Federated Learning for the Design of Parametric Insurance Indices

Fallou Niakh
joint work with François Hu

*ENSAE IP Paris
Milliman France

Insurance Data Science Conference 2026

Introduction

Parametric insurance

1. X_i represents the financial loss of individual i
2. Index
   - Similar to the Consumer Price Index, the index Z quantifies risks for individuals through a linear composite of covariates.
   - A threshold z_0 is fixed beyond which an unanticipated loss occurs
3. Parametric insurance compensation

$$m_i(z) = \mathbb{E}[X_i \mid Z = z], \quad \text{for } z > z_0.$$

4. Basis risk

$$\varepsilon_i = X_i - m_i(Z).$$

Parameterizing the Insurer's index Z

Our method

- Selection of covariates $\boldsymbol{Y} = (Y_1, \ldots, Y_J)$ that impacts X_i:
- We propose a weighted sum that gives greater weight to relevant covariates

$$Z = \sum_{j=1}^{J} a_j Y_j$$

Research question : How to choose $\boldsymbol{a} = (a_1, \ldots, a_J)$?

GLM models

We assume that the distribution of X_i belongs to the class of **Tweedie Generalized Linear Models (GLMs)**

$$\mathbb{E}[X_i \mid \boldsymbol{Y}] = (a_{i0} + \boldsymbol{a}_i' \boldsymbol{Y})^{p_i},$$
$$\text{Var}[X_i \mid \boldsymbol{Y}] = \phi_i (\mathbb{E}[X_i \mid \boldsymbol{Y}])^{q_i},$$

where $a_{i0} \in \mathbb{R}$, $\boldsymbol{a}_i \in \mathbb{R}^J$, $\phi_i \in \mathbb{R}_+$, p_i is the power link parameter, and q_i represents the power variance parameter

First method: see ANOR paper [Nia+25]

Approximation used in [Nia+25]:

$$\mathrm{Var}[X_i \mid Z = z] = \phi_i \left(\mathbb{E}[X_i \mid Z = z]\right)^{q_i},$$

Aggregation by optimization

Find the index Z to minimize:

$$\sum_{i=1}^{n} \mathbb{E}\left[\mathrm{Var}\left[X_i \mid Z\right] \mid Z > z_0\right].$$

We do not know $\mathbb{E}[X_i \mid Z = z]$ but we can approximate it.

Rewriting the linear score:

$$a_{i0} + \boldsymbol{a}_i' \boldsymbol{Y} = a_{i0} + Z + \|\boldsymbol{a}_i - \boldsymbol{a}\| \, \boldsymbol{\alpha}_i' \boldsymbol{Y},$$

where $Z = \boldsymbol{a}'\boldsymbol{Y}$, $\boldsymbol{\alpha}_i = \frac{\boldsymbol{a}_i - \boldsymbol{a}}{\|\boldsymbol{a}_i - \boldsymbol{a}\|}$ and $\|\boldsymbol{a}_i - \boldsymbol{a}\|$ is assumed to be small.

First method: see ANOR paper [Nia+25]

Proposition

$$\mathbb{E}[X_i \mid \boldsymbol{a}_i'\boldsymbol{Y} = z] = \mathbb{E}[X_i \mid Z = z] + \|\boldsymbol{a}_i - \boldsymbol{a}\| L_i(z, \boldsymbol{a}) + \frac{1}{2}\|\boldsymbol{a}_i - \boldsymbol{a}\|^2 F_i(z, \boldsymbol{a}) + o(\|\boldsymbol{a}_i - \boldsymbol{a}\|^2)$$

where

$$L_i(z, \boldsymbol{a}) = -\frac{\partial}{\partial z}\log f(z)\mathrm{Cov}[X_i, \boldsymbol{\alpha}_i'\boldsymbol{Y} \mid Z = z] - \frac{\partial}{\partial z}\mathrm{Cov}[X_i, \boldsymbol{\alpha}_i'\boldsymbol{Y} \mid Z = z] + \mathbb{E}[\boldsymbol{\alpha}_i'\boldsymbol{Y} \mid Z = z]\frac{\partial}{\partial z}\mathbb{E}[X_i \mid Z = z].$$

$$\begin{aligned}F_i(z, \boldsymbol{a}) = {} & \left[\frac{\partial^2}{\partial z^2}\log f(z) + \left(\frac{\partial}{\partial z}\log f(z)\right)^2\right]\mathrm{Cov}\left[X_i, (\alpha_i'\boldsymbol{Y})^2 | Z = z\right] \\ & + 2\frac{\partial}{\partial z}\log f(z)\left[\frac{\partial}{\partial z}\mathrm{Cov}\left[X_i, (\alpha_i'\boldsymbol{Y})^2 | Z = z\right] + \mathbb{E}[(\alpha_i'\boldsymbol{Y})^2 | Z = z]\frac{\partial}{\partial z}\mathbb{E}[X_i | Z = z]\right] \\ & + \frac{\partial^2}{\partial z^2}\mathrm{Cov}\left[X_i, (\alpha_i'\boldsymbol{Y})^2 | Z = z\right] + 2\frac{\partial}{\partial z}\mathbb{E}[(\alpha_i'\boldsymbol{Y})^2 | Z = z]\frac{\partial}{\partial z}\mathbb{E}[X_i | Z = z] \\ & + \mathbb{E}[(\alpha_i'\boldsymbol{Y})^2 | Z = z]\frac{\partial^2}{\partial z^2}\mathbb{E}[X_i | Z = z] \\ & - 2\left[\frac{\partial}{\partial z}\log f(z)\,\mathbb{E}[\alpha_i'\boldsymbol{Y} | Z = z] + \frac{\partial}{\partial z}\mathbb{E}[\alpha_i'\boldsymbol{Y} | Z = z]\right] L_i(z, \boldsymbol{a}).\end{aligned}$$

Estimate Z using federated learning [Nia26]

(A1) *Bounded index coefficients:* optimization is restricted to $\mathcal{A} = \{\boldsymbol{a} : \|\boldsymbol{a}\|_2 \leq M\}$. *(The index cannot generate unbounded payouts, a standard actuarial restriction [MN89].)*

(A2) *Strictly positive index:* $\boldsymbol{a}^\top \boldsymbol{Y} \geq z_0 > 0$ almost surely on $\mathcal{A}$. *(The parametric index always returns a well-defined positive payout, ensuring $\tilde{\mu}_i = Z^{p_i}$ is well-defined.)*

Federated Learning Formulation

We aim to solve the global optimization problem :

$$\min_{\boldsymbol{a}\in\mathcal{A}} \sum_{i=1}^{N} \omega_i \ell_i(\boldsymbol{a}) = \min_{\boldsymbol{a}\in\mathcal{A}} \sum_{i=1}^{N} \omega_i \mathbb{E}\left[\frac{1}{\phi_i}\left(\frac{\tilde{\mu}_i^{2-q_i}}{2-q_i} - \frac{X_i\,\tilde{\mu}_i^{1-q_i}}{1-q_i}\right)\right]$$

under the constraint that individual i observes only its own data.

Federated learning algorithm

- We consider the following pseudo-algorithm of federated learning

---

Algorithm 1 Federated Learning for Parametric Index Design

Input: Number of rounds T, local epochs E, step size η
Output: Global parameter $\boldsymbol{a}^{(T)}$

Server initializes global parameters $\boldsymbol{a}^{(0)}$
for $t = 0$ **to** $T-1$ **do**
  Server broadcasts $\boldsymbol{a}^{(t)}$ to all clients
  for all clients $i = 1, \ldots, N$ **in parallel do**
    $\boldsymbol{a}_i^{(t,0)} \leftarrow \boldsymbol{a}^{(t)}$
    for $e = 1$ **to** E **do**
      Sample mini-batch B_i
      $\boldsymbol{a}_i^{(t,e)} \leftarrow \boldsymbol{a}_i^{(t,e-1)} - \eta \nabla \ell_i(\boldsymbol{a}_i^{(t,e-1)}; B_i)$
    end for
    Client sends $\boldsymbol{a}_i^{(t+1)} \leftarrow \boldsymbol{a}_i^{(t,E)}$ to server
  end for
  Server aggregates

$$\boldsymbol{a}^{(t+1)} \leftarrow \text{Aggregate}\left(\{\boldsymbol{a}_i^{(t+1)}\}_{i=1}^{N}\right)$$

end for

Federated Averaging (FedAvg)

Federated Averaging (FedAvg)

The simplest aggregation rule is the weighted average :

$$\boldsymbol{a}^{(t+1)} = \sum_{i=1}^{N} \omega_i \boldsymbol{a}_i^{(t+1)}.$$

where $\boldsymbol{a}^{(t)}$ the global iterate at round t.

- Gradient dissimilarity and client drift

$$B^2 = \sup_{\boldsymbol{a} \in \mathcal{A}} \sum_{i=1}^{N} \omega_i \left\| \nabla \ell_i(\boldsymbol{a}) - \nabla \ell(\boldsymbol{a}) \right\|_2^2 = \Omega\left(\max_i \phi_i^{-2} \right)$$

which measures how far each client's gradient is from the consensus gradient $\nabla \ell = \sum_{i=1}^{N} \omega_i \nabla \ell_i$.

How FedProx and FedOpt fix the problem

FedProx [Li+20]

Adds a quadratic penalty λ to each client's local objective:

$$\ell_i^{\text{prox}}(\boldsymbol{a}; \boldsymbol{a}^{(t)}) \;=\; \ell_i(\boldsymbol{a}) + \frac{\lambda}{2} \|\boldsymbol{a} - \boldsymbol{a}^{(t)}\|_2^2.$$

FedOpt [Red+21]

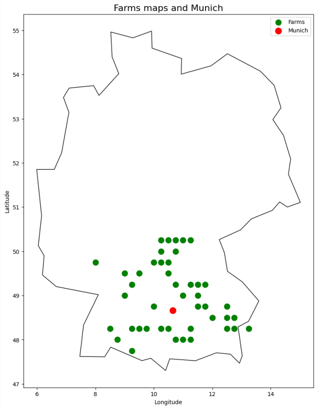
Corrects the imbalance at the server level rather than locally. Let $\boldsymbol{\Delta}^{(t)}$ denote the aggregated pseudo-gradient. The FedAdam variant maintains the second-moment estimate $\boldsymbol{v}^{(t)}$ and applies a coordinate-wise rescaling:

$$\boldsymbol{a}^{(t+1)} \;=\; \boldsymbol{a}^{(t)} - \eta \frac{\hat{\boldsymbol{m}}^{(t)}}{\sqrt{\hat{\boldsymbol{v}}^{(t)}} + \varepsilon}.$$

Case Study/Application

Farms in Germany

- We consider German solar producers



Visualized here: 50 green points corresponding to the solar farms, and the red point is the reference station (Munich) where weather variables are observed to construct the index.

Data collection from ECMWF and initial variables

For each solar park i and each hour h of a day d from January 2012 to December 2022, we collect the following data:

- $\widehat{SSRD}_{i,d,h}$: Day ahead forecast of surface solar radiation downwards.
- $SSRD_{i,d,h}$: Actual surface solar radiation downwards.
- $\widehat{DNI}_{i,d,h}$: Day ahead forecast of direct normal irradiance.
- $DNI_{i,d,h}$: Actual direct normal irradiance.

We also collected weather data for the reference station (Munich):

- $\widehat{SSRD}_{0,d,h}$, $SSRD_{0,d,h}$.
- $\widehat{DNI}_0^{d,h}$, $DNI_{0,d,h}$.

Estimation of energy loss

- Solar production forecasts and actual of each park i are obtained using PVlib in Python:

$$\widehat{P}_{i,d,h} = \mathcal{F}\left(\widehat{SSRD}_{i,d,h}, \widehat{DNI}_{i,d,h}\right)$$
$$P_{i,d,h} = \mathcal{F}\left(SSRD_{i,d,h}, DNI_{i,d,h}\right)$$

  where $\mathcal{F}$ is the conversion function.

- Forecast errors in Munich and energy loss for park i:

$$\Delta P_{i,d,h} = \widehat{P}_{i,d,h} - P_{i,d,h}$$
$$\Delta SSRD_{0,d,h} = \widehat{SSRD}_{0,d,h} - SSRD_{0,d,h}$$
$$\Delta DNI_{0,d,h} = \widehat{DNI}_{0,d,h} - DNI_{0,d,h}$$

Aggregation to daily data

To convert hourly data into daily values, we aggregate the variables as follows:

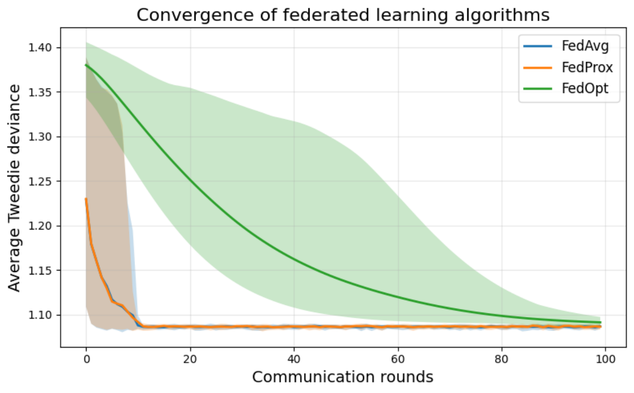
$$\Delta SSRD_{0,d} = \sum_{h=h_{sr}}^{h_{ss}} \Delta SSRD_{0,d,h}, \Delta DNI_{0,d} = \sum_{h=h_{sr}}^{h_{ss}} \Delta DNI_{0,d,h}$$

$$\Delta P_{i,d} = \sum_{h=h_{sr}}^{h_{ss}} \Delta P_{i,d,h}$$

where $h_{sr,d}$ and $h_{ss,d}$ represent the sunrise and sunset hours on day d, respectively.

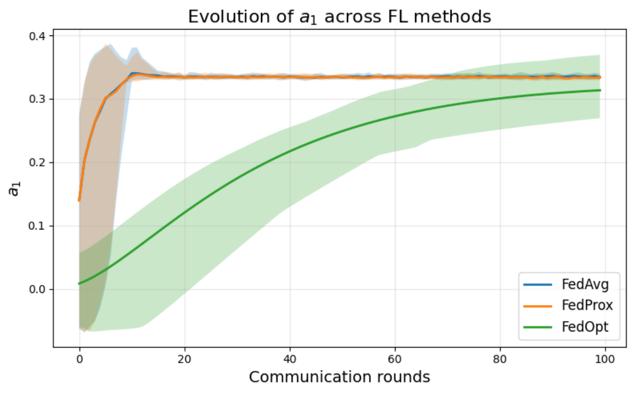
Results

Convergence of Federated Learning



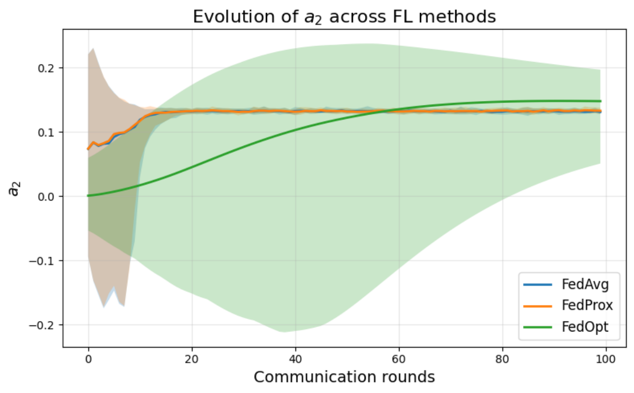
Visualized here: Monte Carlo mean and confidence intervals of the global Tweedie deviance across communication rounds for FedAvg, FedProx, and FedOpt.

Coefficient of SSRD



Visualized here: Monte Carlo mean and confidence intervals of coefficient a_1 of SSRD across federated learning algorithms.

Coefficient of DNI



Visualized here: Monte Carlo mean and confidence intervals of coefficient a_2 of DNI across federated learning algorithms

Conclusion

Conclusion

- We propose a federated optimization scheme that learns a common sensitivity vector to meteorological covariates
- We illustrate the framework on a large-scale case study involving solar power producers in southern Germany
- Federated learning provides a conceptually sound and scalable framework for index calibration that remains valid beyond the regime where analytical approximations hold

References

[Li+20] Tian Li et al. **“Federated Optimization in Heterogeneous Networks”.** In: *Proceedings of Machine Learning and Systems* 2 (2020), pp. 429–450.

[MN89] P. McCullagh and J. A. Nelder. ***Generalized Linear Models.*** 2nd. Chapman & Hall, 1989.

[Nia26] Fallou Niakh. **“Federated Learning for the Design of Parametric Insurance Indices under Heterogeneous Renewable Production Losses”.** In: *arXiv preprint arXiv:2601.12178* (2026).

References II

[Nia+25] Fallou Niakh et al. **“Peer-to-peer basis risk management for renewable production parametric insurance”.** In: *Annals of Operations Research* (2025), pp. 1–47.

[Red+21] Sashank Reddi et al. **“Adaptive Federated Optimization”.** In: *International Conference on Learning Representations (ICLR)*. 2021.