Differential Privacy and Generalization: Sharper Bounds, Theoretically Grounded Algorithms, and Thresholdout

Luca Oneto
www.lucaoneto.com
luca.oneto@unige.it

Summer School on Applied Harmonic Analysis, Genoa, Italy, 24th July 2017
SmartLab

D. Anguita
Associate Prof.

S. Ridella
Emeritus Prof.

L. Oneto
Assistant Professor

P. Sanetti
Res. Assistant

G. Clerico
Res. Assistant

A. Lulli
Postdoc

M. Cambiaso
Res. Assistant

I. Orlandi
PhD Stud.

E. Fumeo
PhD Stud.

F. Cipollini
PhD Stud.
Expertise

Scientific Research

• Topics
  • Neural Networks
  • Kernel Methods
  • Ensemble Methods
  • Statistical Learning Theory
  • Machine Learning
  • Data Mining
  • High Performance Computing
  • Big & Small Data Analysis
  • CBM, EDM, HAR, Sentiment Analysis, Cybersecurity
  • ...

• Publications
  • > 50 International High Ranked Journals
  • > 100 International High Ranked Conferences
  • ...

European Projects

• Basic Research
  • EC NeuroNet I & II - Network of Excellence on Neural Networks
  • EC RAIN - Redundant Array of Inexpensive Workstations for Neurocomputing
  • EC EUNITE - European Network of Excellence on Information Technology for Smart Adaptive Systems
  • EC-FET NiSIS - Nature-inspired Smart Information Systems
  • ...

• Applied Research
  • EC-H2020 IN2DREAMS
  • EC-H2020 In2Rail
  • EC-FP7 MAXBE
  • ...

Technology Transfer

• aizoOn S.r.l.
• Ansaldo STS S.p.A.
• Brembo S.p.A
• Bombardier Transportation
• Cetena S.p.A.
• Damen Shipyards Group
• Ferrari S.p.A. - Scuderia Ferrari
• VarGroup
• ...

University of Genoa – DIBRIS – SmartLab
www.lucaoneto.com - luca.oneto@unige.it
Privacy

• In the last years researchers have studied many ways to access data in a private way (aggregate, noise, etc.)

• Privacy is a bad thing from a data scientist point of view (we cannot access data if not aggregate, etc.)

• The breakthrough was to find a way to exploit privacy as a new regularization method and as a tool for better assessing the generalization performances of a learning algorithm.
Supervised Learning

The only things available for learning is a set of examples of the mapping.

Deterministic Functions/Learning Algorithms

Given a set of data the algorithm always returns the same model.

\[ f = \mathcal{A}(s) \]

The model is a function chosen in a set of functions: given the function and a point, the predicted output is always the same.

\[ x \sim \mathcal{X}, \ y = f(x) \]
Randomized Functions

Given a set of data the algorithm always returns the same model.

\[ \rho \leftarrow \mathcal{A}(s) \]

The model is a distribution over a set of functions. Given the model and a point, the predicted output may be different.

\[ x \sim \mathcal{P} \chi, \ f \sim \rho, \ y = f(x) \]
Randomized Learning Algorithms

Given a set of data the algorithm may return different models.

\[ F = \mathcal{A}(s) \]

The model can be a deterministic or randomized function. In our case the function is deterministic.

\[ x \sim \mathcal{P}x, \quad y = F(x) \]
### Notation

\[ x \in \mathcal{X}, \ y \in \mathcal{Y}, \ z \in \mathcal{Z} = \mathcal{X} \times \mathcal{Y}, \ \mathbb{P}_X, \ \mathbb{P}_Y, \ \mathbb{P}_Z \]

\[ \mathcal{Z}^n = S \ni s = \{z_1, \cdots, z_n\} = (x_1, y_1), \cdots, (x_n, y_n) \] i.i.d. \( \mathbb{P}_S \)

\[ \mathcal{Z} \ni \mathcal{Z} \sim \mathbb{P}_Z, \ S \ni \mathcal{S} \sim \mathbb{P}_S \]

\[ s = \{z_1, \cdots, z_{i-1}, \hat{z}_i, z_{i+1}, \cdots, z_n\} \] \( \hat{z}_i \) i.i.d. \( z_i \)

\[ \hat{s} \subseteq S \]

\[ f : \mathcal{X} \to \mathcal{Y}, \ f \in \mathcal{F} \]

\[ \hat{\mathcal{F}} \subseteq \mathcal{F} \]

\[ \mathcal{A} : \mathcal{S} \to \mathcal{F}, \ \mathbb{P}_\mathcal{A} \]

\[ \hat{\mathcal{D}} : \mathcal{F} \to \hat{\mathcal{S}} \]

\[ \ell : \mathcal{F} \times \mathcal{Z} \to [0, 1] \]

\[ L(f) = \mathbb{E}_\mathcal{Z} \ell(f, \mathcal{Z}), \ V(f) = \mathbb{E}_\mathcal{Z} [\ell(f, \mathcal{Z}) - L(f)]^2 \]

\[ \hat{L}^s_n(f) = \frac{1}{n} \sum_{i=1}^{n} \ell(f, z_i), \ \hat{V}^s_n(f) = \frac{1}{n(n-1)} \sum_{i=1}^{n} \sum_{j=i+1}^{n} [\ell(f, z_i) - \ell(f, z_j)]^2 \]


Goal

Estimate the true (generalization) error of the model based on the empirical data

\[
P\{ | L(f) - \hat{L}_n(f) | \geq \epsilon \} \leq \delta \\
| L(f) - \hat{L}_n(f) | \leq \epsilon, \quad \forall (1 - \delta) \\
\delta \leftrightarrow \epsilon
\]
Differentially Private (DP) Randomized Learning Algorithms

A Randomized Learning Algorithm is $(\epsilon, \delta)$-DP if

$$\mathbb{P}_{\mathcal{A}} \left\{ \mathcal{A}(s) \in \mathcal{F} \right\} \leq e^{\epsilon} \mathbb{P}_{\mathcal{A}} \left\{ \mathcal{A}(\dot{s}) \in \mathcal{F} \right\} + \delta$$

$$\forall \mathcal{F} \subseteq \mathcal{F}, \forall s, \dot{s} \in \mathcal{S}$$

Differentially Private (DP) Randomized Learning Algorithms

A Randomized Learning Algorithm is $\epsilon$-DP if

$$\frac{\mathbb{P}_{\mathcal{A}}\{\mathcal{A}(s) = f\}}{\mathbb{P}_{\mathcal{A}}\{\mathcal{A}(\dot{s}) = f\}} \leq e^\epsilon, \ \forall f \in \mathcal{F}, \ \forall s, \dot{s} \in \mathcal{S}$$

Proof: 

$$\mathbb{P}_{\mathcal{A}}\{\mathcal{A}(s) \in \mathcal{F}\} = \int_{\mathcal{F}} \mathbb{P}_{\mathcal{A}}\{\mathcal{A}(s) = f\} df \leq \int_{\mathcal{F}} e^\epsilon \mathbb{P}_{\mathcal{A}}\{\mathcal{A}(\dot{s}) = f\} df = e^\epsilon \mathbb{P}_{\mathcal{A}}\{\mathcal{A}(\dot{s}) \in \mathcal{F}\}$$

Hold Out, Compression, Complexity, Stability, and… Privacy (I)

The first option to estimate the generalization performance of an algorithm is to split the data

\[ s = s_1 \cup s_2, \quad s_1 \cap s_2 = \emptyset \]

\[ \mathbb{P}_{s_2} \left\{ L(A(s_1)) - \hat{L}_{|s_2|}(A(s_1)) \geq t \right\} \leq e^{-2|s_2|t^2} \]

Hold Out, **Compression**, Complexity, Stability, and... Privacy (II)

Another option is to check how much the algorithm compresses the data

\[ s' \subseteq s \]

\[ \mathbb{P}_s \left\{ L(\mathcal{A}(S)) - \tilde{L}_n^S(\mathcal{A}(S)) \geq t \right\} \leq n \binom{n}{|s'|} e^{-2nt^2} \]

Hold Out, Compression, Complexity, Stability, and… Privacy (III)

Another option is to check how large is the function space in which the algorithm chooses the solution

\[ P_S \left\{ L(\mathcal{A}(S)) - \hat{L}_n^S(\mathcal{A}(S)) \geq t \right\} \leq c_2 \mathcal{C}(\mathcal{F}) e^{-c_1 nt^2} \]

\[ P_S \left\{ L(\mathcal{A}(S)) - \hat{L}_n^S(\mathcal{A}(S)) \geq t \right\} \leq c_2 \mathcal{C} \left( \left\{ f : f \in \mathcal{F}, \hat{L}_n^S(f) \leq c_3 \right\} \right) e^{-c_1 nt^2} \]

\[ \mathcal{C}(\mathcal{F}) : n^{d_{VC}(\mathcal{F})}, e^{nR^2(\mathcal{F})} \]

\[ c_3(L, C, t, n) \]

Hold Out, Compression, Complexity, Stability, and... Privacy (IV)

Another way is to check how close the functions chosen by the algorithm are

\[ |\ell(A(s), \cdot) - \ell(A(\hat{s}), \cdot)|_\infty \leq \beta \]

\[ \mathbb{P}_S \left\{ L(A(S)) - \widehat{L}_n^S(A(S)) \geq t \right\} \leq c_2 e^{n\beta^2 - c_1 nt^2} \]

DP Main Result

If $\mathbb{P}_S\{S \in \tilde{D}(f)\} \leq \beta$

$\forall f \in \mathcal{F}$ and $\epsilon \leq \sqrt{\ln \frac{1}{\beta}/2n}$

$\rightarrow \mathbb{P}_{S,F}\{S \in \tilde{D}(F)\} \leq 3\sqrt{\beta}$

Proof: rather technical…

Hoeffding-type Bounds

\[ \epsilon \leq \sqrt{t^2 - \ln(2)/2n} \]

\[ \rightarrow \mathbb{P}_{S, F}\{|L(F) - \hat{L}_n^S(F)| \geq t\} \leq 3\sqrt{2}e^{-nt^2} \]

Proof: \[ \mathbb{P}_S\{L(f) - \hat{L}_n^S(f) \geq t\} \leq e^{-2nt^2} \]

\[ \tilde{D}(f) = \{s \in S : L(f) - \hat{L}_n^S(f) > t\} \]

\[ \beta = e^{-2nt^2} \]

Chernoff and Bennett-type Bounds

\[ \epsilon \leq \sqrt{t^2 - \frac{\ln(2)}{2n}} \]

\[ \rightarrow \mathbb{P}_{S,F}\{|L(F) - \hat{L}_n^S(F)| \geq \sqrt{6L(F)t}\} \leq 3\sqrt{2}e^{-nt^2} \]

\[ O \left(\frac{1}{\sqrt{n}}\right) \div O \left(\frac{1}{n}\right) \]

Chernoff and **Bennett**-type Bounds

\[ \epsilon \leq \sqrt{t^2 - \frac{\ln(3)}{2n}} \]

\[ \rightarrow \mathbb{P}_{S,F} \left\{ \left| L(F) - \hat{L}_n^S(F) \right| \geq \sqrt{4\hat{V}_n^S(F)t + \frac{14nt^2}{3(n-1)}} \right\} \leq 3\sqrt{3}e^{-nt^2} \]

\[ O \left( \frac{1}{\sqrt{n}} \right) \div O \left( \frac{1}{n} \right) \]

Clopper-Pearson (Binary Classification)

$$\epsilon \leq \sqrt{\frac{\ln (1/2\delta)}{2n}}$$

$$\rightarrow \mathbb{P}_{S, F}\{Q[\delta; n\hat{L}_n^S(F), n - n\hat{L}_n^S(F) + 1] \leq L(F)$$

$$\leq Q[1 - \delta; n\hat{L}_n^S(F) + 1, n - n\hat{L}_n^S(F)]\} \leq 3\sqrt{2\delta}$$

$$O \left(\frac{1}{\sqrt{n}}\right) \div O \left(\frac{1}{n}\right)$$

Clopper-Pearson (Regression)

$$\epsilon \leq \sqrt{\ln(1/2\delta)/2n}$$

$$\rightarrow \mathbb{P}_{S,F} \left\{ Q \left[ \delta; \sum_{i=1}^{n} [\ell(F, Z_i) \geq u_i], n - \sum_{i=1}^{n} [\ell(F, Z_i) \geq u_i] + 1 \right] \leq L(F) \right\} \leq \frac{3\sqrt{2\delta}}{2}$$

Proof: \[
\begin{align*}
\mathbb{P}\{u = \alpha, \alpha \in [0, 1]\} &= 1 \\
\mathbb{P}\{u = \alpha, \alpha \not\in [0, 1]\} &= 1 \\
\mathbb{P}\{h \geq u\} &= \mathbb{E}_{h,u}\{h \geq u\} \\
&= \mathbb{E}_{h}\{\mathbb{E}_u\{h \geq u\}\} = \mathbb{E}_{h}\{h\} \\
\end{align*}
\]

$$O\left(\frac{1}{\sqrt{n}}\right) \div O\left(\frac{1}{n}\right)$$


## Example: DP Random Forest (RF) (I)

www.openml.org

<table>
<thead>
<tr>
<th>Abb.</th>
<th>ID</th>
<th>Name</th>
<th>n</th>
<th>d</th>
<th>Abb.</th>
<th>ID</th>
<th>Name</th>
<th>n</th>
<th>d</th>
</tr>
</thead>
<tbody>
<tr>
<td>D01</td>
<td>40</td>
<td>sonar</td>
<td>208</td>
<td>61</td>
<td>D02</td>
<td>59</td>
<td>ionosphere</td>
<td>351</td>
<td>35</td>
</tr>
<tr>
<td>D03</td>
<td>785</td>
<td>wind_correlations</td>
<td>45</td>
<td>47</td>
<td>D04</td>
<td>882</td>
<td>pollution</td>
<td>60</td>
<td>16</td>
</tr>
<tr>
<td>D05</td>
<td>1104</td>
<td>leukemia</td>
<td>72</td>
<td>7130</td>
<td>D06</td>
<td>1446</td>
<td>CostaMadre1</td>
<td>296</td>
<td>38</td>
</tr>
<tr>
<td>D07</td>
<td>1453</td>
<td>PieChart3</td>
<td>1077</td>
<td>38</td>
<td>D08</td>
<td>1458</td>
<td>arcene</td>
<td>200</td>
<td>10001</td>
</tr>
<tr>
<td>D09</td>
<td>1485</td>
<td>madelon</td>
<td>2600</td>
<td>501</td>
<td>D10</td>
<td>1566</td>
<td>hill-valley</td>
<td>1212</td>
<td>101</td>
</tr>
<tr>
<td>D11</td>
<td>37</td>
<td>diabetes</td>
<td>768</td>
<td>9</td>
<td>D12</td>
<td>1005</td>
<td>glass</td>
<td>214</td>
<td>10</td>
</tr>
<tr>
<td>D13</td>
<td>1494</td>
<td>qsar-biodeg</td>
<td>1055</td>
<td>42</td>
<td>D14</td>
<td>1134</td>
<td>OVA_Kidney</td>
<td>1545</td>
<td>10937</td>
</tr>
<tr>
<td>D15</td>
<td>1217</td>
<td>Click_prediction_small</td>
<td>149639</td>
<td>12</td>
<td>D16</td>
<td>1149</td>
<td>AP_Ovary_Kidney</td>
<td>458</td>
<td>10937</td>
</tr>
<tr>
<td>D17</td>
<td>907</td>
<td>chscase_census4</td>
<td>400</td>
<td>8</td>
<td>D18</td>
<td>976</td>
<td>kdd_JapaneseVowels</td>
<td>9961</td>
<td>15</td>
</tr>
<tr>
<td>D19</td>
<td>1443</td>
<td>PizzaCutter1</td>
<td>61</td>
<td>38</td>
<td>D20</td>
<td>871</td>
<td>pollen</td>
<td>3848</td>
<td>6</td>
</tr>
</tbody>
</table>
Example: DP RF (II)

• the Random Forests (RF): the original RF formulation
• the Random Rotation Ensembles (RRE): a recent improvement over the original RF
• the Random Decision Trees (RDT): a fully random RF implementation which is faster to be trained
• the Differentially Private RDT (DPRDT): a RDT formulation which is also DP

Example: DP RF (III)

- kCV for RF, RRE, and RDT
- DP for DPRDT

\[ n_t = 50, \ n_d = 30, \ k = 3 \]
Randomized Functions or Randomized Algorithms?

For studying Randomized Algorithms we have different options
- Hold out
- Stability
- DP

For studying Randomized Functions, instead we only have one powerful option
- PAC-Bayes theory
PAC-Bayes Theory (I)

\[ \pi : \text{Prior over } \mathcal{F} \]
\[ \rho : \text{Posterior over } \mathcal{F} \]
\[ G_\rho(X) : x \sim \mathcal{P}_x, \ f \sim \rho, \ y = f(x) \]
\[ B_\rho(X) : x \sim \mathcal{P}_x, \ y = \mathbb{E}_\rho \{ f(X) \} \]
\[ L(B_\rho) \leq 2L(G_\rho) \]

PAC-Bayes Theory (II)

\[
\mathbb{P}\left\{ \text{KL} \left[ \hat{L}_n^S(G_Q) \| L(G_Q) \right] \geq \frac{\text{KL} + \ln \left[ \frac{2\sqrt{n}}{\delta} \right]}{n} \right\} \leq \delta
\]

\[
\text{KL}(\rho \| \pi) = \mathbb{E}_\rho \left\{ \ln \left( \frac{\rho}{\pi} \right) \right\} = O \left( \sqrt{\frac{\ln(n)}{n}} \right) \div O \left( \frac{\ln(n)}{n} \right)
\]

Catoni’s Result: Catoni Randomized Function (CRF)

\[ \pi(f) = \frac{1}{Z} \exp(-\gamma L(f)) \]

\[ \rho(f) = \frac{1}{Z'} \exp(-\gamma \hat{L}_n^s(f)) \]

\[ \text{KL}(\rho||\pi) \leq \frac{\gamma^2}{2n} + \gamma \sqrt{\frac{2 \ln \left[ \frac{2\sqrt{n}}{\delta} \right]}{n}}, \quad \text{at:} \ (1 - \delta) \]

Why not a Catoni Randomized Algorithm (CRA)?

Instead of building CDF based on the Catoni’s posterior we can think about a Randomized Algorithm which chooses, inside our space of function, the best function (the one with the smallest empirical error) perturbed with the Catoni’s noise (CRA).

CRA is DP

\[
\begin{align*}
\mathbb{P}\{\mathcal{A}(s) = f\} & = \frac{e^{-\frac{\gamma}{n} \sum_{i=1}^{n} \ell(f, z_i)}}{\sum_{f_1 \in \mathcal{F}} e^{-\frac{\gamma}{n} \sum_{i=1}^{n} \ell(f_1, z_i)}} \\
\mathbb{P}\{\mathcal{A}(\dot{s}) = f\} & = \frac{\sum_{f_1 \in \mathcal{F}} e^{-\frac{\gamma}{n} \sum_{i=1,i \neq j}^{n} \ell(f_1, z_i) + \ell(f, \dot{z}_j)}} {e^{-\frac{\gamma}{n} \sum_{i=1,i \neq j}^{n} \ell(f, z_i) + \ell(f, \dot{z}_j)}}
\end{align*}
\]

\[\leq \frac{e^{0}}{\sum_{f_1 \in \mathcal{F}} e^{-\frac{\gamma}{n} \sum_{i=1,i \neq j}^{n} \ell(f_1, z_i)}} \frac{\sum_{f_1 \in \mathcal{F}} e^{-\frac{\gamma}{n} \sum_{i=1,i \neq j}^{n} \ell(f_1, z_i) e^{0}}}{e^{-\frac{\gamma}{n}}}
\]

\[= e^{\frac{2\gamma}{n}}.
\]

CRF and CRA Generalization Properties

**CRA**

\[
\gamma = \frac{1}{2} \sqrt{n \ln \left( \frac{3\sqrt{2}}{2\delta} \right)}
\]

\[
\rightarrow \mathbb{P}_{S, F} \left\{ Q \left[ \frac{\delta^2}{18}; n \hat{L}_n^S(F), n - n \hat{L}_n^S(F') + 1 \right] \leq L(F') \leq Q \left[ 1 - \frac{\delta^2}{18}; n \hat{L}_n^S(F) + 1, n - n \hat{L}_n^S(F') \right] \right\} \leq \delta
\]

**CRF**

\[
\gamma = \frac{1}{2} \sqrt{n \ln \left( \frac{3\sqrt{2}}{2\delta} \right)}
\]

\[
\rightarrow \mathbb{P}_S \left\{ k_1 [\hat{L}_n^S(G_Q) || L(G_Q)] \geq \frac{\ln \left( \frac{3\sqrt{2}}{\delta} \right)}{8n} + \sqrt{\frac{2 \ln \left( \frac{3\sqrt{2}}{\delta} \right) \ln \left( \frac{2\sqrt{n}}{\delta} \right)}{4n} + \ln \left( \frac{2\sqrt{n}}{\delta} \right)} \right\} \leq 2\delta.
\]

\[
O \left( \frac{1}{\sqrt{n}} \right) \div O \left( \frac{1}{n} \right)
\]

\[
O \left( \frac{4}{\sqrt{\ln(n)/n}} \right) \div O \left( \sqrt{\frac{\ln(n)}{n}} \right)
\]
Example: CRF and CRA

Functions space counts of trees built with an hold out set.

Generalization Error

Dataset

\( n_t = 50, k = 3 \)
Non-Adaptive Data Analysis (I)

NAS: the non-adaptive setting is the case when the procedures for building our models exploit just the training set.

---

**Algorithm 1: Union Bound for the NAS**

**Input:** $s_t, s_h$, and $P_1, \ldots, P_m$

**Output:** $\hat{L}_n^{s_h}(f_1), \ldots, \hat{L}_n^{s_h}(f_m)$

1. for $i \leftarrow 1$ to $m$ do
2. \hspace{1em} $f_i = P_i(s_t)$ and compute $\hat{L}_n^{s_h}(f_i)$;

---

Non-Adaptive Data Analysis (II)

NAS: the non-adaptive setting is the case when the procedures for building our models exploit just the training set.

Then we can use the Bonferroni Correction:

\[
P_{S_h} \left\{ \exists i \in \mathcal{I}_m : \left| L(\mathcal{P}_i(s_t)) - \hat{L}_{n}^{S_h}(\mathcal{P}_i(s_t)) \right| \geq \sqrt{\frac{\ln \left( \frac{2m}{\delta} \right)}{2n}} \right\} \leq \delta
\]

\[
O \left( \sqrt{\frac{\ln(m)}{n}} \right) \div O \left( \frac{\ln(m)}{n} \right)
\]

Adaptive Data Analysis (I)

AS: the adaptive setting is the case when the procedures for building our models exploit both the training set and the performance of the procedure at previous step over the hold out set.

Algorithm 1: Hold out for the AS

<table>
<thead>
<tr>
<th>Input: $s_t, s_h, \text{ and } P_1, \cdots, P_m$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Output: $\hat{L}^{s_1}<em>{n_h}(f_1), \cdots, \hat{L}^{s_m}</em>{n_h}(f_m)$</td>
</tr>
<tr>
<td>1 Split $s_h$ in $s^i_h$ with $i \in \mathcal{I}_m$;</td>
</tr>
<tr>
<td>2 for $i \leftarrow 1$ to $m$ do</td>
</tr>
<tr>
<td>3 $f_i = \mathcal{P}<em>i \left( s_t, \hat{L}^{s_1}</em>{n_h}(f_1), \cdots, \hat{L}^{s_{i-1}}<em>{n_h}(f</em>{i-1}) \right)$ and compute $\hat{L}^{s_i}_{n_h}(f_i)$;</td>
</tr>
</tbody>
</table>

Adaptive Data Analysis (II)

AS: the adaptive setting is the case when the procedures for building our models exploit both the training set and the performance of the procedure at previous step over the hold out set.

Then we need one test set at each step:

\[ f_i = P_i(s_t, P_{i-1}, \ldots, P_1) \]

\[
\mathbb{P}_{S_i^h} \left\{ \exists i \in I_m : \left| L(f_i) - \hat{L}_{n^h}^S(f_i) \right| \leq \sqrt{\frac{m \ln \left( \frac{2}{\delta} \right)}{2n}} \right\} \leq \delta
\]

\[
O \left( \sqrt{\frac{m}{n}} \right) \div O \left( \frac{m}{n} \right)
\]

Thresholdout

The idea is to look at the test set error only when is far from the one on the training set, but, when we look at the error on the test set, we look at it in a “private way”.

Algorithm 1: Thresholdout for the AS

Input: \(s_t, s_h, T, \sigma, B\), and \(P_1, \ldots, P_m\)

Output: \(a_1, \ldots, a_m\)

1. \(\gamma \sim \text{Lap}(2\sigma), \quad \hat{T} = T + \gamma;\)
2. for \(i \leftarrow 1\) to \(m\) do
3. \hspace{1em} if \(B < 1\) then
4. \hspace{1em} \hspace{1em} \(a_i = \perp;\)
5. \hspace{1em} else
6. \hspace{1em} \hspace{1em} \(f_i = P_i(s_t, a_1, \ldots, a_{i-1}), \quad \eta \sim \text{Lap}(4\sigma);\)
7. \hspace{1em} \hspace{1em} if \(\hat{L}_n^{s_h}(f_i) - \hat{L}_n^{s_t}(f_i) \geq \hat{T} + \eta\) then
8. \hspace{1em} \hspace{1em} \hspace{1em} \(\xi \sim \text{Lap}(\sigma), \quad \gamma \sim \text{Lap}(2\sigma), \quad \hat{T} = T + \gamma, \quad B = B - 1;\)
9. \hspace{1em} \hspace{1em} \hspace{1em} \(a_i = \hat{L}_n^{s_h}(f_i) + \xi;\)
10. \hspace{1em} \hspace{1em} else
11. \hspace{1em} \hspace{1em} \hspace{1em} \(a_i = \hat{L}_n^{s_t}(f_i);\)


Hoeffding-type Bound

We obtain a result which is analogous to the one of the NAS setting

$$\mathbb{P}_{A_i, F_i} \left\{ \exists i \in \mathcal{I}_m : A_i \neq \bot \land |A_i - L(F_i)| \geq 40 \sqrt{B \ln \left( \frac{12m}{\beta} \right)} \right\} \leq \beta$$

$$O \left( \sqrt{\frac{\ln(m)}{n}} \right)$$

Chernoff-type Bound

We can obtain a sharp result (at least asymptotically).

\[ t = 40 \sqrt{\frac{B \ln \left( \frac{12m}{\beta} \right)}{n}} \]

\[ \mathbb{P}_{A_i, F_i} \left\{ \exists i \in \mathcal{I}_m : A_i \neq \bot \land |A_i - L(F_i)| \geq 30 \sqrt{A_i t} + 50t^2 \right\} \leq \beta \]

\[ O \left( \sqrt{\frac{\ln(m)}{n}} \right) \div O \left( \frac{\ln(m)}{n} \right) \]

Open Problems

• Is privacy reducing our ability to learn something from data?

• Can we improve the rate of convergence and the constants in the bounds?

• How can we exploit privacy to derive new learning algorithms?

• Can we improve the Thresholdout?

• How many times can we access the data without compromising the privacy?