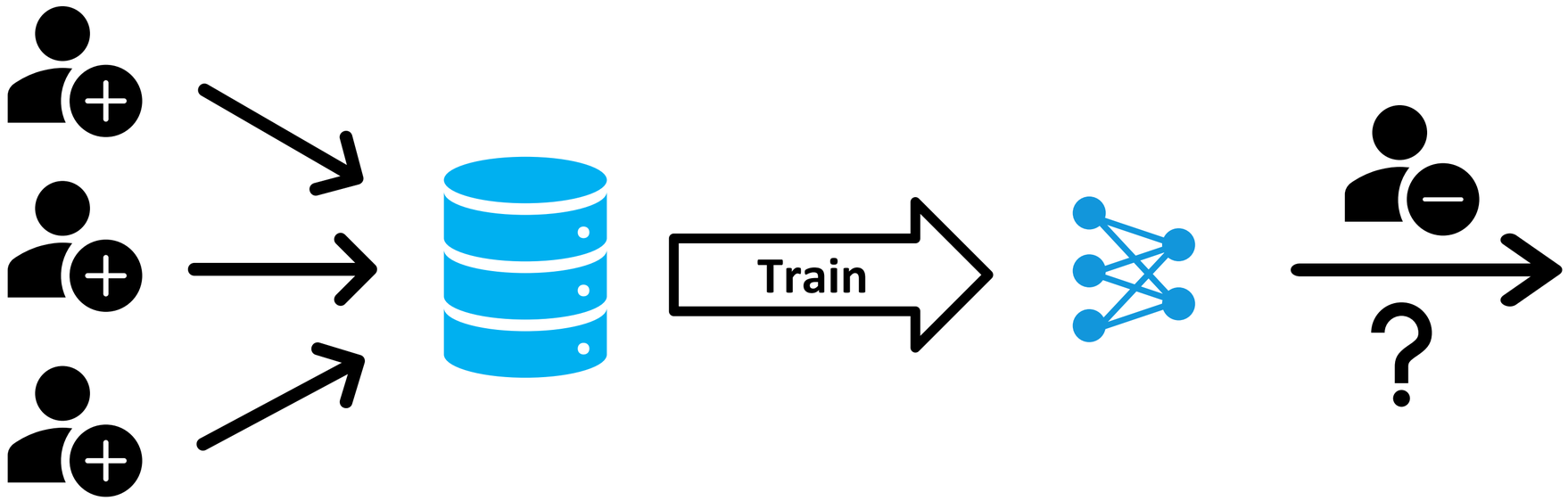
Machine Unlearning

Yufang Liu

East China Normal University

Background

- Data Protection & Right to Erasure (GDPR Article 17)



Certified Data Removal from Machine Learning Models

Chuan Guo [1] **Tom Goldstein** [2] **Awni Hannun** [2] **Laurens van der Maaten** [2]

Cornell University[1], Facebook AI Research[2]

ICML 2020

Setup

Training Dataset

$$\mathcal{D} = \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n)\}$$

Learning Algorithm

$$h^* \leftarrow A(\mathcal{D})$$

Remove Mechanism

$$h^- = M(A(\mathcal{D}), \mathcal{D}, \mathbf{x})$$

Remove Goal

$$M(A(\mathcal{D}), \mathcal{D}, \mathbf{x}) = A(\mathcal{D} \setminus \mathbf{x})$$

Re-training from scratch is expensive !

Certified Removal

- Distributional indistinguishability: For $\epsilon > 0$

$$e^{-\epsilon} \leq \frac{P(M(A(\mathcal{D}), \mathcal{D}, \mathbf{x}) \in \mathcal{T})}{P(A(\mathcal{D} \setminus \mathbf{x}) \in \mathcal{T})} \leq e^{\epsilon}.$$

  for any subset of hypotheses $\mathcal{T} \subseteq \mathcal{H}$ and any $\mathbf{x} \in \mathcal{D}$
- Compare with differential privacy:

$$e^{-\epsilon} \leq \frac{P(A(\mathcal{D}) \in \mathcal{T})}{P(A(\mathcal{D}') \in \mathcal{T})} \leq e^{\epsilon},$$

  - Can also relax to allow a small failure probability δ
  - Such M is said to be an (ϵ, δ)-certified remove mechanism

Linear Models

- Learning algorithm A minimizes the loss:

$$L(\mathbf{w}; \mathcal{D}) = \sum_{i=1}^{n} \ell(\mathbf{w}^\top \mathbf{x}_i, y_i) + \frac{\lambda n}{2} \|\mathbf{w}\|_2^2,$$

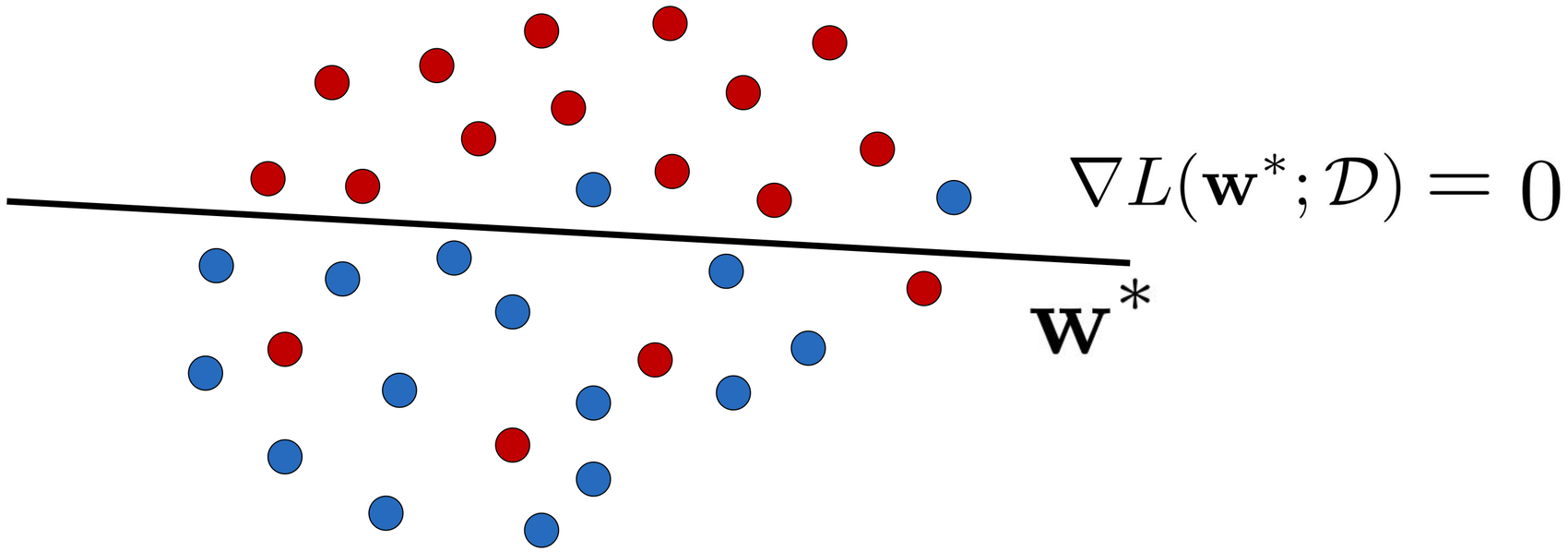
- Assume ℓ is convex

$$\mathbf{w}^* = A(\mathcal{D}) = \text{argmin}_{\mathbf{w}} L(\mathbf{w}; \mathcal{D})$$

  - Assume remove $(\mathbf{x}_n, y_n)$ and let $\mathcal{D}' = \mathcal{D} \setminus (\mathbf{x}_n, y_n)$

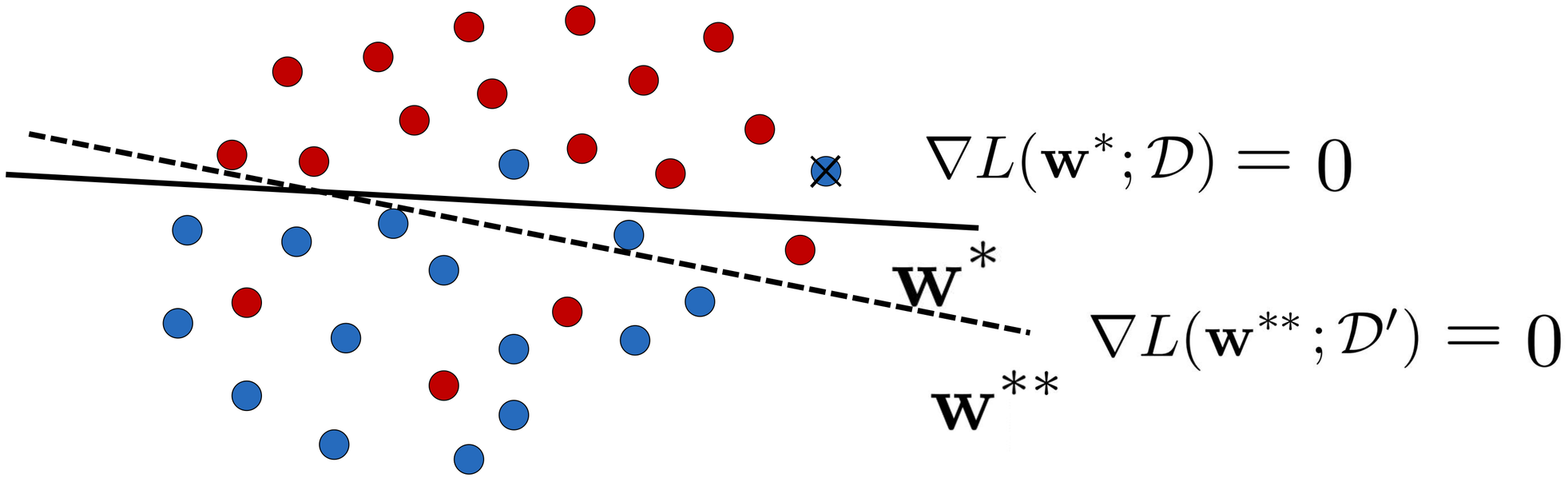
Strategy Overview

- Optimal solutions are characterized by zero gradients



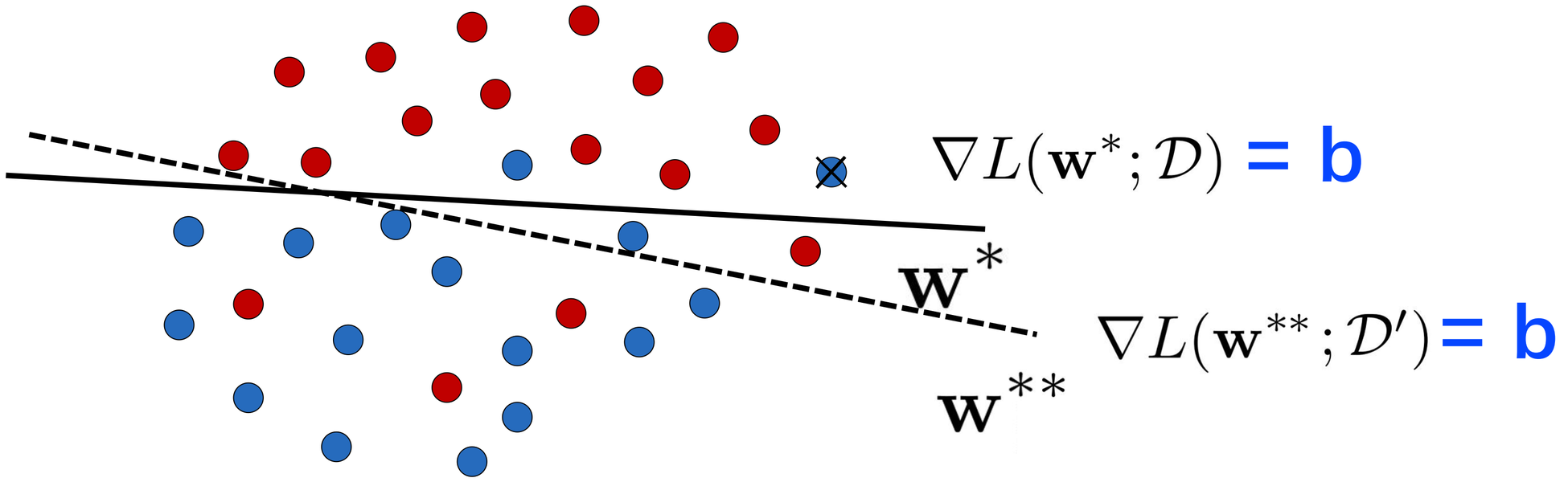
Strategy Overview

- Optimal solutions are characterized by zero gradients
- Apply a fast but inexact update from $\mathbf{w}^*$ to $\mathbf{w}^{**}$
  - Problem: Small but non-zero gradient residual could leak information



Strategy Overview

- Idea: Mask the gradient residual with random noise
  - Masks $\mathbf{w}^-$ indistinguishable from $\mathbf{w}^{**}$ without knowing b
  - Amount of noise added depends on $\nabla L(\mathbf{w}^-; \mathcal{D}')$



Methods

- Sample a random noise vector b and optimize loss

$$L_{\mathbf{b}}(\mathbf{w}; \mathcal{D}) = \sum_{i=1}^{n} \ell\left(\mathbf{w}^\top \mathbf{x}_i, y_i\right) + \frac{\lambda n}{2}\|\mathbf{w}\|_2^2 + \mathbf{b}^\top \mathbf{w},$$

- For every remove request, apply a newton update

$$\mathbf{w}^- = M(\mathbf{w}^*, \mathcal{D}, (\mathbf{x}_n, y_n)) := \mathbf{w}^* + H_{\mathbf{w}^*}^{-1}\Delta,$$

$$H_{\mathbf{w}^*} = \nabla^2 L(\mathbf{w}^*; \mathcal{D}') \quad \Delta = \lambda \mathbf{w}^* + \nabla \ell((\mathbf{w}^*)^\top \mathbf{x}_n, y_n)$$

- …until a "remove budget" is exhausted
- $O(d^2 n)$ offline computation, $O(d^3)$ online computation
- Can be reduced to $O(nd)$ using conjugate gradient iterations

Select Noise Distribution

- Derive a guarantee for gaussian noise :

$$\textit{Suppose that } \|\nabla L(\mathbf{w}^-; \mathcal{D}')\|_2 \leq \epsilon' \textit{ for some } \epsilon' > 0. \textit{ If } \mathbf{b} \sim \mathcal{N}(0, c\epsilon'/\epsilon)^d$$
$$\textit{with } c > 0, \textit{ then } M \textit{ is } (\epsilon, \delta)\textit{-CR for } A \textit{ with } \delta = 1.5 \cdot e^{-c^2/2}.$$

- Multiple removals results in additive gradient residual bound
- Accounting for remove budget:
  - Begin with total budget $\sigma\epsilon/c$
  - Accumulate gradient residual norm

$$\beta \leftarrow \beta + \text{upper bound of } \|\nabla L(\mathbf{w}^-; \mathcal{D}')\|_2$$

  - Once $\beta > \sigma\epsilon/c$, re-train from scratch

Gradient Norm Bound 1

Theorem 1. *Suppose that* $\forall(\mathbf{x}_i, y_i) \in \mathcal{D}, \mathbf{w} \in \mathbb{R}^d : \|\nabla \ell(\mathbf{w}^\top \mathbf{x}_i, y_i)\|_2 \leq C$. *Suppose also that* ℓ'' *is* γ*-Lipschitz and* $\|\mathbf{x}_i\|_2 \leq 1$ *for all* $(\mathbf{x}_i, y_i) \in \mathcal{D}$.

Goal: upper bound of $\|\nabla L(\mathbf{w}^-; \mathcal{D}')\|_2$

$$G(\mathbf{w}) = \nabla L(\mathbf{w}; \mathcal{D}')$$

$$\mathbf{w}_\eta = \mathbf{w}^* + \eta H_{\mathbf{w}^*}^{-1} \Delta$$

By Taylor's Theorem, $\eta \in [0, 1]$

$$\begin{aligned}
G(\mathbf{w}^-) &= G(\mathbf{w}^* + H_{\mathbf{w}^*}^{-1}\Delta) \\
&= G(\mathbf{w}^*) + \nabla G(\mathbf{w}^* + \eta H_{\mathbf{w}^*}^{-1}\Delta) H_{\mathbf{w}^*}^{-1}\Delta \\
&= G(\mathbf{w}^*) + H_{\mathbf{w}_\eta} H_{\mathbf{w}^*}^{-1}\Delta \\
&= (G(\mathbf{w}^*) + \Delta) + H_{\mathbf{w}_\eta} H_{\mathbf{w}^*}^{-1}\Delta - \Delta \\
&= 0 + H_{\mathbf{w}_\eta} H_{\mathbf{w}^*}^{-1}\Delta - H_{\mathbf{w}^*} H_{\mathbf{w}^*}^{-1}\Delta \\
&= (H_{\mathbf{w}_\eta} - H_{\mathbf{w}^*}) H_{\mathbf{w}^*}^{-1}\Delta.
\end{aligned}$$

Gradient Norm Bound 1

$$\|G(\mathbf{w}^-)\|_2 = \|(H_{\mathbf{w}_\eta} - H_{\mathbf{w}^*})H_{\mathbf{w}^*}^{-1}\Delta\|_2$$
$$\leq \|H_{\mathbf{w}_\eta} - H_{\mathbf{w}^*}\|_2 \|H_{\mathbf{w}^*}^{-1}\Delta\|_2 \quad \boxed{\leq \gamma(n-1)\|H_{\mathbf{w}^*}^{-1}\Delta\|_2^2.}$$

Using the Lipschitz-ness of ℓ'', we have for every i:

$$\begin{aligned}
\|\nabla^2\ell(\mathbf{w}_\eta^\top\mathbf{x}_i, y_i) - \nabla^2_{\mathbf{w}}\ell((\mathbf{w}^*)^\top\mathbf{x}_i, y_i)\|_2 &= \|[\ell''(\mathbf{w}_\eta^\top\mathbf{x}_i, y_i) - \ell''((\mathbf{w}^*)^\top\mathbf{x}_i, y_i)]\mathbf{x}_i\mathbf{x}_i^\top\|_2 \\
&\leq |\ell''(\mathbf{w}_\eta^\top\mathbf{x}_i, y_i) - \ell''((\mathbf{w}^*)^\top\mathbf{x}_i, y_i)| \cdot \|\mathbf{x}_i\|_2^2 \\
&\leq \gamma\|\mathbf{w}_\eta - \mathbf{w}^*\|_2 \qquad \text{since } \|\mathbf{x}_i\|_2 \leq 1 \\
&= \gamma\|\eta H_{\mathbf{w}^*}^{-1}\Delta\|_2 \\
&\leq \gamma\|H_{\mathbf{w}^*}^{-1}\Delta\|_2.
\end{aligned}$$

$$\|H_{\mathbf{w}_\eta} - H_{\mathbf{w}^*}\|_2 \leq \sum_{i=1}^{n-1}\left\|\nabla^2\ell(\mathbf{w}_\eta^\top\mathbf{x}_i, y_i) - \nabla^2\ell((\mathbf{w}^*)^\top\mathbf{x}_i, y_i)\right\|_2 \quad \leq \gamma(n-1)\|H_{\mathbf{w}^*}^{-1}\Delta\|_2.$$

Gradient Norm Bound 1

$$\|G(\mathbf{w}^-)\|_2 \le \gamma(n-1)\|H_{\mathbf{w}^*}^{-1}\Delta\|_2^2 \le \frac{4\gamma C^2}{\lambda^2(n-1)}$$

$L(\cdot;\mathcal{D}')$ is $\lambda(n-1)$-strongly convex

$$\|H_{\mathbf{w}^*}\|_2 \ge \lambda(n-1)$$

$$\|H_{\mathbf{w}^*}^{-1}\|_2 \le \frac{1}{\lambda(n-1)}$$

$$0 = \nabla L(\mathbf{w}^*;\mathcal{D}) = \sum_{i=1}^{n} \nabla\ell\left((\mathbf{w}^*)^\top\mathbf{x}_i, y_i\right) + \lambda n\mathbf{w}^*$$

+

$$\|\nabla\ell(\mathbf{w}^\top\mathbf{x}, y)\|_2 \le C$$

$$\|\mathbf{w}^*\|_2 = \frac{\|\sum_{i=1}^{n}\nabla\ell((\mathbf{w}^*)^\top\mathbf{x}_i, y_i)\|_2}{\lambda n} \le \frac{C}{\lambda}$$

+

$$\Delta = \lambda\mathbf{w}^* + \nabla\ell\left((\mathbf{w}^*)^\top\mathbf{x}_n, y_n\right)$$

$$\|\Delta\|_2 \le \lambda\|\mathbf{w}^*\|_2 + \|\nabla\ell((\mathbf{w}^*)^\top\mathbf{x}_n, y_n)\|_2 \le 2C$$

Gradient Norm Bound 2

- Data-dependent Bound

$$H_{\mathbf{w}} = (X^-)^\top D_{\mathbf{w}} X^- + \lambda(n-1) I_d \qquad (D_{\mathbf{w}})_{ii} = \ell''\left(\mathbf{w}^\top \mathbf{x}_i, y_i\right)$$

$$\begin{aligned} \|\nabla L(\mathbf{w}^-; \mathcal{D}')\|_2 &= \|(H_{\mathbf{w}_\eta} - H_{\mathbf{w}^*}) H_{\mathbf{w}^*}^{-1} \Delta\|_2 \\ &= \|(X^-)^\top (D_{\mathbf{w}_\eta} - D_{\mathbf{w}^*}) X^- H_{\mathbf{w}^*}^{-1} \Delta\|_2 \\ &\leq \|X^-\|_2 \|D_{\mathbf{w}_\eta} - D_{\mathbf{w}^*}\|_2 \|X^- H_{\mathbf{w}^*}^{-1} \Delta\|_2 \end{aligned}$$

$$+ \qquad \|D_{\mathbf{w}_\eta} - D_{\mathbf{w}^*}\|_2 \leq \gamma \|\mathbf{w}_\eta - \mathbf{w}^*\|_2 \leq \gamma \|H_{\mathbf{w}^*}^{-1} \Delta\|_2$$

$$\|\nabla L(\mathbf{w}^-; \mathcal{D}')\|_2 \leq \gamma \|X^-\|_2 \|H_{\mathbf{w}^*}^{-1} \Delta\|_2 \|X^- H_{\mathbf{w}^*}^{-1} \Delta\|_2$$

Methods

Algorithm 1 Training of a certified removal-enabled model.

Input: Dataset $\mathcal{D}$, loss ℓ, parameters $\sigma, \lambda > 0$.
Sample $\mathbf{b} \sim \mathcal{N}(0, \sigma)^d$.
Return: $\operatorname{argmin}_{\mathbf{w} \in \mathbb{R}^d} \sum_{i=1}^{n} \ell(\mathbf{w}^\top \mathbf{x}_i, y_i) + \lambda n \|\mathbf{w}\|_2^2 + \mathbf{b}^\top \mathbf{w}$.

Algorithm 2 Repeated certified removal of data batches.

Input: Dataset $\mathcal{D}$, loss ℓ, parameters $\epsilon, \delta, \sigma, \lambda > 0$.
  Lipschitz constant γ of ℓ'.
  Solution $\mathbf{w}$ computed by Algorithm 1.
  Sequence of batches of training sample
  indices to be removed: $B_1, B_2, \ldots$
Gradient residual bound $\beta \leftarrow 0$.
$c \leftarrow \sqrt{2 \log(1.5/\delta)}$.
$K \leftarrow \sum_{i=1}^{n} \mathbf{x}_i \mathbf{x}_i^\top$.
$X \leftarrow [\mathbf{x}_1 | \mathbf{x}_2 | \cdots | \mathbf{x}_n]^\top$.
for $j = 1, 2, \ldots$ **do**
  $\Delta \leftarrow |B_j| \lambda \mathbf{w} + \sum_{i \in B_j} \nabla \ell(\mathbf{w}^\top \mathbf{x}_i, y_i)$.
  $H \leftarrow \sum_{i : i \notin B_1, B_2, \ldots, B_j} \nabla^2 \ell(\mathbf{w}^\top \mathbf{x}_i, y_i)$.
  $X \leftarrow \texttt{remove\_rows}(X, B_j)$.
  $K \leftarrow K - \sum_{i \in B_j} \mathbf{x}_i \mathbf{x}_i^\top$.
  $\beta \leftarrow \beta + \gamma \sqrt{\|K\|_2} \cdot \|H^{-1} \Delta\|_2 \cdot \|X H^{-1} \Delta\|_2$.
  if $\beta > \sigma \epsilon / c$ **then**
    Re-train from scratch using Algorithm 1.
  else
    $\mathbf{w} \leftarrow \mathbf{w} + H^{-1} \Delta$.
  end if
end for

Linear Logistic Regression

- MNIST Binary Classification (3 & 8) $\delta = 1e{-}4$

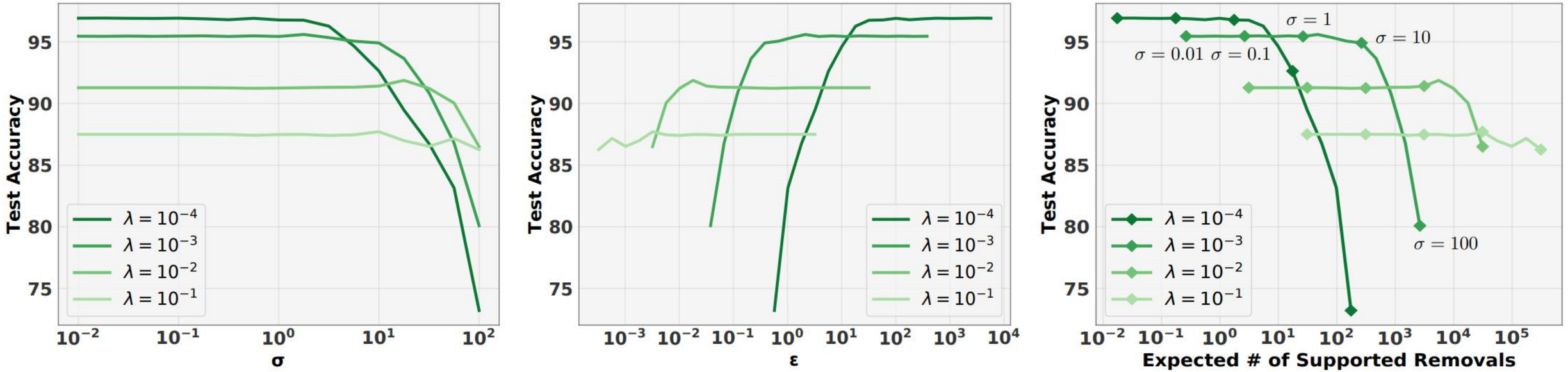
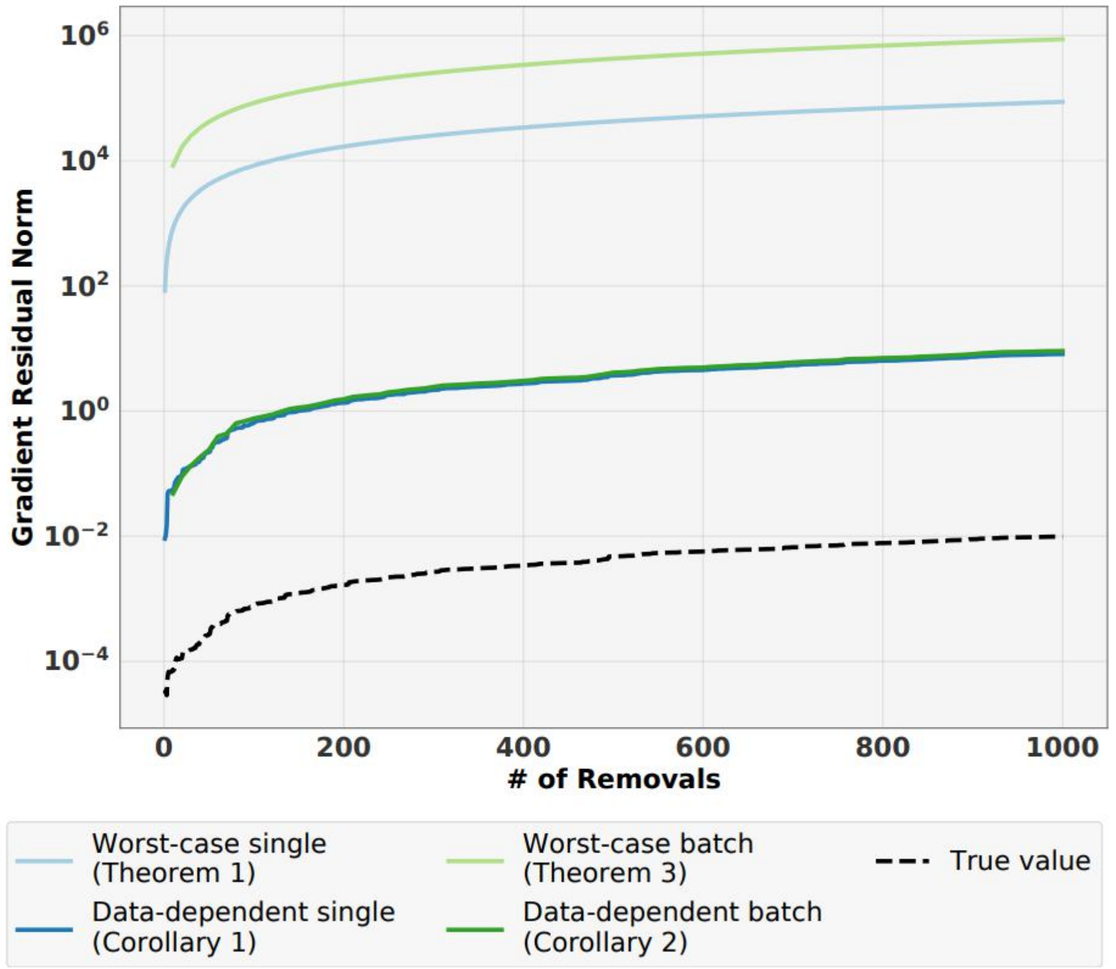


Figure 1. **Linear logistic regression on MNIST. Left**: Effect of L_2-regularization parameter, λ, and standard deviation of the objective perturbation, σ, on test accuracy. **Middle**: Effect of ϵ on test accuracy when supporting 100 removals. **Right**: Trade-off between accuracy and supported number of removals at $\epsilon = 1$. At a given ϵ, higher λ and σ values reduce test accuracy but allow for many more removals.

Linear Logistic Regression

the number of supported removals is several orders of magnitude higher when using the data-dependent bounds.



Batch size m=10

Linear Logistic Regression

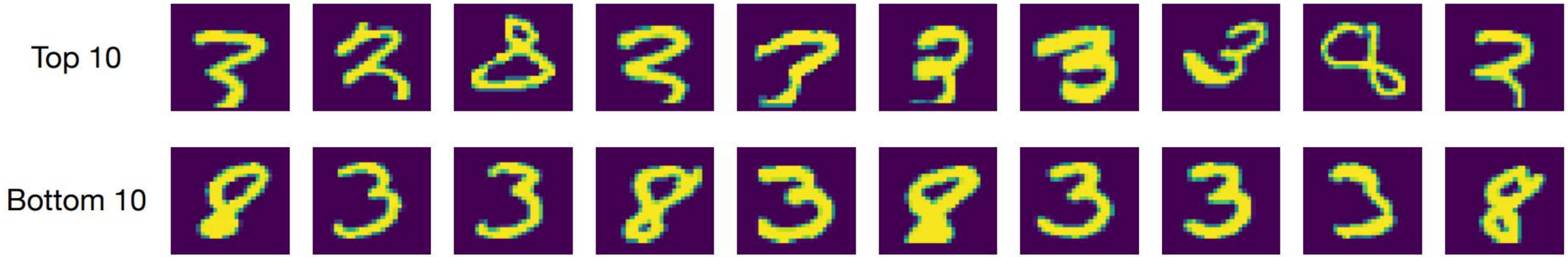


Figure 3. **MNIST training digits sorted by norm of the removal update $\|\mathbf{H}_{w^*}^{-1}\Delta\|_2$.** The samples with the highest norm (**top**) appear to be atypical, making it harder to undo their effect on the model. The samples with the lowest norm (**bottom**) are prototypical 3s and 8s, and hence are much easier to remove.

outliers is harder to remove, because the model tends to memorize their details and their impact on the model is easy to distinguish from other samples.

On Public Feature Extractors

- scene classification
  - LSUN dataset (200k)
  - ResNeXt-101 model
- Sentiment classification
  - SST dataset
  - RoBERTa model

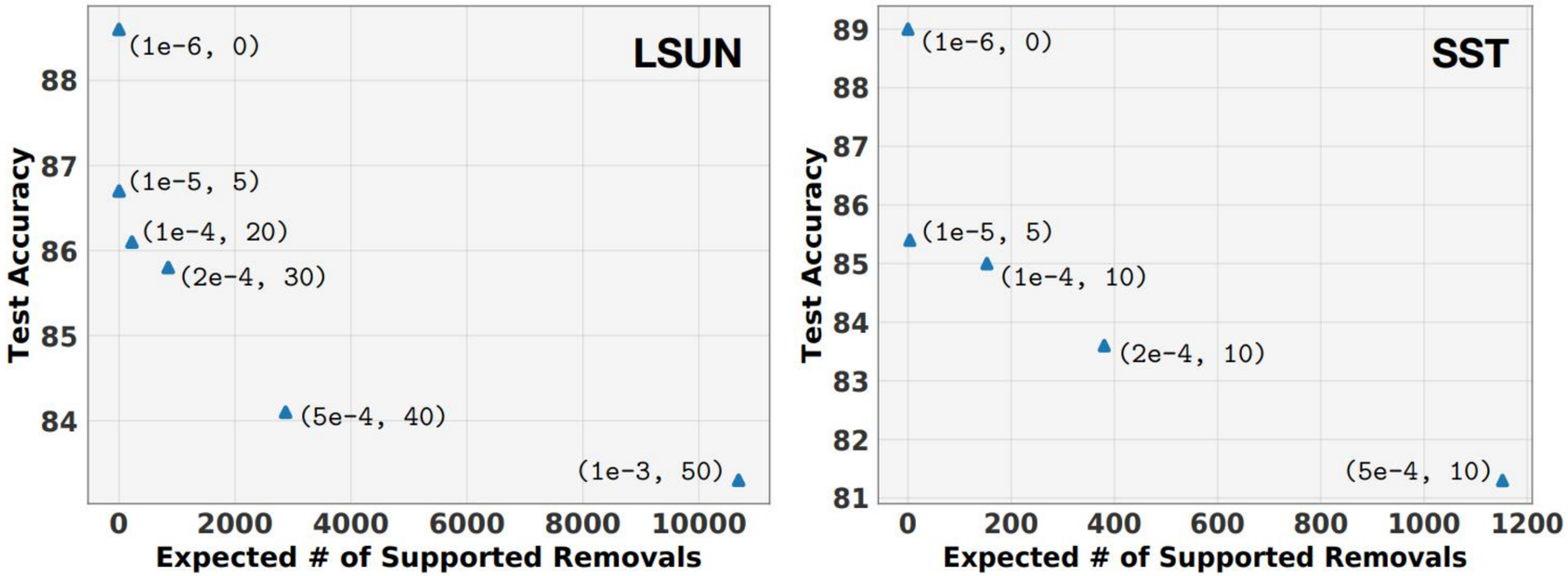


Figure 4. **Linear models trained on public feature extractors.** Trade-off between test accuracy and the expected number of supported removals (at $\epsilon = 1$) on LSUN (**left**) and SST (**right**). The setting of (λ, σ) is shown next to each point. The number of supported removals rapidly increases when accuracy is slightly sacrificed.

Time Compare

Dataset	**MNIST (§4.1)**	**LSUN (§4.2)**	**SST (§4.2)**
Removal setting	CR Linear	Public Extractor + CR Linear	Public Extractor + CR Linear
Removal time	0.04s	0.48s	0.07s
Training time	15.6s	124s	61.5s

Hard to Forget: Poisoning Attacks on Certified Machine Unlearning

Neil G. Marchant,[1] Benjamin I. P. Rubinstein,[1] Scott Alfeld[2]

[1] School of Computing and Information Systems, University of Melbourne, Parkville, Australia
[2] Department of Computer Science, Amherst College, Amherst, USA
nmarchant@unimelb.edu.au, brubinstein@unimelb.edu.au, salfeld@amherst.edu

AAAI 2022

Motivation

- Attack on certified machine unlearning
  - Posion attack
  - increase the computational cost

Methods

Give $D = D_{\text{psn}} \cup D_{\text{cln}}$ $\hat{\boldsymbol{\theta}}(D) = \arg\min_{\boldsymbol{\theta}} R(\boldsymbol{\theta}; D)$

Attack Goal

$$\min_{\mathbf{X}_{\text{psn}} \in \mathbb{R}^{m \times d}} -C(\hat{\boldsymbol{\theta}}(D), D_{\text{psn}})$$

computational cost $C(\hat{\boldsymbol{\theta}}, D_{\text{psn}}) = \|\mathbf{X}\|_2 \|\Delta\boldsymbol{\theta}\|_2 \|\mathbf{X}\Delta\boldsymbol{\theta}\|_2$

constraints

$$g_1(D_{\text{psn}}) = \sup_{\mathbf{x} \in \text{rows}(\mathbf{X}_{\text{psn}} - \mathbf{V}_1)} \|\mathbf{x}\|_\infty - r_1 \leq 0,$$

$$g_2(D_{\text{psn}}) = \sup_{\mathbf{x} \in \text{rows}(\mathbf{X}_{\text{psn}} - \mathbf{X}_{\text{ref}})} \|\mathbf{x}\|_p - r \leq 0.$$

Methods

- Faster cost surrogates 1

$$C(\hat{\boldsymbol{\theta}}, D_{\text{psn}}) = \|\mathcal{I}(D_{\text{psn}}; D, \hat{\boldsymbol{\theta}})\|_2 = \|\Delta\boldsymbol{\theta}\|_2$$

- Fast cost surrogates 2

$$\begin{aligned}\|\mathcal{I}(D_{\text{psn}}; D, \hat{\boldsymbol{\theta}})\|_2 &\leq \|\mathrm{H}_{\boldsymbol{\theta}} R_{\mathbf{b}}(\hat{\boldsymbol{\theta}}; D)\|_2 \cdot \|\nabla_{\boldsymbol{\theta}} R(\hat{\boldsymbol{\theta}}; D_{\text{psn}})\|_2 \\ &\leq \frac{1}{\lambda(|D| - 1)} \cdot \|\nabla_{\boldsymbol{\theta}} R(\hat{\boldsymbol{\theta}}; D_{\text{psn}})\|_2\end{aligned}$$

$$C(\hat{\boldsymbol{\theta}}, D_{\text{psn}}) = \|\nabla_{\boldsymbol{\theta}} R(\hat{\boldsymbol{\theta}}; D_{\text{psn}})\|_2$$

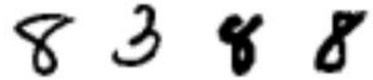
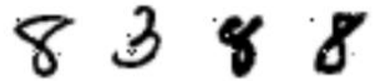
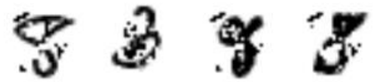
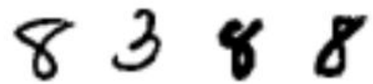
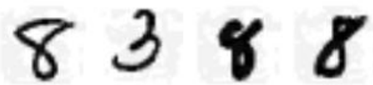
Experiment

Table 2: Attack effectiveness on Binary-MNIST as a function of the regularization strength λ and magnitude of the objective perturbation σ. The accuracy is reported for the initial model, prior to processing erasure requests.

σ	λ	Accuracy		Retrain interval		
		Benign	Attack	Benign	Attack	% ↓
1	10^{-5}	0.962	0.962	3.58	0.07	98.0
	10^{-4}	0.968	0.968	5.80	0	100
	10^{-3}	0.958	0.959	16.4	0.22	98.7
	10^{-2}	0.926	0.926	188	48.4	74.3
10	10^{-5}	0.919	0.919	0	0	–
	10^{-4}	0.932	0.931	9.32	0.27	97.1
	10^{-3}	0.954	0.955	132	8.15	93.8
	10^{-2}	0.926	0.920	1640	512	68.8

Retrain interval: the number of erasure requests handled before re-train triggered

Experiment

Constraint		Retrain interval	Poisoned examples
Norm	r		
ℓ_1	0	131	
	$d/200$	72.3	
	$d/20$	8.15	
	$d/2$	3.42	
	d	3.54	
ℓ_∞	0.000	131	
	0.050	82.2	
	0.100	48.9	
	0.500	5.63	
	1.000	3.54	

Experiment

- Ignoring model dependence

$$\hat{\theta} = \arg\max_{\theta} R(\theta; D_{\text{cln}})$$

Table 4: Attack effectiveness and computation time for different choices of the cost function.

Cost function	Ignore model dep.	Retrain interval	Attack time (s)
GRNB (10)	No	6.96	39.2
	Yes	7.08	24.4
Influence norm (11)	No	7.98	29.0
	Yes	8.15	8.72
Gradient norm (12)	No	16.34	23.1
	Yes	19.21	3.54

Approximate Data Deletion from Machine Learning Models

Zachary Izzo
Dept. of Mathematics
Stanford University
zizzo@stanford.edu

Mary Anne Smart
Department of CS&E
UC San Diego
msmart@eng.ucsd.edu

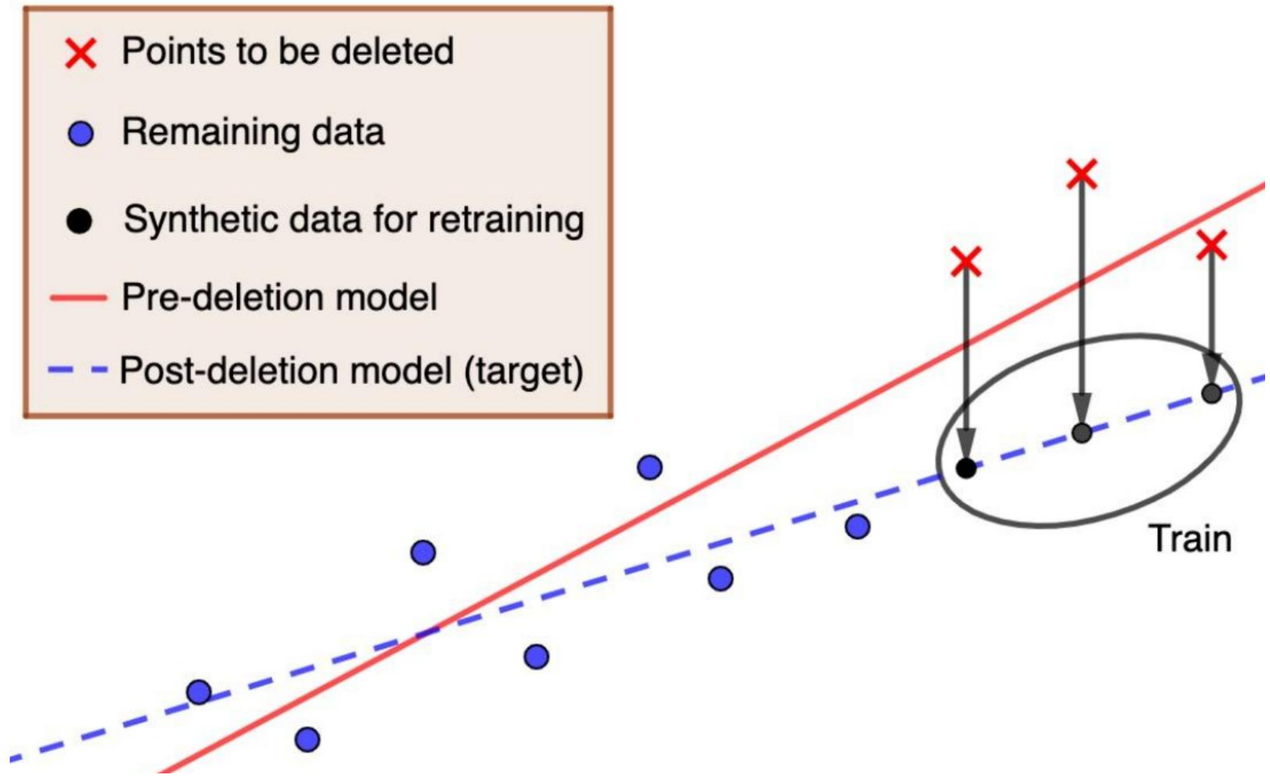
Kamalika Chaudhuri
Department of CS&E
UC San Diego
kamalika@cs.ucsd.edu

James Zou
Deptartment of BDS
Stanford University
jamesz@stanford.edu

AISTATS *2021*

Motivation

- Previous methods: runtimes scale with square of data dimension
- accurately and efficiently



Methods

- PRU-> assume remove the first k points

$O(k^2 d)$

Algorithm 1 The projective residual update

procedure PRU($X, Y, H, \theta^{\text{full}}, k$)
  $\hat{y}'_1, \ldots, \hat{y}'_k \leftarrow \text{LKO}(X, Y, H, k)$
  $S^{-1} \leftarrow \textsc{PseudoInv}(\sum_{i=1}^{k} x_i x_i^\intercal)$
  $\nabla L \leftarrow \sum_{i=1}^{k} (\theta^{\text{full}\intercal} x_i - \hat{y}'_i) x_i$
  return $\theta^{\text{full}} - \textsc{FastMult}(S^{-1}, \nabla L)$
end procedure

Methods

- Caculate $\hat{y}_i^{\setminus k}$

linear regression

$$\theta^{\text{full}} = (X^\intercal X)^{-1} XY$$

$$\hat{Y} = X\theta^{\text{full}} = \underbrace{(X(X^\intercal X)^{-1}X^\intercal)}_{H} Y,$$

$$\theta^{\setminus k} = \underset{\theta}{\operatorname{argmin}} \left[\sum_{i=1}^{k} (\theta^\intercal x_i - \hat{y}_i^{\setminus k})^2 + \sum_{i=k+1}^{n} (\theta^\intercal x_i - y_i)^2 \right]$$

$$HY' = \hat{Y}_{\setminus k} \qquad Y' = (\hat{y}_1^{\setminus k}, \ldots, \hat{y}_k^{\setminus k}, y_{k+1}, \ldots, y_n)^\intercal$$

$$\hat{Y}_{\setminus k} = (\hat{y}_1^{\setminus k}, \ldots, \hat{y}_n^{\setminus k})^\intercal$$

Methods

- Caculate $\hat{y}_i^{\setminus k}$

$$\text{define } r_i = y_i - \hat{y}_i,\ r = (r_1, \ldots, r_k)^\intercal,\ r_i^{\setminus k} = y_i - \hat{y}_i^{\setminus k},\ \text{and } r^{\setminus k} = (r_1^{\setminus k}, \ldots, r_k^{\setminus k})^\intercal$$

$$\rightarrow \quad r_i^{\setminus k} = \frac{r_i + \sum_{j \neq i} h_{ij} r_j^{\setminus k}}{1 - h_{ii}}$$

$$r_i^{\setminus k}(1 - h_{ii}) = r_i + \sum_{j \neq i} h_{ij} r_j^{\setminus k}$$

$$r_i^{\setminus k} - r_i = \sum h_{ij} r_j^{\setminus k}$$

$$\left(y_i - \hat{y}_i^{\setminus k}\right) - (y_i - \hat{y}_i) = \sum h_{ij}\left(y_i - \hat{y}_i^{\setminus k}\right)$$

$$\hat{y}_i - \hat{y}_i^{\setminus k} = \hat{y}_i - H\hat{y}_i^{\setminus k}$$

$$\hat{y}_i^{\setminus k} = H\hat{y}_i^{\setminus k}$$

$$r^{\setminus k} = (I - T)^{-1}\left(\frac{r_1}{1-h_{11}}, \ldots, \frac{r_k}{1-h_{kk}}\right)^\intercal,$$

$$T_{ij} = \mathbf{1}\{i \neq j\}\frac{h_{ij}}{1-h_{jj}}$$

k linear equations in k unknowns

$O(k^3)$ via simple Gaussian elimination.

Methods

- Caculate $\hat{y}_i^{\setminus k}$

Algorithm 2 Leave-k-out predictions

procedure LKO($X, Y, H, \theta^{\text{full}}, k$)
  $R \leftarrow Y_{1:k} - X_{1:k}\theta^{\text{full}}$
  $D \leftarrow \text{diag}(\{(1 - H_{ii})^{-1}\}_{i=1}^{k})$
  $T_{ij} \leftarrow \mathbf{1}\{i \neq j\} \frac{H_{ij}}{1-H_{jj}}$
  $T \leftarrow (T_{ij})_{i,j=1}^{k}$
  $\hat{Y}^{\setminus k} \leftarrow Y_{1:k} - (I - T)^{-1} DR$
  return $\hat{Y}^{\setminus k}$
end procedure

Experiment

- Evaluation Metrics
  - L2 distance
  - Feature injection test

$$L^{\text{full}}(\theta) = \sum_{i=1}^{n} \ell(x_i, y_i; \theta) + \frac{\lambda}{2}\|\theta\|_2^2$$

1. deleted points all belong to the positive class, first k points
2. $\tilde{x}_i = [x_i^\intercal,\ 1]^\intercal$ for $1 \le i \le k$, $\tilde{x}_i = [x_i^\intercal,\ 0]^\intercal$ for $k < i \le n$
3. train a linear regression classifier on $\{(\tilde{x}_i, y_i)\}_{i=1}^{n}$

Experiment

- Synthetic dataset

$$x_i \overset{\text{i.i.d.}}{\sim} N(0, \Sigma) \qquad Y = X\theta^* + \varepsilon, \text{ where } \varepsilon \sim N(0, \sigma^2 I_n)$$

$X[i, j] = 0$ with probability 1-p except the last column

Table 8: Median baseline weights on injected feature for table 3.

	$p = 0.25$	0.1	0.05
$k = 5$	8.97	10.61	11.03
$k = 50$	10.23	9.73	10.10
$k = 100$	9.51	9.99	10.01

	$p = 0.25$	0.1	0.05
$k = 5$ (INF)	1.09	0.99	1.01
$k = 5$ (PRU)	**0.98**	**0.96**	**0.93**
$k = 50$ (INF)	**0.84**	0.97	2.32^{**}
$k = 50$ (PRU)	0.86	**0.67**	**0.35**
$k = 100$ (INF)	0.76	0.92	0.98
$k = 100$ (PRU)	**0.72**	**0.32**	$\mathbf{0.00}^{*}$

d = 1500, n=10d

Experiment

- Synthetic dataset

Outlier Detection

$$D^{\text{full}} = \{(\lambda x_1, \lambda y_1)\} \cup D^{\setminus 1}$$

Outlier is different to remove

Table 4: Mean results for the L^2 test on synthetic data (50 trials). The L^2 distance is given as fraction of baseline distance ($\|\theta^{\text{full}} - \theta^{\setminus k}\|$; the values of the starting distance can be found in the appendix). Results are for $d = 1500$ for various group sizes (k) and outlier sizes (λ, see Theorem 5).

	$\lambda = 1$	$\lambda = 10$	$\lambda = 100$
$k = 5$ (INF)	**0.38**	0.93	0.99
$k = 5$ (PRU)	0.92	**0.92**	**0.92**
$k = 50$ (INF)	**0.16**	0.91	0.99
$k = 50$ (PRU)	0.88	**0.88**	**0.88**
$k = 100$ (INF)	**0.14**	0.90	0.99
$k = 100$ (PRU)	0.88	**0.88**	**0.88**

Experiment

- Runtime compare

	$d = 1000$	$d = 1500$	$d = 2000$	$d = 2500$	$d = 3000$
$k = 1$ (INF)	0.0085	0.0053	0.0041	0.0036	0.0028
$k = 1$ (PRU)	**0.0062**	**0.0017**	**0.0008**	**0.0004**	**0.0003**
$k = 5$ (INF)	**0.0092**	0.0052	0.0043	0.0033	0.0028
$k = 5$ (PRU)	0.0112	**0.0035**	**0.0019**	**0.0011**	**0.0007**
$k = 10$ (INF)	**0.0098**	0.0054	0.0045	0.0033	0.0031
$k = 10$ (PRU)	0.0155	**0.0049**	**0.0025**	**0.0015**	**0.0010**
$k = 25$ (INF)	**0.0105**	**0.0058**	**0.0050**	**0.0035**	0.0032
$k = 25$ (PRU)	0.0365	0.0121	0.0067	0.0037	**0.0026**
$k = 50$ (INF)	**0.0122**	**0.0065**	**0.0051**	**0.0036**	**0.0033**
$k = 50$ (PRU)	0.0794	0.0273	0.0151	0.0085	0.0059

Experiment

- Yelp dataset
  - Select 2000 reviews
  - Vocabulary with 1500 common words (word counts as feature)

Since the Yelp dataset does not have large outliers, IF > PRU for L2 distance test