Privacy-Preserving Logistic Regression

Harshavardhan Thorat (21D180013)
Team: ReLUlu

Under the guidance of

Prof. P Balamurugan

Industrial Engineering and Operations Research

Indian Institute of Technology, Bombay

IE 506 Course Project (2025)

Outline

- Introduction
- Differential Privacy
  - Sensitivity
- Privacy in logistic regression
- Solution Approach
  - Algorithm 1
  - Algorithm 2
- Comparison between algorithm 1 and algorithm 2
- Results
- References

Introduction

- The research paper by Chaudhuri and Monteleoni focuses on the critical trade-off between privacy and learnability in machine learning algorithms that use private databases.

- The privacy problem arises from the increasing digitization of sensitive personal data

- Simply removing names or other obvious details from a dataset is not enough since attackers can still use auxiliary information to figure out identities.

Practical Applications:

- Ensuring privacy-preserving machine learning is critical in domains like healthcare, finance, and social networking, where sensitive data must remain confidential while still allowing useful predictive models

Differential Privacy

- ϵ-differential privacy is a mathematical definition of privacy introduced by Dwork et al. (2006). It ensures that the output of any randomized algorithm (also called a "mechanism") does not significantly depend on any single individual's data point.

$$\frac{\Pr[M(D_1) = t]}{\Pr[M(D_2) = t]} \leq e^{\epsilon}$$

- a randomized mechanism *M* satisfies *ϵ*-differential privacy if, for all databases, *D*1 and *D*2 differing in at most one element (i.e., differing in exactly one individual's data), and for any possible output *t*

Sensitivity

- A key concept in differential privacy is the sensitivity of a function. Sensitivity measures how much the output of a function can change when altering a single input.

$$S(f) = \max_{(a,a')} |f(x_1, \ldots, x_{n-1}, a) - f(x_1, \ldots, x_{n-1}, a')|$$

- According to the authors, for logistic regression with regularization parameter λ, the sensitivity is bounded by 2/(nλ). This bound is crucial for the first algorithm, which adds noise proportional to sensitivity.

Problem considered in the paper

- The paper specifically addresses the problem of designing privacy-preserving logistic regression algorithms under the ϵ-differential privacy model. Logistic regression is widely used for binary classification tasks, finding a linear decision boundary w that minimizes the regularized logistic loss

$$f_\lambda(w) = \frac{1}{2}\lambda\|w\|^2 + \frac{1}{n}\sum_{i=1}^{n}\log(1 + e^{-y_i w^T x_i})$$

Where:

- w is the weight vector (classifier)
- x_i are feature vectors with $\|x_i\| \leq 1$
- $y_i \in \{-1, 1\}$ are binary labels
- λ is the regularization parameter

The challenge is to output a classifier w^* that achieves good predictive accuracy while ensuring differential privacy.

Solution Approaches

Algorithm 1 (Sensitivity based approach)

The first algorithm applies the standard Dwork et al. approach:

1. Compute Optimal Classifier: Train a regularized logistic regression model on the dataset to obtain the optimal weight vector

$$w^* = \arg\min_w \left[\frac{1}{2}\lambda \|w\|^2 + \frac{1}{n}\sum_{i=1}^{n} \log(1 + e^{-y_i w^T x_i}) \right]$$

2. Generate Noise Vector: we pick a random vector η from the noise density function $h(\eta) \propto e^{-(n\varepsilon\lambda/2)\|\eta\|}$

3. Output the perturbed classifier: $w^* + \eta$ is the privacy preserving classifier

Algorithm 1

Learning Performance:

Given a logistic regression problem with regularization parameter λ, let w* be the classifier that minimizes f_λ, and w* + η be the classifier output by Algorithm 1 respectively.

Then, with probability 1 − δ over the randomness in the privacy mechanism,

$$\hat{f}_\lambda(w^* + \eta) \leq \hat{f}_\lambda(w^*) + \frac{2d^2(1+\lambda)\log^2(d/\delta)}{\lambda^2 n^2 \epsilon^2}$$

Performance degrades for small λ, as weaker regularization increases sensitivity.

Thus, regularization not only prevents overfitting but also inherently improves privacy by limiting sensitivity.

Algorithm 1

Limitations:

- When regularization is weak, sensitivity $S(f) = 2/(\eta\lambda)$ becomes large, requiring excessive noise

- The term d^2 makes the method less practical for high dimensional data

Algorithm 2

Algorithm 2 (Objective Perturbation Method)

In this algorithm, instead of adding noise to the final classifier, noise is added during the process of training the classifier. This approach has the advantage of not depending on sensitivity and can be applied to a broader class of machine learning problems.

1. Generate a random variable b $\in R^d$ from the density $h(b) \propto e^{-(n\varepsilon\lambda/2)||b||}$

2. Calculate the classifier *w**

$$w^* = \arg\min_w \underbrace{\frac{1}{2}\lambda w^T w}_{\text{Regularization}} + \underbrace{\frac{b^T w}{n}}_{\text{Perturbation}} + \underbrace{\frac{1}{n}\sum_{i=1}^{n}\log(1+e^{-y_i w^T x_i})}_{\text{Logistic Loss}}$$

This is a convex optimization problem identical in structure to standard logistic regression

3. w* is the privacy-preserving classifier

Algorithm 2

- The term ($b^T w/n$) introduces random linear perturbation to the objective function
- The noise vector b is sampled from a multivariate laplace distribution

Privacy Proof:

1. let D1 and D2 be two databases differing in one example (a, y) and (a', y')
2. For the same *w** the perturbation vectors b1 (for D1) and b2 (for D2) must satisfy:

$$b_1 - \frac{ya}{1 + e^{yw^{*T}a}} = b_2 - \frac{y'a'}{1 + e^{y'w^{*T}a'}}$$

3. Since ||a||, ||a'|| < 1, the norm difference satisfies ||b1 - b2|| < 2
4. The probability ratio of observing *w** under D1 and D2 is bounded by:

$$\frac{\Pr[w^*|D_1]}{\Pr[w^*|D_2]} = e^{-\frac{\epsilon}{2}(\|b_1\| - \|b_2\|)} \leq e^{\epsilon}$$

5. Thus, ε-differential privacy is preserved. Also, privacy guarantee for the algorithm does not depend on λ

Algorithm 2

Empirical Loss Bound:

With probability 1-δ, the empirical loss of algorithm 2's classifier w_2 satisfies:

$$\hat{f}_\lambda(w_2) \leq \hat{f}_\lambda(w_1) + \frac{8d^2 \log^2(d/\delta)}{\lambda n^2 \epsilon^2}$$

w1 = non-private classifier

w2 = Algorithm 2 output

For λ<1, algorithm does degrade with decreasing λ, but the degradation is better than that of Algorithm 1.

In algorithm 1, the error scales as $(1+\lambda)/\lambda^2$

Algorithm 1	Algorithm 2
Modifies the output classifier post-training	Perturbs the optimization objective pre-training
Noise vector η added to w^*	Random term $b^T w/n$ added to the objective
Depends on the sensitivity $S(f) = 2/(\eta\lambda)$	Independent of sensitivity
Regularization λ reduces sensitivity	Regularization λ has no direct privacy link
Optimal use case: Large λ	Optimal use case: small λ (most common)

Results

The paper includes some simulations on synthetic data that compare the two privacy-preserving methods

Data 1: Uniform separable data

Data 2: Non-separable data

The experiments used 17,500 points and five-fold cross-validation

$\lambda = 0.01$ for both cases

	Algorithm 1	Algorithm 2	Standard Logistic regression
Test Error (Separable data)	0.2962 ± 0.0617	0.1426 ± 0.1284	Approx 0
Test Error (non-separable data)	0.3257 ± 0.0536	0.1903 ± 0.1105	0.05 ± 0.1105

Results

Learning Curves: Graph of test error after each increment of 1000 points, averaged over five-fold cross validation.

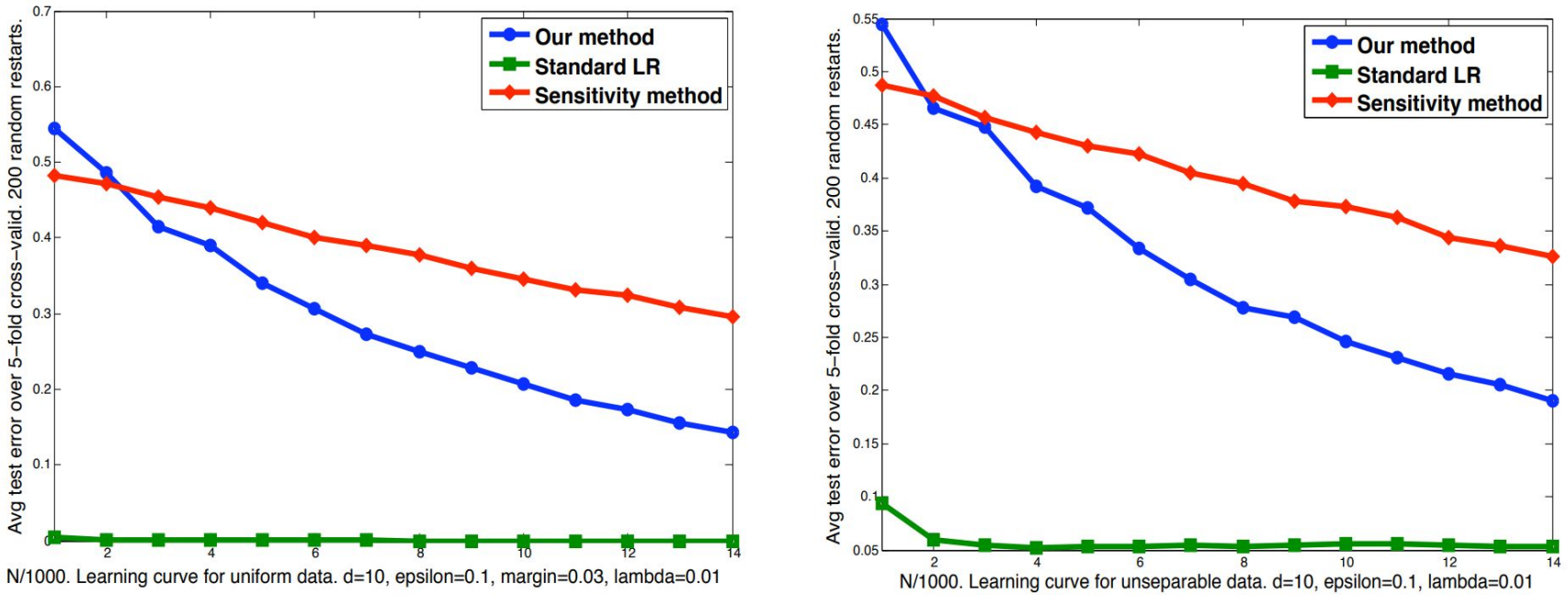

Figure 1: Red: Algorithm 1, Blue: Algorithm 2

- The graph shows that Algorithm 2 reaches a lower final error than sensitivity method.

Results

Privacy-performance tradeoff

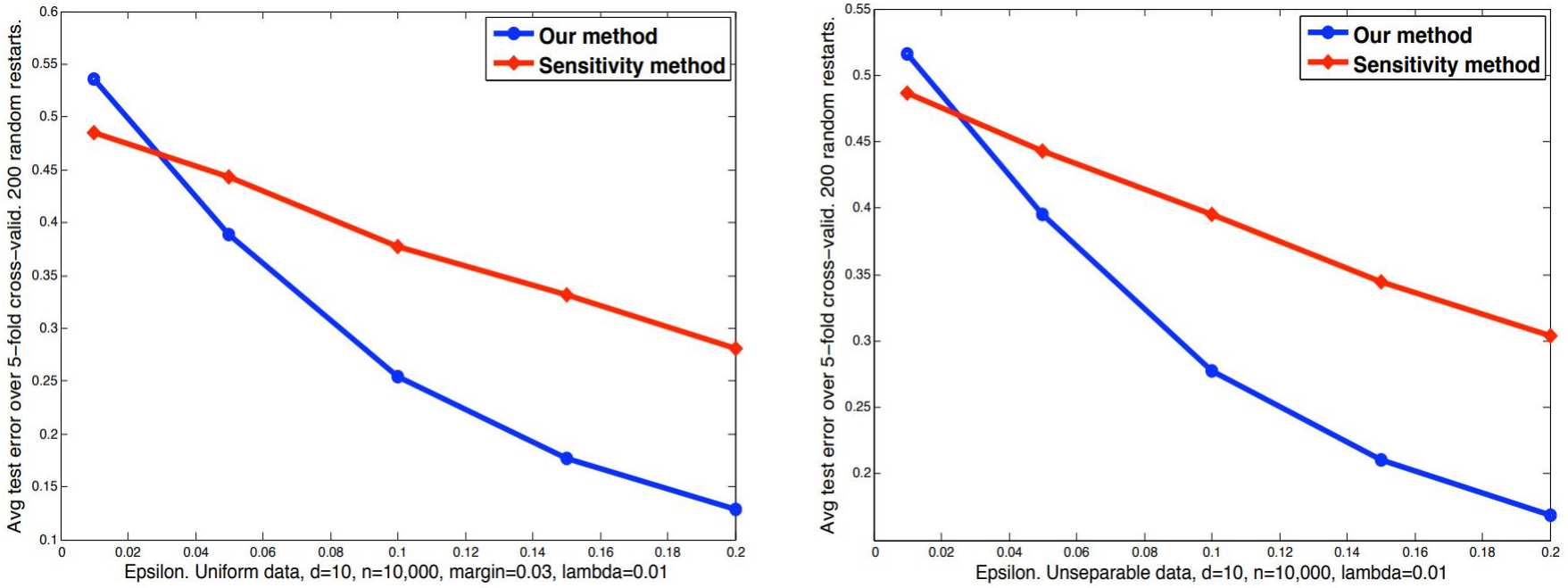

Figure 2: Red: Algorithm 1, Blue: Algorithm 2

- Lower ε decreases (increased privacy) performance degrades in both methods but algorithm 2 maintains better accuracy.
- For very small ε (ε <0.1), the algorithm 1 error is very high making it ineffective.

References

- Dwork, Cynthia. "Differential privacy." *International colloquium on automata, languages, and programming*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006.

- Chaudhuri, Kamalika, and Claire Monteleoni. "Privacy-preserving logistic regression." *Advances in neural information processing systems* 21 (2008).

- AI Tools: Deepseek R1 for equation generation