Differentially Private Compressive Learning

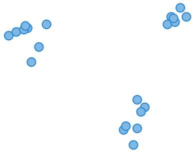
Antoine Chatalic

MaLGa & DIBRIS, University of Genoa (Italy)

Statistical Learning and Differential Privacy Workshop
University of Bath – September 2022

Setting: unsupervised parametric learning

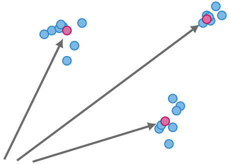
Clustering



Applications: community detection, anomaly detection…

Setting: unsupervised parametric learning

Clustering

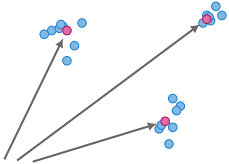


Model: set of k points.

Applications: community detection, anomaly detection...

Setting: unsupervised parametric learning

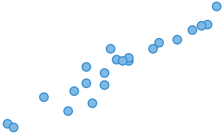
Clustering



Model: set of k points.

Applications: community detection, anomaly detection...

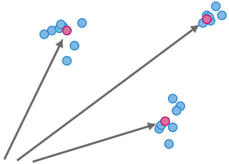
Principal component analysis (PCA)



Applications: compression, data visualization, preprocessing...

Setting: unsupervised parametric learning

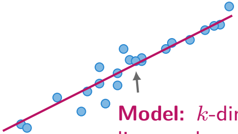
Clustering



Model: set of k points.

Applications: community detection, anomaly detection...

Principal component analysis (PCA)

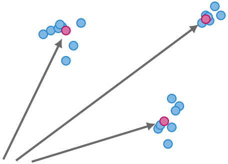


Model: k-dimensional linear subspace.

Applications: compression, data visualization, preprocessing...

Setting: unsupervised parametric learning

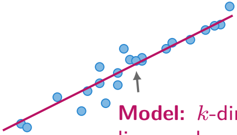
Clustering



Model: set of k points.

Applications: community detection, anomaly detection...

Principal component analysis (PCA)



Model: k-dimensional linear subspace.

Applications: compression, data visualization, preprocessing...

Goal: find the hypothesis h which best "fits" the data:

$$h^* = \arg\min_{h \in H} \mathbf{E}_{\mathbf{x} \sim p_X} \ell(\mathbf{x}, h).$$

Hypothesis space

Loss function measuring how a model "fits" the data

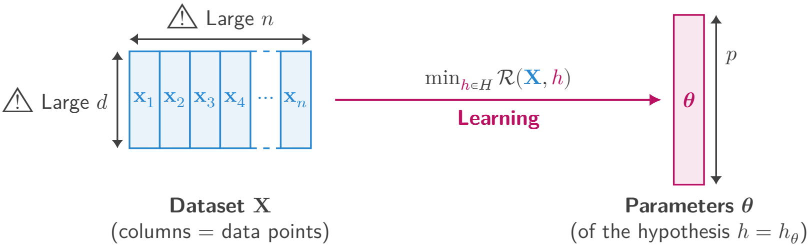
Main challenges

Dataset $\mathbf{X}$
(columns = data points)

Parameters $\boldsymbol{\theta}$
(of the hypothesis $h = h_{\theta}$)

Challenges:

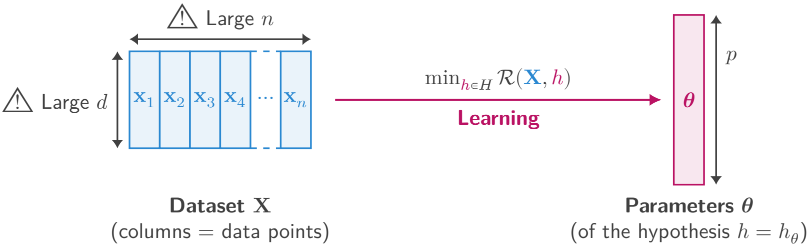
Main challenges



Challenges:

- Large data collections.
- High-dimensional features.
- Distributed datasets.
- Data streams.
- Sensitive data
  (e.g. emails, medical data).

Main challenges



Challenges:

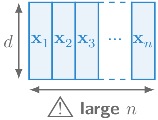
- Large data collections.
- High-dimensional features.
- Distributed datasets.
- Data streams.
- Sensitive data (e.g. emails, medical data).

Limitations of “standard” methods:

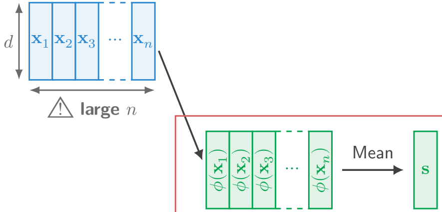
- Multiple passes on the data.
- Computationally expensive.
- High energy consumption.

Can we do better?

Various approaches for Large-Scale Learning

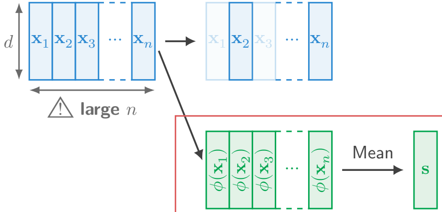


Various approaches for Large-Scale Learning



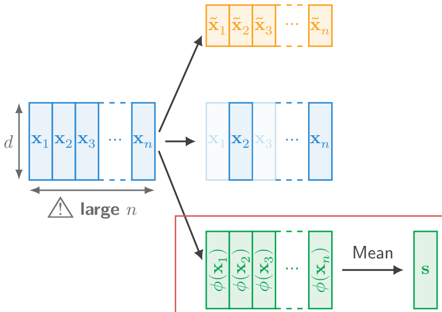
▷ **Vector of moments**
Generalized method of moments, compressive learning.

Various approaches for Large-Scale Learning



- **Coresets**
  Reduce the number n of samples.
- **Vector of moments**
  Generalized method of moments, compressive learning.

Various approaches for Large-Scale Learning

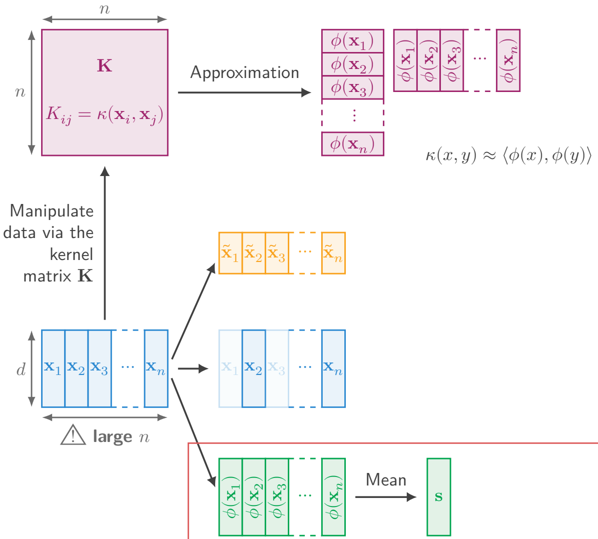


- **Dimensionality reduction**
  PCA, random projections, non-linear reduction.

- **Coresets**
  Reduce the number n of samples.

- **Vector of moments**
  Generalized method of moments, compressive learning.

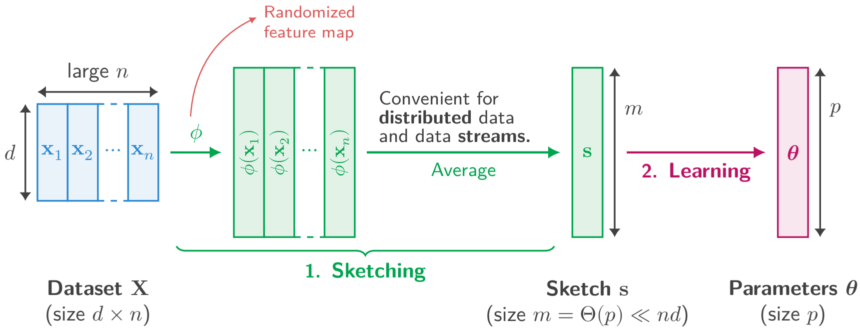
Various approaches for Large-Scale Learning



- **Kernel matrix approximation**
  Low-rank, structured, sparse approximations.
- **Dimensionality reduction**
  PCA, random projections, non-linear reduction.
- **Coresets**
  Reduce the number n of samples.
- **Vector of moments**
  Generalized method of moments, compressive learning.

The Compressive Learning Framework

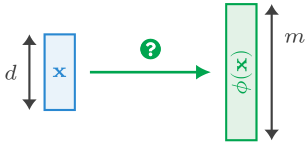
Compressive learning



The sketch is just a vector of "generalized" moments!

[Gribonval et al., 2021. **"Compressive Statistical Learning with Random Feature Moments"**]

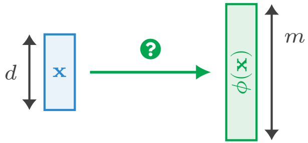
Which feature map Φ can we use?



Random features approximations: $\phi(\mathbf{x}) \triangleq \rho(\mathbf{\Omega}^T\mathbf{x})$ where

- $\mathbf{\Omega} = [\boldsymbol{\omega}_1, ..., \boldsymbol{\omega}_m] \in \mathbb{R}^{d\times m}$ is a **random** matrix (e.g., i.i.d. normal entries);
- ρ is a **deterministic non-linear** function applied pointwise.

Which feature map Φ can we use?



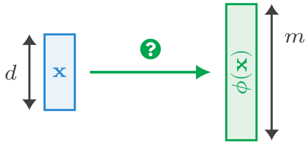
Random features approximations: $\phi(\mathbf{x}) \triangleq \rho(\mathbf{\Omega}^T\mathbf{x})$ where

- $\mathbf{\Omega} = [\boldsymbol{\omega}_1, ..., \boldsymbol{\omega}_m] \in \mathbb{R}^{d\times m}$ is a **random** matrix (e.g., i.i.d. normal entries);
- ρ is a **deterministic non-linear** function applied pointwise.

Example:

- For clustering/density estimation: $\rho(t) \triangleq \exp(-\imath t)$ **(random Fourier features)** [Rahimi and Recht, 2008. **"Random Features for Large-Scale Kernel Machines"**] (The sketch is just m random samples of the empirical **characteristic function** φ, as $\mathbf{s}_j = \frac{1}{n}\sum_{i=1}^{n} e^{-i\omega_j^T x_i} = \varphi(\omega_j)$.)

Which feature map Φ can we use?



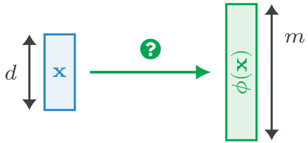
Random features approximations: $\phi(\mathbf{x}) \triangleq \rho(\boldsymbol{\Omega}^T\mathbf{x})$ where

- $\boldsymbol{\Omega} = [\boldsymbol{\omega}_1, ..., \boldsymbol{\omega}_m] \in \mathbb{R}^{d\times m}$ is a **random** matrix (e.g., i.i.d. normal entries);
- ρ is a **deterministic non-linear** function applied pointwise.

Example:

- For clustering/density estimation: $\rho(t) \triangleq \exp(-\imath t)$ **(random Fourier features)** [Rahimi and Recht, 2008. **"Random Features for Large-Scale Kernel Machines"**] (The sketch is just m random samples of the empirical **characteristic function** φ, as $\mathbf{s}_j = \frac{1}{n}\sum_{i=1}^n e^{-i\omega_j^T x_i} = \varphi(\omega_j)$.)
- Variant: **quantized** RFF: $\phi_j(\mathbf{x}) = \mathrm{sign}(\cos(\boldsymbol{\omega}_\mathbf{j}^T\mathbf{x} + \mathbf{b}_\mathbf{j}))$ with random dithering $\mathbf{b}_\mathbf{j} \in [0, 2\pi[$

Which feature map Φ can we use?



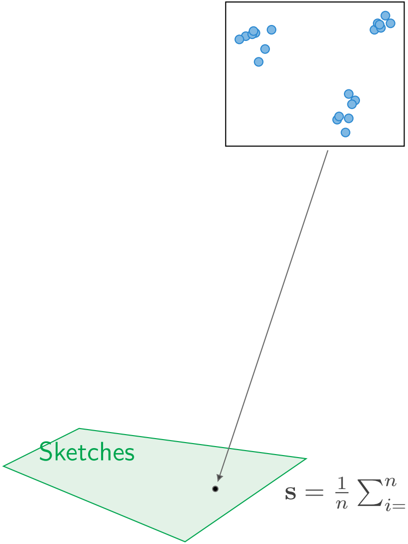
Random features approximations: $\phi(\mathbf{x}) \triangleq \rho(\mathbf{\Omega}^T\mathbf{x})$ where

- $\mathbf{\Omega} = [\boldsymbol{\omega}_1, ..., \boldsymbol{\omega}_m] \in \mathbb{R}^{d\times m}$ is a **random** matrix (e.g., i.i.d. normal entries);
- ρ is a **deterministic non-linear** function applied pointwise.

Example:

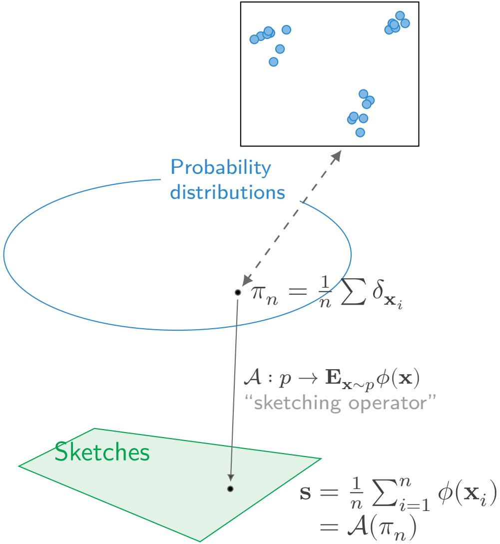
- For clustering/density estimation: $\rho(t) \triangleq \exp(-\imath t)$ **(random Fourier features)**
  [Rahimi and Recht, 2008. **"Random Features for Large-Scale Kernel Machines"**] (The sketch is just m random samples of the empirical **characteristic function** φ, as $\mathbf{s}_j = \frac{1}{n}\sum_{i=1}^{n} e^{-i\omega_j^T x_i} = \varphi(\omega_j)$.)
- Variant: **quantized** RFF: $\phi_j(\mathbf{x}) = \operatorname{sign}(\cos(\boldsymbol{\omega}_\mathbf{j}^T\mathbf{x} + \mathbf{b_j}))$ with random dithering $\mathbf{b_j} \in [0, 2\pi[$
- For PCA: $\rho(t) \triangleq t^2$ **(random quadratic features)**
  (Sketch = rank-one linear measurements of the covariance matrix for centered data.)

Learning as an inverse problem

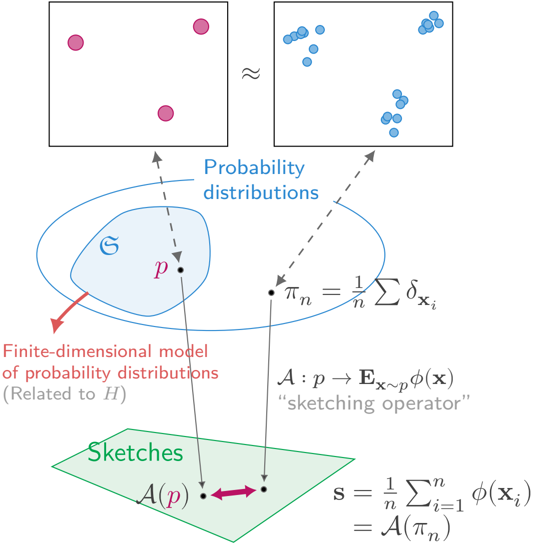


$\mathbf{s} = \frac{1}{n} \sum_{i=1}^{n} \phi(\mathbf{x}_i)$

Learning as an inverse problem



Learning as an inverse problem

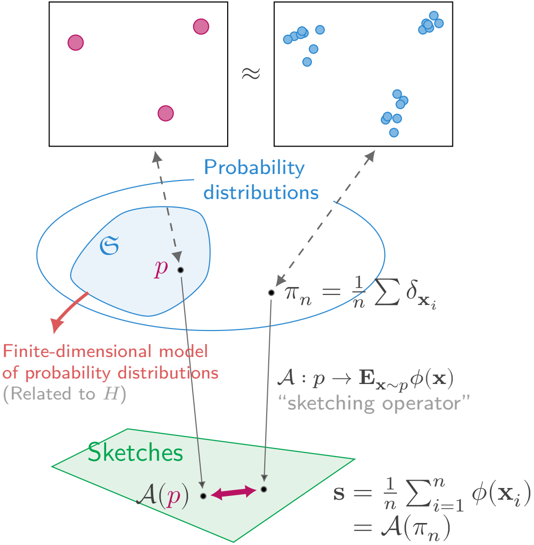


Moment-matching problem:

$$\arg\min_{p \in \mathfrak{S}} \Big\| \underbrace{\mathcal{A}(p)}_{\text{sketch of } p} - \underbrace{\mathbf{s}}_{\text{empirical sketch}} \Big\|_2$$

Cf. generalized method of moments [Hall, 2005] .

Learning as an inverse problem



Moment-matching problem:

$$\arg\min_{p\in\mathfrak{S}} \Big\| \underbrace{\mathcal{A}(p)}_{\text{sketch of } p} - \underbrace{\mathbf{s}}_{\text{empirical sketch}} \Big\|_2$$

Cf. generalized method of moments [Hall, 2005] .

⚠ Difficult/non-convex problem!

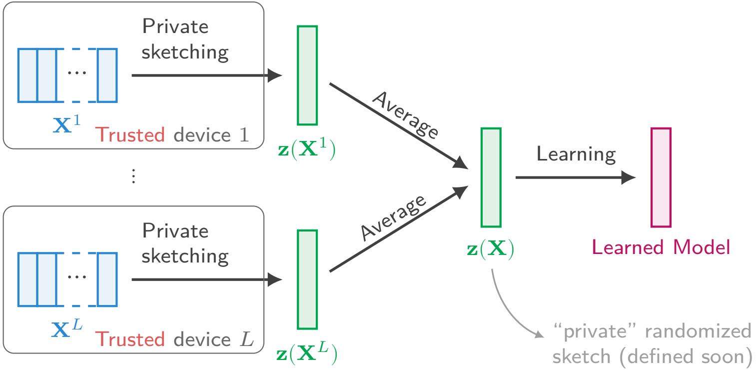
Heuristics can be used, e.g.:

- "Continuous" matching pursuit.
  [Bourrier et al., 2013] [Keriven et al., 2017]
- Approximate message passing
  [Byrne et al., 2019]

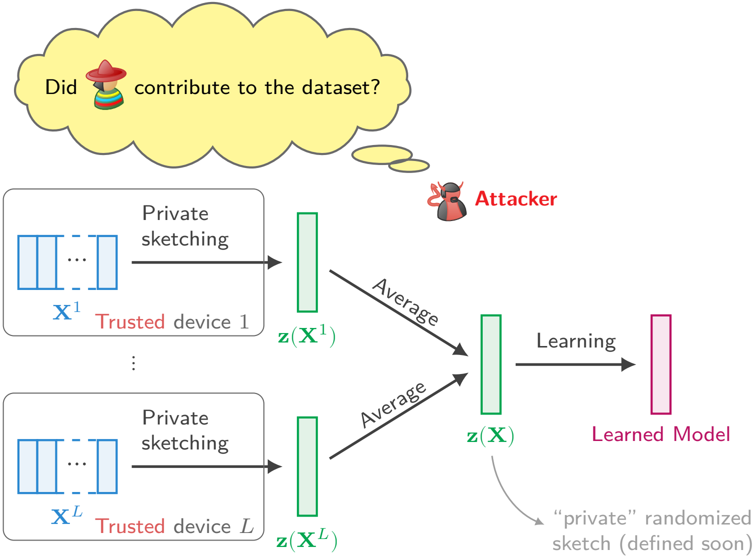
Privacy-Preserving Compressive Learning

(Joint work with V. Schellekens, F. Houssiau, R. Gribonval, L. Jacques and Y.-A. de Montjoye)

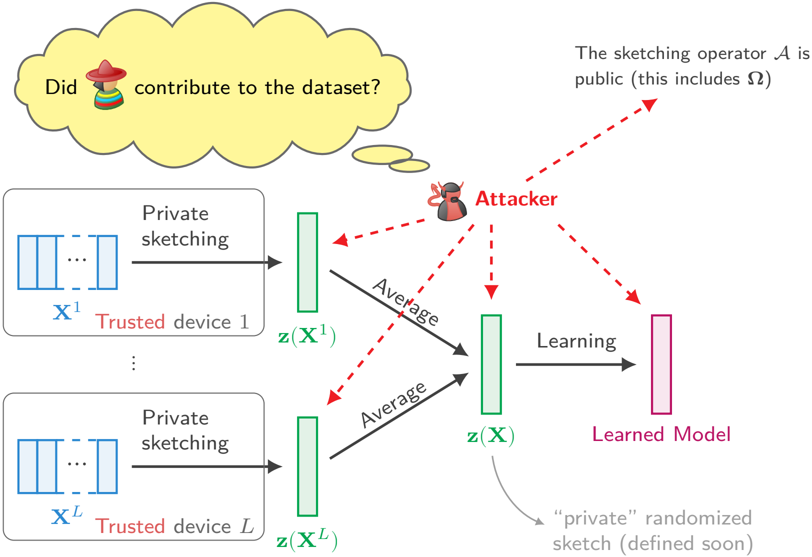
Privacy preservation: what are we talking about?



Privacy preservation: what are we talking about?



Privacy preservation: what are we talking about?



Defining and quantifying privacy

[Dwork et al., 2006. **"Calibrating Noise to Sensitivity in Private Data Analysis"**]

Definition: The randomized mechanism $\mathbf{z}(\cdot)$ achieves (ε, δ)-differential privacy (DP) iff for any (input) neighbor datasets $\mathbf{X}_1 \sim \mathbf{X}_2$ and set S:

$$\mathbb{P}[\mathbf{z}(\mathbf{X}_1) \in S] \leq \exp(\varepsilon)\, \mathbb{P}[\mathbf{z}(\mathbf{X}_2) \in S] + \delta$$

relaxation ("approximate DP" when $\delta > 0$)

privacy "budget" (smaller ε = more privacy)

Notation:

- (ε, δ)-DP in general;
- ε-DP when $\delta = 0$.

Defining and quantifying privacy

[Dwork et al., 2006. **"Calibrating Noise to Sensitivity in Private Data Analysis"**]

Definition: The randomized mechanism $\mathbf{z}(\cdot)$ achieves (ε, δ)-differential privacy (DP) iff for any (input) neighbor datasets $\mathbf{X}_1 \sim \mathbf{X}_2$ and set S:

$$\mathbb{P}[\mathbf{z}(\mathbf{X}_1) \in S] \leq \exp(\varepsilon)\, \mathbb{P}[\mathbf{z}(\mathbf{X}_2) \in S] + \delta$$

relaxation ("approximate DP" when $\delta > 0$)

privacy "budget" (smaller ε = more privacy)

Examples of neighboring relations:

- replacement of one element (BDP):

- add/removal of one element (UDP):

$\mathbf{X}_1$ $\qquad$ $\mathbf{X}_2 = \mathbf{X}_1 +$ 

Notation:

- (ε, δ)-DP in general;
- ε-DP when $\delta = 0$.

Defining and quantifying privacy

[Dwork et al., 2006. **"Calibrating Noise to Sensitivity in Private Data Analysis"**]

Definition: The randomized mechanism $\mathbf{z}(\cdot)$ achieves (ε, δ)-differential privacy (DP) iff for any (input) neighbor datasets $\mathbf{X}_1 \sim \mathbf{X}_2$ and set S:

$$\mathbb{P}[\mathbf{z}(\mathbf{X}_1) \in S] \leq \exp(\varepsilon)\, \mathbb{P}[\mathbf{z}(\mathbf{X}_2) \in S] + \delta$$

relaxation ("approximate DP" when $\delta > 0$)

privacy "budget" (smaller ε = more privacy)

Examples of neighboring relations:

- replacement of one element (BDP):

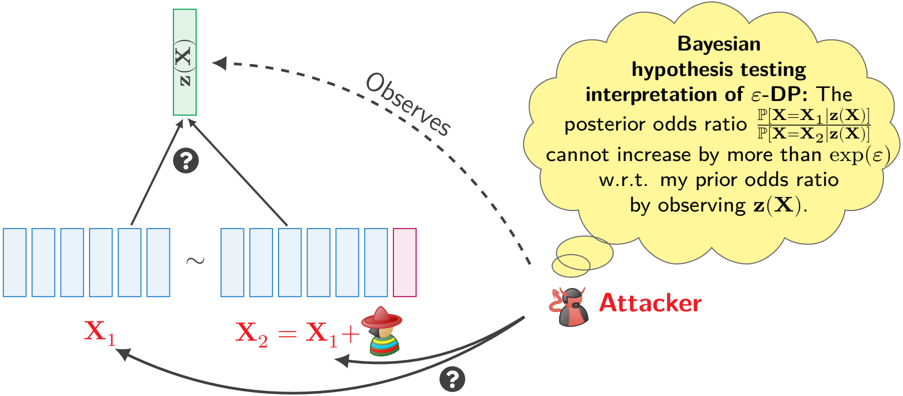
- add/removal of one element (UDP):

$\mathbf{X}_1$ $\quad$ $\mathbf{X}_2 = \mathbf{X}_1 +$ 

Notation:

- (ε, δ)-DP in general;
- ε-DP when $\delta = 0$.

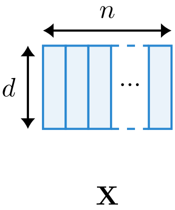
Interpretation of ε-DP



Differential privacy by additive perturbation

Simple way to satisfy DP: add noise to the output.

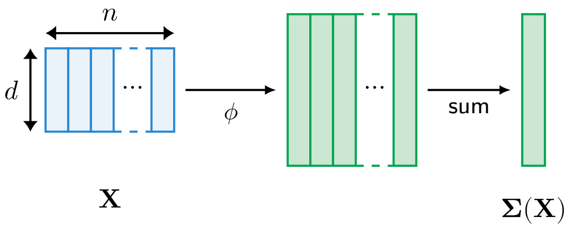
Proposed mechanism



Differential privacy by additive perturbation

Simple way to satisfy DP: add noise to the output.

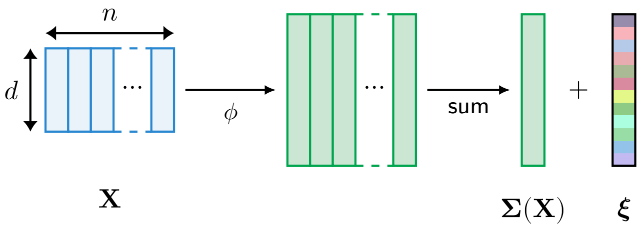
Proposed mechanism



Differential privacy by additive perturbation

Simple way to satisfy DP: add noise to the output.

Proposed mechanism

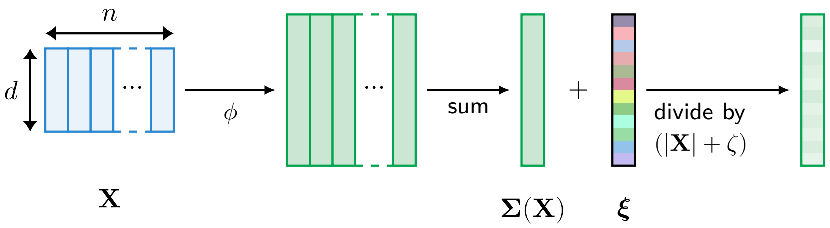


- Add noise ξ on the sum of features.

Differential privacy by additive perturbation

Simple way to satisfy DP: add noise to the output.

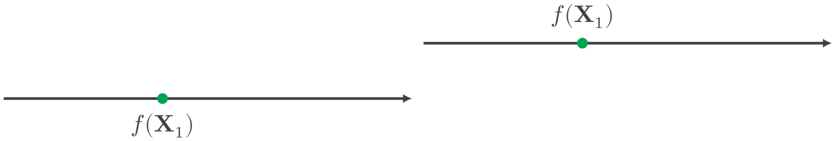
Proposed mechanism



- Add noise $\boldsymbol{\xi}$ on the sum of features.
- Add noise ζ on $|\mathbf{X}|$.

Which noise to ensure privacy? (Common knowledge)

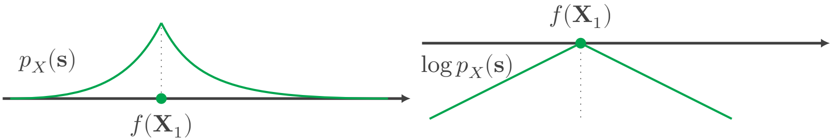
- Laplacian noise for pure ε-DP.



[Dwork et al., 2006. **"Calibrating Noise to Sensitivity in Private Data Analysis"**]

Which noise to ensure privacy? (Common knowledge)

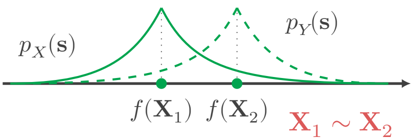
- Laplacian noise for pure ε-DP.

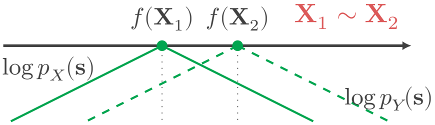


[Dwork et al., 2006. **"Calibrating Noise to Sensitivity in Private Data Analysis"**]

Which noise to ensure privacy? (Common knowledge)

- Laplacian noise for pure ε-DP.

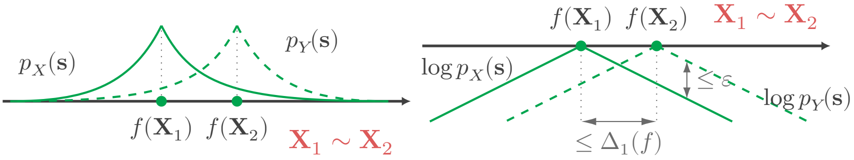




[Dwork et al., 2006. **"Calibrating Noise to Sensitivity in Private Data Analysis"**]

Which noise to ensure privacy? (Common knowledge)

- Laplacian noise for pure ε-DP.

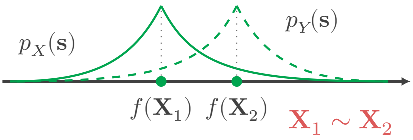


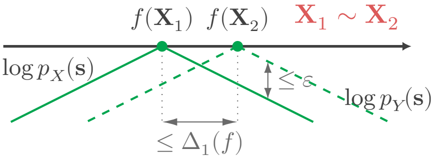
Noise level: $b^* = \frac{\Delta_1(f)}{\varepsilon}$ with $\Delta_1(f) \triangleq \sup_{\mathbf{X}_1 \sim \mathbf{X}_2} \|f(\mathbf{X}_1) - f(\mathbf{X}_2)\|_1$.

[Dwork et al., 2006. **"Calibrating Noise to Sensitivity in Private Data Analysis"**]

Which noise to ensure privacy? (Common knowledge)

- Laplacian noise for pure ε-DP.





Noise level: $b^* = \frac{\Delta_1(f)}{\varepsilon}$ with $\Delta_1(f) \triangleq \sup_{\mathbf{X}_1 \sim \mathbf{X}_2} \|f(\mathbf{X}_1) - f(\mathbf{X}_2)\|_1$.

[Dwork et al., 2006. **"Calibrating Noise to Sensitivity in Private Data Analysis"**]

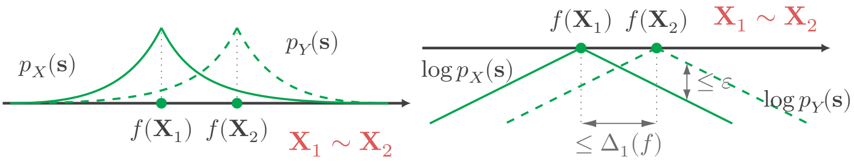
- Gaussian noise for approximate (ε, δ)-DP.

  The noise scales with $\Delta_2(f) \triangleq \sup_{\mathbf{X}_1 \sim \mathbf{X}_2} \|f(\mathbf{X}_1) - f(\mathbf{X}_2)\|_2$.

[Balle and Wang, 2018. **"Improving the Gaussian Mechanism for Differential Privacy"**]

Which noise to ensure privacy? (Common knowledge)

- Laplacian noise for pure ε-DP.



Noise level: $b^* = \frac{\Delta_1(f)}{\varepsilon}$ with $\boxed{\Delta_1(f) \triangleq \sup_{\mathbf{X}_1 \sim \mathbf{X}_2} \|f(\mathbf{X}_1) - f(\mathbf{X}_2)\|_1.}$

[Dwork et al., 2006. **"Calibrating Noise to Sensitivity in Private Data Analysis"**]

l_1/l_2 "sensitivities"

- Gaussian noise for approximate (ε, δ)-DP.

  The noise scales with $\boxed{\Delta_2(f) \triangleq \sup_{\mathbf{X}_1 \sim \mathbf{X}_2} \|f(\mathbf{X}_1) - f(\mathbf{X}_2)\|_2.}$

[Balle and Wang, 2018. **"Improving the Gaussian Mechanism for Differential Privacy"**]

Privacy results

	Pure ε-DP	Approximate (ε, δ)-DP
	$\Delta_1(\boldsymbol{\Sigma})$	$\Delta_2(\boldsymbol{\Sigma})$
Fourier features	$\leq \sqrt{2m}$	$= \sqrt{m}$
$+$ $\boldsymbol{\Omega}$ nonresonant	$= \sqrt{2m}$	$= \sqrt{m}$
Quadratic features	$= \|\boldsymbol{\Omega}\|_2^2$	$= \|\boldsymbol{\Omega}^T\|_{2\to4}^2$
$+$ $\boldsymbol{\Omega}$ union of orthogonal bases.	$= m/d$	No particular closed form.

(Results for the "replacement" neighboring relation (BDP) can be found in the paper.)

[Chatalic et al., 2021. **"Compressive Learning with Privacy Guarantees"**]

Privacy results

Holds almost surely!

	Pure ε-DP	Approximate (ε, δ)-DP
	$\Delta_1(\boldsymbol{\Sigma})$	$\Delta_2(\boldsymbol{\Sigma})$
Fourier features	$\leq \sqrt{2m}$	$= \sqrt{m}$
+ $\boldsymbol{\Omega}$ nonresonant	$= \sqrt{2m}$	$= \sqrt{m}$
Quadratic features	$= \lVert\boldsymbol{\Omega}\rVert_2^2$	$= \lVert\boldsymbol{\Omega}^T\rVert_{2\to 4}^2$
+ $\boldsymbol{\Omega}$ union of orthogonal bases.	$= m/d$	No particular closed form.

Order-4 tensor approximation problem (NP-hard)

(Results for the "replacement" neighboring relation (BDP) can be found in the paper.)

Different problems:

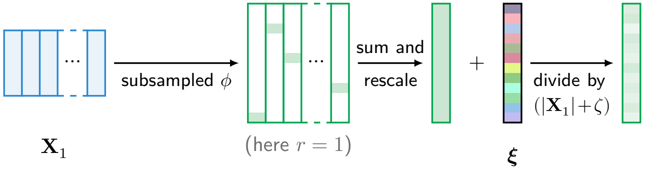
- obtaining upper bounds (easy);
- obtaining sharp bounds (🧩). Nonresonant = linearly independent over $\mathbb{Q}$;
- computing numerically the bound (🧩 in some settings).

[Chatalic et al., 2021. **"Compressive Learning with Privacy Guarantees"**]

Subsampling

Compute only $r < m$ features of ϕ when sketching.

Proposed mechanism (with subsampling)



More precisely $\Sigma_H(\mathbf{X}) = \frac{1}{\alpha}\sum_{i=1}^{n} \phi(\mathbf{x_i}) \odot \mathbf{h}_i$ where the $(\mathbf{h}_i)_{1\leq i\leq |\mathbf{X}|}$ are e.g. Poisson with parameter α, uniform over masks with fixed size r ($\alpha = r/m$), uniform over masks with block structure.

$$\mathbf{z}(X) = \frac{\Sigma_H(X)+\boldsymbol{\xi}}{|\mathbf{X}|+\zeta}$$

Goal 1: Reduce the computational complexity.
Goal 2: Reduce the amount of released information.

Privacy results (with subsampling)

	Pure ε-DP	Approximate (ε, δ)-DP
	Laplace with parameter b	Gaussian with parameter σ
Fourier features	$b^* \leq \sqrt{2}\frac{m}{\varepsilon}$	$\sigma^* \leq \frac{\eta(\varepsilon,\delta)}{\sqrt{2\varepsilon}} \frac{m}{\sqrt{r}}$
+ $\mathbf{\Omega}$ nonresonant	$b^* = \sqrt{2}\frac{m}{\varepsilon}$	not covered
Quadratic features	$b^* \leq \frac{1}{\alpha\varepsilon} \sup_{\mathbf{h}} \lVert \Omega_{\mathbf{h}} \rVert_2^2$	$\sigma^* \leq \frac{\eta(\varepsilon,\delta)}{\sqrt{2\varepsilon}} \frac{m}{r} \sup_{\mathbf{h}} \lVert \Omega_{\mathbf{h}} \rVert_{2\to 4}^2$
+ $\mathbf{\Omega}$ union of orthogonal bases.	$b^* = \frac{m}{d\varepsilon}$	no particular form

(Results for the "replacement" neighboring relation (BDP) can be found in the paper.)

[Chatalic et al., 2021. **"Compressive Learning with Privacy Guarantees"**]

Record subsampling vs feature supsampling

One can also subsample the **data:** an ε-UDP mechanism applied after Poisson-subsampling the dataset with parameter α is $\log(1+\alpha(\exp(\varepsilon)-1))$-UDP ($< \varepsilon$).
[Balle et al., 2018. **"Privacy Amplification by Subsampling"**] .

Lemma

Both types of subsampling "**do not improve** privacy" when properly rescaling the sketch.
In most settings, previous bounds however remain valid (no loss of privacy).

Note: In spite of that, subsampling improves the complexity-privacy tradeoff!

Utility Guarantees under Differential Privacy

Role of the noise-to-signal ratio

Noise-to-signal ratio:

private sketch

$$\mathrm{NSR} \triangleq \frac{\mathbf{E}\|\mathbf{z}(\mathbf{X}) - \mathbf{s}\|_2^2}{\|\mathbf{s}\|^2}.$$

empirical sketch

Role of the noise-to-signal ratio

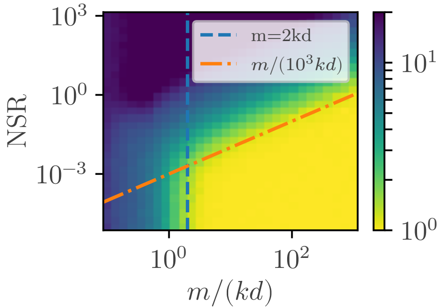
Noise-to-signal ratio:

$$\text{NSR} \triangleq \frac{\mathbf{E}\|\mathbf{z}(\mathbf{X}) - \mathbf{s}\|_2^2}{\|\mathbf{s}\|^2}.$$

private sketch

empirical sketch

Empirical correlation (clustering task):



Color $=$ relative error.

For m large enough and fixed, the NSR is a **good indicator of the error.**

Recall: $m =$ sketch size
$kd \approx$ number of parameters to learn

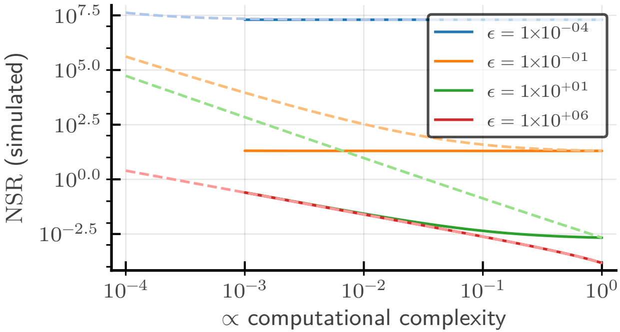
Record subsampling vs feature subsampling

Legend:
— feature subsampling
- - record subsampling

Observation: feature subsampling yields a better utility in some regimes!

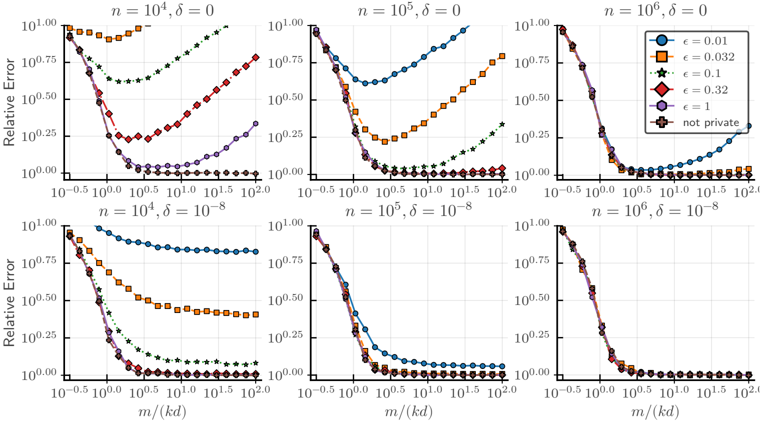
When doing β-data sampling and α-feature subsampling:

$$\mathrm{NSR}_\xi \propto \frac{m^3}{n^2\|z\|^2} \frac{1}{\beta^2 \log^2(1 + (\exp(\varepsilon) - 1)/\beta)}$$

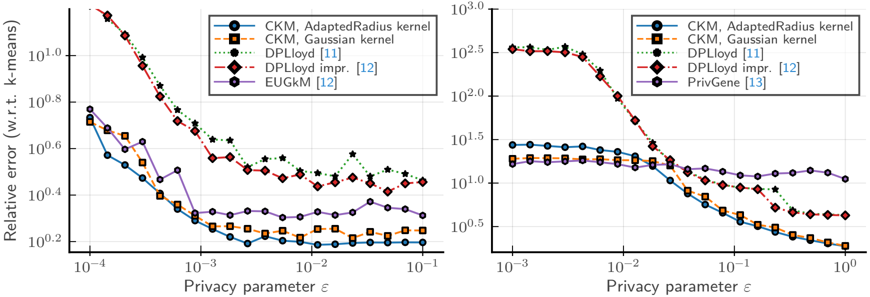


Hyperparameter tuning: choice of the sketch size

Choosing m in the pure DP setting can be tricky.
Analysis of the NSR is helpful in this regard!



Experimental results (clustering problem)

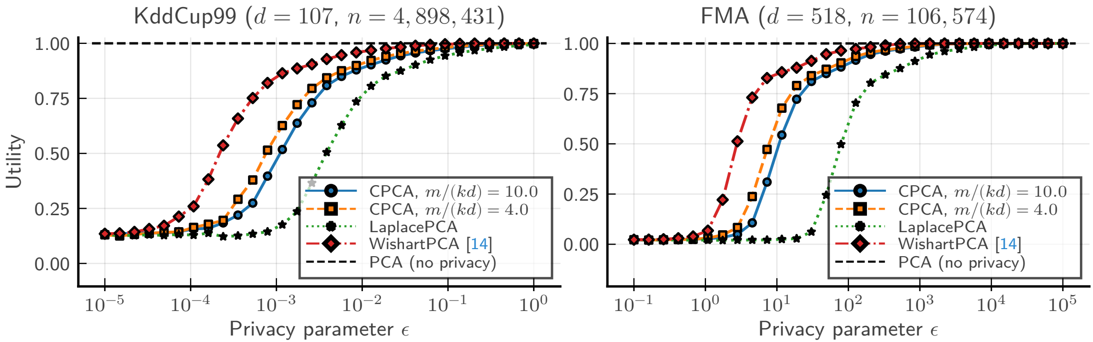


(Left) Gowalla dataset, $d = 2, n \approx 10^6$; (Right) FMA dataset, MFCC features. Medians over 100 trials.

Observations:

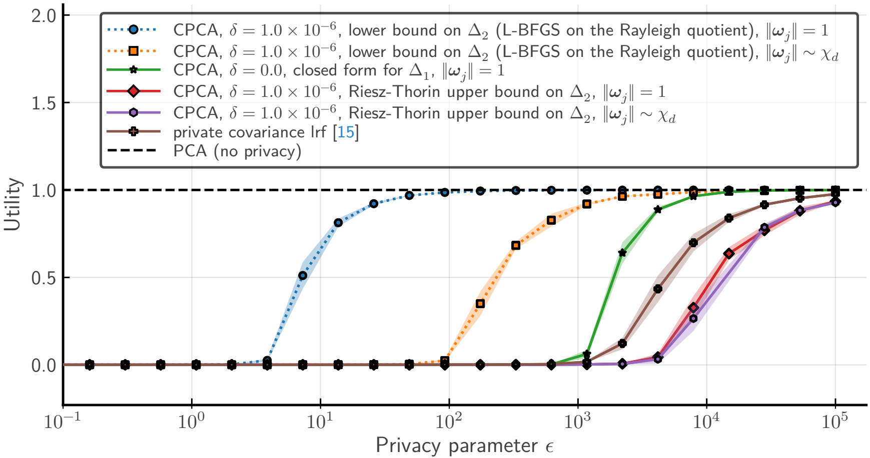
- Competitive results with other methods from the literature.
- DPLloyd suffers from its “iterative” nature.

Experimental results, PCA (1)



- LaplacePCA: simple baseline ($O(n^2)$!).
- WishartPCA: adding noise following a Wishart distribution (still $O(n^2)$!).

Experimental results, PCA (2)



$d = 2^{15}$ here, synthetic data.

⚠ The two dotted curves are not DP. Efficiently computing Δ_2 remains a challenge.

The Moment-to-Moment Method

(Joint work with F. Houssiau, V. Schellekens, S. K. Annamraju and Y.-A. de Montjoye)

Learning generalized moments

Goal: learn a function $F(X) = \frac{1}{n}\sum_{i=1}^{n} f(\mathbf{x}_i)$.

Idea: use a linear model in the feature space

- Find $a \in \mathbb{R}^m$ s.t. $f_a(\mathbf{x}) = \langle \phi(\mathbf{x}), a\rangle$ is a good approximation of f on the considered domain.
- Compute $F_a(X) = \langle \mathbf{s}, a\rangle = \frac{1}{n}\sum_{i=1}^{n} \langle \phi(\mathbf{x}_i), a\rangle$

One can get a universal approximator taking $\phi_i(\mathbf{x}) = \rho(\mathbf{a}_i^T\mathbf{x} + \mathbf{b}_i)$ with random $\mathbf{a}_i, \mathbf{b}_i$ and a bounded non-constant piecewise continuous ρ.

[Zhang et al., 2012. **"Universal Approximation of Extreme Learning Machine With Adaptive Growth of Hidden Nodes"**]

Learning generalized moments (2)

Problem formulation:

$$\min \mathbf{E}_{X\sim p}(f(X) - \langle a, \phi(X)\rangle)^2 + \lambda\|a\|^2$$

where p is ideally the true data distribution.

In practice:

- We draw a finite sample from p.
- λ is chosen to compensate the noise added for privacy.

Limitations...

- Distributional shift: in practice p differs from the true data distribution.
- Approximation error.
- Sampling error.
- DP noise.

M2M for Empirical Risk Minimization

Problem: $\min_\theta \sum_{i=1}^n \ell(\mathbf{x}_i, \theta)$.

Idea: use M2M to approximate the loss $\ell(\cdot, \theta)$.

We end up with the following bilevel optimization problem:

$$\min_\theta \langle a_\theta, \mathbf{s} \rangle \quad \text{s.t.} \quad a_\theta \in \arg\min_a \mathbf{E}_{X\sim p}(\ell(X,\theta) - \langle a, \phi(X)\rangle)^2 + \lambda \|a\|^2$$

We use n samples $\tilde{\mathbf{x}}_1, ..., \tilde{\mathbf{x}}_{n_s} \sim p$ and get

$$\theta^* \in \arg\min_\theta \sum_{i=1}^{n_s} \underbrace{\phi(\tilde{\mathbf{x}}_i)^T S\, \mathbf{s}}_{w(\tilde{\mathbf{x}}_i)} \ell(\tilde{\mathbf{x}}_i, \theta)$$

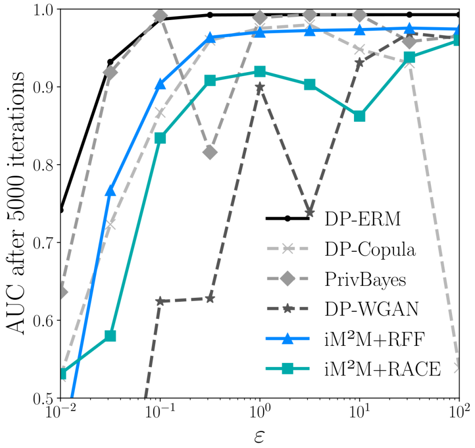
where $S = \left(\frac{1}{n_s} \sum_{i=1}^{n_s} \phi(\tilde{\mathbf{x}}_i)\phi(\tilde{\mathbf{x}}_i)^T + \lambda I\right)^{-1}$.

M2M for Empirical Risk Minimization

Example: logistic regression

Comparing with:

- DP-Copula (Li et al., Fitting a Gaussian Copula)
- PrivBayes (Zhang et al.)
- DP-ERM (Chaudhuri et al., objective perturbation)
- DP-WGAN (Xie et al.)



Conclusion

Efficiency and privacy can be **achieved with similar tools**.

Some advantages of sketches...

- One can learn measuring only one bit of information / data sample !
- (Data-agnostic) sketches of generalized moments can easily be privatized by noise addition.
- Sketches are generic enough to approximate various function classes.
- One single privatized sketch can thus be **used for multiple analyses**.

Perspectives

- Studying more finely M2M.
- Impact of subsampling / quantizing for other privacy definitions?

Appendix

References I

[1] Rémi Gribonval et al. "Compressive Statistical Learning with Random Feature Moments." In: *Mathematical Statistics and Learning* 3.2 (Aug. 20, 2021), pp. 113–164. ISSN: 2520-2316.

[2] Ali Rahimi and Benjamin Recht. "Random Features for Large-Scale Kernel Machines." In: *Advances in Neural Information Processing Systems*. 2008, pp. 1177–1184.

[3] Alastair R. Hall. *Generalized Method of Moments*. Oxford university press, 2005.

[4] Anthony Bourrier et al. "Compressive Gaussian Mixture Estimation." In: ICASSP-38th International Conference on Acoustics, Speech, and Signal Processing. 2013, pp. 6024–6028.

[5] Nicolas Keriven et al. "Compressive K-means." In: International Conference on Acoustics, Speech and Signal Processing (ICASSP). Mar. 5, 2017.

[6] Evan Byrne et al. "Sketched Clustering via Hybrid Approximate Message Passing." In: *IEEE Transactions on Signal Processing* 67.17 (Sept. 2019), pp. 4556–4569.

[7] Cynthia Dwork et al. "Calibrating Noise to Sensitivity in Private Data Analysis." In: *Theory of Cryptography Conference*. Springer, 2006, pp. 265–284.

References II

[8] Borja Balle and Yu-Xiang Wang. "Improving the Gaussian Mechanism for Differential Privacy: Analytical Calibration and Optimal Denoising." In: *International Conference on Machine Learning*. International Conference on Machine Learning. July 3, 2018, pp. 394–403.

[9] Antoine Chatalic et al. "Compressive Learning with Privacy Guarantees." In: *Information and Inference: A Journal of the IMA* (iaab005 May 15, 2021). ISSN: 2049-8772.

[10] Borja Balle et al. "Privacy Amplification by Subsampling: Tight Analyses via Couplings and Divergences." In: *Advances in Neural Information Processing Systems 31*. Ed. by S. Bengio et al. Curran Associates, Inc., 2018, pp. 6277–6287.

[11] Avrim Blum et al. "Practical Privacy: The SuLQ Framework." In: *Proceedings of the Twenty-Fourth ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems*. ACM, 2005, pp. 128–138.

[12] Dong Su et al. "Differentially Private K-Means Clustering and a Hybrid Approach to Private Optimization." In: *ACM Trans. Priv. Secur.* 20.4 (Oct. 2017), 16:1–16:33. ISSN: 2471-2566.

References III

[13] Jun Zhang et al. "PrivGene: Differentially Private Model Fitting Using Genetic Algorithms." In: *Proceedings of the 2013 International Conference on Management of Data - SIGMOD '13*. The 2013 International Conference. New York, New York, USA: ACM Press, 2013, p. 665. ISBN: 978-1-4503-2037-5.

[14] Wuxuan Jiang et al. "Wishart Mechanism for Differentially Private Principal Components Analysis." In: *Thirtieth AAAI Conference on Artificial Intelligence*. 2016.

[15] Jalaj Upadhyay. "The Price of Privacy for Low-Rank Factorization." In: *Advances in Neural Information Processing Systems*. 2018, pp. 4176–4187.

[16] Rui Zhang et al. "Universal Approximation of Extreme Learning Machine With Adaptive Growth of Hidden Nodes." In: *IEEE Transactions on Neural Networks and Learning Systems* 23.2 (Feb. 2012), pp. 365–371. ISSN: 2162-2388.