Symmetric Matrix Perturbation For Differentially-Private Principal Component Analysis

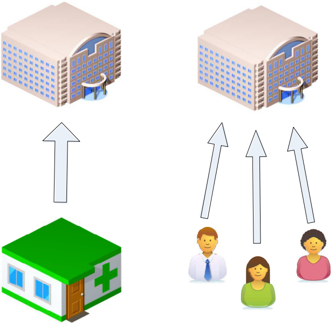
Hafiz Imtiaz & Anand D. Sarwate

Department of Electrical and Computer Engineering
Rutgers, the State University of New Jersey

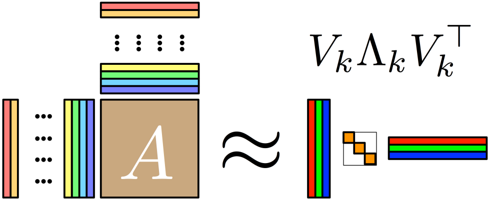
March 25, 2016

Outline

Why learn from private data?

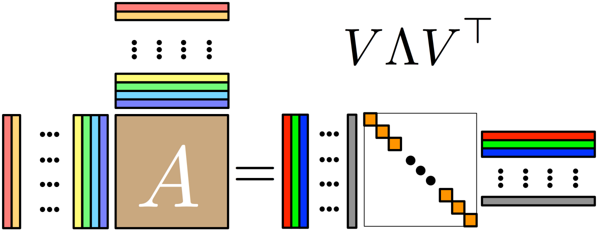


- Much of private/sensitive data is being digitized
- Using/reusing data - learn about populations
- Free and open sharing - ethical, legal, and technological obstacles



Principal Component Analysis

The PCA problem

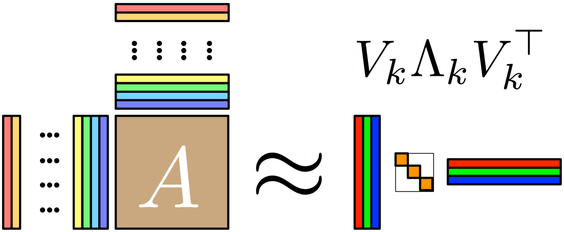


Data matrix: $X = [x_1 \ x_2 \ \ldots \ x_n]$, samples are in columns
Second-moment matrix $A = XX^\top$.
We can decompose A as

$$A = V\Lambda V^\top$$

where $\Lambda = \mathrm{diag}(\lambda_1, \lambda_2, \ldots, \lambda_n)$ and $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_n$

The PCA problem

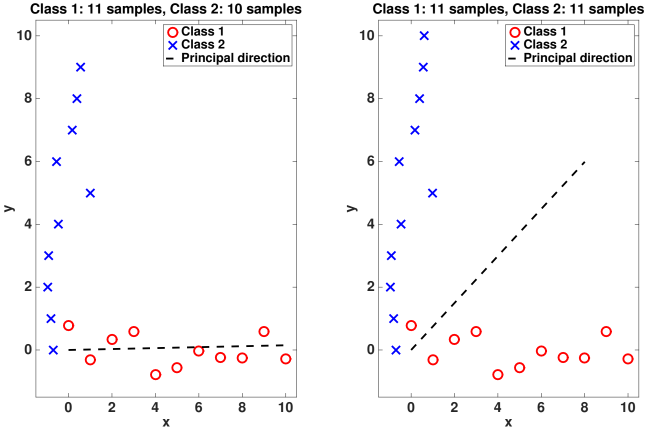


The rank-k approximation of A:

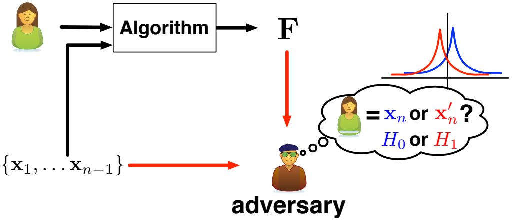
$$A_k = V_k \Lambda_k V_k^\top$$

The top-k PCA subspace is the span of the corresponding columns of V.

Why we need privacy in PCA?

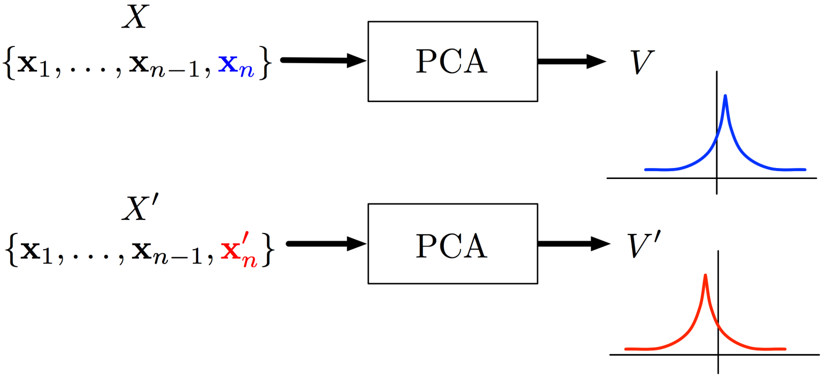


Changing one sample can significantly change the principal direction



Differential Privacy

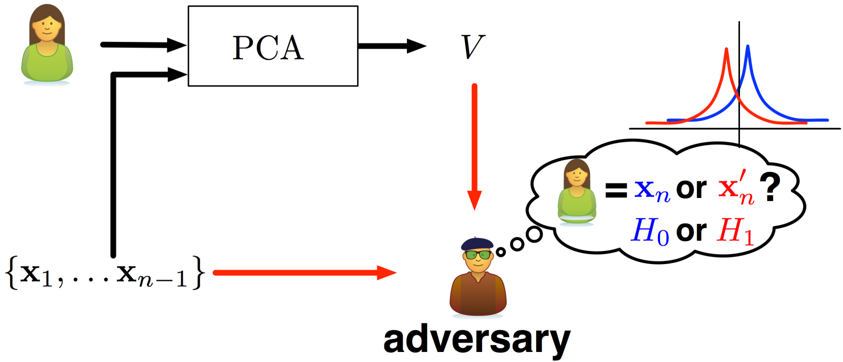
Differential privacy: a definition



[Dwork et al. 2006] An algorithm $\mathcal{A}$ is (ε, δ)-differentially private if for any set of outputs $\mathcal{F}$, and all $(\mathcal{D}, \mathcal{D}')$ differing in a single point,

$$\mathbb{P}\left(\mathcal{A}(\mathcal{D}) \in \mathcal{F}\right) \leq \exp(\varepsilon) \cdot \mathbb{P}\left(\mathcal{A}(\mathcal{D}') \in \mathcal{F}\right) + \delta$$

Differential privacy: hypothesis testing



$$\log \frac{\mathbb{P}\left(\mathcal{A}(\mathcal{D}) \in \mathcal{F}\right)}{\mathbb{P}\left(\mathcal{A}(\mathcal{D}') \in \mathcal{F}\right)} \leq \varepsilon$$

Privacy-utility tradeoff

Tradeoff between privacy and utility. With more data:

- Stronger evidence for structure $\rightarrow$ more accuracy/utility
- Less dependence on individuals $\rightarrow$ less privacy risk
- How much data do we need?
- What is the tradeoff in practice?

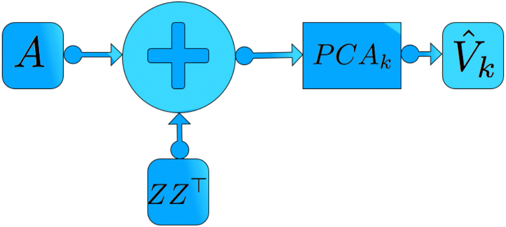
Differentially-private PCA Algorithms

Several algorithms are available:

- (ϵ, δ): Analyze Gauss [Dwork et al. 2014]
- (ϵ, δ): Private Power Method [Hardt et al. 2014]
- $(\epsilon, 0)$: PPCA [Chaudhuri et. al. 2013, McSherry et. al. 2007]
- $(\epsilon, 0)$: Proposed Symmetric Noise (SN) algorithm
- (ϵ, δ): Wishart noise [Sheffet 2015] (linear regression)
- $(\epsilon, 0)$: Wishart noise [Jiang 2016] (in a parallel effort)

	AG	**PPM**	**PPCA**	**SN**
Estimates $\hat{A}$	✓	✗	✗	✓
$\hat{A}$ PSD	✗	–	–	✓
$\delta > 0$	✓	✓	✗	✗
$\delta = 0$	✗	✓	✓	✓

Table: Comparison of Algorithms



Proposed SN **algorithm**

Proposed SN Algorithm: Wishart noise addition

Input: $d \times n$ data matrix X, privacy parameter ϵ, dimension k.

1. Compute $A = XX^\top$.
2. Generate $d \times p$ matrix $Z = [z_1, z_2, \ldots, z_p]$ where $z_i \sim \mathcal{N}(0, \frac{1}{2\epsilon}I)$ and $p = d + 1$.

Output: $\hat{A} = A + ZZ^\top$. Set $\hat{V}_k$ using PCA on $\hat{A}$.

Remark: Adding wishart noise preserves the PSD structure of A, which is not the case for AG [Dwork et al. 2014]

Analysis of SN algorithm

Privacy of SN Algorithm

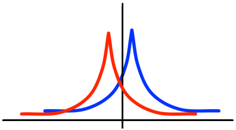
- z_i are iid $\sim \mathcal{N}(0, \frac{1}{2\epsilon} I_d)$ where $\{z_i : i = 1, 2, \ldots, d+1\}$
- $Z = [z_1, z_2, \ldots, z_p]$
- The positive semidefinite $E = ZZ^\top$ is distributed $\sim$ *Wishart* $W_d(\Sigma, p)$ where $\Sigma = \frac{1}{2\epsilon} I_d$ and $p = d+1$

$$
\begin{aligned}
f_E(E) &\propto (\det(E))^{\frac{p-d-1}{2}} \exp\left(-\frac{1}{2}\operatorname{tr}\left(\Sigma^{-1}E\right)\right) \\
&\propto \exp\left(-\epsilon \operatorname{tr}(E)\right)
\end{aligned}
$$

Privacy of SN Algorithm

- Two neighboring databases with A and A', an output Y from SN.
- Data samples satisfy $\|x_i\|_2 \leq 1$ and therefore, $\|A - A'\|_2 \leq 1$.

$$\frac{f_E(Y - A)}{f_E(Y - A')} = \frac{\exp\left(-\epsilon\, \mathrm{tr}(Y - A)\right)}{\exp\left(-\epsilon\, \mathrm{tr}(Y - A')\right)} \leq \exp\left(\epsilon\right).$$



Empirical performance of SN algorithm

What do we mean by performance?

The *performance* can be different in different applications:

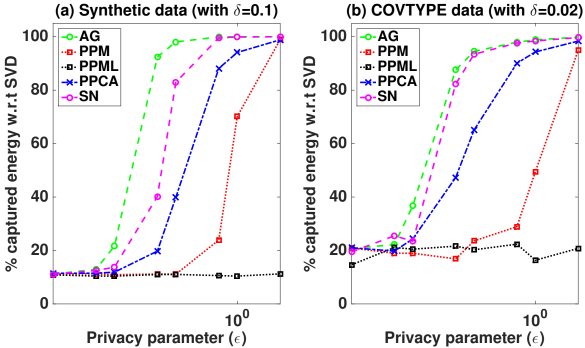
- captured energy of A in the private subspace
- classification performance of projected data on $\hat{V}_k$
- difference between the A and $\hat{A}$

$$\text{Percentage of captured energy w.r.t SVD} = \frac{\operatorname{tr}(\hat{V}_k^\top A \hat{V}_k)}{\operatorname{tr}(V_k^\top A V_k)} \times 100$$

Datasets used

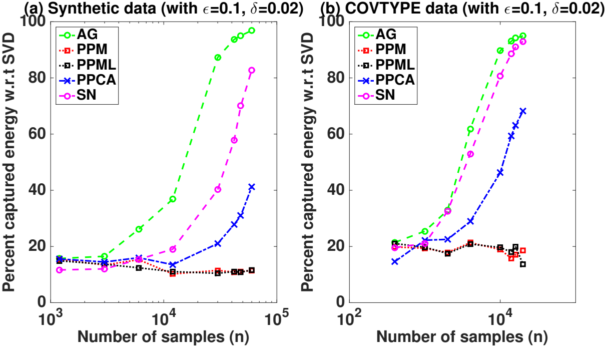
- *Synthetic data* set ($d = 100$, $n = 60000$, $k = 10$) was generated with a pre-determined covariance matrix
- The *Covertype* dataset ($d = 54$, $k = 10$) contains Forest CoverTypes - was collected by Department of Forest Sciences of Colorado State University. Has 5,81,012 samples.
- The *MNIST* ($d = 784$, $k = 50$) - database of handwritten digits. Has 60,000 training and 10,000 testing samples

Dependence on privacy parameter ϵ



- AG, PPM and SN - standard deviation of noise is inversely proportional to ϵ
- Smaller ϵ means more noise and lower privacy risk.
- For PPCA, an increase in ϵ means skewing the probability density function more towards the optimal subspace.

Dependence on number of samples n



- Intuitively, it should be easier to guarantee smaller privacy risk ϵ and higher utility $q(\cdot)$ when the number of samples is large.

Classification

- We projected the d-dimensional data samples onto the private k-dimensional subspace $\hat{V}_k$.

Table: Percentage error in classification

	Synthetic		COVTYPE		MNIST	
	70%	50%	70%	50%	70%	50%
SVD	6.63	6.34	0.08	0.08	0.61	0.32
AG	6.58	6.32	1.08	0.85	2.72	2.38
PPM	10.48	10.06	2.05	1.26	2.67	2.48
PPCA	7.43	7.21	5.21	4.85	3.16	2.91
SN	7.99	7.48	0.05	0.05	2.22	2.09

Some concluding remarks

Conclusions

- The AG and the SN - best performance among (ϵ, δ) and $(\epsilon, 0)$-private methods, respectively.
- In some regimes SN achieved as much utility as AG, even though SN provides stricter privacy guarantee.
- When there's a large *eigengap* - SN provided a very good approx. to $V_k(A)$
- Also, SN provided a very good approx. to A_k
- We found, [Sheffet 2015] and [Jiang 2016] outperform PPM and PPCA, but did not have empirical utility better than that of SN.
- Results suggest: the asymptotic guarantees for differentially-private algorithms may not always reflect their empirical performance

Future Works

- Application in distributed PCA and thus, fMRI analysis
- Can we add less noise?
- When data dimension is large, can we compute $V_k(A)$ in any other way?

Thank you!