NEURAL INFORMATION PROCESSING SYSTEMS

Covariances for Free: Exploiting Mean Distributions for Training-free Federated Learning

Dipam Goswami, Simone Magistri, Kai Wang, Bartłomiej Twardowski, Andrew D. Bagdanov and Joost van de Weijer

Code: https://github.com/dipamgoswami/FedCOF

Computer Vision Center

UAB Universitat Autònoma de Barcelona

Universitas Studiorum Florentina

MICC Media Integration and Communication Center

ideas Research Institute

CityU 香港城市大學 City University of Hong Kong

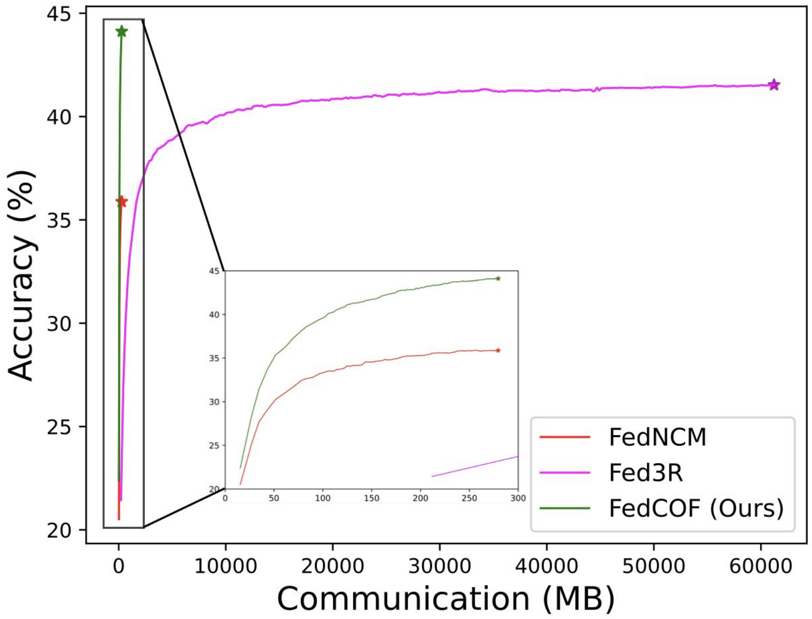
Training-free Federated Learning Methods

- Federated NCM (FedNCM) - Shares first-order statistics (class means and class counts) from each client to server.
- Federated Ridge Regression (Fed3R) - Shares second-order feature statistics (F.F^T) and the sum of class features.

Method	Client Shares	Server Uses	Comm. Cost
FedNCM	$\{\hat{\mu}_{k,c}, n_{k,c}\}_{c=1}^{C'}$	$\{\{\hat{\mu}_{k,c}, n_{k,c}\}_{c=1}^{C'}\}_{k=1}^{K}$	$dC'K$
Fed3R	G_k, B_k	$\{G_k, B_k\}_{k=1}^{K}$	$(dC' + d^2)K$
FedCOF	$\{\hat{\mu}_{k,c}, n_{k,c}\}_{c=1}^{C'}$	$\{\{\hat{\mu}_{k,c}, n_{k,c}\}_{c=1}^{C'}\}_{k=1}^{K}, \{\hat{\Sigma}_c\}_{c=1}^{C'}$	$dC'K$

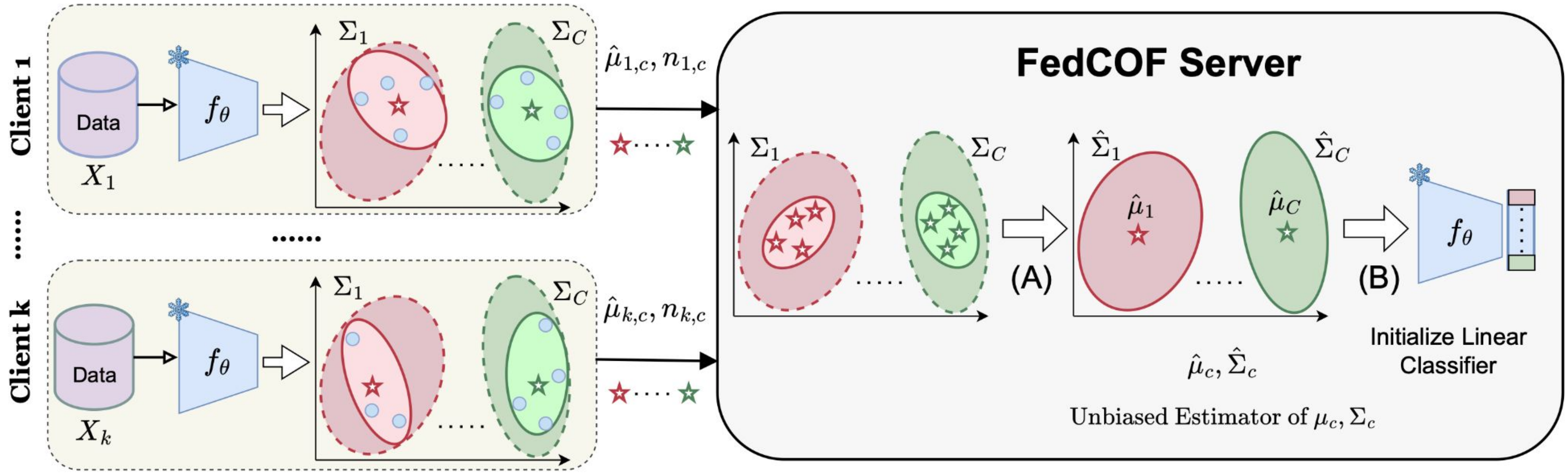
Training-free Federated Learning Methods

➔ We propose **FedCOF** (Federated Learning with COvariances for Free), which shares only first-order statistics from clients and estimates an unbiased global class covariance at the server to initialize the classifier weights.



Pre-trained MobileNetv2 on iNaturalist-Users-120K

Federated Learning with COvariances for Free



Estimating Covariances Using Only Client Means

For client k, let $\{F_{k,c}^j\}_{j=1}^{n_{k,c}}$ denote the feature vectors of class c, where $n_{k,c}$ is the number of samples from class c assigned to client k. The sample mean of these features, $\overline{F}_{k,c} = \frac{1}{n_{k,c}} \sum_{j=1}^{n_{k,c}} F_{k,c}^j$, is itself a random variable with expectation and variance given by:

$$\mathbb{E}[\overline{F}_{k,c}] = \mu_c, \quad \mathrm{Var}[\overline{F}_{k,c}] = \frac{\Sigma_c}{n_{k,c}} \tag{4}$$

Proposition 1. *Let K be the number of clients, each with $n_{k,c}$ features, and let C be the total number of classes. Let $\hat{\mu}_c = \frac{1}{N_c} \sum_{j=1}^{N_c} F^j$ be the unbiased estimator of the population mean μ_c and $N_c = \sum_{k=1}^{K} n_{k,c}$ be the total number of features for a single class. Assuming the features for class c are iid across clients at initialization, the estimator*

$$\hat{\Sigma}_c = \frac{1}{K-1} \sum_{k=1}^{K} n_{k,c} (\overline{F}_{k,c} - \hat{\mu}_c)(\overline{F}_{k,c} - \hat{\mu}_c)^\top \tag{5}$$

is an unbiased estimator of the population covariance Σ_c, for all $c \in 1, \ldots, C$.

Classifier Initialization with Estimated Covariances

Proposition 2. *Let* $F \in \mathbb{R}^{d\times N}$ *be a feature matrix with empirical global mean* $\hat{\mu}_g \in \mathbb{R}^d$, *and* $Y \in \mathbb{R}^{N\times C}$ *be a label matrix. The optimal ridge regression solution* $W^* = (G + \lambda I_d)^{-1}B$, *where* $B \in \mathbb{R}^{d\times C}$ *and* $G \in \mathbb{R}^{d\times d}$ *can be written in terms of class means and covariances as follows:*

$$B = [\hat{\mu}_c N_c]_{c=1}^{C}\,, \quad G = \sum_{c=1}^{C}(N_c - 1)\hat{S}_c + \sum_{c=1}^{C} N_c(\hat{\mu}_c - \hat{\mu}_g)(\hat{\mu}_c - \hat{\mu}_g)^\top + N\hat{\mu}_g\hat{\mu}_g^\top, \tag{7}$$

where the first two terms $G_{with} = \sum_{c=1}^{C}(N_c - 1)\hat{S}_c$ *and* $G_{btw} = \sum_{c=1}^{C} N_c(\hat{\mu}_c - \hat{\mu}_g)(\hat{\mu}_c - \hat{\mu}_g)^\top$ *represents the within-class and between class scatter respectively, while* $\hat{\mu}_c$, $\hat{S}_c$ *and* N_c, *denote the empirical mean, covariance and sample size for class* c, *respectively.*

Removing between-class scatter

- We remove the between-class scatter and obtain the following formulation for initializaing the classifier weights:

$$W^* = \hat{G}^{-1}B; \qquad \hat{G} \quad = \sum_{c=1}^{C}(N_c - 1)\hat{\Sigma}_c + N\hat{\mu}_g\hat{\mu}_g^\top .$$

Dataset	G_{btw}	G_{with}	Acc.(↑)
CIFAR-100	✓	✓	57.1
	✗	✓	**57.3**
ImageNet-R	✓	✓	37.6
	✗	✓	**38.6**
CUB200	✓	✓	50.4
	✗	✓	**53.7**
Stanford Cars	✓	✓	41.4
	✗	✓	**44.8**

Federated Learning with COvariances for Free

Algorithm 1 FedCOF: Federated Learning with COvariances for Free

Client-Side (Client k):

Input: C: set of all classes, f_θ: pre-trained model, $X_{k,c}$: samples of class c in client k, $n_{k,c}$: number of samples in $X_{k,c}$

for $c = 1$ to C **do**

$$\hat{\mu}_{k,c} = \frac{1}{n_{k,c}} \sum_{x \in X_{k,c}} f_\theta(x)$$

end for

Send the class means $\hat{\mu}_{k,c}$ and sample counts $n_{k,c}$ to the Server

Server-Side:

Input: $\hat{\mu}_{k,c}, n_{k,c}$ sent from K clients, $\gamma > 0$

for $c = 1 \ldots C$ **do**

$$\hat{\mu}_c = \frac{1}{N_c} \sum_{k=1}^{K} n_{k,c}\hat{\mu}_{k,c};\ N_c = \sum_{k=1}^{K} n_{k,c}$$

$$\hat{\Sigma}_c = \frac{1}{K-1} \sum_{k=1}^{K} n_{k,c}(\hat{\mu}_{k,c} - \hat{\mu}_c)(\hat{\mu}_{k,c} - \hat{\mu}_c)^\top + \gamma I_d, \text{ Eq.(6)}$$

end for

$$\hat{\mu}_g = \frac{1}{N} \sum_{c=1}^{C} N_c\hat{\mu}_c, \qquad N = \sum_{c=1}^{C} N_c$$

$$B = [\hat{\mu}_c N_c]_{c=1}^{C}, \qquad \hat{G} = \sum_{c=1}^{C} (N_c - 1)\hat{\Sigma}_c + N\hat{\mu}_g\hat{\mu}_g^\top$$

$$W^* = \hat{G}^{-1}B, \text{ Eq. (8)}$$

Normalize W^*: $W_c^* \leftarrow W_c^* / \|W_c^*\| \quad c = 1, \ldots, C$

Experiments

Dataset	Method	SqueezeNet ($d = 512$) Acc (↑)	SqueezeNet ($d = 512$) Comm. (↓)	MobileNetv2 ($d = 1280$) Acc (↑)	MobileNetv2 ($d = 1280$) Comm. (↓)	ViT-B/16 ($d = 768$) Acc (↑)	ViT-B/16 ($d = 768$) Comm. (↓)
CIFAR-100	FedNCM [29]	41.5±0.1	**5.9**	55.6±0.1	**14.8**	55.2±0.1	**8.9**
	Fed3R [11]	**56.9**±0.1	110.2	62.7±0.1	670.1	**73.9**±0.1	244.8
	FedCOF (Ours)	56.1±0.2	**5.9**	**63.5**±0.1	**14.8**	73.2±0.1	**8.9**
	FedCOF Oracle (Full Covs)	56.4±0.1	3015.3	63.9±0.1	18823.5	73.8±0.1	6780.0
IN-R	FedNCM [29]	23.8±0.1	**7.1**	37.6±0.2	**17.8**	32.3±0.1	**10.7**
	Fed3R [11]	37.6±0.2	111.9	46.0±0.3	673.1	**51.9**±0.2	246.6
	FedCOF (Ours)	**37.8**±0.4	**7.1**	**47.4**±0.1	**17.8**	51.8±0.3	**10.7**
	FedCOF Oracle (Full Covs)	38.2±0.1	3645.7	48.0±0.3	22758.8	52.7±0.1	8197.4
CUB200	FedNCM [29]	37.8±0.3	**4.8**	58.3±0.3	**12.0**	75.7±0.1	**7.2**
	Fed3R [11]	50.4±0.3	109.6	58.6±0.2	667.3	77.7±0.1	243.1
	FedCOF (Ours)	**53.7**±0.3	**4.8**	**62.5**±0.4	**12.0**	**79.4**±0.2	**7.2**
	FedCOF Oracle (Full Covs)	54.4±0.1	2472.1	63.1±0.5	15432.7	79.6±0.2	5558.6
Cars	FedNCM [29]	19.8±0.2	**5.4**	30.0±0.1	**13.5**	26.2±0.4	**8.1**
	Fed3R [11]	39.9±0.2	110.2	41.6±0.1	668.8	47.9±0.3	244.0
	FedCOF (Ours)	**44.0**±0.3	**5.4**	**47.3**±0.5	**13.5**	**52.5**±0.3	**8.1**
	FedCOF Oracle (Full Covs)	44.6±0.1	2767.3	47.2±0.3	17275.7	53.1±0.1	6222.5
iNat-120K	FedNCM [29]	21.2±0.1	**111.8**	36.0±0.1	**279.5**	53.9±0.1	**167.7**
	Fed3R [11]	32.1±0.1	9837.3	41.5±0.1	61064.1	62.5±0.1	22050.2
	FedCOF (Ours)	**32.5**±0.1	**111.8**	**44.1**±0.1	**279.5**	**63.1**±0.1	**167.7**
	FedCOF Oracle (Full Covs)	32.4±0.1	57k	43.6±0.1	358k	62.9±0.1	128k

Experiments

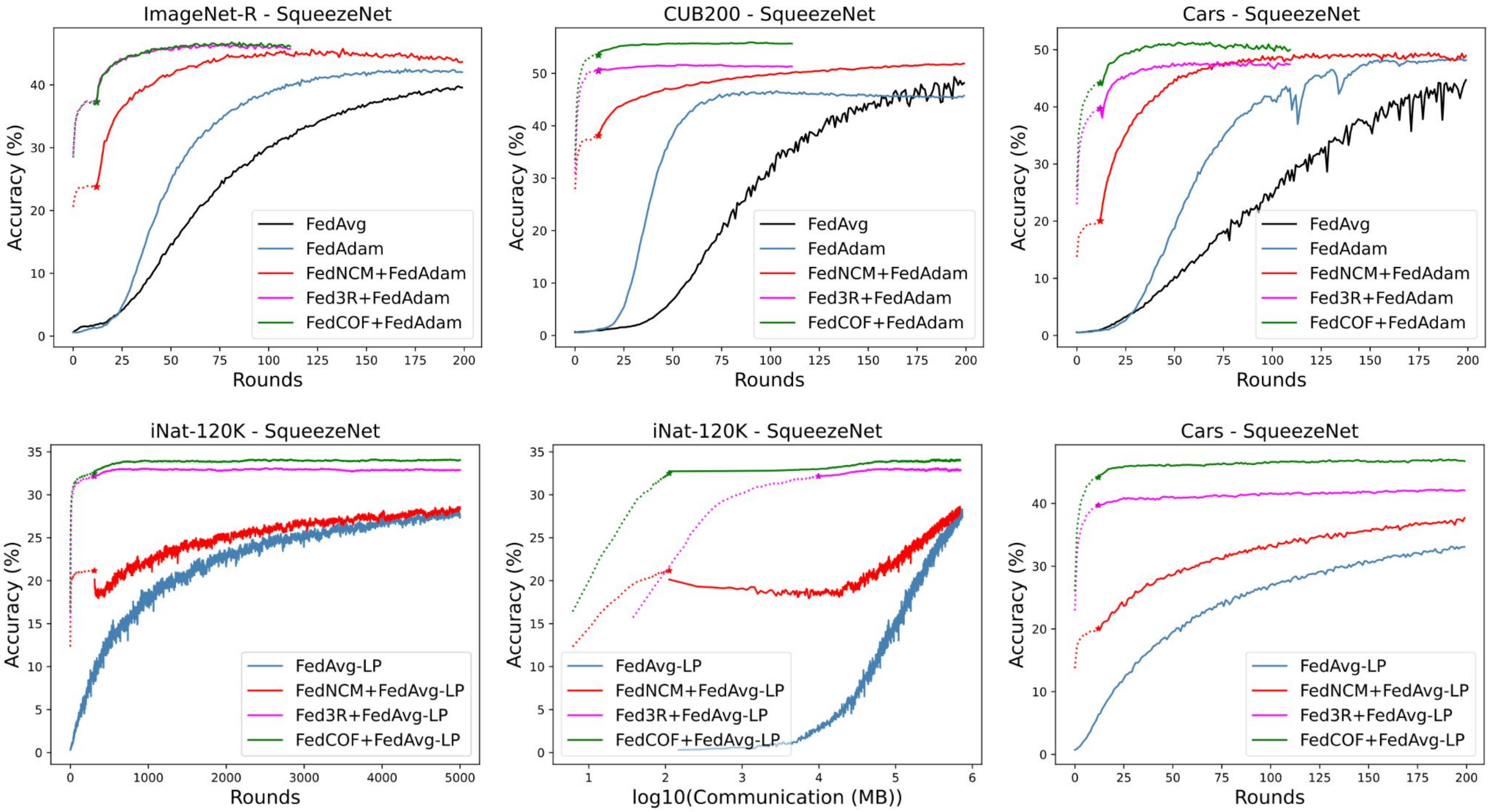
Comparison with prompt-tuning methods

Method	CIFAR-100		IN-R		CUB200		Cars	
	Acc (↑)	Comm. (↓)	Acc (↑)	Comm. (↓)	Acc (↑)	Comm. (↓)	Acc (↑)	Comm. (↓)
FedAvg-PT [54]	74.5±0.5	884.7	47.6±1.3	1622.0	37.0±2.0	1622.0	13.6±1.7	1592.5
FedProx-PT [54]	73.6±0.4	884.7	47.9±0.5	1622.0	38.5±0.8	1622.0	13.7±1.5	1592.5
PFPT [54]	75.1±0.5	846.5	50.7±0.2	1794.4	38.6±0.9	1765.5	12.9±1.1	1736.1
FedCOF (Ours)	**75.3**±0.1	**8.9**	**54.9**±0.2	**10.7**	**65.0**±0.1	**7.2**	**50.4**±0.1	**8.1**

Comparison with training-based methods

Method	Training	ImageNet-R	CUB200	Cars
FedAvg	✓	30.0±0.6	30.3±6.7	24.9±1.6
FedAdam	✓	38.8±0.6	46.4±0.8	41.8±0.6
FedNCM	✗	23.8±0.1	37.8±0.3	19.8±0.2
Fed3R	✗	37.6±0.2	50.4±0.3	39.9±0.2
FedCOF (Ours)	✗	37.8±0.4	53.7±0.3	44.0±0.3
FedNCM+FedAdam	✓	44.7±0.1	50.2±0.2	48.7±0.2
Fed3R+FedAdam	✓	45.9±0.3	51.2±0.3	47.4±0.4
FedCOF+FedAdam	✓	**46.0**±0.4	**55.7**±0.4	**49.6**±0.6

Experiments



Thanks