Analyze Gauss: Optimal Bounds for Privacy-Preserving Principal Component Analysis

Cynthia Dwork, Kunal Talwar, Abhradeep Thakurta, Li Zhang

Published 2014 in STOC

https://docs.google.com/presentation/d/118XzZlBe1MyouyXHRcEpI-_vDVOkFo_kfCVdQEdKYzo/edit?usp=sharing

Presenter:Weiting Zhan

Outline

1. Goal:

Utility guarantee  Privacy guarantee

2. Method: Gaussian noise

Principal Component Analysis (PCA)

3.Evaluation method: Perturbed leader algorithm

Regret guarantee

Goal vs result

utility guarantee : to compute a subspace that captures the covariance of A as much as possible.

Theorem 13 (Subspace convergence in spectral norm). *Let $\sigma_1 \geq \cdots \geq \sigma_n$ be the singular values of the data matrix A. Under the randomness of the algorithm, with probability at least $1 - 2\delta$, Algorithm 2 outputs a k-dimensional subspace $\widehat{V}_k$ such that*

$$\left\| V_k V_k^T - \widehat{V}_k \widehat{V}_k^T \right\|_2 = O\left(\frac{\Delta_{\epsilon,\delta}\sqrt{n}}{\sigma_k^2 - \sigma_{k+1}^2 - \log(1/\delta)/\epsilon} \right).$$

privacy guarantee : to learn "about" A without compromising the privacy of any individual.

(2ε, 2δ)-*differentially private.*

Regret guarantee: Gaussian noise for the regularization noise,

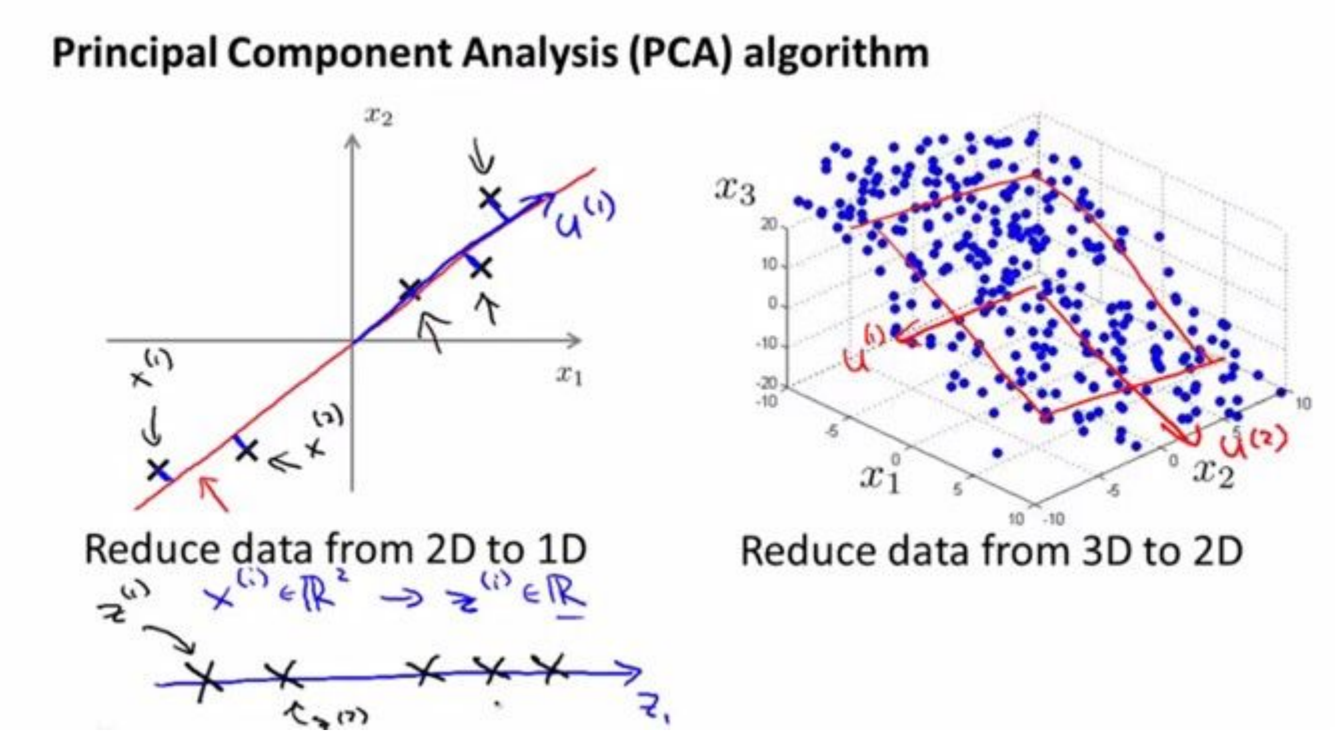
Online problem regret bound:

$$\widetilde{O}(\sqrt{k\mathsf{OPT}}n^{1/4})$$

Offline problem regret bound:

$$\widetilde{O}(\sqrt{k\mathsf{OPT}}n^{1/4})$$

Principal Component Analysis:reduce Data dimension

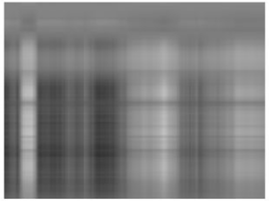


Gaussian Noise + Principal Component Analysis

In this example, Gaussian noise and PCA is applied in the compression of 512-by-512 grayscale image. The image is represented by a matrix $X \in \mathbb{R}^{512\times512}$.

(a) Input original image

(b) Gaussian noise image $\sigma = 20$



(a) 1 principal component

(b) 5 principal component

(c) 9 principal component

(d) 13 principal component

(e) 17 principal component

(f) 21 principal component

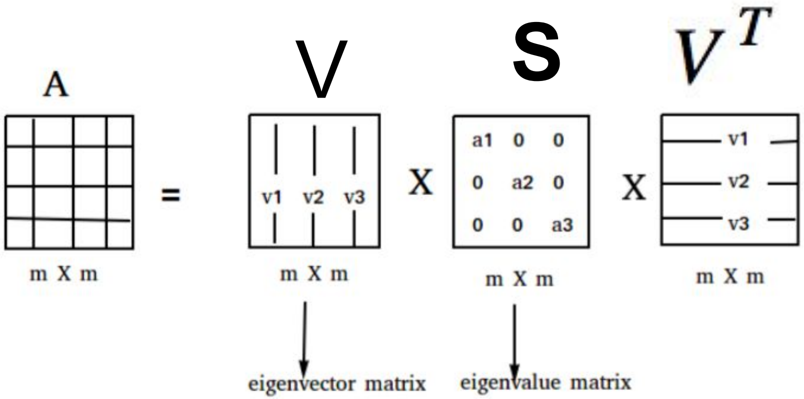
Trade-off between utility and privacy

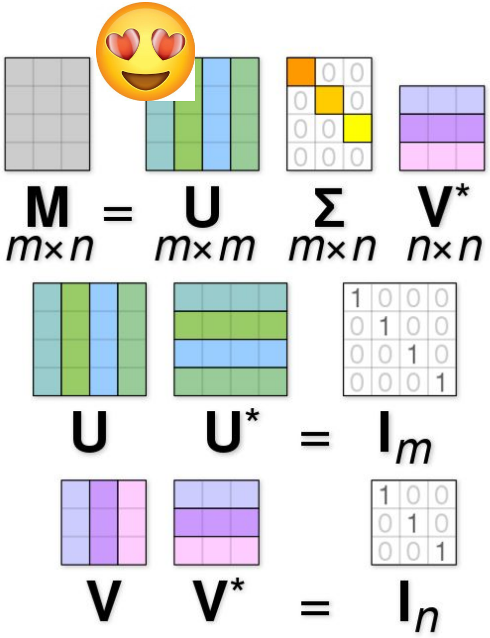
(g) 25 principal component

(h) 29 principal component

S. Voloshynovskiy, O. Koval, and T. Pun, "Image denoising based on the edge-process model", *Signal Processing*, vol. 85, iss. 10, pp. 1950-1969, 2005.
https://www.projectrhea.org/rhea/index.php/PCA_Theory_Examples

Eigenvalue decomposition and Singular value decomposition





$$A^T A = V S V^T$$

$$A = U S V^T$$

$$U = A V S^T$$

Algorithm 1: The Gaussian Mechanism

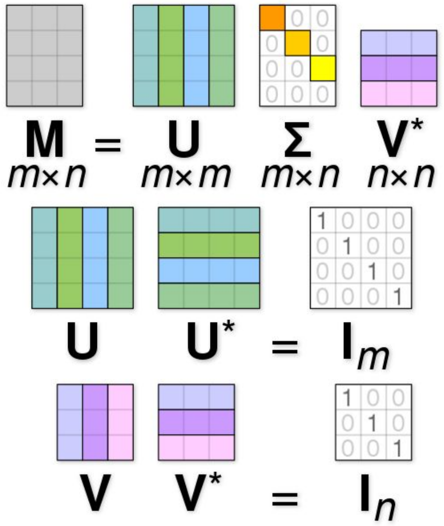
Algorithm 1 The Gaussian Mechanism: releasing the covariance matrix privately

Input: matrix $A \in \Re^{m \times n}$, and privacy parameters $\epsilon, \delta > 0$.

1: $E \in \Re^{n \times n}$ be a symmetric matrix where the upper triangle (including the diagonal) is i.i.d. samples from $\mathcal{N}\left(0, {\Delta_{\epsilon,\delta}}^2\right)$, and each lower triangle entry is copied from its upper triangle counterpart.

2: Output $\widehat{C} \leftarrow A^T A + E$.

Algorithm 2: Private Subspace Recovery



Algorithm 2 Private Subspace Recovery

Input: matrix: $A \in \Re^{m \times n}$, rank parameter: k, and privacy parameters: $\epsilon, \delta > 0$.

1: $V \Sigma V^T \leftarrow$ Eigenvalue decomposition of $A^T A$. Let $|\lambda_1| \geq \cdots \geq |\lambda_n|$ be the eigenvalues.

2: $\hat{d} \leftarrow (|\lambda_k| - |\lambda_{k+1}|) + \mathrm{Lap}\left(\frac{2}{\epsilon}\right)$.

3: $V_k \leftarrow$ Top k eigenvectors of $A^T A$ (as a column matrix).

4: $\hat{W} \leftarrow V_k V_k^T + E$, where $E \in \Re^{n \times n}$ is a symmetric matrix where the upper triangle is i.i.d. sample from $\mathcal{N}\left(0, \frac{\Delta_{\epsilon,\delta}{}^2}{(\hat{d} - 2(1+\log(1/\delta)/\epsilon))^2}\right)$, where $\Delta_{\epsilon,\delta} = \frac{1+\sqrt{2\log(1/\delta)}}{\epsilon}$.

5: Let $\widehat{V} \hat{\Sigma} \widehat{V}^T$ be the eigenvalue decomposition of $\hat{W}$ and let $\widehat{V}_k$ be the top k eigenvectors of $\widehat{V}$ (as a row matrix). Output $\widehat{V}_k \widehat{V}_k^T$.

Proof

Theorem 4 (Worst case utility guarantee). *Let V_k be the principal rank-k right singular subspace of A and let $\widehat{V}_k$ be the principal rank-k subspace of the matrix $\widehat{C}$ (output by Algorithm 1). Then with high probability,*

$$\|A\widehat{V}_k\|_F^2 \geq \|AV_k\|_F^2 - O\left(k\sqrt{n}\Delta_{\epsilon,\delta}\right) .$$

Eigenvalue Properties

- $|A| = \prod_{i=1}^{n} \lambda_i$
- The rank of a matrix is equal to the number of its non-zero eigenvalues
- Eigenvalues of a diagonal matrix, are simply the diagonal entries
- A matrix is said to be diagonalizable if we can write

$$A = X\Lambda X^{-1}$$

iTutor

Trace of a Matrix

- The *trace* of an n x n matrix $\boldsymbol{A}$ is defined to be the sum of the elements on the main diagonal of $\boldsymbol{A}$:

  trace($\mathbf{A}$) = tr ($\mathbf{A}$) = $\Sigma_i\, a_{ii}$.

  where a_{ii} is the entry on the ith row and ith column of A.

- Properties:
  - $tr(\boldsymbol{A} + \boldsymbol{B}) = tr(\boldsymbol{A}) + tr(\boldsymbol{B})$
  - $tr(c\boldsymbol{A}) = c\; tr(\boldsymbol{A})$
  - $tr(\boldsymbol{AB}) = tr(\boldsymbol{BA})$
  - $tr(\boldsymbol{ABC}) = tr(\boldsymbol{CAB})$ (*invariant* under *cyclic* permutations.)
  - $tr(\boldsymbol{A}) = tr(\boldsymbol{A}^T)$
  - d $tr(\boldsymbol{A}) = tr(\mathrm{d}\boldsymbol{A})$ (differential of trace)
  - $tr(\boldsymbol{A}) = rank(\boldsymbol{A})$ when $\boldsymbol{A}$ is idempotent –i.e., $\boldsymbol{A} = \boldsymbol{A}^2$.

Theorem 4 (Worst case utility guarantee). *Let V_k be the principal rank-k right singular subspace of A and let $\widehat{V}_k$ be the principal rank-k subspace of the matrix $\widehat{C}$ (output by Algorithm 1). Then with high probability,*

$$\|A\widehat{V}_k\|_F^2 \geq \|AV_k\|_F^2 - O\left(k\sqrt{n}\Delta_{\epsilon,\delta}\right) .$$

Proof. We have the following with the noise matrix E in Algorithm 1.

$$\begin{aligned} \mathsf{tr}(V_kV_k^T(A^TA+E)) &= \mathsf{tr}(V_kV_k^T(A^TA)) + \sum_{i=1}^{k} v_iEv_i^T \\ &\geq \sum_{i=1}^{k}\sigma_i^2 - k\|E\|_2. \end{aligned} \tag{1}$$

By definition, the highest singular subspace captures the maximum variance. Therefore,

$$\mathsf{tr}(\widehat{V}_k^T(A^TA+E)\widehat{V}_k) \geq \mathsf{tr}(V_k^T(A^TA+E)V_k). \tag{2}$$

Combining (1) and (2), we get the following.

$$\begin{aligned} &\mathsf{tr}(\widehat{V}_k^T(A^TA+E)\widehat{V}_k) \geq \sum_{i=1}^{k}\sigma_i^2 - k\|E\|_2 \\ \Leftrightarrow\ &\mathsf{tr}(\widehat{V}_k^T(A^TA)\widehat{V}_k) \geq \sum_{i=1}^{k}\sigma_i^2 - k\|E\|_2 - \mathsf{tr}(\widehat{V}_k^TE\widehat{V}_k) \\ \Rightarrow\ &\mathsf{tr}(\widehat{V}_k^T(A^TA)\widehat{V}_k) \geq \sum_{i=1}^{k}\sigma_i^2 - 2k\|E\|_2 \end{aligned} \tag{3}$$

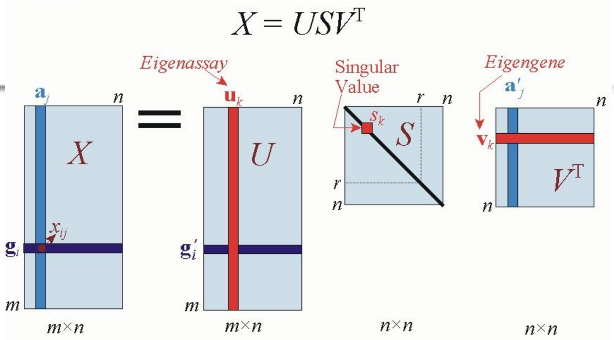
Since E is a symmetric Gaussian ensemble, by Corollary 2.3.6 from [42], with probability at least $1 - negl(n)$, $\|E\|_2 = O\left(\sqrt{n}\Delta_{\epsilon,\delta}\right)$. This completes the proof. ☐

Algorithm 3: Online singular subspace computation

Algorithm 3 Online singular subspace computation.

Input: Vectors $a_1, \ldots, a_m \in \Re^n$ where $\|a_t\| \le 1$, rank parameter: k, regularization parameter: ϵ, δ.
Output: k-dimensional subspaces $\widehat{V}_1, \ldots, \widehat{V}_m$.

1: Choose an arbitrary rank k subspace $\widehat{V}_1$.
2: **for** $t \leftarrow 1$ to m **do**
3: Get a *reward* $R_t = \|\widehat{V}_t^T a_t\|_2^2 = \mathrm{tr}(a_t^T \widehat{V}_t \widehat{V}_t^T a_t)$ and receive input a_t.
4: Compute $C_t = \sum_{\tau=1}^{t} a_\tau a_\tau^T$
5: Compute $\widehat{C}_t = C_t + E_t$, where E_t is sampled as in Algorithm 1 using the parameters ϵ, δ.
6: Compute $\widehat{V}_{t+1}$ as the top k singular subspace of $\widehat{C}_t$.
7: **end for**



http://public.lanl.gov/mewall/kluwer2002.html