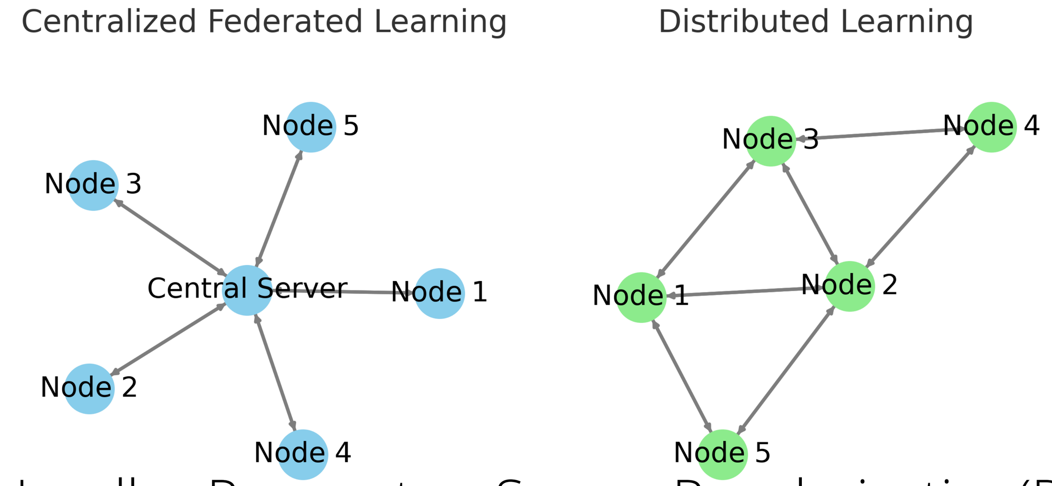
Improving performance of distributed learning through density ensimation

Kacper Trębacz[1 2], Jack Henry Good[2], Artur Dubrawski[2]
[1] Warsaw University of Technology, Warsaw, Poland
[2] Carnegie Mellon University, Pittsburgh, PA, USA

Background

We work in a distributed learning scenario, where we have a set of agents that want to learn a common model without sharing their data. It is similar to federated learning, but there is no central node. We assume extreme imbalance case, where each client data from only 1 class.



Usually, Parameter Space Regularization(PSR) is used to exchange information between models. If θ are model parameters, then it can be stated as:

$$\theta_i = \arg\min_{\theta_i} \left(R_i(\theta_i) + \sum_{j=1}^{m} \lambda_{i,j} \|\theta_i - \theta_j\|^2 \right)$$

Ongoing work at Auton lab proposes using Function Space Regularization(FSR) to exchange knowledge. If $f : \mathbb{R}^n \to \mathbb{R}^m$ is a machine leaning model, then i can be states as:

$$f_i = \arg\min_{f_i} \left(R_i(f_i) + \sum_{j=1}^{m} \lambda_{i,j} \|f_i - f_j\|^2 \right)$$

Objectives

Problem

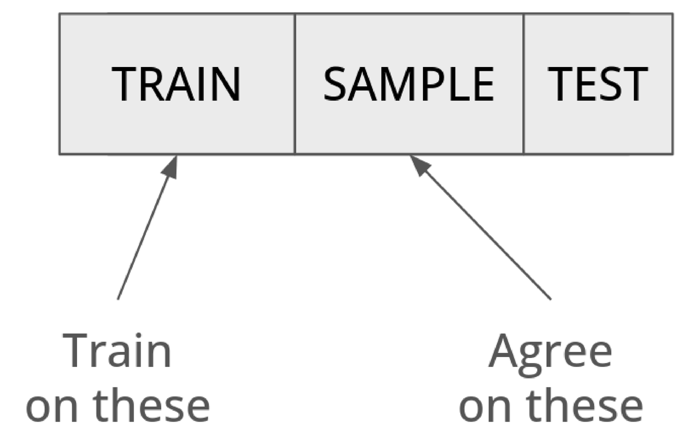
- Although FSR outperforms PSR in non-iid settings it still performs much worse than a centralized model, in some cases

Solution

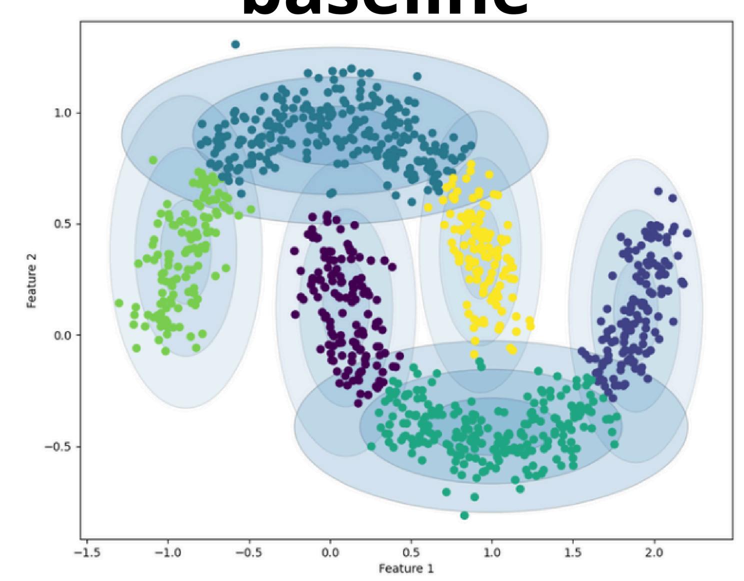
- Leverage density estimation of the data to weight FSR and enforce stronger agreement on parts of the domain with data and weaken in other places
- Add two sampling baselines for comparison

Method

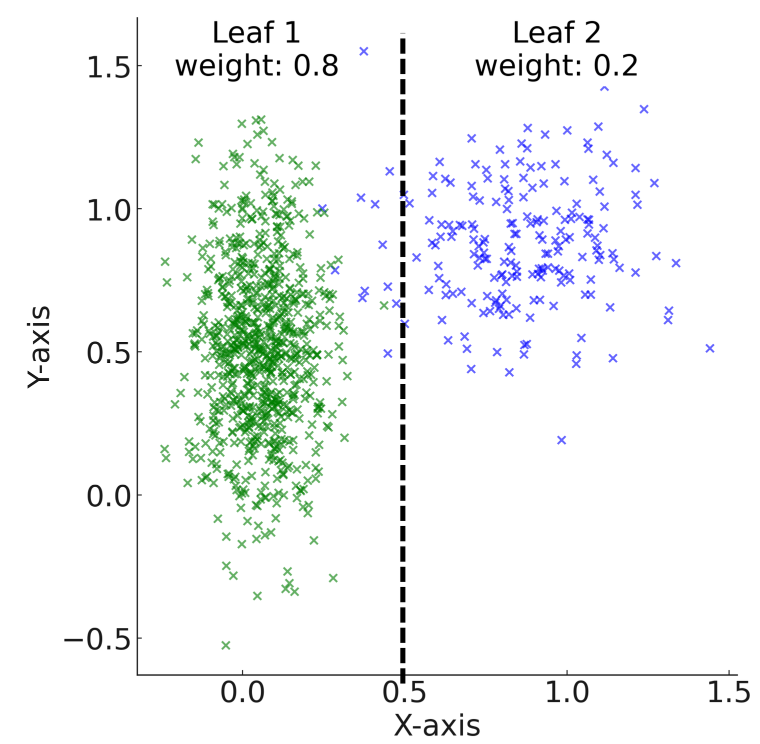
Holdout baseline



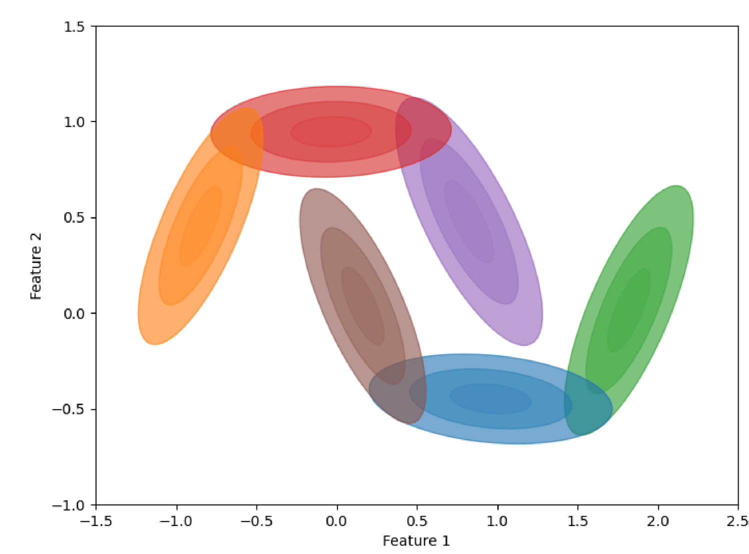
GMM sampling baseline



Weighting by leaf density



Weighting by GMM density



Results

dataset	central	holdout baseline	GMM sampling baseline	leaf density estimate	function space	GMM density estimate
mc16	0.77 (78.0)	0.76 (206.0)	0.72 (135.0)	0.62 (202.0)	0.72 (184.0)	**0.77 (196.0)**
mc32	0.76 (44.0)	0.73 (87.0)	0.70 (152.0)	0.59 (137.0)	0.61 (140.0)	**0.76 (113.0)**
mc64	0.64 (32.0)	0.65 (87.0)	0.62 (89.0)	0.57 (59.0)	0.59 (95.0)	**0.65 (87.0)**
diabetes	0.78 (114)	0.70 (65.0)	0.56 (38.0)	**0.60 (48.0)**	0.58 (28.0)	0.58 (32.0)
glass	0.63 (34.0)	0.21 (19.0)	0.28 (41.0)	0.31 (24.0)	**0.39 (59.0)**	0.34 (62.0)
iris	0.96 (4.0)	0.31 (21.0)	0.89 (6.0)	0.94 (8.0)	0.94 (8.0)	**0.95 (8.0)**
vehicle	0.68 (68.0)	0.35 (61.0)	0.32 (53.0)	**0.43** (93.0)	**0.43** (120)	0.42 (112.0)
wdbc	0.92 (4.0)	0.61 (68.0)	0.87 (8.0)	0.87 (8.0)	0.81 (4.0)	**0.89 (4.0)**
digits	0.85 (123)	0.85 (123)	0.18 (~160)	? (123)	0.26 (~160)	**0.51 (~130)**

Discussion

- On synthetic datasets, weighting by data density significantly improves over base FSR approach and is close to central and holdout model performance
- On real-world datasets, the GMM density method shows a slight increase in performance in most bases and a more than doubles the score for digits dataset
- Good density estimate leads to a significant increase in performance

Future Work

- Consider distributions other than Gaussian and logistic
- Develop tools for choosing a density estimate

Acknowledgment

- This material is based upon work supported by the U.S. Army Research Office and the U.S. Army Futures Command under Contract No. W911NF-20-D-0002. The content of the information does not necessarily reflect the position or the policy of the government, and no official endorsement should be inferred.
- The authors extend their heartfelt appreciation to the CMU RISS program team, particularly Rachel Burcin and Dr. John M. Dolan

References

[1] Almeida, Inês, and Joao Xavier. "Djam: Distributed jacobi asynchronous method for learning personal models." IEEE Signal Processing Letters 25.9 (2018): 1389-1392.

Table 1. **F1 score and (number of leaves)** of the trained decision trees for each proposed method and dataset