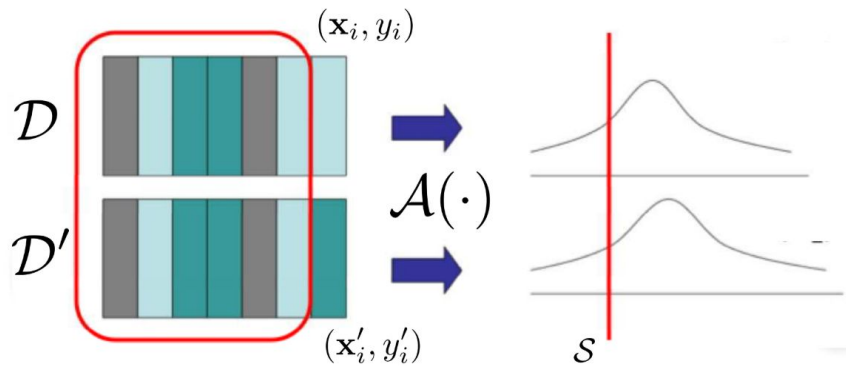


Privacy: Differentially Private ML

Cheryl Bai, Jingyi Cui, Jade Gregoire, Ganesh Nanduru, Srikar Mutnuri

Privacy

- Anonymization and aggregate results are not enough
- ϵ -differential privacy
 - A statistical procedure satisfies this if changing a single data point does not shift the output distribution “by too much” (some epsilon degree)
 - This makes it difficult to infer the value of any particular data point



Classification

The Empirical Risk Minimization (ERM) Framework

- ERM: goal is to minimize the average over the training data of the prediction loss (with respect to the label) of the classifier in predicting each training data point
- Output perturbation: in this method, noise is added to the output of the standard ERM algorithm
- Objective perturbation: adding noise to the regularized ERM objective function prior to minimizing loss
- End-to-end privacy: each step in the learning process can cause additional risk of privacy violation, so you need privacy guarantees for training and parameter tuning (intermediate steps)



Output Perturbation

- This method alters the output of the function computed on the database, before releasing it
- Specifically, the sensitivity method makes an algorithm differentially private by adding noise to its output

Algorithm 1 ERM with output perturbation (sensitivity)

Inputs: Data $\mathcal{D} = \{z_i\}$, parameters ϵ_p, Λ .

Output: Approximate minimizer \mathbf{f}_{priv} .

Draw a vector \mathbf{b} according to (4) with $\beta = \frac{n\Lambda\epsilon_p}{2}$.

Compute $\mathbf{f}_{\text{priv}} = \text{argmin} J(\mathbf{f}, \mathcal{D}) + \mathbf{b}$.

random noise


Objective Perturbation

- A method in which noise is added to the objective function before optimizing over the space classifiers
- The privacy parameter does not depend on the sensitivity of the classification algorithm

Algorithm 2 ERM with objective perturbation

Inputs: Data $\mathcal{D} = \{z_i\}$, parameters ϵ_p, Λ, c .

Output: Approximate minimizer \mathbf{f}_{priv} .

Let $\epsilon'_p = \epsilon_p - \log(1 + \frac{2c}{n\Lambda} + \frac{c^2}{n^2\Lambda^2})$.  privacy parameter

If $\epsilon'_p > 0$, then $\Delta = 0$, else $\Delta = \frac{c}{n(e^{\epsilon_p/4} - 1)} - \Lambda$, and $\epsilon'_p = \epsilon_p/2$.

Draw a vector \mathbf{b} according to (4) with $\beta = \epsilon'_p/2$.

Compute $\mathbf{f}_{\text{priv}} = \text{argmin} J_{\text{priv}}(\mathbf{f}, \mathcal{D}) + \frac{1}{2}\Delta\|\mathbf{f}\|^2$.



Privacy Guarantees

Output Perturbation

Theorem: If $N(\cdot)$ is differentiable, and 1-strongly convex, and l is convex and differentiable, with $|l'(z)| \leq 1$ for all z , then, output perturbation provides ϵ -differential privacy.

Objective Perturbation

Theorem: If $N(\cdot)$ is 1-strongly convex and doubly differentiable, and $l(\cdot)$ is convex and doubly differentiable, with $|l'(z)| \leq 1$ and $|l''(z)| \leq c$ for all z , then objective perturbation is ϵ -differentially private.



Kernel Methods

- The kernel trick maps data to a higher dimension, such that linear methods can be applied to nonlinear problems

$$\mathbf{f}^*(\mathbf{x}) = \sum_{i=1}^n a_i k(\mathbf{x}(i), \mathbf{x}).$$

- However, this releases the coefficients a_i and individual data points $\mathbf{x}(i)$ → violates differential privacy!



Kernel Methods

- Solution: apply output or objective perturbations for classification with kernels, by transforming to linear classification problem

Algorithm 3 Private ERM for nonlinear kernels

Inputs: Data $\{(\mathbf{x}_i, y_i) : i \in [n]\}$, positive definite kernel function $k(\cdot, \cdot)$, sampling function $\bar{K}(\theta)$, parameters ϵ_p, Λ, D

Outputs: Predictor \mathbf{f}_{priv} and pre-filter $\{\theta_j : j \in [D]\}$.

Draw $\{\theta_j : j = 1, 2, \dots, D\}$ iid according to $\bar{K}(\theta)$.

Set $\mathbf{v}(i) = \sqrt{2/D}[\phi(\mathbf{x}(i); \theta_1) \cdots \phi(\mathbf{x}(i); \theta_D)]^T$ for each i .

Run Algorithm 1 or Algorithm 2 with data $\{(\mathbf{v}(i), y(i))\}$ and parameters ϵ_p, Λ .

Parameter Tuning

- In practice, regularization constant selection is based on the data itself
 - Adversary could infer the value of the regularization constant and therefore the data
- Two possible solutions
 - Use smaller, publicly available dataset as holdout set
 - Train for the regularization constant on different subsets of the training data

Algorithm 4 Privacy-preserving parameter tuning

Inputs: Database \mathcal{D} , parameters $\{\Lambda_1, \dots, \Lambda_m\}, \epsilon_p$.

Outputs: Parameter \mathbf{f}_{priv} .

Divide \mathcal{D} into $m+1$ equal portions $\mathcal{D}_1, \dots, \mathcal{D}_{m+1}$, each of size $\frac{|\mathcal{D}|}{m+1}$.

For each $i = 1, 2, \dots, m$, apply a privacy-preserving learning algorithm (for example Algorithms 1, 2, or 3) on \mathcal{D}_i with parameter Λ_i and ϵ_p to get output \mathbf{f}_i .

Evaluate z_i , the number of mistakes made by \mathbf{f}_i on \mathcal{D}_{m+1} . Set $\mathbf{f}_{\text{priv}} = \mathbf{f}_i$ with probability

$$q_i = \frac{e^{-\epsilon_p z_i / 2}}{\sum_{i=1}^m e^{-\epsilon_p z_i / 2}}.$$

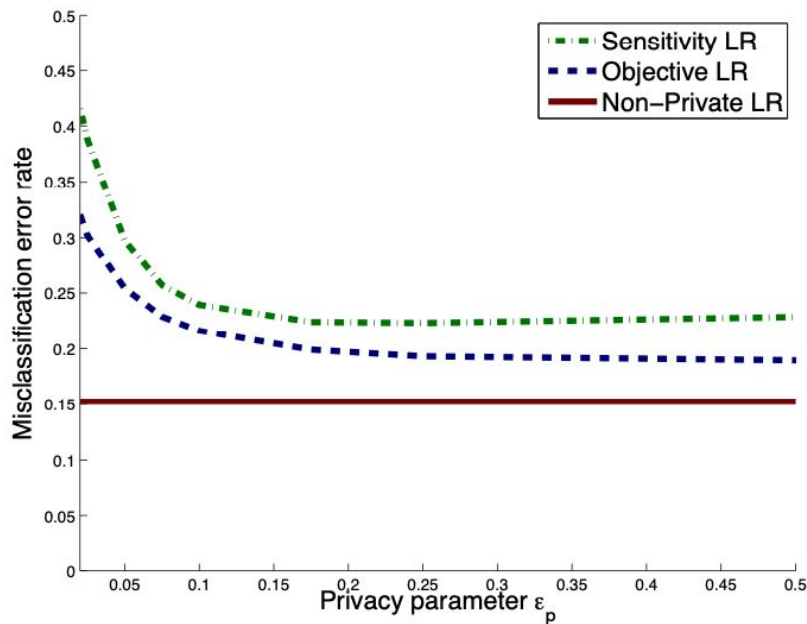


Experiments

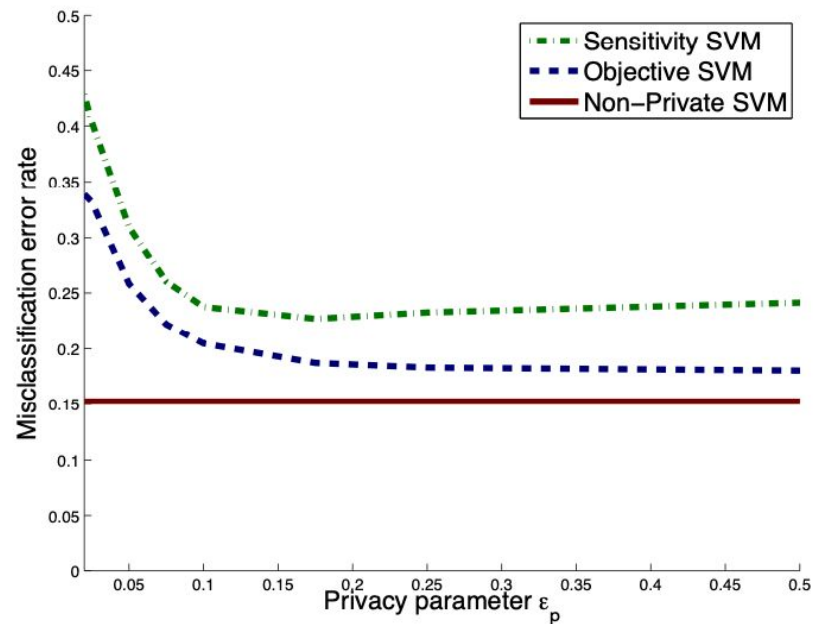
- Consider classification with output and objective perturbation on two real datasets
 - UCI Adult dataset - predict annual income
 - KDDCup99 dataset - predict network denial-of-service attack

age	workclass	marital-status	race	class
39	State-gov	Never-married	White	$\leq 50K$
49	Self-emp-inc	Married-civ-spouse	White	$> 50K$
28	Private	Married-civ-spouse	Other	$\leq 50K$
35	Private	Divorced	White	$> 50K$
38	Private	Divorced	White	$\leq 50K$
53	Local-gov	Never-married	White	$\leq 50K$
28	Private	Married-civ-spouse	Black	$\leq 50K$
37	Private	Married-civ-spouse	Black	$> 50K$
37	Private	Married-civ-spouse	White	$\leq 50K$
49	Private	Married-spouse-absent	Black	$\leq 50K$
38	Federal-gov	Married-civ-spouse	White	$> 50K$
42	Private	Married-civ-spouse	White	$> 50K$

Privacy-Accuracy Tradeoff

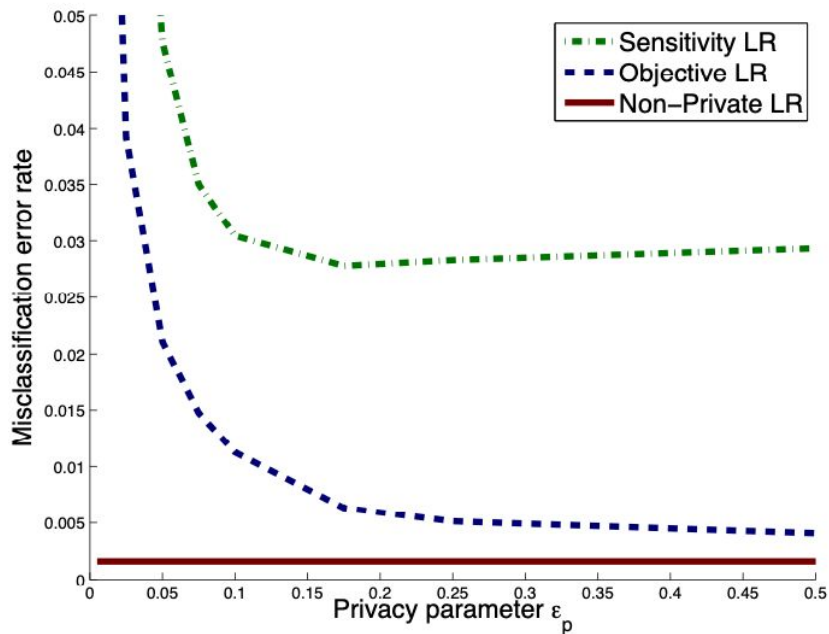


(a) Regularized logistic regression, Adult

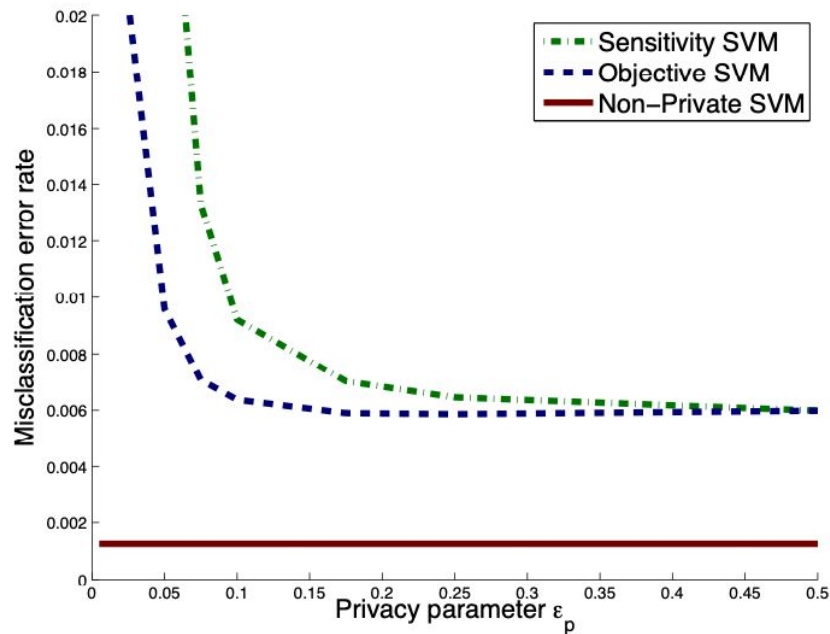


(b) Regularized SVM, Adult

Privacy-Accuracy Tradeoff

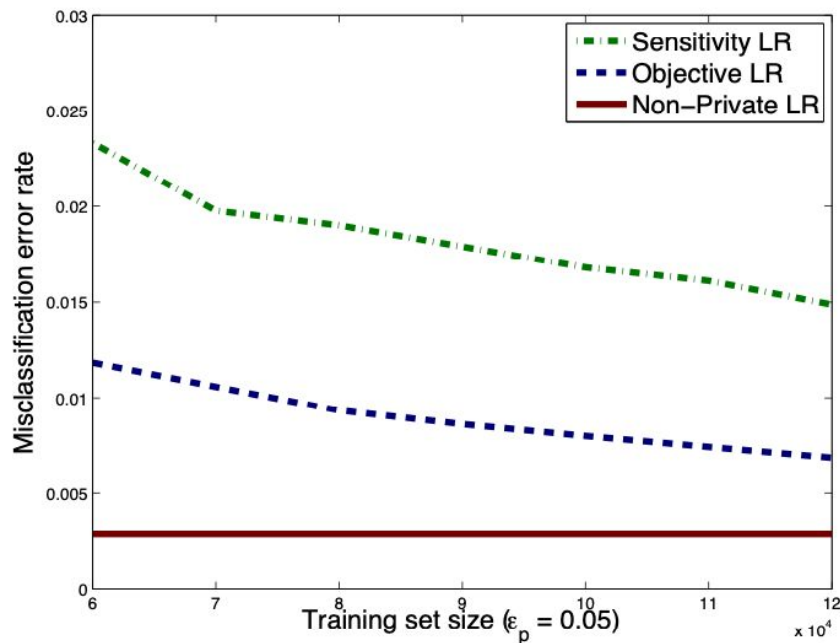


(a) Regularized logistic regression, KDDCup99

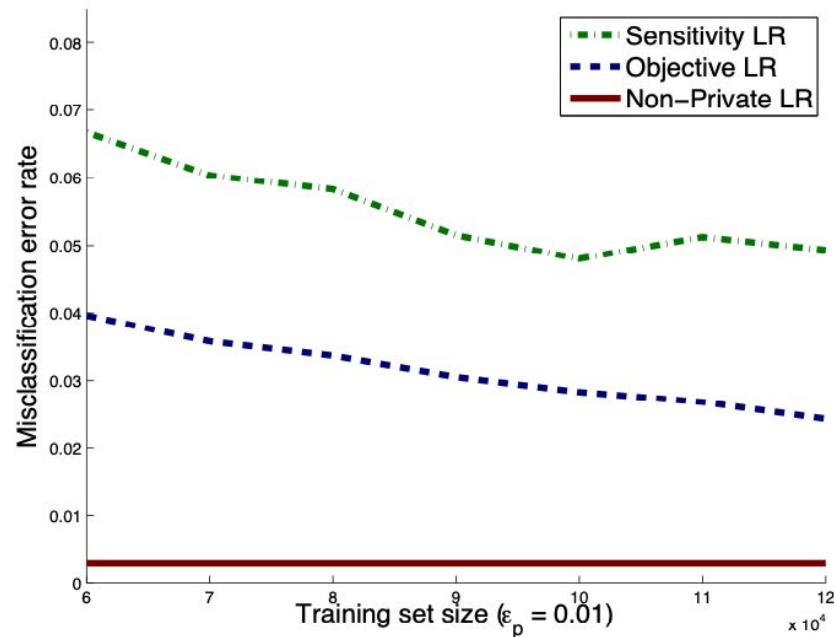


(b) Regularized SVM, KDDCup99

Accuracy vs. Training Data Size Tradeoff - LR

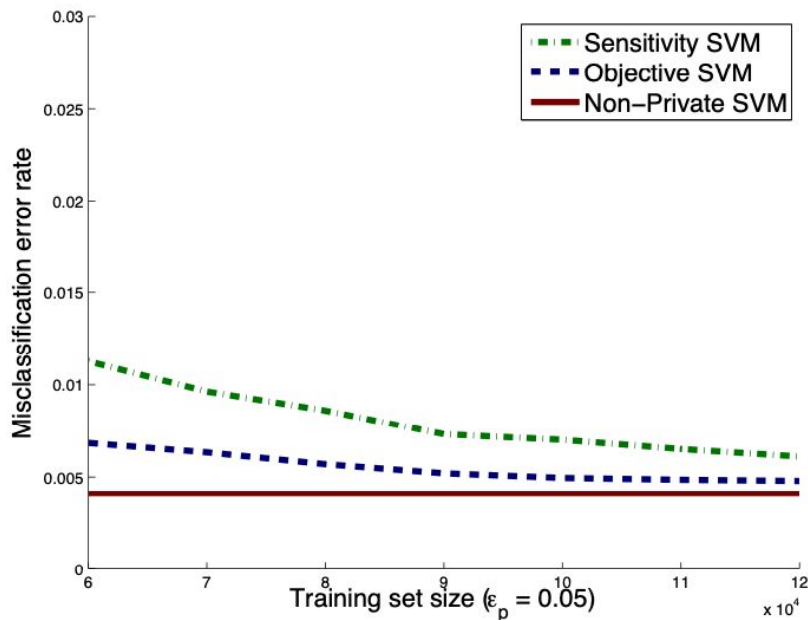


(a) $\epsilon_p = 0.05$

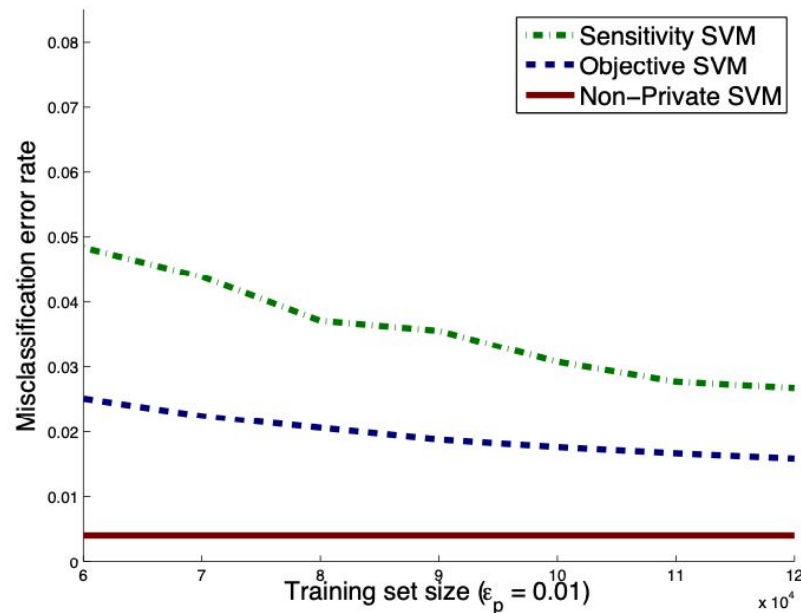


(b) $\epsilon_p = 0.01$

Accuracy vs. Training Data Size Tradeoff - SVM



(a) $\epsilon_p = 0.05$



(b) $\epsilon_p = 0.01$

DP-ERM: Conclusion

- Applied both output and objective perturbation methods to logistic regression, SVMs, and kernel methods
- Objective-perturbation is generally better at managing the privacy-accuracy tradeoff
 - It introduces less noise by perturbing the objective function directly, leading to better generalization in most cases.
- Limitations
 - This work only applies to strongly convex regularizers → objective perturbation could be further generalized
 - Current privacy-preserving methodology for kernels could be statistically inefficient



Discussion

- Output perturbation introduced noise in the outputs, while objective perturbation added noise to the objective function. Where else could you add noise in the model pipeline to help with privacy?
- When would you be more likely to use:
 - Output perturbation?
 - Objective perturbation?

Deep Learning

DP-SGD: Introduction

- Model training requires large datasets for proper training and generalizability
- These datasets can contain sensitive information
 - e.g. crowdsourced datasets like CommonCrawl
- We need to preserve privacy while maintaining model performance
- A solution: Differentially Private Stochastic Gradient Descent (DP-SGD)



DP-SGD: Method

Gradient clipping limits the influence an individual data point can have during training

Noise added to the gradient helps to increase anonymity while keeping the dataset intact

Uses (ϵ, δ) -differential privacy:

ϵ : the probability of any output differs by at most a factor of $\exp(\epsilon)$, **except for...**

δ : probability of a significant privacy breach

Algorithm 1 Differentially private SGD (Outline)

Input: Examples $\{x_1, \dots, x_N\}$, loss function $\mathcal{L}(\theta) = \frac{1}{N} \sum_i \mathcal{L}(\theta, x_i)$. Parameters: learning rate η_t , noise scale σ , group size L , gradient norm bound C .

Initialize θ_0 randomly

for $t \in [T]$ **do**

Take a random sample L_t with sampling probability L/N

Compute gradient

For each $i \in L_t$, compute $\mathbf{g}_t(x_i) \leftarrow \nabla_{\theta_t} \mathcal{L}(\theta_t, x_i)$

Clip gradient

$\bar{\mathbf{g}}_t(x_i) \leftarrow \mathbf{g}_t(x_i) / \max(1, \frac{\|\mathbf{g}_t(x_i)\|_2}{C})$

Add noise

$\tilde{\mathbf{g}}_t \leftarrow \frac{1}{L} (\sum_i \bar{\mathbf{g}}_t(x_i) + \mathcal{N}(0, \sigma^2 C^2 \mathbf{I}))$

Descent

$\theta_{t+1} \leftarrow \theta_t - \eta_t \tilde{\mathbf{g}}_t$

Output θ_T and compute the overall privacy cost (ϵ, δ) using a privacy accounting method.

DP-SGD: Method

How is privacy accounting done?

Introducing the moments accountant:

$$c(o; \mathcal{M}, \text{aux}, d, d') \triangleq \log \frac{\Pr[\mathcal{M}(\text{aux}, d) = o]}{\Pr[\mathcal{M}(\text{aux}, d') = o]}.$$

o: outcome

M: mechanism

aux: auxiliary input

(d, d'): neighboring datasets



DP-SGD: Method

The accountant uses a moment generating function to create an adaptive composition of the privacy loss:

$$\alpha_{\mathcal{M}}(\lambda; \text{aux}, d, d') \triangleq \log \mathbb{E}_{o \sim \mathcal{M}(\text{aux}, d)} [\exp(\lambda c(o; \mathcal{M}, \text{aux}, d, d'))]. \quad (2)$$

1. **[Composability]** Suppose that a mechanism \mathcal{M} consists of a sequence of adaptive mechanisms $\mathcal{M}_1, \dots, \mathcal{M}_k$ where $\mathcal{M}_i: \prod_{j=1}^{i-1} \mathcal{R}_j \times \mathcal{D} \rightarrow \mathcal{R}_i$. Then, for any λ

$$\alpha_{\mathcal{M}}(\lambda) \leq \sum_{i=1}^k \alpha_{\mathcal{M}_i}(\lambda).$$

2. **[Tail bound]** For any $\varepsilon > 0$, the mechanism \mathcal{M} is (ε, δ) -differentially private for

$$\delta = \min_{\lambda} \exp(\alpha_{\mathcal{M}}(\lambda) - \lambda \varepsilon).$$



DP-SGD: Method

Implementation

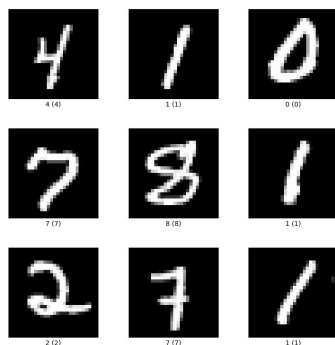
- Added PCA layer for dimensionality reduction
- Tensorflow model trained until privacy budget was exceeded

```
def DPTrain(loss, params, batch_size, noise_options):  
    accountant = PrivacyAccountant()  
    sanitizer = Sanitizer()  
    dp_opt = DPSGD_Optimizer(accountant, sanitizer)  
    sgd_op = dp_opt.Minimize(  
        loss, params, batch_size, noise_options)  
    eps, delta = (0, 0)  
    # Carry out the training as long as the privacy  
    # is within the pre-set limit.  
    while within_limit(eps, delta):  
        sgd_op.run()  
        eps, delta = accountant.GetSpentPrivacy()
```

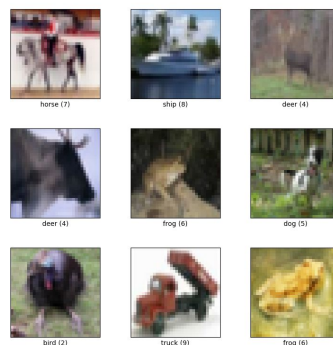


DP-SGD: Experimental Setup

- Evaluated on MNIST and CIFAR-10 with varying privacy budgets
- Metrics: training accuracy, test accuracy



MNIST



CIFAR-10

DP-SGD: Results

- Increasing the privacy budget increases model accuracy, with diminishing returns
- The accuracy vs. epsilon curve gets steeper as delta decreases
- DP-SGD achieved 97% training accuracy on MNIST, 73% training accuracy on CIFAR-10

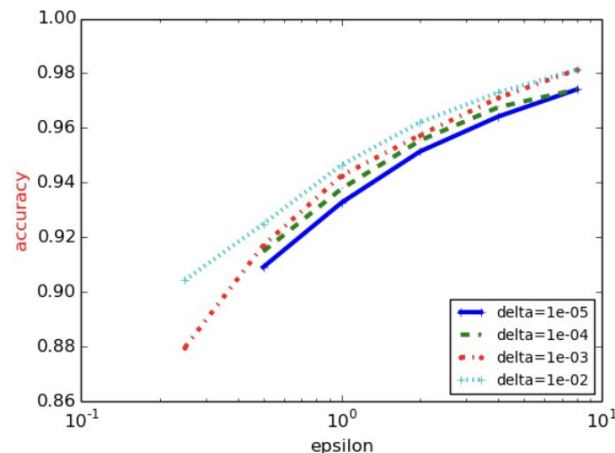


Figure 4: Accuracy of various (ϵ, δ) privacy values on the MNIST dataset. Each curve corresponds to a different δ value.



DP-SGD: Results

- Compared to theoretical privacy bounds, the moment accountant calculates a smaller upper bound for epsilon during training

$$\sigma \geq c_2 \frac{q \sqrt{T \log(1/\delta)}}{\epsilon}.$$

Strong composition theorem

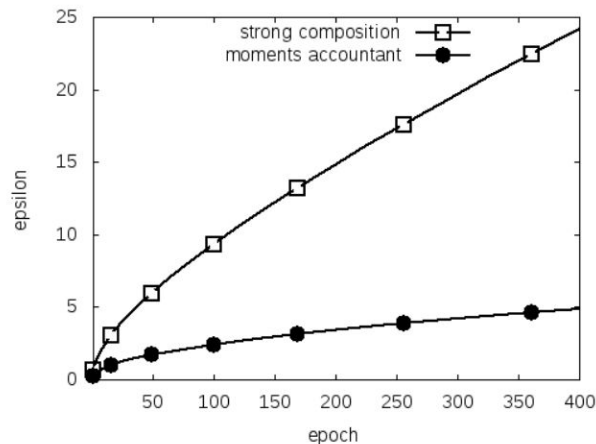


Figure 2: The ϵ value as a function of epoch E for $q = 0.01$, $\sigma = 4$, $\delta = 10^{-5}$, using the strong composition theorem and the moments accountant respectively.

DP-SGD: Conclusion

- DP-SGD strikes a balance between privacy and accuracy
- The moments accountant tracks privacy loss efficiently and with strict bounds
- Strengths
 - Implements a new privacy-maintaining framework that can train on non-convex objectives
- Weaknesses
 - Results on CIFAR are relatively weak, and more robust metrics could be reported
 - Benchmarks are rudimentary and don't demonstrate privacy-performance balancing for more sophisticated tasks



Discussion

This paper covered their method of differential privacy on MNIST and CIFAR. What other image dataset do you think could benefit from differential privacy in a real-life scenario?

Transfer Learning

Scenario

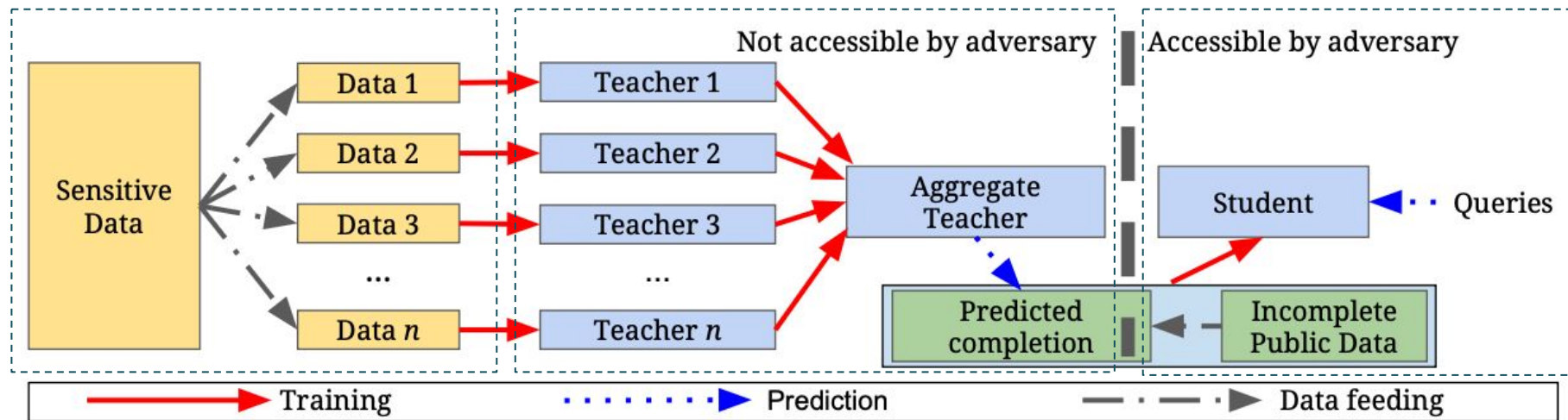
Assume the Census Bureau wants to release a bunch of models trained on PII data collected from across the country. The goal is to help decide if someone should receive certain benefits.

Each center on the map holds a part of that data. What could be the best way for them to use this aggregate data for training without compromising privacy?



Federal Statistical Research Data Centers (RDC)

PATE: Private Aggregation of Teacher Ensembles



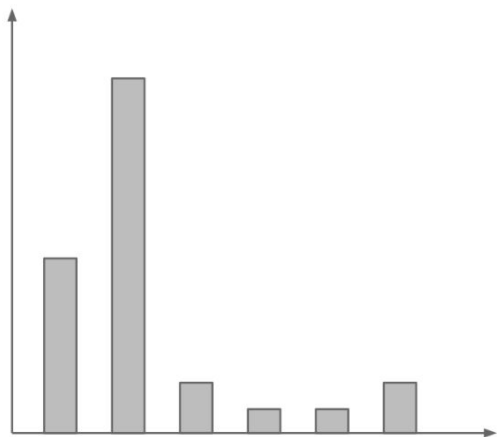
Disjoint data (sensitive): n disjoint sets (X_n, Y_n) and train a model separately on each set

Teacher Model: trained on partitioned sensitive data.

Student Model: learns from the **noisy aggregation** of the teachers predictions through polling

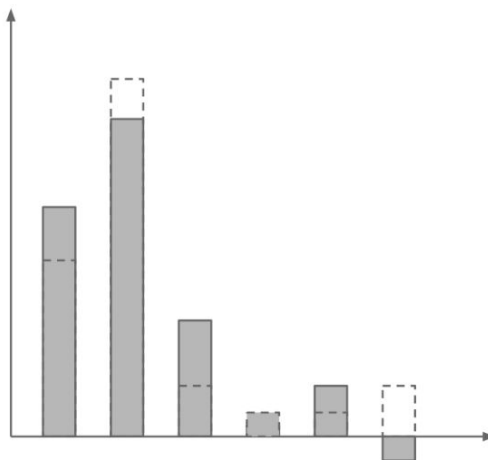
Model agnostic!!

PATE: Noisy Aggregation



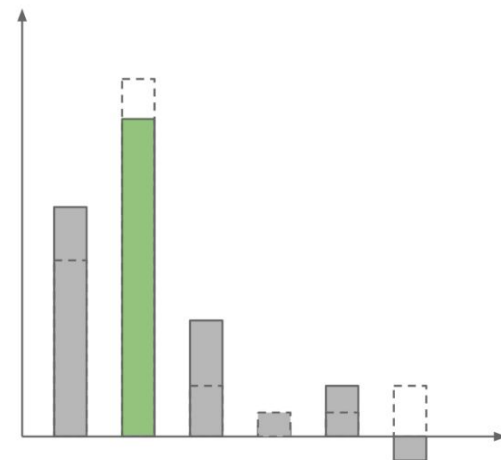
Count votes

$$n_j(\vec{x}) = |\{i : i \in 1..n, f_i(\vec{x}) = j\}|$$



Add Laplacian noise

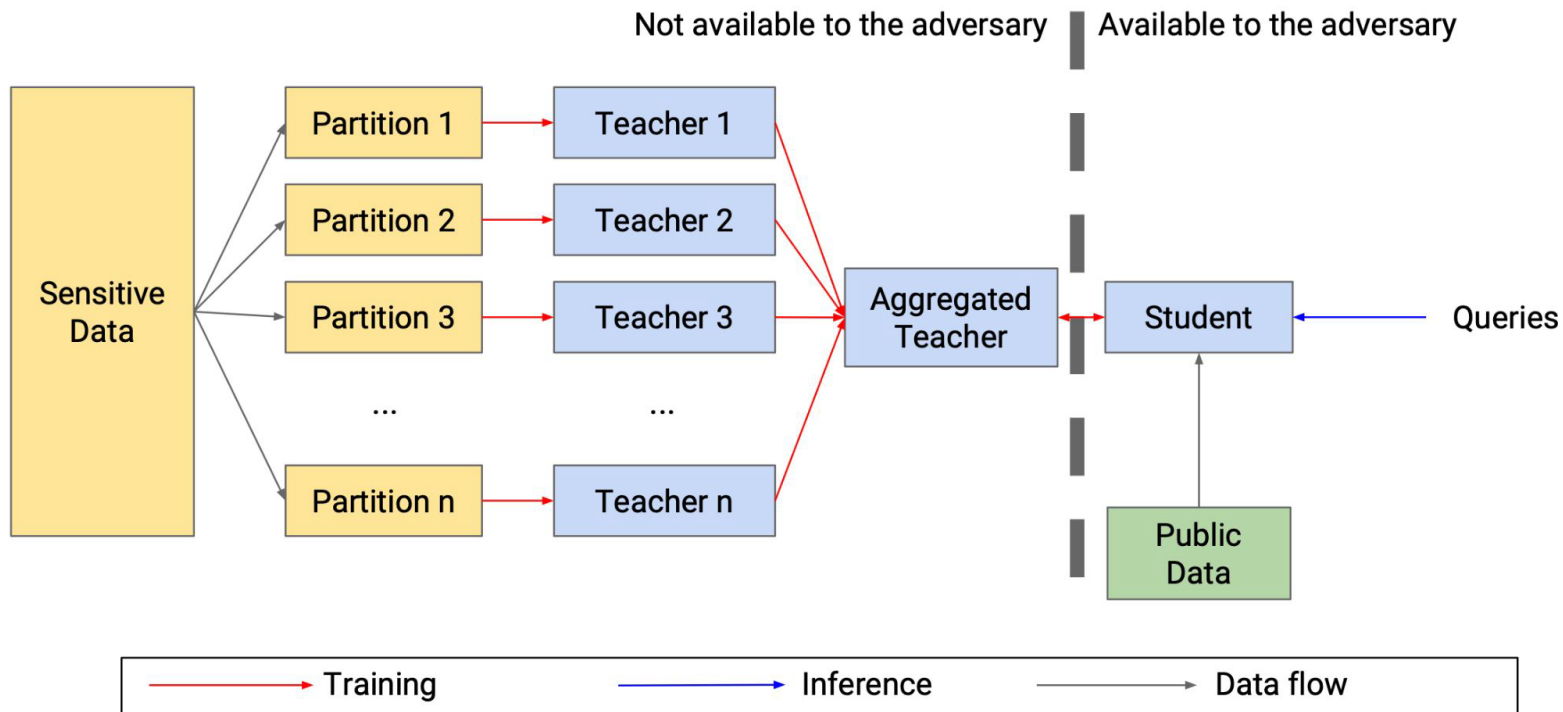
$$\text{Lap}\left(\frac{1}{\epsilon}\right)$$



Take maximum

$$f(x) = \arg \max_j \left\{ n_j(\vec{x}) + \text{Lap}\left(\frac{1}{\epsilon}\right) \right\}$$

PATE: Training

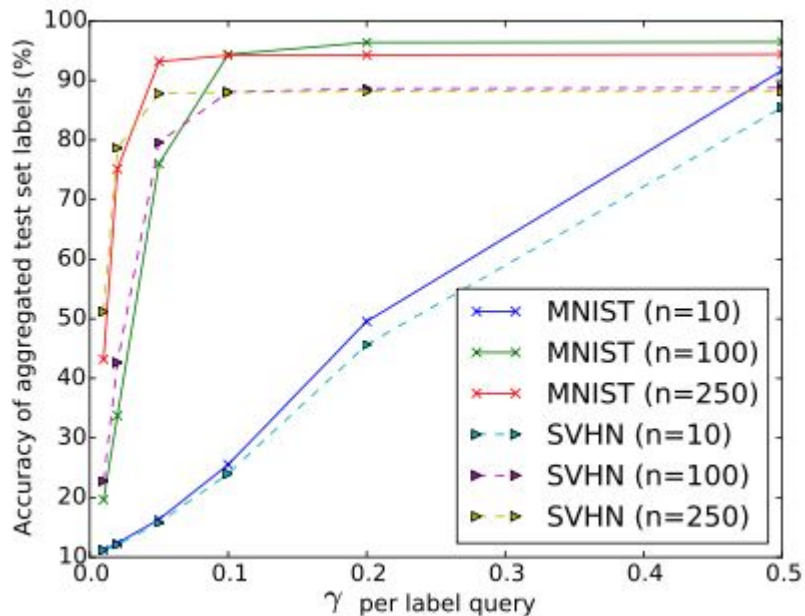


PATE: Comments

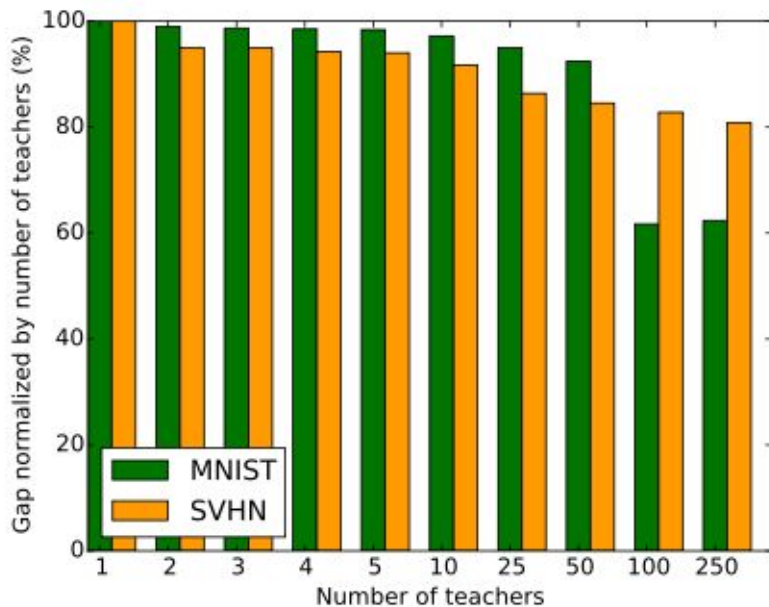
- For adjacent datasets, teachers generally get the same partitions, except for one whose partition differs.
 - Implies that the label counts for any example in these differ by one in at most two locations
 - Yields loose guarantees for bounding privacy loss!
- Strong quorum \Rightarrow small privacy cost
- More teachers \Rightarrow lower privacy cost
 - Additive gap between two largest values of n_j increases with the number of teachers
 - Larger gap \Rightarrow smaller privacy cost
 - Trade-off: large # teachers means each teacher has too little training data to be accurate.
- Constraints on the data:
 - Teachers are trained on *disjoint* partitions
 - Students use public, non-sensitive and potentially unlabelled data



PATE: Result Analysis for Aggregate Teachers

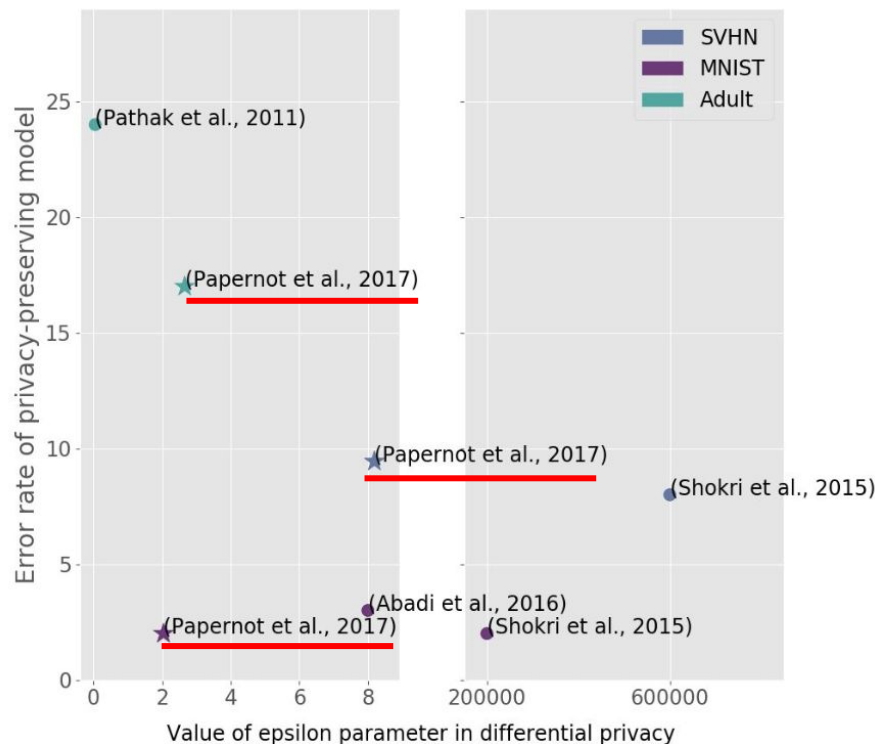


How much noise can be injected to a query?



How certain is the aggregation of teacher predictions?

PATE: Student Accuracy vs Privacy trade-off



UCI Diabetes	
ϵ	1.44
δ	10^{-5}
Non-private baseline	93.81%
Student accuracy	93.94%

PATE: Result Analysis

Dataset	ϵ	δ	Queries	Non-Private Baseline	Student Accuracy
MNIST	2.04	10^{-5}	100	99.18%	98.00%
MNIST	8.03	10^{-5}	1000	99.18%	98.10%
SVHN	5.04	10^{-6}	500	92.80%	82.72%
SVHN	8.19	10^{-6}	1000	92.80%	90.66%

Figure 4: **Utility and privacy of the semi-supervised students:** each row is a variant of the student model trained with generative adversarial networks in a semi-supervised way, with a different number of label queries made to the teachers through the noisy aggregation mechanism. The last column reports the accuracy of the student and the second and third column the bound ϵ and failure probability δ of the (ϵ, δ) differential privacy guarantee.

UCI Adult dataset - 83% accuracy with (ϵ, δ) of $(2.66, 10^{-5})$, non-private @ 85%

UCI Diabetes dataset - 93.94% accuracy with (ϵ, δ) of $(1.44, 10^{-5})$, non-private @ 93.81%

Can achieve SOTA privacy guarantees while maintaining high utility on challenging benchmark datasets

Discussion

Assume a certain characteristic X is only seen in a very small % of the population who would require more benefits than usual.

- If the teacher models are trained with the obvious constraints in place (# teachers vs accuracy), would the student model still be able to accurately catch this higher need? Why or why not?
- Specifically focusing on X within the dataset will compromise the privacy of those having it.
 - How can we ensure utility for rare traits without compromising privacy across the board? How can we tune our privacy budget in these cases?
 - Is this feasible or does it remain unavoidable?

Discussion Questions

- If you were to create a model that ensured data privacy, what methods would you use (differential privacy, data anonymization, data aggregation, etc)?
 - How would this change based on what your model is used for?
- Should differential privacy be a requirement for models?
 - Legal requirement? Only for companies?
 - Why or why not?
 - Who should bear the costs of making models differentially private if it becomes a legal requirement?

Discussion Scenario

An educational platform wants to recommend personalized courses based on a student's past web activities, quiz scores, and learning speed. If this system were trained using real student data;

- What kinds of harms could occur to the students if their data was leaked?
- How would you implement privacy?
- Would making the recommendation system an opt-in feature be more fair? Your data would only be used if you opted in to use the feature yourself.

Summary

- Objective perturbation (adding noise to the objective function) proved to outperform output perturbation (adding noise to output) in terms of privacy-accuracy trade-offs
- DP-SGD using gradient clipping & noise, with moments accountant can allow for tighter privacy analysis in deep networks.
- PATE built on these to give a model agnostic architecture for private deep learning via noisy knowledge transfer.