Differential Privacy For Adaptive Queries

Michael Lee
Medinah,Ill

1 Introduction

This document contains notes and expositions of the proofs used to justify and prove the efficacy of using differential privacy to reuse a holdout or validation data set in an optimization paradigm where the metrics computed on the validation or holdout data set are used to modify or optimize an algorithm on a separate training. This allows one to train on the training set and validate on the holdout data set continually, up to a defined, but parameterized, limit or "budget".

Note that this is a very deep subject and I make no claims of completeness or rigor. This is a conceptual document that takes the mathematical exposition as far as necessary to facilitate a deeper understanding.

1.1 Definitions and Problem Setup

We have a function, $\phi : \chi \to [0,1]$, on the distribution P, referred to as a linear functional of P. A request for an approximation to the expectation of a bounded function on χ, $P[\phi] = \mathbb{E}_{x\sim P}\phi(x)$ of some function of P, is called a statistical query.

A data set, $S = (x_1, \ldots, x_n)$ consists of n samples drawn randomly and independently from the distribution P over a discrete universe χ of possible data points. A natural estimator of $P[\phi]$ is $\mathcal{E}_S \equiv \frac{1}{n}\sum_{i=1}^{n}\phi(x_i)$. The Hoeffding inequality says that for a fixed function ϕ the probability (over the choice of data set) that the estimator $\mathcal{E}_S$ has an error greater than τ is no more than $2e^{-2\tau^2 n}$.

2 Max-Information and Differential Privacy

While this section may be very technical, it illustrates, in detail the concepts that reusable holdout methodologies are based on. It provides the machinery necessary to quantify the amount of information transferred back-and-forth in an adaptive query scenario.

2.1 max-information

Definition 2.1. Let $\mathbf{X}$ and $\mathbf{Y}$ be jointly distributed random variables. The max-information between $\mathbf{X}$ and $\mathbf{Y}$, denoted $I_\infty(\mathbf{X};\mathbf{Y})$, is the minimal value of k such that for every x in the support of $\mathbf{X}$ and every y in support of $\mathbf{Y}$ we have $P[\mathbf{X}=x|\mathbf{Y}=y] \le 2^k P[\mathbf{X}=x]$

This definition is a lower bound on the amount of dependence that $\mathbf{X}$ has on $\mathbf{Y}$. The random variable $\mathbf{S}$ is drawn i.i.d. from P^n and random variable ϕ is as above, where an analyst may arrive at ϕ with foreknowledge of other ϕ from ϕ. Let's say for each function ϕ in support of ϕ we have a set of bad data sets, $R(\phi)$, which cause the empirical value $\mathcal{E}_S[\phi]$ to be far from the true value $P[\phi]$, i.e. ϕ overfits to S. The max-information we defined above will allow us to bound the probability of overfitting, $P[S \in R(\phi)]$:

Theorem 2.1. *For $k = I_\infty(\mathbf{S},\phi)$, $P[\mathbf{S}\in R(\phi)] \le 2^k \max_\phi P[\mathbf{S}\in R(\phi)]$*

Proof. Since $k = I_\infty(\mathbf{S},\phi)$, we need $P[\mathbf{S}=S|\phi=\phi] \le 2^k P[\mathbf{S}=s]$ to be true for all S and ϕ. Then,

$$\begin{aligned}
P[\mathbf{S}\in R(\phi)] &= \sum_\phi P[\mathbf{S}\in R(\phi)|\phi=\phi]P[\phi=\phi] \\
&= \sum_\phi P[\phi=\phi] \sum_{S\in R(\phi)} P[\mathbf{S}=S|\phi=\phi] \\
&\le \sum_\phi P[\phi=\phi] \sum_{S\in R(\phi)} 2^k P[\mathbf{S}=S)] \\
&= 2^k \sum_\phi P[\phi=\phi] \sum_{S\in R(\phi)} P[\mathbf{S}=S] \\
&= 2^k \sum_\phi P[\phi=\phi] P[\mathbf{S}\in R(\phi)] \\
&\le 2^k \sum_\phi P[\phi=\phi] \max_\phi P[\mathbf{S}\in R(\phi)] \\
&\le 2^k \max_\phi P[\mathbf{S}\in R(\phi)] \sum_\phi P[\phi=\phi] \\
&\le 2^k \max_\phi P[\mathbf{S}\in R(\phi)]
\end{aligned}$$

□

where $\sum_\phi P[\phi=\phi] = 1$ has been used in the last step. What does this mean? If we limit the mutual information between $\mathbf{S}$ and ϕ, we can bound the probability of the non-desirable outcome $\mathbf{S}\in R(\phi)$.

If we define this non-desirable outcome as

$$R_\tau = \{S \in \chi^n : \mathcal{E}_S[\phi] - p[\phi] > \tau\} \tag{1}$$

and we use Hoeffding's inequality, then we can bound the probability of overfitting by a function of accuracy (τ) and the size of the data set (n):

Corollary 2.1.1. *If $I_\infty(\mathbf{S};\phi) \le (\log_2 e)\tau^2 n$, then $P[\mathbf{S}\in R_\tau(\phi)] \le \exp(-\tau^2 n)$*

2.2 Differential Privacy

Let's say we have two data sets, x and y that differ by only one record or data point. We then refer to x and y as *adjacent*.

Definition 2.2. For randomized algorithm, $\mathcal{M}$ with domain $\mathcal{X}^n$ is (ε,δ) differentially private if for all $\mathbf{S} \in Range(\mathcal{M})$ and for all pairs of adjacent data sets $x, y \in \mathcal{X}^n$:

$$P[\mathcal{M}(x)\in S] \le \exp(\varepsilon)P[\mathcal{M}(y)\in S] + \delta \tag{2}$$

where the probability is over the random variable $\mathcal{M}$. The case where $\delta = 0$ is the *pure differential privacy* case and is sometimes referred to as ε-differentially private.

Intuition: If $|\varepsilon|$ is small, then the probability of getting S for $\mathcal{M}(x)$ is pretty much the same as for $\mathcal{M}(y)$. In other words, it's difficult, from a statistical perspective, to learn much about the record in which x and y differ by.

2.3 Using Differential Privacy to bound max-information

Lemma 2.2. *Let $\mathcal{M}$ be an ε-differentially private algorithm. Let $\mathbf{S}$ be any random variable over n-element input data sets for $\mathcal{M}$ and let $\mathbf{Y}$ be the corresponding output distribution $\mathbf{Y} = \mathcal{M}(\mathbf{S})$. Then, $I_\infty(\mathbf{S};\mathbf{Y}) \le (\log_2 e)\varepsilon n$*

Proof. $I_\infty(\mathbf{S};\mathbf{Y}) = I_\infty(\mathbf{Y};\mathbf{S})$ by Bayes' rule. Since any two data sets S and S' differ by at most n records,

$$P[\mathbf{Y}=y|\mathbf{S}=S] \le e^{\varepsilon n}P[\mathbf{Y}=y|\mathbf{S}=S'] \tag{3}$$

See [2] . Since for every data set S', there must exist a y such that $P[\mathbf{Y}=y|\mathbf{S}=S'] \le P[\mathbf{S}=S']$, we have

$$P[\mathbf{Y}=y|\mathbf{S}=S] \le e^{\varepsilon n}P[\mathbf{Y}=y] \tag{4}$$

which gives $I_\infty(\mathbf{S};\mathbf{Y}) \le (\log_2 e)\varepsilon n$ □

From 2.2 and 2.1.1, if $\mathcal{M}$ is such that the interaction with a data set is τ^2 differentially private, then we're able to specify the probability that that interaction causes us to overfit.

2.4 Stronger bounds

We required that $\varepsilon = \tau^2$ to bound the probability of overfitting in the previous section. τ^2 It's possible to use differential privacy to achieve $\varepsilon = \tau$ and to use (ε, δ)-differential privacy with $\delta > 0$. The following are stated without proof, but detailed proofs can be found in [1]. We define the probability of that a function overfits a random function, or overfitting, as $\beta = e^{-2\tau^2 n}$

Theorem 2.3. *Let $\mathcal{M}$ be an ε-differentially private algorithm and let $\mathbf{S}$ be a random variable drawn from a distribution $\mathcal{P}^n$ ranging over $\mathcal{X}^n$. Let $Y = \mathcal{M}(S)$ be the corresponding output distribution. Assume that for each element y in the range of $\mathcal{M}$ there is subset $R(y) \in \mathcal{X}^n$ so that $\max_y P[\mathbf{S} \in R(y)] < \beta$. Then for $\varepsilon \leq \sqrt{\frac{\log \frac{1}{\beta}}{2n}}$ we have $P[\mathbf{S} \in R(\mathbf{Y})] \leq \frac{3}{\sqrt{\beta}}$*

When we have an algorithm that generates a function $f : \mathcal{X} \mapsto [0,1]$, we have

Corollary 2.3.1. *Let $\mathcal{M}$ be an ε-differentially private algorithm that outputs a function from $\mathcal{X} \mapsto [0,1]$. For a random variable $\mathbf{S}$ distributed according to $\mathcal{P}$ we let $\phi = \mathcal{M}(\mathbf{S})$. Then, for any $\tau > 0$, setting $\varepsilon \leq \tau$ ensures that $P[|\mathcal{P}(\phi) - \mathcal{E}_{\mathcal{S}}(\phi)| \geq \tau] \leq 6e^{-\tau^2 n}$*

and relaxing the differential privacy requirement, we have

Theorem 2.4. *Let $\mathcal{M}$ be an (ε, δ)-differentially private algorithm that outputs a function from $\mathcal{X} \mapsto [0,1]$. For a random variable $\mathbf{S}$ distributed according to $\mathcal{P}^n$, we let $\phi = \mathcal{M}(\mathbf{S})$. Then for any tau > 0 and $n \geq 48 \log(8/\beta)/\tau^2$, setting $\varepsilon \leq \tau/4$ and $\delta \leq (\beta/8)^{(4/\tau)}$ ensures*

$$P[|\mathcal{P}(\phi) - \mathcal{E}_{\mathcal{S}}(\phi)| \geq \tau] \leq \beta \tag{5}$$

Note that τ is the "approximately" part of a PAC learnable algorithm and β is the "probably" part.

3 Thresholdout

Theorem 2.4 is for a differentially private algorithm, but our use case is where an analyst has unfettered access to the training data set, S_t, but interacts in a differentially private way with the holdout data set, S_h via a "fuzzified" estimate of $\phi(S_h)$. To reconcile this, we have the following lemma:

Lemma 3.1. *If $\mathcal{A}$ is an (ε, δ)-differentially private algorithm with domain $\mathcal{X}^n$ and range $\mathcal{O}$, and $\mathcal{B}$ is any, possibly randomized, algorithm with domain $\mathcal{O}$ and range $\mathcal{O}'$, then the algorithm $\mathcal{B} \circ \mathcal{A}$ with domain $\mathcal{X}^n$ and range $\mathcal{O}'$ is also (ε, δ)-differentially private.*

Armed with this knowledge, we enumerate the *Thresholdout* algorithm:

3.1 Algorithm

Algorithm 1 Thresholdout

```
For a given function $\phi : \mathcal{X} \mapsto [0,1]$
$\hat{T} \leftarrow T + \gamma$ for $\gamma \sim Lap(2\sigma)$
while $B > 0$ do
    Sample $\xi \sim Lap(\sigma)$, $\gamma \sim Lap(2\sigma)$, $\eta \sim Lap(4\sigma)$
    if $|\mathcal{E}_{S_h}[\phi] - \mathcal{E}_{S_t}[\phi]| > \hat{T} + \eta$ then
        Output $\mathcal{E}_{S_h}[\phi] + \xi$        ▷ where privacy gets implemented
        $B \leftarrow B - 1$, $\hat{T} \leftarrow T + \gamma$
    else
        Output $\mathcal{E}_{S_t}[\phi]$
```

Now, we state the guarantees that *Thresholdout* provides

Lemma 3.2. *(1)Thresholdout satisfies $(2B/(\sigma n), 0)$-differential privacy. (2)Thresholdout also satisfies $(\sqrt{32B \log(2/\delta)}/(\sigma n), \delta)$-differential privacy for any $\delta > 0$.*

The first guarantee states that to achieve the probability and error of Corollary 2.3.1, we need n large enough to achieve $(\varepsilon, 0)$-differential privacy for $\varepsilon = \tau$, or $n > 2B/(\sigma\tau)$. Also, by setting $n > \log(6/\beta)/\tau^2$, we also have that $6e^{-\tau^2 n} < \beta$. Taking those two statements together we can say that n needs to be $n \geq n_0(B, \sigma, \tau, \beta) = \max 2B/(\sigma\tau), \log(6/\beta)/\tau^2$.

The second guarantee can be used to achieve 2.4 by setting $\varepsilon = \tau/4$ and $\delta = (\beta/8)^{4/\tau}$. Then n is required to be $n \geq n_0(B, \sigma, \tau, \beta) = \frac{32\sqrt{2B\log(8/\beta)}}{\tau^{3/2}\sigma} + \frac{16\sqrt{\log(2)B}}{\tau\sigma}$.

The first depends linearly on B and the second has a better dependence on B, but slightly worse dependence on τ ($1/\tau$ vs. $1/\tau^{3/2}$).

Using the previous results we have an overall generalization bound (i.e. τ and β) for *Thresholdout*.

Theorem 3.3. *Let $\beta, \tau > 0$ and $m \geq B > 0$. Set $T = 3\tau/4$ and $\sigma = \tau/(96 \log(4m/\beta))$. Let $\mathbf{S}_h$ denote a holdout data set of size n drawn i.i.d. from a distribution $\mathcal{P}$ and S_t be any additional (training) set over $\mathcal{X}$. Consider an algorithm that is given access to S_t and adaptively chooses functions $\phi_1, \phi_2, \ldots, \phi_m$ while interacting with Threshold which is given data sets S_t, $\mathbf{S}_h$ and values σ, B, T. For every $i \in [m]$, let $\mathbf{a}_i$ denote the answer on function $\phi_i : \mathcal{X} \mapsto [0,1]$ and for every $i \in [m]$, we define the counter $\mathbf{Z}_i$*

$$\mathbf{Z}_i = |\{j \leq i : |\mathcal{P}[\phi_j] - \mathcal{E}_{S_t}[\phi_j]\}| \tag{6}$$

Then,

$$P[\exists i \in [m], \mathbf{Z}_i < B \wedge |\mathbf{a}_i - \mathcal{P}[\phi_i]| \geq \tau] \leq \beta \tag{7}$$

whenever

$$\begin{aligned} n &\geq \min\{n_0(B, \sigma, \tau/8, \beta/(2m)), n_1(B, \sigma, \tau/8, \beta/(2m))\} \\ &= O\left(\frac{\log m/\beta}{\tau^2}\right) \min\left\{B, \sqrt{B\log(m/\beta)/\tau}\right\} \end{aligned}$$

3.2 Experiment

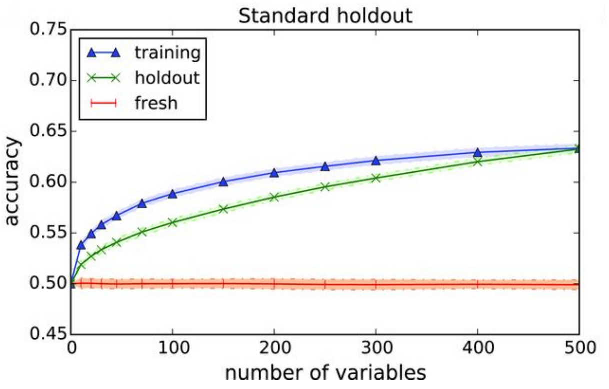

Figure 1: No feature correlation, Standard Holdout

Then, we look at the use of Thresholdout to choose features:

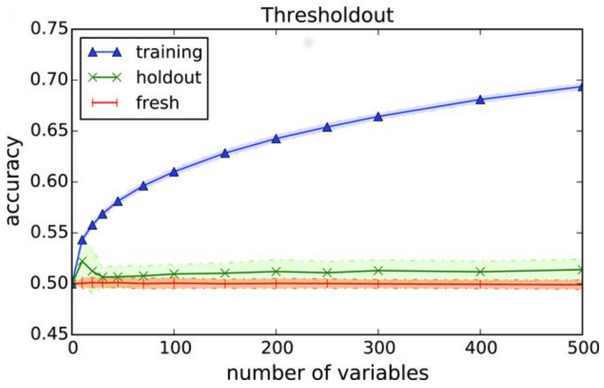

Figure 2: No feature correlation, Thresholdout

Making the first twenty variables correlated:

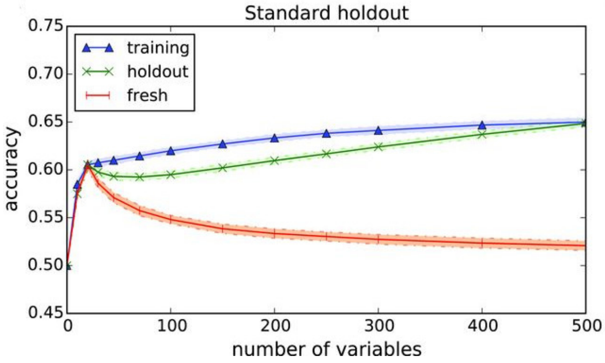

Figure 3: First 20 features correlated, Thresholdout

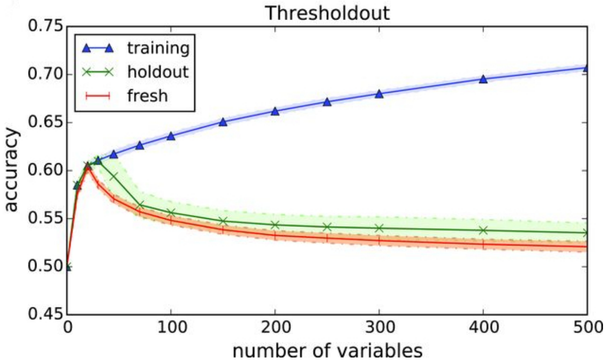

Figure 4: First 20 features correlated, Thresholdout

[1] Moritz Hardt Toniann Pitassi Cynthia Dwork, Vitaly Feldman and Aaron Roth. Preserving statistical validity in adaptive data anlysis. *CoRR*, page abs/1411.2664, 2014.

[2] Cynthia Dwork and Aaron Roth. The algorithmic foundations of differential privacy. *EUROCRYPT*, pages 486–503, 2006.