Improved Coresets for Vertical Federated Learning : Regularized Linear and Logistic Regressions

Supratim Shit, Gurmehak Kaur Chadha, Surendra Kumar, Bapi Chatterjee

Department of Computer Science and Engineering, IIIT, New Delhi

July 8, 2025

DISTRIBUTED COMPUTING and LEARNING lab

Infosys CENTRE FOR ARTIFICIAL INTELLIGENCE

Finding patterns to change the world for the better...

ICML

Abstract

We propose novel coreset construction algorithms for regularized logistic regression in both centralized and vertical federated learning (VFL) settings. Our methods improve coreset size bounds for regularized linear regression in VFL by removing dependence on data partition properties. The improvements stem from leveraging reduced model complexity due to regularization. Empirical results support our theoretical guarantees, showing that training on coresets yields performance comparable to full data training.

Introduction

Coreset

Coresets are weighted samples of datasets with provable theoretical guarantees.
More formally, let $\mathbf{Z} = \{(\mathbf{x}_i, y_i)\}_{i=1}^n$ be a dataset of n points with $\mathbf{X} \in \mathbb{R}^{n \times d}$, labels $\mathbf{y} \in \mathbb{R}^n$. Let $\mathcal{Q}$ denote the model space as $\mathbf{q} \in \mathcal{Q}$. With that, a subset $\mathcal{S} \subset \mathbf{Z}$ with weights w is an (ε, δ)-coreset if, for all $\mathbf{q} \in \mathcal{Q}$,

$$(1 - \varepsilon) f(\mathbf{Z}_v, \mathbf{q}) \leq f(\mathcal{S}_w, \mathbf{q}) \leq (1 + \varepsilon) f(\mathbf{Z}_v, \mathbf{q})$$

holds with probability $\geq 1 - \delta$.

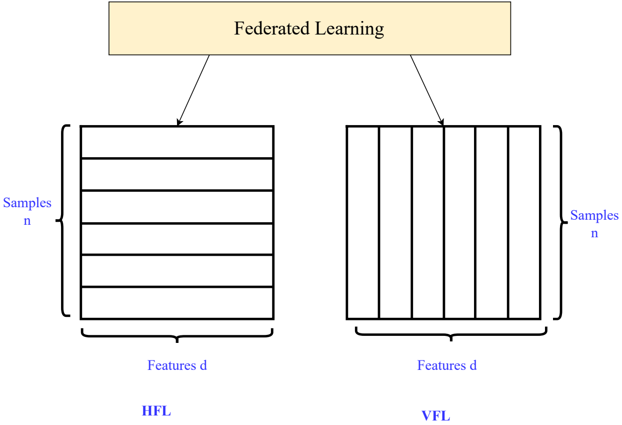
Introduction

Federated Learning

Federated learning has become a go-to approach for training machine learning models across distributed clients without sharing raw data. A central server coordinates model updates by selecting clients and synchronizing their contributions. FL comes in two main forms:

- → **Horizontal FL (HFL):** Clients have data with the same features but different samples.
- → **Vertical FL (VFL):** Clients hold different features of the same set of samples.

Federated Learning



Introduction

Horizontal Federated Learning (HFL)

HFL: Consider a model $\mathbf{q}$ and a set of clients T. A basic federated learning procedure is described as

$$
\begin{aligned}
\mathbf{q}_{r,k+1}^{(j)} &= \mathbf{q}_{r,k}^{(j)} - \eta \nabla_{\mathbf{q}^{(j)}} \mathsf{cost}(\mathbf{Z}^{(j)}, \mathbf{q}_{r,k}^{(j)}) \\
&\quad \forall j \in \mathbf{S}_r \subseteq [T], \forall k \in [K-1],\ \forall r \in [R] \\
\mathbf{q}_{r+1} &= \frac{1}{|\mathbf{S}_r|} \sum_{j \in \mathbf{S}_r} \mathbf{q}_{r,K}^{(j)},
\end{aligned}
$$

where at each synchronization round $r \in [R]$, $S_r \subseteq [T]$ clients participate in local training for $K-1$ steps.$\eta > 0$ is the learning rate.

Introduction

Vertical Federated Learning (VFL)

In VFL, the feature space $\mathbf{X}$ is partitioned across clients $j \in [T]$, where each holds $\mathbf{X}^{(j)}$ such that $\bigcup_{j\in[T]} \mathbf{X}^{(j)} = \mathbf{X}$.

$$\text{Client } j \in [T] \text{ computes } \nabla_{\mathbf{q}^{(j)}} \mathsf{cost}(\mathbf{X}^{(j)}, \mathbf{q}_r^{(j)}).$$

$$\mathbf{q}_{r+1} = \mathbf{q}_r - \eta \bigcup_{j\in[T]} \nabla_{\mathbf{q}^{(j)}} \mathsf{cost}(\mathbf{X}^{(j)}, \mathbf{q}_r^{(j)})$$

$$\forall r \in [R],\ \forall j \in [T].$$

Here, $\mathbf{q} \in \mathcal{Q}$ denotes the model parameter.

Introduction

Vertical Regularized Logistic Regression (VRLog)

$$\text{ClassLoss}(\mathbf{Z}, \mathbf{q}, \lambda) = \sum_{i=1}^{n} \ln \left(1 + \exp(-y_i \mathbf{x}_i^\top \mathbf{q}) \right) + \lambda \|\mathbf{q}\|_1$$

For a given dataset $\mathbf{Z}$ consisting of $\mathbf{X}$ representing the points and $\mathbf{y}$ be their labels in the VFL model, a regularization parameter $\lambda > 0$, the goal of the VRLog problem is to compute a vector $\mathbf{q} \in \mathbb{R}^d$ on the server that (approximately) minimizes $\text{ClassLoss}(\mathbf{Z}, \mathbf{q}, \lambda)$ while maintaining minimum total communication complexity.

Introduction

Coreset Loss for VRLog

$$\mathsf{ClassLoss}(\mathbf{S}_w, \mathbf{q}, \lambda) \in (1 \pm \varepsilon) \cdot \mathsf{ClassLoss}(\mathbf{Z}, \mathbf{q}, \lambda)$$

Where $\mathbf{S}_w$ be a weighted set, comprising of a subset $\mathbf{S} \subseteq \mathbf{Z}$ with an associated weight function w. We call $\mathbf{S}_w$ an ε-coreset for VRLog if with at least 0.99 probability, it guarantees the above requirement for ClassLoss for every $\mathbf{q} \in \mathbb{R}^d$.

Introduction

Vertical Ridge Linear Regression (VRLR)

$$\text{RegLoss}(\mathbf{Z}, \mathbf{q}, \lambda) = \sum_{i=1}^{n} (\mathbf{x}_i^\top \mathbf{q} - y_i)^2 + \lambda \|\mathbf{q}\|_2^2$$

Where $\lambda > 0$regularization parameter ,the goal of the VRLR problem is to compute a vector $\mathbf{q} \in \mathbb{R}^d$ on the server that (approximately) minimizes ClassLoss$(\mathbf{Z}, \mathbf{q}, \lambda)$ while maintaining minimum total communication complexity.

Introduction

Coreset Loss for VRLR

$$\mathsf{RegLoss}(\mathbf{S}_w, \mathbf{q}, \lambda) \in (1 \pm \varepsilon) \cdot \mathsf{RegLoss}(\mathbf{Z}, \mathbf{q}, \lambda)$$

Where $\mathbf{S}_w$ be a weighted set, comprising of a subset $\mathbf{S} \subseteq \mathbf{Z}$ with an associated weight function w. We call $\mathbf{S}_w$ an ε-coreset for VRLR if with at least 0.99 probability, it guarantees the above requirement for $\mathsf{ClassLoss}$ for every $\mathbf{q} \in \mathbb{R}^d$.

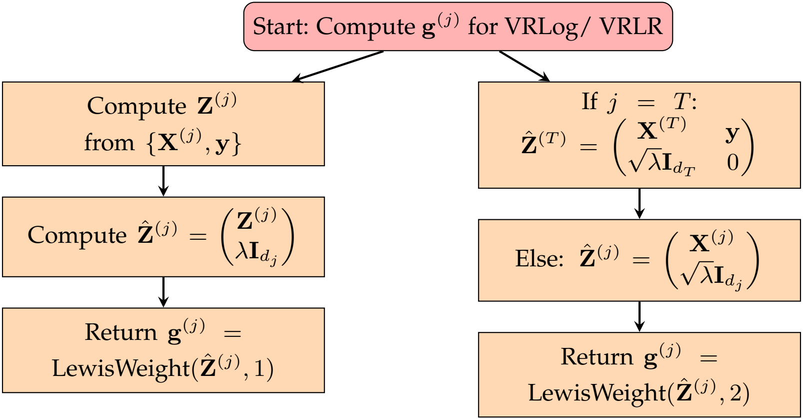
Regularized Sensitivity

For every point $i \in [n]$, the sensitivity score is defined as:

$$s_i = \sup_{\mathbf{q} \in \mathcal{Q}} \frac{f(\mathbf{z}_i, \mathbf{q})}{\text{loss}(\mathbf{Z}, \mathbf{q}, \lambda)}$$

In the above definition, the importance of every point $i \in [n]$ is quantified by s_i, which is the supremum of the relative loss of the point to the complete regularized loss over all feasible models. The sensitivity scores can be any value between 0 and 1. Further, as λ increases, the sensitivity score decreases.

Flowchart for Computing $\mathrm{g}^{(j)}$ for VRLog and VRLR



Start: Compute $\mathbf{g}^{(j)}$ for VRLog/ VRLR

Compute $\mathbf{Z}^{(j)}$ from $\{\mathbf{X}^{(j)}, \mathbf{y}\}$

Compute $\hat{\mathbf{Z}}^{(j)} = \begin{pmatrix} \mathbf{Z}^{(j)} \\ \lambda \mathbf{I}_{d_j} \end{pmatrix}$

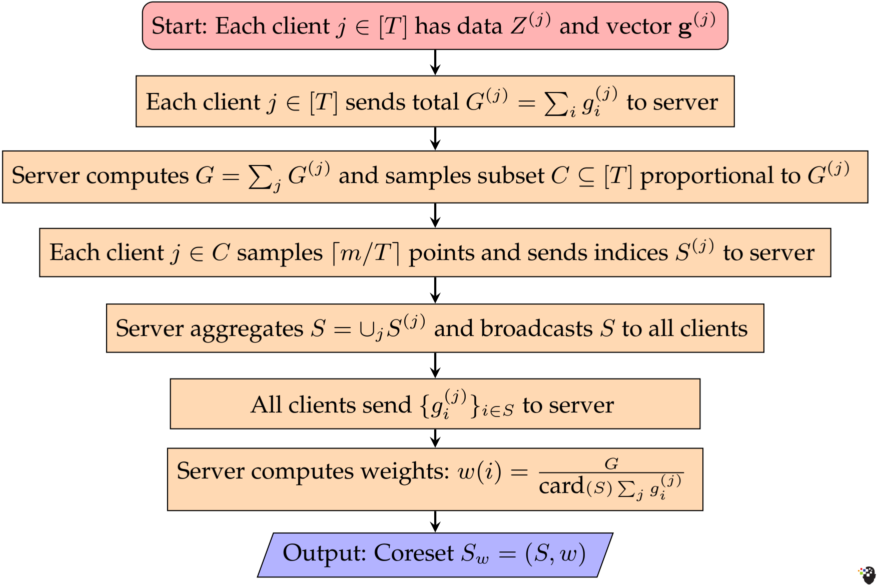
Return $\mathbf{g}^{(j)} =$ LewisWeight$(\hat{\mathbf{Z}}^{(j)}, 1)$

If $j = T$: $\hat{\mathbf{Z}}^{(T)} = \begin{pmatrix} \mathbf{X}^{(T)} & \mathbf{y} \\ \sqrt{\lambda}\mathbf{I}_{d_T} & 0 \end{pmatrix}$

Else: $\hat{\mathbf{Z}}^{(j)} = \begin{pmatrix} \mathbf{X}^{(j)} \\ \sqrt{\lambda}\mathbf{I}_{d_j} \end{pmatrix}$

Return $\mathbf{g}^{(j)} =$ LewisWeight$(\hat{\mathbf{Z}}^{(j)}, 2)$

Compute Coreset for VRLR / VRLog



Start: Each client $j \in [T]$ has data $Z^{(j)}$ and vector $\mathbf{g}^{(j)}$

Each client $j \in [T]$ sends total $G^{(j)} = \sum_i g_i^{(j)}$ to server

Server computes $G = \sum_j G^{(j)}$ and samples subset $C \subseteq [T]$ proportional to $G^{(j)}$

Each client $j \in C$ samples $\lceil m/T \rceil$ points and sends indices $S^{(j)}$ to server

Server aggregates $S = \cup_j S^{(j)}$ and broadcasts S to all clients

All clients send $\{g_i^{(j)}\}_{i \in S}$ to server

Server computes weights: $w(i) = \frac{G}{\text{card}(S) \sum_j g_i^{(j)}}$

Output: Coreset $S_w = (S, w)$

VRLR Coresets

We propose a tighter ε-coreset for VRLR that improves on [Huang et al., 2022], with better theoretical guarantees and reduced dependence on ill-conditioned data partitions.

$\rightarrow$ Coreset size is reduced by avoiding dependence on $\left(\frac{\sigma_{\max}}{\sigma_{\min}}\right)^2$.Each client computes local ℓ_2 Lewis scores using regularized data matrices.

$\rightarrow$ Final coreset size depends only on the number of clients T and statistical dimension.

VRLR Coresets

Coreset Size

The algorithm returns a ε-coreset for VRLR of size

$$m = O\left(\frac{T\sum_{j=i}^{T} sd(\,Z^{(j)}, \lambda, 2)\log(d)}{\varepsilon^2}\right)$$

The computable in input-sparsity time $O(nnz(\hat{Z}))$ with probability at least 0.99.Training on this coreset requires communication complexity $O(mT)$ and achieves $(1 \pm \varepsilon)$ approximation for ridge regression.

Statistical dimension

Statistical dim: For a matrix $\mathbf{A} \in \mathbb{R}^{n\times d}$ and $\lambda > 0$, the ℓ_2 statistical dimension is defined as $sd(\mathbf{A}, \lambda, 2) = \sum_{i=1}^{d} \frac{1}{1+\lambda/\sigma_i^2}$, where $\{\sigma_i\}$ are singular values of $\mathbf{M} = \mathbf{U}^\top \mathbf{D}^{p/2-1}\mathbf{A}$ and $\mathbf{U}$ is the ℓ_2 Lewis basis of $\mathbf{A}$.

VRLog Coresets

We propose a provably tight ε-coreset construction for regularized logistic regression (VRLog) using local ℓ_1 Lewis weights with communication-efficient computation.[Feldman and Langberg, 2011]

$\rightarrow$ Each client computes scores from regularized local features using ℓ_1 Lewis weights.

$\rightarrow$ Coreset size depends on the μ-complexity and ℓ_1 statistical dimension of each partition.

VRLog Coresets

Enables efficient federated training with bounded sensitivity and no dependence on $\sqrt{n}$.

Coreset Size

the algorithm computes an ε-co reset in $\tilde{O}(nd^2)$ of size

$$m = O\left(\frac{\mu^2 T \sum_{j=1}^{T} sd(\mathbf{Z}^{(j)}, \lambda, 1)}{\varepsilon^2}\right)$$

For some $\varepsilon \in (0, 1)$ and the model can be trained with communication complexity $O(mT)$

This coreset yields a $(1 \pm \varepsilon)$ approximation to the logistic regression loss with probability at least 0.99, while scaling gracefully with T and feature dimensions.

Our Core Contribution

Sensitivity and Regularization Effect. Consider a dataset $\mathbf{A}$ with n points in $\mathbb{R}^d$ such that $n/d = c$, and assume the response vector is $\mathbf{0}$. Let $\mathbf{A}^\top = \begin{bmatrix} I & \cdots & I \end{bmatrix}^\top$, where I is the identity matrix in $\mathbb{R}^d$. Following Huang et al. [Huang et al., 2022], the sensitivity score per point is $1/c$, and total sensitivity for n points is $n/c = d$, regardless of whether $\lambda = 0$ or $\lambda > 0$.

Our Core Contribution

We have clearly motivated why a smaller coreset size is expected for the case when $\lambda > 0$, the regularized sensitivity becomes $1/(c+\lambda)$, and total sensitivity reduces to $n/(c+\lambda) < d$. In fact, for higher values of λ, the total sensitivity score could be significantly smaller. Therefore, regularization reduces the size of the coreset by at least a factor of $c/(c+\lambda)$.

Experiments

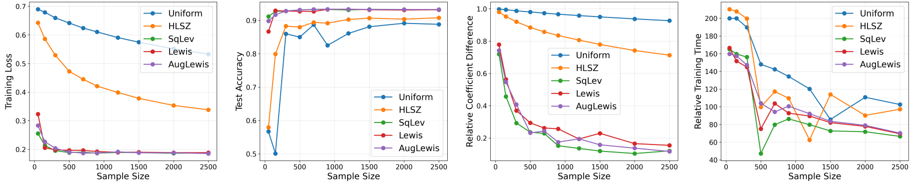


Figura 1: VRLog Coreset Performance (Credit Card)

VRLog (Credit Card Dataset): Our method, AugLewis, consistently outperforms existing baselines (Uniform, HLSZ, SqLev, Lewis) in F1 score and model recovery, while achieving up to 100× faster training with strong approximation guarantees on training loss. [1]

[1] Codes available at `https://github.com/dcll-iiitd/CoresetForVFL`

Experiments

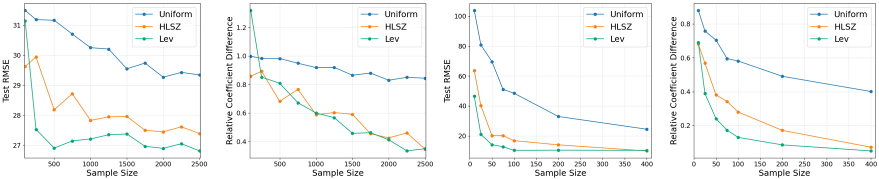


Figura 2: VRLR Coreset Performance (Blog Feedback, Financial Dataset)

VRLR (Blog Feedback Datasets): Our sampling method, Lev, significantly reduces test RMSE and improves model closeness compared to Uniform and leverage-based approaches, aligning with theoretical expectations from regularized sensitivity scores

References

Feldman, D. and Langberg, M. (2011).
A unified framework for approximating and clustering data.
In *Proceedings of the forty-third annual ACM symposium on Theory of computing*, pages 569–578. ACM.

Huang, L., Li, Z., Sun, J., and Zhao, H. (2022).
Coresets for vertical federated learning: Regularized linear regression and k-means clustering.
Advances in Neural Information Processing Systems, 35:29566–29581.