Oracle-Efficient Differentially Private Learning with Public Data

Adam Block, Mark Bun, Rathin Desai, Abhishek Shetty, and Steven Wu

Warm-up: (Non-Private) Binary Classification

PAC Model [Valiant84]

Known:

Space of examples X

Concept class $C = \{f: X \to \{0,1\}\}$

Unknown:

Distribution P over X

Target concept $c \in C$

$P \xrightarrow{\text{i.i.d.}} D =$

Example	Label
x_1	$c(x_1)$
x_2	$c(x_2)$
…	…
x_n	$c(x_n)$

→ Learner L → Hypothesis $h: X \to \{0,1\}$

C is PAC learnable if there is a L such that for all P and all c

$$\Pr[h(x) \neq c(x)] \leq 0.01$$

with randomness over D and L

Differential Privacy

[DN03, DMNS06]

D =

D_1
D_2
D_3
…
D_n

D' =

D_1
D_2
D'_3
…
D_n

D and D' are *neighbors* if they differ by at most one row

A private algorithm needs to have close output distributions on any pair of neighbors

Definition: A (randomized) algorithm A is (ε, δ)-differentially private if for all neighbors D, D' and every $S \subseteq \text{Range}(A)$

$$\Pr[A(D) \in S] \leq e^{\varepsilon} \Pr[A(D') \in S] + \delta$$

Private Binary Classification

Differentially Private PAC Model

[Kasiviswanathan-Lee-Raskhodnikova-Nissim-Smith08]

$P \xrightarrow{\text{i.i.d.}} D =$

Example	Label
x_1	$c(x_1)$
x_2	$c(x_2)$
…	…
x_n	$c(x_n)$

→ (ϵ, δ)-*Differentially Private* Learner L → Hypothesis $h: X \to \{0,1\}$

Definition: An algorithm *L* is (ϵ, δ)-differentially private if for all pairs of *D*, *D'* that differ by one example and every S ⊆ Range(L)

$$\Pr[L(D) \in S] \leq \exp(\epsilon) \Pr[L(D') \in S] + \delta$$

But…

- Statistical Feasibility:
  - Littlestone dimension is a pessimistic *worst-case measure*
  - Rules out simple functions (e.g., thresholds, half-spaces)
  - Does not reflect recent empirical advances in DP ML

Can we leverage external information to sidestep Littlestone dimension lower bound?

Private Learning with Unlabeled Public Data

[Beimel-Nissim-Stemmer'14, Alon-Bassily-Moran'19…]

$P \xrightarrow{\text{i.i.d.}} D =$

Example	Label
x_1	y_1
x_2	y_2
$\dots$	$\dots$
x_n	y_n

→ Learner L

(ϵ, δ)-

Differentially Private

w.r.t. D only

$P_{pub} \xrightarrow{\text{i.i.d.}} D_{\text{pub}} =$

Example
$\tilde{x}_1$
$\tilde{x}_2$
$\dots$
$\tilde{x}_m$

Unlabeled Public Data:

$P_{\text{pub}} \approx P_X$ (by some f-divergence)

Analogous to smooth online learning

[Rakhlin-Sridharan-Tewari'11,

Haghtalab-Roughgarden-Shetty'20,

Haghtalab-Han-Shetty-Yang'22,

Block-Simchowitz'22,

Block-Simchowitz-Rakhlin'23…]

Formulation of Computational Efficiency

Oracle Efficiency:

Abstraction for powerful solvers for non-convex optimization (e.g., SGD, integer program solvers…)

Assume access to an *oracle* that can solve (non-private) empirical risk minimization problem of the form:

$$\arg\min_{h \in C} \sum_{i=1}^{n} \ell(h(x_i), y_i)$$

Solve the private learning problem efficiently (in polynomial time)

Prime oracle-efficient in online learning:
Follow-the-perturbed-leader (FTPL)
[Kalai-Vempala'02]

DP Oracle-Efficient Learner with Unlabeled Public Data

[Block-Bun-Desai-Shetty-Wu24]

D_{pub} → Private Learner L

Unlabeled Public Data for Stability

D_{pub}-FTPL:

$$\tilde{f} = \arg\min_{h \in C} \sum_{i=1}^{n} \ell(h(x_i), y_i) + \underbrace{\sum_{\tilde{x}_j \in D_{\text{pub}}} w_j \, \ell(h(\tilde{x}_j), \tilde{y}_j)}_{\text{"Perturbation"}}$$

w_j : Laplace noise; $\tilde{y}_j$: random label in $\{0,1\}$

Label D_{pub} with $\tilde{f}$ →

Example	Label
$\tilde{x}_1$	$\tilde{f}(\tilde{x}_1)$
$\tilde{x}_2$	$\tilde{f}(\tilde{x}_2)$
$\dots$	$\dots$
$\tilde{x}_m$	$\tilde{f}(\tilde{x}_m)$

↓ Solve ERM

Output accurate predictor h with sample size $\approx$ VC-Dim$(C)^2$

Beyond Classification:
Also extends regression with convex, Lipschitz loss function ℓ

Summary

- Designed algorithms based on FTPL, FTRL from online learning that ensure stability to get private learning algorithms.
- Improved previous set of results by giving
  - Oracle efficient algorithms for more general function classes
  - Using public unlabelled data as opposed to public labelled data
  - Minimizing number of calls to the oracle
- First to design learning algorithms for real valued functions.