Learning the Privacy-Utility Trade-off with Bayesian Optimization

Borja Balle

Joint work with B. Avent, J. Gonzalez, T. Diethe and A. Paleyes

Privacy

Theory *vs* Practice

Plot from J. M. Abowd "Disclosure Avoidance for Block Level Data and Protection of Confidentiality in Public Tabulations" (CSAC Meeting, December 2018)

Example: DP-SGD clipping operation ensures that kclip*L*(*v*)k² *L* so that the `2-sensitivity of any gradient to a change in one datapoint in *z*

Input: dataset $z = (z_1, \ldots, z_n)$ **Hyperparameters:** learning rate η , mini-batch size m , number of epochs T, noise variance σ^2 , clipping norm L Initialize $w \leftarrow 0$ for $t \in [T]$ do for $k \in [n/m]$ do Sample $S \subset [n]$ with $|S| = m$ uniformly at random Let $g \leftarrow \frac{1}{m}$ $\sum_{j \in S} \text{clip}_L(\nabla \ell(z_j, w)) + \frac{2L}{m} \mathcal{N}(0, \sigma^2 I)$ Update $w \leftarrow w - \eta g$

- 5+ hyper-parameters affecting both privacy and utility
- For convex problems can be set to achieve near-optimal rates
- For deep learning applications we don't have (good) utility bounds

return *w*

[Bassily et al. 2014; Abadi et al. 2016]

- 1. Efficient to compute $\overline{\mathsf{H}}$ \overline{a}
- 2. Use empirical utility measurements
- 3. Enable fine-grained comparisons $\frac{1}{\sqrt{2}}$ \Box

Desiderata 399 400

Privacy-Utility Pareto Front the Adult and MNIST datasets (respectively) by both DPARETO and random sampling. *Center right:* Pareto fronts learned for MLP2 architecture on the MNIST dataset with DPARETO and random sampling, including the shared points they were both initialized with. *Far right:* Adult dataset DPARETO sampled points and its Pareto front compared to larger set of random sampling points and its Pareto front.

Problem Formulation

 $\lambda \in \Lambda \}$

Parametrized Algorithm Class

$$
\mathcal{A}=\{A_{\lambda}:Z\rightarrow W
$$

Error (Utility) Oracle

 $E: \Lambda \rightarrow [0,1]$

Privacy Oracle

 $P: \Lambda \rightarrow [0, \infty)$

Eg. DP-SGD

Eg. Expected classification error

Eg. Epsilon for fixed delta

Hyper-parameter Space

Privacy Loss

Error

Hyper-parameter Space

Hyper-parameter Space

Hyper-parameter Space

Hyper-parameter Space

Hyper-parameter Space

Hyper-parameter Space

Hyper-parameter Space

Hyper-parameter Space

Bayesian Optimization (BO)

- Gradient-free optimization for blackbox functions
- Widely used in applications (HPO in ML, scheduling & planning, experimental design, etc)
-
-
-
-
- -
-

Bayesian Optimization (BO)

Goal: $\lambda^* = \argmin F(\lambda)$ λ∈Λ

- Gradient-free optimization for blackbox functions
- Widely used in applications (HPO in ML, scheduling & planning, experimental design, etc)

Input: $F : \Lambda \subset \mathbb{R}^p \to \mathbb{R}$

Expensive, non-convex, smooth

Bayesian Optimization (BO)

Goal: $\lambda^* = \argmin F(\lambda)$ λ∈Λ

- Gradient-free optimization for blackbox functions
- Widely used in applications (HPO in ML, scheduling & planning, experimental design, etc)

Input: $F : A \subset \mathbb{R}^p \to \mathbb{R}$

Expensive, non-convex, smooth

Bayesian Optimization Loop:

Given k evaluations $(\lambda_1, F(\lambda_1)), \ldots, (\lambda_k, F(\lambda_k))$

1. Build a surrogate model for F (eg. Gaussian process)

2. Find most promising next evaluation

The DPareto Algorithm our notation. *k* times until the optimization budget is used up. Further implementation details are provided in Appx. E.1.

Fit a GP to the transformed privacy using *D* Fit a GP to the transformed utility using *D* Optimize the HVPoI acquisition function in Eq. (2) using anti-ideal point v^{\dagger} and obtain a new query point λ Evaluate oracles $v \leftarrow (P(\lambda), E(\lambda))$ Augment dataset $\mathcal{D} \leftarrow \mathcal{D} \cup \{(\lambda, v)\}$

return *Pareto front* $\mathcal{PF}(\lbrace v | (\lambda, v) \in \mathcal{D} \rbrace)$

- 255 $\overline{}$ using *multi-objective* Bayesian 258 259 from the data in *V*. Furthermore, hypervolume can be used • Find privacy-utility Pareto front increment in the hypervolume given a new point *^v* ² ^R*^p*: optimization
- \overline{a} **2** USE transformed daussian processes to model privacy and error oracles 265 lecting a new hyperparameter can be computed using the • Use transformed Gaussian processes to model privacy and
- Acquisition function optimizes 268 $\overline{}$ improvement l model trained on *I* as follows: Poir fanolion oplinises
Pluma heead prohebility of $\overline{}$ *^v*2˜ \overline{V} hyper-volume based probability of *^j* (; *v^j*)*dv^j , improvement* [Couckuyt et al. 2014]
- **Input:** hyperparameter set Λ , privacy oracle P , error oracle E, anti-ideal point *v†*, number of initial points k_0 , number of iterations k , prior GP
- Initialize dataset $\mathcal{D} \leftarrow \emptyset$
- for $i \in [k_0]$ do
	- Sample random point $\lambda \in \Lambda$
	- Evaluate oracles $v \leftarrow (P(\lambda), E(\lambda))$
	- Augment dataset $\mathcal{D} \leftarrow \mathcal{D} \cup \{(\lambda, v)\}$

for $i \in [k]$ do

Example: Sparse Vector Technique more details).

- 100 queries with 0/1 output, sensitivity 1
- 10% queries return 1 (randomly selected) ly selected)
- 10² 10¹ 10⁰ 10¹ 10² • Privacy: SVT analysis
- b Error: 1 F-score (avg. over 50 runs)

25 return *w* 30 \mathbf{I} if $c \geq C$ then return w 1 *F*¹ Input: dataset *z*, queries *q*1*,...,q^m* Hyperparameters: noise *b*, bound *C* $c \leftarrow 0, w \leftarrow (0, \ldots, 0) \in \{0, 1\}^m$ $b_1 \leftarrow b/(1 + (2C)^{1/3}), b_2 \leftarrow b - b_1, \rho \leftarrow \textsf{Lap}(b_1)$ for $i \in [m]$ do $\nu \leftarrow \textsf{Lap}(b_2)$ if $q_i(z)+\nu\geq \frac{1}{2}+\rho$ then $w_i \leftarrow 1, c \leftarrow c + 1$ *[Lyu et al. 2017]*

3 **Setup**

Example: Sparse Vector Technique

Implementing the Oracles

Privacy Oracle

- Epsilon for fixed delta / Others DP variants / Attacks success metrics
- Closed-form expression / Numerical calculation (eg. moments accountant)

- Fixed input / Distribution over inputs / Worst-case (over a set of) inputs
- On expectation / With high probability
- Exact expression / Empirical evaluation

Error Oracle

Implementing the Oracles

Privacy Oracle

-
-

• Epsilon for fixed delta / Others DP variants / Attacks success metrics

• Closed-form expression / Numerical calculation (eg. moments accountant)

• Fixed input / Distribution over inputs / Worst-case (over a set of) inputs

Error Oracle

-
- On expectation / With high probability
- Exact expression / Empirical evaluation

Machine Learning Experiments

- Adult dataset (n=40K, d=123)
	- Logistic regression (SGD and ADAM)
	- Linear SVM (SGD)
- MNIST dataset (n=60K, d=784)
	- MLP1 (1000 hidden)
	- MLP2 (128-64 hidden)

DPareto *vs* Random Sampling

Conclusion

- Empirical privacy-utility trade-off evaluation enables application-specific decisions
- Bayesian optimization provides computationally efficient method to recover the Pareto front (esp. with large number of hyper-parameters)
- **Future work:**
	- Address leakage in Pareto front (when error oracle is input-specific)
	- Include further criteria (eg. running time of parametrized algorithm)