivity being $\Delta_f = \max_{\mathbf{X}\sim\tilde{\mathbf{X}}} \|f(\mathbf{X}) - f(\tilde{\mathbf{X}})\|$ and $Y_i \overset{\text{iid}}{\sim} Lap(0, \frac{\Delta_f}{\epsilon})$, $f(\mathbf{X}) + (Y_1, ..., Y_k)$ is $(\epsilon) - DP$, and for any $\beta > 0$, we have $\Pr\left(\max_i |Y_i| \geq \frac{\Delta_f}{\epsilon} \log \frac{k}{\beta}\right) \leq \beta$

Then, for the eigenvectors estimation, they will use the exponential mechanism. The latter can be defined as follows:

Given the sensitivity $\Delta_g = \max_{\mathbf{X}\sim\tilde{\mathbf{X}},y} |g(\mathbf{X}, y) - g(\tilde{\mathbf{X}}, y)|$, the exponential mechanism samples y from the density proportional to $f_{\exp}(y) = \exp\left(\frac{\epsilon}{2\Delta_g} g(\mathbf{X}, y)\right)$ with respect to the base measure μ. The exponential

mechanism preserves $(\epsilon - DP)$*. Let* $OPT = \max_y g(\mathbf{X}, y)$ *and* $G_\tau = \{y \in \mathcal{Y} : g(\mathbf{X}, y) \geq OPT - \tau\}$*. If* $\hat{y}$ *is the output of the exponential mechanism, we have* $\Pr(\hat{y} \notin G_{2\tau}) \leq \exp\left(-\frac{\epsilon\tau}{2\Delta_g}\right) \cdot \mu(G_\tau)$*.*

4. Algorithm

4.1. How does their approach and algorithm work?

The main goal of the proposed algorithm is to estimate a differentially private version of the covariance matrix $C = XX^T$ while preserving the structure of the data. Instead of directly adding noise to C, the approach involves decomposing it into its eigenvalues and eigenvectors, perturbing these components separately, and then reconstructing a private covariance matrix $\hat{C}$. This method ensures better accuracy compared to direct perturbation.

4.1.1. Algorithm Overview

The algorithm follows an iterative procedure:

1. Initialization The covariance matrix C is computed from the data, and the initial projection matrix is set as $P_1 = I_d$. Noise is then added to the eigenvalues using the Laplace mechanism:

$$\hat{\lambda}_i = \lambda_i + \text{Lap}\left(\frac{2}{\epsilon_0}\right) \tag{1}$$

where λ_i is the i-th eigenvalue of C.

2. Iterative Eigenvector Estimation and Projection For each iteration $i = 1, \ldots, d$ (or up to rank-k approximation):

(a) Eigenvector Selection using the Exponential Mechanism: A unit vector u_i is sampled from the unit sphere S^{d-i+1}, with probability density:

$$P(u_i) \propto \exp\left(\frac{\epsilon_i}{4} u_i^T C_i u_i\right) \tag{2}$$

This ensures that directions with high variance are more likely to be chosen while maintaining privacy.

(b) Mapping Back to the Original Space: The sampled vector u_i is projected back using the current projection matrix:

$$\hat{\theta}_i = P_i^T u_i \tag{3}$$

ensuring it remains in the original d-dimensional space.

(c) Updating the Projection: A new orthonormal basis is formed for the subspace orthogonal to $\{\hat{\theta}_1, \ldots, \hat{\theta}_i\}$. The updated covariance matrix is projected onto this space:

$$C_{i+1} = P_{i+1} C P_{i+1}^T \tag{4}$$

3. Reconstruction of the Private Covariance Matrix Once all eigenvalues and eigenvectors are estimated, the final private covariance matrix is reconstructed as:

$$\hat{C} = \sum_{i=1}^{d} \hat{\lambda}_i \hat{\theta}_i \hat{\theta}_i^T \tag{5}$$

This ensures usability in machine learning tasks while maintaining differential privacy.

4.2. How do they add noise?

Noise is introduced in a targeted manner to optimize the trade-off between privacy and accuracy.

4.2.1. 1. Laplace Mechanism for Eigenvalues

Eigenvalues are perturbed using the Laplace mechanism, as they have low sensitivity:

$$\hat{\lambda}_i = \lambda_i + \text{Lap}\left(\frac{2}{\epsilon_0}\right) \tag{6}$$

This ensures that small changes in individual data points do not significantly affect the eigenvalues.

4.2.2. 2. Exponential Mechanism for Eigenvectors

Eigenvectors are sampled probabilistically using the exponential mechanism:

$$P(u_i) \propto \exp\left(\frac{\epsilon_i}{4} u_i^T C u_i\right) \tag{7}$$

This mechanism ensures that the algorithm favors directions with high variance while preserving privacy.

4.2.3. 3. Optimized Allocation of the Privacy Budget

To minimize reconstruction error, the privacy budget is allocated as:

$$\epsilon_i = \frac{\epsilon}{2} \frac{\sqrt{\hat{\lambda}_i + \tau}}{\sum_j \sqrt{\hat{\lambda}_j + \tau}} \tag{8}$$

where τ is a correction factor. Larger eigenvalues receive more privacy budget, as they contribute more significantly to the covariance structure.

4.3. Summary

Instead of perturbing C directly, the algorithm decomposes it into eigenvalues and eigenvectors. Laplace noise is added to the eigenvalues to maintain privacy. Eigenvectors are sampled using the exponential mechanism to balance accuracy and privacy. The privacy budget is allocated strategically to minimize reconstruction error. The final covariance matrix $\hat{C}$ retains statistical utility while ensuring privacy. This approach improves upon previous methods by preserving the fundamental structure of the data while adhering to strong differential privacy guarantees.

5. Experiment

In this section, we replicate the experimental evaluation of **Amin2019** using the same three datasets from the UCI repository: Wine, Adult, and Airfoil. Our implementation follows the paper's iterative eigenvalue/eigenvector sampling algorithm for differentially private covariance estimation (see Algorithm 1 in **Amin2019**). In addition to the proposed method (denoted as "DP (Iterative)"), we compare two baselines:

- **Laplace Mechanism:** Adds independent Laplace noise to every entry of the covariance matrix C.
- **Gaussian Mechanism:** Adds Gaussian noise, yielding (ϵ, δ)-DP rather than pure ϵ-DP.

5.1. Implementation and Datasets

Code Overview. We load the datasets in Python, normalize each data point (so that each column of X has $\|x_j\|_2 \leq 1$), and compute the true covariance matrix $C = XX^\top$. We then run:

(i) **Iterative Eigenvector Sampling (DP):**
- Split ϵ into $\epsilon_0 = \epsilon/2$ for eigenvalues (Laplace mechanism) and ϵ_i for eigenvectors (Exponential mechanism).
- Estimate the eigenvalues with Laplace noise.
- Iteratively select eigenvectors using the exponential mechanism, updating the projection to remain orthogonal to previously chosen directions.
- Reconstruct the private covariance matrix $\widehat{C} = \sum_{i=1}^{d} \widehat{\lambda}_i \, \widehat{\theta}_i \, \widehat{\theta}_i^\top$.

(ii) **Laplace Mechanism (L):** Perturb C directly by adding Laplace$\left(\frac{2d}{\epsilon}\right)$ noise to each entry.

(iii) **Gaussian Mechanism (G):** Perturb C with Gaussian noise, using standard deviation $\sigma \propto d^{1.5}\sqrt{\log(1/\delta)}/\epsilon$.

Datasets.

- **Wine:** $\dim(X) = (13, 178)$. Each of the 178 data points has 13 numerical features.
- **Adult:** $\dim(X) = (97, 45222)$. After one-hot encoding of categorical features, each of the 45,222 data points is embedded in 97 dimensions.
- **Airfoil:** $\dim(X) = (5, 1503)$. We use the first 5 features from the airfoil self-noise dataset, yielding 1503 data points.

In all cases, the columns of X are normalized so that each data point has ℓ_2-norm at most 1.

5.2. Results

We repeat each experiment for a range of privacy parameters $\epsilon \in \{0.01, 0.1, 0.2, 0.5, 1.0, 2.0, 4.0\}$ and plot the average **Normalized Frobenius Error**:

$$\frac{\|\widehat{C} - C\|_F}{n},$$

where n is the number of columns (data points). Each data point in the plots below is an average over multiple independent trials (e.g., 20). Figures **??**, **??**, and **??** show the performance on the Wine, Adult, and Airfoil datasets respectively.

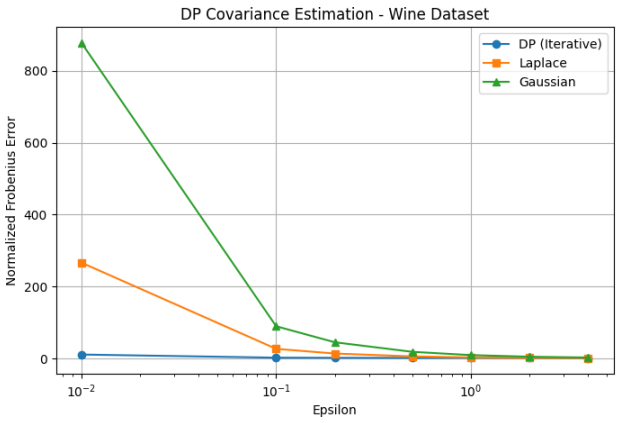

Figure 1. DP Covariance Estimation — Wine Dataset ($d = 13$, $n = 178$).

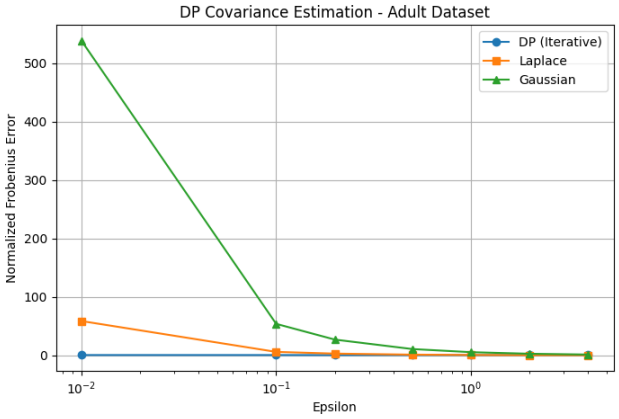

Figure 2. DP Covariance Estimation — Adult Dataset ($d = 97$, $n = 45222$).

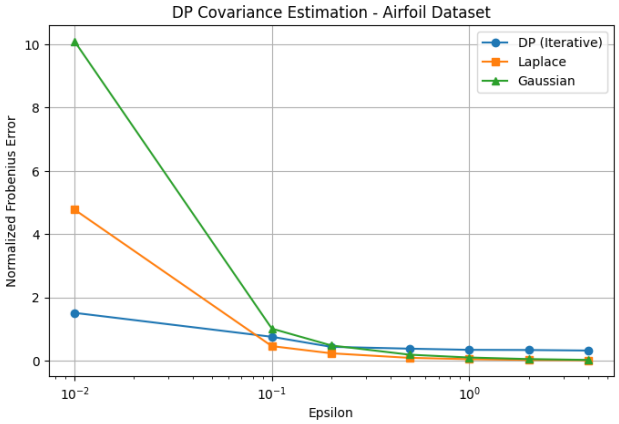

Figure 3. DP Covariance Estimation — Airfoil Dataset ($d = 5$, $n = 1503$).

5.3. Discussion

Overall, we observe that:

- **Iterative (DP) vs. Laplace:** The iterative algorithm consistently yields lower error than simple Laplace perturbation, especially at higher privacy levels (smaller ϵ).
- **Iterative (DP) vs. Gaussian:** Despite Gaussian noise achieving (ϵ, δ)-DP, the proposed method frequently outperforms Gaussian for moderate or small ϵ. This aligns with the paper's claim that their method excels in high-privacy regimes.
- **Dataset Dependence:** The difference in shapes $\dim(X)$ influences the noise scale.

For instance, Adult is higher-dimensional ($d = 97$) than Wine ($d = 13$) or Airfoil ($d = 5$), which can lead to larger noise magnitudes for the same ϵ. Yet the iterative approach maintains relatively lower error even as dimension grows.

These results closely replicate the findings of Amin et al. (2019), confirming that their proposed algorithm offers a strong privacy–utility trade-off compared to simpler output perturbation methods. The method's advantage is especially pronounced for small ϵ (where privacy demands are greatest). For less stringent privacy regimes (larger ϵ), the difference narrows, though the iterative method generally remains competitive.

For further informations, you can consult our GitHub repository at the following address : mydarkbluehttps://github.com/kaderrsami/diff_priv/tree/main.

6. Conclusion

The new algorithm provides a novel way of estimating the covariance matrix under differential privacy. Through the use of eigen decomposition, the algorithm minimizes noise distortion and thus improves accuracy compared to the use of simple perturbation techniques. The Laplace mechanism is applied to eigenvalues for noise addition with controlled noise, and the exponential mechanism selects eigenvectors in a way that preserves their statistical properties. In addition, the iterative projection methodology enhances stability by ensuring orthogonality among the projected eigenvectors, thus offering a more accurate and systematic reconstruction of the covariance matrix.

This approach has explicit implications for data analysis with privacy guarantees. Provides an efficient framework for statistical learning algorithms such as principal component analysis and linear regression, without compromising privacy guarantees. The use of structured noise is also less redundant in information loss and thus an appropriate choice compared to noise injection.

Although with these advantages, the method has some drawbacks. Iterative computation of eigenvectors and projection of the covariance matrix can be computationally intensive, especially for high-dimensional data. This can pose practical challenges to large-scale applications. Also, the usefulness of reconstructed covariance matrix relies on proper allocation of the privacy budget. Improper allocation can result in excessive noise on significant components and result in lost utility.

Future work can explore budget allocation mechanisms that optimally balance utility and privacy more. Other approaches that reduce computational overhead without diminishing privacy guarantees can be explored as well. Another line of research is to apply this approach to adaptive privacy-preserving mechanisms, where noise scales are adjusted dynamically based on dataset characteristics.

Overall, while the given algorithm certainly improves differentially private covariance estimation by a large margin, careful handling of computation complexity and privacy budget allocation still plays a critical role in making it useful at large scales.