One-shot federated conformal prediction

Pierre Humbert

Joint work with Batiste Le Bars, Aurelien Bellet, Sylvain Arlot

UQSay #67

December 14, 2023

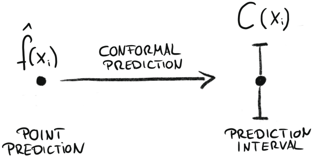
Goal of the paper

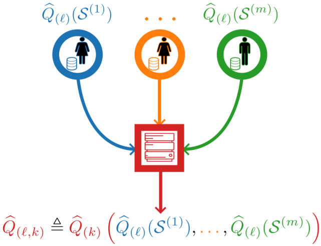
We want to use

- Conformal Prediction methods

In a

- Federated Learning environment




$\widehat{Q}_{(\ell)}(\mathcal{S}^{(1)})$
$\dots$
$\widehat{Q}_{(\ell)}(\mathcal{S}^{(m)})$

$$\widehat{Q}_{(\ell,k)} \triangleq \widehat{Q}_{(k)}\left(\widehat{Q}_{(\ell)}(\mathcal{S}^{(1)}), \dots, \widehat{Q}_{(\ell)}(\mathcal{S}^{(m)})\right)$$

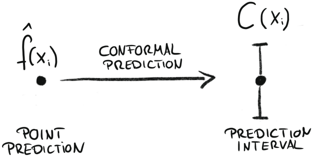
Goal of the paper

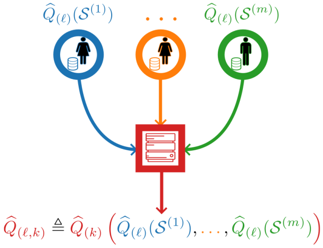
We want to use

- Conformal Prediction methods

In a

- Federated Learning environment




$\widehat{Q}_{(\ell)}(\mathcal{S}^{(1)})$
$\dots$
$\widehat{Q}_{(\ell)}(\mathcal{S}^{(m)})$

$$\widehat{Q}_{(\ell,k)} \triangleq \widehat{Q}_{(k)}\left(\widehat{Q}_{(\ell)}(\mathcal{S}^{(1)}), \dots, \widehat{Q}_{(\ell)}(\mathcal{S}^{(m)})\right)$$

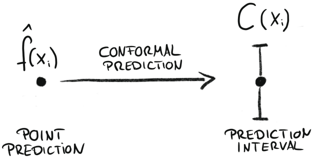
Goal of the paper

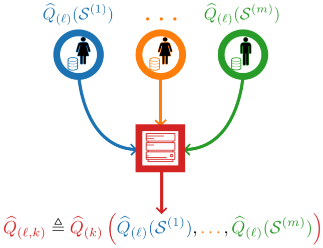
We want to use

- Conformal Prediction methods

In a

- Federated Learning environment




$\widehat{Q}_{(\ell)}(\mathcal{S}^{(1)})$
$\dots$
$\widehat{Q}_{(\ell)}(\mathcal{S}^{(m)})$
$\widehat{Q}_{(\ell,k)} \triangleq \widehat{Q}_{(k)}\left(\widehat{Q}_{(\ell)}(\mathcal{S}^{(1)}), \dots, \widehat{Q}_{(\ell)}(\mathcal{S}^{(m)})\right)$

Conformal Prediction (CP)

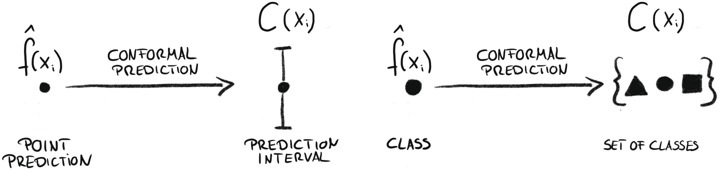
CP provides uncertainty evaluation in the prediction of an algorithm

In a supervised problem

- Given a new observation
  ⟶ Predict its associated response (point prediction)

In conformal prediction

- Given a new observation
  ⟶ Construct a set containing the true response with high probability



Conformal Prediction (CP)

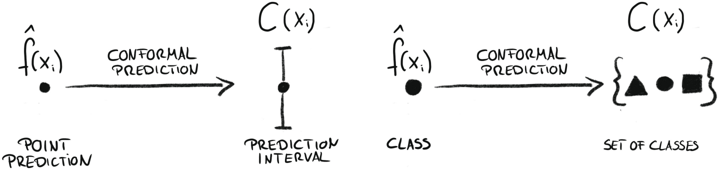
CP provides uncertainty evaluation in the prediction of an algorithm

In a supervised problem

- Given a new observation
  ⟶ Predict its associated response (point prediction)

In conformal prediction

- Given a new observation
  ⟶ Construct a set containing the true response with high probability



Conformal Prediction (CP)

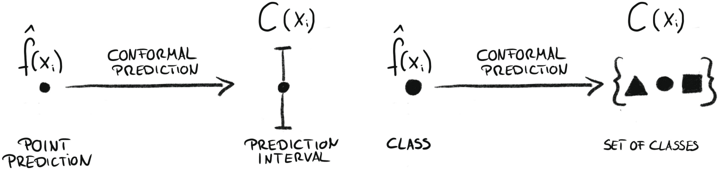
CP provides uncertainty evaluation in the prediction of an algorithm

In a supervised problem

- Given a new observation
  $\longrightarrow$ Predict its associated response (point prediction)

In conformal prediction

- Given a new observation
  $\longrightarrow$ Construct a set containing the true response with high probability



Conformal Prediction (CP)

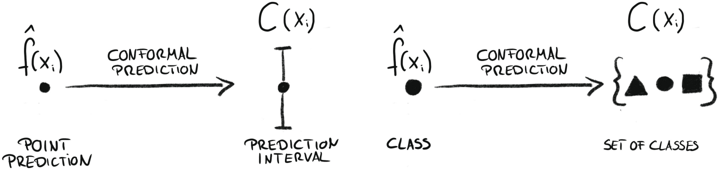
CP provides uncertainty evaluation in the prediction of an algorithm

In a supervised problem

- Given a new observation
  $\longrightarrow$ Predict its associated response (point prediction)

In conformal prediction

- Given a new observation
  $\longrightarrow$ Construct a set containing the true response with high probability



Conformal Prediction (CP)

Motivations

1. Point predictions are uncertain and not sufficiently conservative
   Ex: We want to be conservative when diagnosing a disease

2. Non-conformal techniques have poor statistical guarantees
   $\longrightarrow$ CP allows the calibration of algorithms
   (e.g. quantile regression)

Conformal Prediction (CP)

Setup

n i.i.d. random variables $Z_1 = (X_1, Y_1), \ldots, Z_n = (X_n, Y_n) \sim P$

Goal

For $Z = (X, Y) \sim P$ and a given $\alpha \in (0, 1)$, construct a prediction set $C(X)$ such that

$$\mathbb{P}(Y \in C(X)) \geq 1 - \alpha \;, \tag{1}$$

for **any distribution** P and **any sample size** n.

$\longrightarrow$ If $C(X)$ satisfies equation (1), it is called *marginally valid*.

Question: How to construct $C(X)$?

Conformal Prediction (CP)

Setup

n i.i.d. random variables $Z_1 = (X_1, Y_1), \dots, Z_n = (X_n, Y_n) \sim P$

Goal

For $Z = (X, Y) \sim P$ and a given $\alpha \in (0, 1)$, construct a prediction set $C(X)$ such that

$$\mathbb{P}(Y \in C(X)) \geq 1 - \alpha \ , \tag{1}$$

for **any distribution** P and **any sample size** n.

$\longrightarrow$ If $C(X)$ satisfies equation (1), it is called *marginally valid*.

Question: How to construct $C(X)$?

Conformal Prediction (CP)

Setup

n i.i.d. random variables $Z_1 = (X_1, Y_1), \ldots, Z_n = (X_n, Y_n) \sim P$

Goal

For $Z = (X, Y) \sim P$ and a given $\alpha \in (0, 1)$, construct a prediction set $C(X)$ such that

$$\mathbb{P}(Y \in C(X)) \geq 1 - \alpha \ , \tag{1}$$

for **any distribution** P and **any sample size** n.

$\longrightarrow$ If $C(X)$ satisfies equation (1), it is called *marginally valid.*

Question: How to construct $C(X)$?

Conformal Prediction (CP)

Setup

n i.i.d. random variables $Z_1 = (X_1, Y_1), \ldots, Z_n = (X_n, Y_n) \sim P$

Goal

For $Z = (X, Y) \sim P$ and a given $\alpha \in (0, 1)$, construct a prediction set $C(X)$ such that

$$\mathbb{P}(Y \in C(X)) \geq 1 - \alpha \ , \tag{1}$$

for **any distribution** P and **any sample size** n.

$\longrightarrow$ If $C(X)$ satisfies equation (1), it is called *marginally valid.*

Question: How to construct $C(X)$?

Conformal Prediction (CP)

Setup

n i.i.d. random variables $Z_1 = (X_1, Y_1), \ldots, Z_n = (X_n, Y_n) \sim P$

Goal

For $Z = (X, Y) \sim P$ and a given $\alpha \in (0, 1)$, construct a prediction set $C(X)$ such that

$$\mathbb{P}(Y \in C(X)) \geq 1 - \alpha \ , \tag{1}$$

for **any distribution** P and **any sample size** n.

$\longrightarrow$ If $C(X)$ satisfies equation (1), it is called *marginally valid.*

Question: How to construct $C(X)$?

Some candidates for C

Two marginally valid sets

- $C(X) = \{y \mid y \leq q_Y(1-\alpha)\}$ where $q_Y(1-\alpha)$ is the true quantile of order $(1-\alpha)$ of the law of Y.
  $\longrightarrow$ We need to know P_Y.
- $C(X) = \mathbb{R}$, $(1-\alpha) \cdot 100\%$ of the time and $C(X) = \emptyset$ else.
  $\longrightarrow$ Not informative.

Question: How to construct **"a good"** $C(X)$?
(marginally valid **and** as small as possible)

Some candidates for C

Two marginally valid sets

- $C(X) = \{y \mid y \leq q_Y(1-\alpha)\}$ where $q_Y(1-\alpha)$ is the true quantile of order $(1-\alpha)$ of the law of Y.
  $\longrightarrow$ We need to know P_Y.
- $C(X) = \mathbb{R}$, $(1-\alpha) \cdot 100\%$ of the time and $C(X) = \emptyset$ else.
  $\longrightarrow$ Not informative.

Question: How to construct **"a good"** $C(X)$?
(marginally valid **and** as small as possible)

Some candidates for C

Two marginally valid sets

- $C(X) = \{y \mid y \leq q_Y(1-\alpha)\}$ where $q_Y(1-\alpha)$ is the true quantile of order $(1-\alpha)$ of the law of Y.
  $\longrightarrow$ We need to know P_Y.
- $C(X) = \mathbb{R}$, $(1-\alpha) \cdot 100\%$ of the time and $C(X) = \emptyset$ else.
  $\longrightarrow$ Not informative.

Question: How to construct **"a good"** $C(X)$?
(marginally valid **and** as small as possible)

Conformal Prediction

Two important methods

- Split Conformal Prediction (Papadopoulos et al., 2002)
  Good theoretical guarantees and very low computational cost

- Full Conformal Prediction (Vovk et al., 2005)
  Better marginal theoretical guarantees but high computational cost

In practice ⟶ Split Conformal Prediction

Conformal Prediction

Two important methods

- Split Conformal Prediction (Papadopoulos et al., 2002)
  Good theoretical guarantees and very low computational cost
- Full Conformal Prediction (Vovk et al., 2005)
  Better marginal theoretical guarantees but high computational cost

In practice ⟶ Split Conformal Prediction

Conformal Prediction

Two important methods

- Split Conformal Prediction (Papadopoulos et al., 2002)
  Good theoretical guarantees and very low computational cost

- Full Conformal Prediction (Vovk et al., 2005)
  Better marginal theoretical guarantees but high computational cost

In practice ⟶ Split Conformal Prediction

Conformal Prediction

Two important methods

- Split Conformal Prediction (Papadopoulos et al., 2002)
  Good theoretical guarantees and very low computational cost
- Full Conformal Prediction (Vovk et al., 2005)
  Better marginal theoretical guarantees but high computational cost

In practice $\longrightarrow$ Split Conformal Prediction

Split Conformal Prediction

Input: $Z_1, \ldots, Z_n$, test point X, and $\alpha \in (0,1)$.

1. Randomly split $\{1, \ldots, n\}$ into two equal-sized subsets $\mathcal{I}_1$ and $\mathcal{I}_2$ (A **training** set and a **calibration** set)
2. Learn a predictor $\widehat{f}$ on $\{Z_i, i \in \mathcal{I}_1\}$
3. Compute scores $S_i = s_{\widehat{f}}(X_i, Y_i)$ for $i \in \mathcal{I}_2$

   Example: absolute residuals $\quad s_{\widehat{f}}(X_i, Y_i) = |Y_i - \widehat{f}(X_i)|$
4. $\hat{\boldsymbol{q}}_{\boldsymbol{k}}$: the k-th smallest value in $\{S_i\}_{i \in \mathcal{I}_2}$ with $k = \lceil (1-\alpha)(|\mathcal{I}_2| + 1) \rceil$ (computation of the empirical **quantile**)
5. Return the set
$$\widehat{C}(X) = \{y : s(y, X) \leq \hat{\boldsymbol{q}}_{\boldsymbol{k}}\}$$

Split Conformal Prediction

Input: $Z_1, \ldots, Z_n$, test point X, and $\alpha \in (0, 1)$.

1. Randomly split $\{1, \ldots, n\}$ into two equal-sized subsets $\mathcal{I}_1$ and $\mathcal{I}_2$ (A **training** set and a **calibration** set)
2. Learn a predictor $\widehat{f}$ on $\{Z_i, i \in \mathcal{I}_1\}$
3. Compute scores $S_i = s_{\widehat{f}}(X_i, Y_i)$ for $i \in \mathcal{I}_2$

   Example: absolute residuals $\quad s_{\widehat{f}}(X_i, Y_i) = |Y_i - \widehat{f}(X_i)|$
4. $\hat{\boldsymbol{q}}_{\boldsymbol{k}}$: the k-th smallest value in $\{S_i\}_{i \in \mathcal{I}_2}$ with $k = \lceil (1-\alpha)(|\mathcal{I}_2| + 1) \rceil$ (computation of the empirical **quantile**)
5. Return the set

$$\widehat{C}(X) = \{y : s(y, X) \leq \hat{\boldsymbol{q}}_{\boldsymbol{k}}\}$$

Split Conformal Prediction

Input: $Z_1, \dots, Z_n$, test point X, and $\alpha \in (0, 1)$.

1. Randomly split $\{1, \dots, n\}$ into two equal-sized subsets $\mathcal{I}_1$ and $\mathcal{I}_2$ (A **training** set and a **calibration** set)
2. Learn a predictor $\widehat{f}$ on $\{Z_i, i \in \mathcal{I}_1\}$
3. Compute scores $S_i = s_{\widehat{f}}(X_i, Y_i)$ for $i \in \mathcal{I}_2$

   Example: absolute residuals $\quad s_{\widehat{f}}(X_i, Y_i) = |Y_i - \widehat{f}(X_i)|$
4. $\hat{\boldsymbol{q}}_{\boldsymbol{k}}$: the k-th smallest value in $\{S_i\}_{i \in \mathcal{I}_2}$ with $k = \lceil (1 - \alpha)(|\mathcal{I}_2| + 1) \rceil$ (computation of the empirical **quantile**)
5. Return the set

$$\widehat{C}(X) = \{y : s(y, X) \leq \hat{\boldsymbol{q}}_{\boldsymbol{k}}\}$$

Split Conformal Prediction

Input: $Z_1, \dots, Z_n$, test point X, and $\alpha \in (0,1)$.

1. Randomly split $\{1, \dots, n\}$ into two equal-sized subsets $\mathcal{I}_1$ and $\mathcal{I}_2$ (A **training** set and a **calibration** set)
2. Learn a predictor $\widehat{f}$ on $\{Z_i, i \in \mathcal{I}_1\}$
3. Compute scores $S_i = s_{\widehat{f}}(X_i, Y_i)$ for $i \in \mathcal{I}_2$

   Example: absolute residuals $\quad s_{\widehat{f}}(X_i, Y_i) = |Y_i - \widehat{f}(X_i)|$
4. $\hat{\boldsymbol{q}}_{\boldsymbol{k}}$: the k-th smallest value in $\{S_i\}_{i \in \mathcal{I}_2}$ with $k = \lceil (1-\alpha)(|\mathcal{I}_2| + 1) \rceil$ (computation of the empirical **quantile**)
5. Return the set

$$\widehat{C}(X) = \{y : s(y, X) \leq \hat{\boldsymbol{q}}_{\boldsymbol{k}}\}$$

Split Conformal Prediction

Input: $Z_1, \ldots, Z_n$, test point X, and $\alpha \in (0,1)$.

1. Randomly split $\{1, \ldots, n\}$ into two equal-sized subsets $\mathcal{I}_1$ and $\mathcal{I}_2$
   (A **training** set and a **calibration** set)
2. Learn a predictor $\widehat{f}$ on $\{Z_i, i \in \mathcal{I}_1\}$
3. Compute scores $S_i = s_{\widehat{f}}(X_i, Y_i)$ for $i \in \mathcal{I}_2$
   Example: absolute residuals $\quad s_{\widehat{f}}(X_i, Y_i) = |Y_i - \widehat{f}(X_i)|$
4. $\hat{\boldsymbol{q}}_{\boldsymbol{k}}$: the k-th smallest value in $\{S_i\}_{i \in \mathcal{I}_2}$ with $k = \lceil (1-\alpha)(|\mathcal{I}_2| + 1) \rceil$
   (computation of the empirical **quantile**)
5. Return the set

$$\widehat{C}(X) = \{y : s(y, X) \leq \hat{\boldsymbol{q}}_{\boldsymbol{k}}\}$$

Split Conformal Prediction

Input: $Z_1, \ldots, Z_n$, test point X, and $\alpha \in (0,1)$.

1. Randomly split $\{1, \ldots, n\}$ into two equal-sized subsets $\mathcal{I}_1$ and $\mathcal{I}_2$ (A **training** set and a **calibration** set)
2. Learn a predictor $\widehat{f}$ on $\{Z_i, i \in \mathcal{I}_1\}$
3. Compute scores $S_i = s_{\widehat{f}}(X_i, Y_i)$ for $i \in \mathcal{I}_2$
   Example: absolute residuals $\quad s_{\widehat{f}}(X_i, Y_i) = |Y_i - \widehat{f}(X_i)|$
4. $\boldsymbol{\hat{q}_k}$: the k-th smallest value in $\{S_i\}_{i \in \mathcal{I}_2}$ with $k = \lceil (1-\alpha)(|\mathcal{I}_2| + 1) \rceil$
   (computation of the empirical **quantile**)
5. Return the set
$$\widehat{C}(X) = \{y : s(y, X) \leq \boldsymbol{\hat{q}_k}\}$$

Main result on the split method

Theorem

(Vovk et al., 2005; Lei et al., 2018)

The set returns by the Split Conformal Prediction method satisfies

$$\mathbb{P}(Y \in \widehat{C}(X)) \geq 1 - \alpha \ , \tag{2}$$

for any distribution P and any sample size n (distribution-free !).

Moreover, if we assume that the scores $\{S_i\}_{i \in \mathcal{I}_2}, S := s(X, Y)$ are continuous, then

$$\mathbb{P}(Y \in \widehat{C}(X)) \leq 1 - \alpha + \frac{1}{|\mathcal{I}_2| + 1} \ , \tag{3}$$

with $|\mathcal{I}_2|$ the size of the second subset.

Main result on the split method

Quick proof

When the scores are continuous:

$$\text{rank}(S) := 1 + \textstyle\sum_{i \in \mathcal{I}_2} 1\{S_i \leq S\} \sim U(1, \dots, |\mathcal{I}_2| + 1)$$

$\longrightarrow$ distribution-free statistic.

$$\begin{aligned}
\mathbb{P}(Y \in \widehat{C}(X)) &= \mathbb{P}(S \leq \hat{q}) \\
&= \mathbb{P}\left(\text{rank}(S) \leq \lceil (1-\alpha)(|\mathcal{I}_2| + 1) \rceil\right) \\
&= \frac{\lceil (1-\alpha)(|\mathcal{I}_2| + 1) \rceil}{|\mathcal{I}_2| + 1}.
\end{aligned}$$

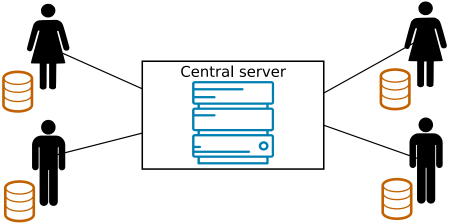
Finally

$$1 - \alpha \leq \frac{\lceil (1-\alpha)(|\mathcal{I}_2| + 1) \rceil}{|\mathcal{I}_2| + 1} \leq 1 - \alpha + \frac{1}{|\mathcal{I}_2| + 1} .$$

Federating Learning

One-shot federated learning

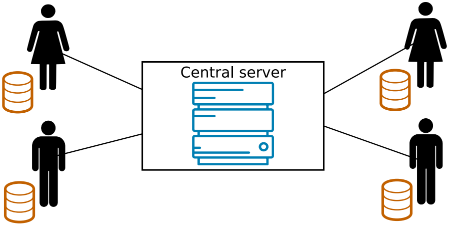
- Some agents connected to a central server
  Local datasets ⟶ Decentralized data set
- **One-shot:** only one round of communication between the agents and the server is allowed



Federating Learning

One-shot federated learning

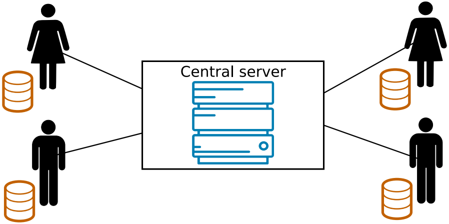
- Some agents connected to a central server
  Local datasets ⟶ Decentralized data set
- **One-shot:** only one round of communication between the agents and the server is allowed



Federating Learning

One-shot federated learning

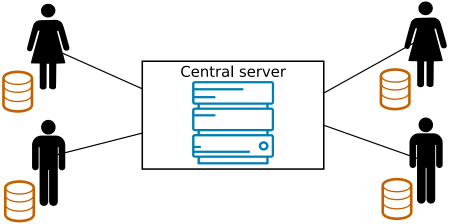
- Some agents connected to a central server
  Local datasets ⟶ Decentralized data set
- **One-shot:** only one round of communication between the agents and the server is allowed



Federating Learning

One-shot federated learning

- Some agents connected to a central server
  Local datasets ⟶ Decentralized data set
- **One-shot:** only one round of communication between the agents and the server is allowed



Federating Learning and Conformal Prediction

Objective in FL

- Find how the agents need to collaborate to improve a particular objective.

  Ex: Learn a regressor using all the data but without sharing them

Objective in FL + CP

- Improve the coverage/length of the final set computed by the server

Federating Learning and Conformal Prediction

Objective in FL

- Find how the agents need to collaborate to improve a particular objective.

  Ex: Learn a regressor using all the data but without sharing them

Objective in FL + CP

- Improve the coverage/length of the final set computed by the server

Federating Learning and Conformal Prediction

Objective in FL

- Find how the agents need to collaborate to improve a particular objective.

  Ex: Learn a regressor using all the data but without sharing them

Objective in FL + CP

- Improve the coverage/length of the final set computed by the server

One-shot federated CP

Setup

1. m agents and a central server
2. n i.i.d. random variables per agent
   $\longrightarrow$ i-th data of agent j: $Z_i^{(j)} = (X_i^{(j)}, Y_i^{(j)}) \sim P$
3. We assume $\hat{f}$ is given (size of the calibration set is mn)

Goal

Construct $\widehat{C}(X)$ such that

$$\mathbb{P}(Y \in \widehat{C}(X)) \geq 1 - \alpha \,, \tag{4}$$

for any distribution P, any sample size, and in **only one round of communication (one-shot FL)**.

Problem: Split CP need to order all the scores $\longrightarrow$ Impossible in one-shot.

One-shot federated CP

Setup

1. m agents and a central server
2. n i.i.d. random variables per agent
   $\longrightarrow$ i-th data of agent j: $Z_i^{(j)} = (X_i^{(j)}, Y_i^{(j)}) \sim P$
3. We assume $\hat{f}$ is given (size of the calibration set is mn)

Goal

Construct $\widehat{C}(X)$ such that

$$\mathbb{P}(Y \in \widehat{C}(X)) \geq 1 - \alpha \ , \tag{4}$$

for any distribution P, any sample size, and in **only one round of communication (one-shot FL)**.

Problem: Split CP need to order all the scores $\longrightarrow$ Impossible in one-shot.

One-shot federated CP

Setup

1. m agents and a central server
2. n i.i.d. random variables per agent
   $\longrightarrow$ i-th data of agent j: $Z_i^{(j)} = (X_i^{(j)}, Y_i^{(j)}) \sim P$
3. We assume $\hat{f}$ is given (size of the calibration set is mn)

Goal

Construct $\widehat{C}(X)$ such that

$$\mathbb{P}(Y \in \widehat{C}(X)) \geq 1 - \alpha \ , \tag{4}$$

for any distribution P, any sample size, and in **only one round of communication (one-shot FL)**.

Problem: Split CP need to order all the scores $\longrightarrow$ Impossible in one-shot.

One-shot federated CP

Setup

1. m agents and a central server
2. n i.i.d. random variables per agent
   $\longrightarrow$ i-th data of agent j: $Z_i^{(j)} = (X_i^{(j)}, Y_i^{(j)}) \sim P$
3. We assume $\hat{f}$ is given (size of the calibration set is mn)

Goal

Construct $\widehat{C}(X)$ such that

$$\mathbb{P}(Y \in \widehat{C}(X)) \geq 1 - \alpha \, , \tag{4}$$

for any distribution P, any sample size, and in **only one round of communication (one-shot FL)**.

Problem: Split CP need to order all the scores $\longrightarrow$ Impossible in one-shot.

One-shot federated CP

The idea

In the split conformal method, we construct

$$\widehat{C}(X) = \{y : s(y, X) \leq \widehat{q}_k\} \ .$$

In One-shot FL, we also construct

$$\widehat{C}(X) = \{y : s(y, X) \leq ?\} \ .$$

The main question

Which $\widehat{q}$

1. is computable in one round of communication (one-shot)
2. and gives a coverage $\geq 1 - \alpha$?

One-shot federated CP

The idea

In the split conformal method, we construct

$$\widehat{C}(X) = \{y : s(y, X) \leq \widehat{q}_k\} \ .$$

In One-shot FL, we also construct

$$\widehat{C}(X) = \{y : s(y, X) \leq ?\} \ .$$

The main question

Which $\widehat{q}$

1. is computable in one round of communication (one-shot)
2. and gives a coverage $\geq 1 - \alpha$?

One-shot federated CP

The idea

In the split conformal method, we construct

$$\widehat{C}(X) = \{y : s(y, X) \leq \widehat{q}_k\} \ .$$

In One-shot FL, we also construct

$$\widehat{C}(X) = \{y : s(y, X) \leq ?\} \ .$$

The main question

Which $\widehat{q}$

1. is computable in one round of communication (one-shot)
2. and gives a coverage $\geq 1 - \alpha$?

One-shot federated CP

The idea

In the split conformal method, we construct

$$\widehat{C}(X) = \{y : s(y, X) \leq \widehat{q}_k\} \ .$$

In One-shot FL, we also construct

$$\widehat{C}(X) = \{y : s(y, X) \leq ?\} \ .$$

The main question

Which $\widehat{q}$

1. is computable in one round of communication (one-shot)
2. and gives a coverage $\geq 1 - \alpha$?

Intuition

Two extreme cases

1. $n = 1$:
   central server need to compute a "quantile" of order $\lceil (m+1)(1-\alpha) \rceil$
2. $m = 1$:
   Standard case, so we compute a "quantile" of order $\lceil (n+1)(1-\alpha) \rceil$

And in the generalize case $(n \geq 1, m \geq 1)$?

Should we compute quantiles ?

$\longrightarrow$ The answer is yes !

Intuition

Two extreme cases

1. $n = 1$:
   central server need to compute a "quantile" of order $\lceil (m+1)(1-\alpha) \rceil$
2. $m = 1$:
   Standard case, so we compute a "quantile" of order $\lceil (n+1)(1-\alpha) \rceil$

And in the generalize case $(n \geq 1, m \geq 1)$?

Should we compute quantiles ?

$\longrightarrow$ The answer is yes !

Intuition

Two extreme cases

1. $n = 1$:
   central server need to compute a "quantile" of order $\lceil (m+1)(1-\alpha) \rceil$
2. $m = 1$:
   Standard case, so we compute a "quantile" of order $\lceil (n+1)(1-\alpha) \rceil$

And in the generalize case $(n \geq 1, m \geq 1)$?

Should we compute quantiles ?

$\longrightarrow$ The answer is yes !

Intuition

Two extreme cases

1. $n = 1$:
   central server need to compute a "quantile" of order $\lceil (m+1)(1-\alpha) \rceil$
2. $m = 1$:
   Standard case, so we compute a "quantile" of order $\lceil (n+1)(1-\alpha) \rceil$

And in the generalize case $(n \geq 1, m \geq 1)$?

Should we compute quantiles ?

$\longrightarrow$ The answer is yes !

How to construct $\widehat{C}$?

Input: $k \in \{1, \ldots, m\}$, $\ell \in \{1, \ldots, n\}$, and $\alpha \in (0, 1)$

1. For j in $\{1, \ldots, m\}$

   Agent j computes local scores $S_i^j = s_{\widehat{f}}(X_i, Y_i)$ for $i \in \{1, \ldots, n\}$

   Agent sends $S_{(\ell)}^j =$ the ℓ-th smallest value in $\{S_i^j\}_{i=1}^n$ to the server

2. Central server computes the k-th smallest value in $(S_{(\ell)}^1, \ldots, S_{(\ell)}^m)$, denoted $\widehat{\boldsymbol{Q}}_{(\boldsymbol{\ell},\boldsymbol{k})}$

3. Return

$$\widehat{C}_{\ell,k}(X) = \{y \mid s(y, X) \leq \widehat{\boldsymbol{Q}}_{(\boldsymbol{\ell},\boldsymbol{k})}\}$$

How to construct $\widehat{C}$?

Input: $k \in \{1, \dots, m\}$, $\ell \in \{1, \dots, n\}$, and $\alpha \in (0, 1)$

1. For j in $\{1, \dots, m\}$

   Agent j computes local scores $S_i^j = s_{\widehat{f}}(X_i, Y_i)$ for $i \in \{1, \dots, n\}$

   Agent sends $S_{(\ell)}^j =$ the ℓ-th smallest value in $\{S_i^j\}_{i=1}^n$ to the server

2. Central server computes the k-th smallest value in $(S_{(\ell)}^1, \dots, S_{(\ell)}^m)$, denoted $\widehat{\boldsymbol{Q}}_{(\boldsymbol{\ell},\boldsymbol{k})}$

3. Return

$$\widehat{C}_{\ell,k}(X) = \{y \mid s(y, X) \leq \widehat{\boldsymbol{Q}}_{(\boldsymbol{\ell},\boldsymbol{k})}\}$$

How to construct $\widehat{C}$?

Input: $k \in \{1, \dots, m\}$, $\ell \in \{1, \dots, n\}$, and $\alpha \in (0, 1)$

1. For j in $\{1, \dots, m\}$

   Agent j computes local scores $S_i^j = s_{\widehat{f}}(X_i, Y_i)$ for $i \in \{1, \dots, n\}$

   Agent sends $S_{(\ell)}^j =$ the ℓ-th smallest value in $\{S_i^j\}_{i=1}^n$ to the server

2. Central server computes the k-th smallest value in $(S_{(\ell)}^1, \dots, S_{(\ell)}^m)$, denoted $\boldsymbol{\widehat{Q}_{(\ell,k)}}$

3. Return

$$\widehat{C}_{\ell,k}(X) = \{y \mid s(y, X) \leq \boldsymbol{\widehat{Q}_{(\ell,k)}}\}$$

How to construct $\widehat{C}$?

Input: $k \in \{1, \dots, m\}$, $\ell \in \{1, \dots, n\}$, and $\alpha \in (0, 1)$

1. For j in $\{1, \dots, m\}$

   Agent j computes local scores $S_i^j = s_{\widehat{f}}(X_i, Y_i)$ for $i \in \{1, \dots, n\}$

   Agent sends $S_{(\ell)}^j$ = the ℓ-th smallest value in $\{S_i^j\}_{i=1}^n$ to the server

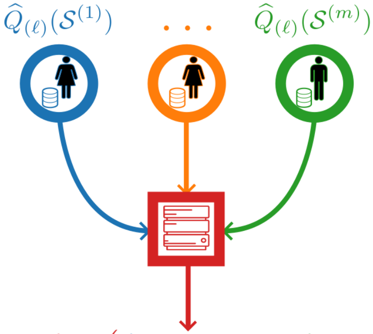
2. Central server computes the k-th smallest value in $(S_{(\ell)}^1, \dots, S_{(\ell)}^m)$, denoted $\boldsymbol{\widehat{Q}_{(\ell,k)}}$

3. Return

$$\widehat{C}_{\ell,k}(X) = \{y \mid s(y, X) \leq \boldsymbol{\widehat{Q}_{(\ell,k)}}\}$$

One-shot federated CP

Formally, we compute the *Quantile-of-Quantiles (QQ)*.


$\widehat{Q}_{(\ell)}(\mathcal{S}^{(1)})$
...
$\widehat{Q}_{(\ell)}(\mathcal{S}^{(m)})$

$$\widehat{Q}_{(\ell,k)} \triangleq \widehat{Q}_{(k)}\left(\widehat{Q}_{(\ell)}(\mathcal{S}^{(1)}), \dots, \widehat{Q}_{(\ell)}(\mathcal{S}^{(m)})\right)$$

Which (ℓ, k) we need to choose ?

Main result

Theorem

For any $(\ell, k) \in \{1, \dots, n\} \times \{1, \dots, m\}$, *the set*

$$\widehat{C}_{\ell,k}(X) = \{y \mid s(y, X) \leq \widehat{Q}_{(\ell,k)}\} \qquad \textit{satisfies:}$$

$$\begin{aligned} \mathbb{P}\left(Y \in \widehat{C}_{\ell,k}(X)\right) &\geq M_{\ell,k} \qquad (5) \\ &\triangleq 1 - \frac{1}{mn+1} \sum_{j=k}^{m} \binom{m}{j} \sum_{I_{1,j}=\ell}^{n} \sum_{I_{1,j}^c=0}^{\ell-1} \frac{\binom{n}{i_1} \cdots \binom{n}{i_m}}{\binom{mn}{i_1+\cdots+i_m}} . \end{aligned}$$

Moreover, when the associated scores $\{S_i^j\}_{i,j=1}^{n,m}$ *and* $S \triangleq s(X, Y)$ *have continuous c.d.f,* (5) *is an equality.*

⟶ As in the centralized case, also a distribution-free bound !

The final set with FedCP-QQ

From the theorem, we know that

$$\mathbb{P}\left(Y \in \widehat{C}_{\ell,k}(X)\right) \geq M_{\ell,k}$$

FedCP-QQ algorithm

FedCP-QQ computes $\widehat{Q}_{(\ell^*,k^*)}$ and returns $\widehat{C}_{\ell^*,k^*}(X)$ where

$$(\ell^*, k^*) = \arg\min_{\ell,k} \{M_{\ell,k} : M_{\ell,k} \geq 1 - \alpha\} \ .$$

$\longrightarrow$ By construction,

$$\mathbb{P}\left(Y \in \widehat{C}_{\ell^*,k^*}(X)\right) \geq 1 - \alpha$$

The final set with FedCP-QQ

From the theorem, we know that

$$\mathbb{P}\left(Y \in \widehat{C}_{\ell,k}(X)\right) \geq M_{\ell,k}$$

FedCP-QQ algorithm

FedCP-QQ computes $\widehat{Q}_{(\ell^*,k^*)}$ and returns $\widehat{C}_{\ell^*,k^*}(X)$ where

$$(\ell^*, k^*) = \arg\min_{\ell,k} \{M_{\ell,k} : M_{\ell,k} \geq 1-\alpha\} \ .$$

$\longrightarrow$ By construction,

$$\mathbb{P}\left(Y \in \widehat{C}_{\ell^*,k^*}(X)\right) \geq 1-\alpha$$

The final set with FedCP-QQ

From the theorem, we know that

$$\mathbb{P}\left(Y \in \widehat{C}_{\ell,k}(X)\right) \geq M_{\ell,k}$$

FedCP-QQ algorithm

FedCP-QQ computes $\widehat{Q}_{(\ell^*,k^*)}$ and returns $\widehat{C}_{\ell^*,k^*}(X)$ where

$$(\ell^*, k^*) = \arg\min_{\ell,k} \{M_{\ell,k} : M_{\ell,k} \geq 1-\alpha\} \ .$$

⟶ By construction,

$$\mathbb{P}\left(Y \in \widehat{C}_{\ell^*,k^*}(X)\right) \geq 1-\alpha$$

Remaining questions

1. How to compute $M_{\ell,k}$?
2. Behavior of (ℓ^*, k^*) when m or $n \longrightarrow +\infty$?
3. Upper bound ?

"Fast" algorithm to compute M

$$M_{\ell,k} \triangleq 1 - \frac{1}{mn+1} \sum_{j=k}^{m} \binom{m}{j} \sum_{I_{1,j}=\ell}^{n} \sum_{I_{1,j}^c=0}^{\ell-1} \frac{\binom{n}{i_1}\cdots\binom{n}{i_m}}{\binom{mn}{i_1+\cdots+i_m}} ,$$

We recognize the p.m.f. of a multivariate hypergeometric distribution:

$$\sum_{I_{1,j}=\ell}^{n} \sum_{I_{1,j}^c=0}^{\ell-1} \frac{\binom{n}{i_1}\cdots\binom{n}{i_m}}{\binom{mn}{i_1+\cdots+i_m}} 1\{i_1+\cdots+i_m=c\}$$

$$= \mathbb{P}(a_1 \leq H_1 \leq b_1, \cdots, a_m \leq H_m \leq b_m)$$

where

$$(a_i, b_i) = \begin{cases} (\ell, n) & \text{if } i \in \{1,\ldots,j\} \\ (0, \ell-1) & \text{if } i \in \{j+1,\ldots,m\} \end{cases} ,$$

and $(H_1, \ldots, H_m)$ follows a multivariate hypergeometric distribution with known parameters $\longrightarrow$ fast evaluation with e.g. (Lebrun, 2013)

Illustration of M

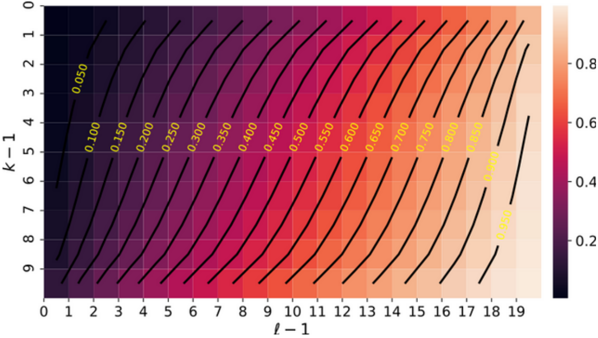


Figure: $m = 10$, $n = 20$; (No need to compute all values of M)

▷ The server computes $M_{\ell,k}$ for all ℓ and k (only once for given m and n)

▷ Quick search because values are ordered by column and row.

Asymptotic behavior of (ℓ^*, k^*)

1. When, $n \longrightarrow +\infty, \quad \ell^*/(n+1) \longrightarrow (1-\alpha)$
   i.e. the agents compute the "true" quantile of order $(1-\alpha)$

2. When $\min(m,n) \longrightarrow +\infty, \quad k^*/(m+1) \longrightarrow 1/2$
   i.e. the server compute the median

Asymptotically, agents send quantiles of order $(1-\alpha)$ and the server takes the median of these quantiles.

Empirical upper bound

Lower bound ⟶ Our theorem. And the upper bound ?

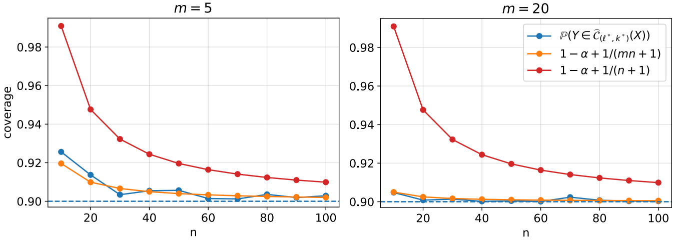


Figure: Comparison of the exact value of $\mathbb{P}(Y \in \widehat{C}_{\ell*,k*}(X))$ (blue) with the upper bound when: data are centralized (orange), there is only one agent (red). Parameters are $\alpha = 0.1, m = \{5, 20\}$, and $n = \{10, \dots, 100\}$.

⟶ Upper bound in $\quad 1 - \alpha + \mathcal{O}(1/(mn+1))$?

Training-conditional coverage in centralized CP

For the marginal guarantee:

$$\mathbb{P}(Y \in \widehat{C}(X)) \geq 1 - \alpha \ , \tag{6}$$

the probability is taken on (X, Y) **and** $\mathcal{D}_n = (X_i, Y_i)_{i=1}^n$.

Problem

In practice, we only have access to one data set

$\longrightarrow$ We want guarantee for this particular data set

Training-conditional coverage in centralized CP

For the marginal guarantee:

$$\mathbb{P}(Y \in \widehat{C}(X)) \geq 1 - \alpha \ , \tag{6}$$

the probability is taken on (X, Y) **and** $\mathcal{D}_n = (X_i, Y_i)_{i=1}^n$.

Problem

In practice, we only have access to one data set

$\longrightarrow$ We want guarantee for this particular data set

Training-conditional coverage in centralized CP

For the marginal guarantee:

$$\mathbb{P}(Y \in \widehat{C}(X)) \geq 1 - \alpha \ , \tag{6}$$

the probability is taken on (X, Y) **and** $\mathcal{D}_n = (X_i, Y_i)_{i=1}^n$.

Problem

In practice, we only have access to one data set

$\longrightarrow$ We want guarantee for this particular data set

Training-conditional coverage in centralized CP

For the marginal guarantee:

$$\mathbb{P}(Y \in \widehat{C}(X)) \geq 1 - \alpha \ , \tag{6}$$

the probability is taken on (X, Y) **and** $\mathcal{D}_n = (X_i, Y_i)_{i=1}^n$.

Problem

In practice, we only have access to one data set

$\longrightarrow$ We want guarantee for this particular data set

Training-conditional coverage in CP

Definition

The conditional miscoverage rate is:

$$\alpha(\mathcal{D}_n) = \mathbb{P}(Y \notin \widehat{C}(X) \mid \mathcal{D}_n) \tag{7}$$

Remark: $\mathbb{E}(1 - \alpha(\mathcal{D}_n)) = \mathbb{P}(Y \in \widehat{C}(X)) \geq 1 - \alpha$

Marginal guarantees control only the expectation and **not** the variance

For given α and δ in $(0, 1)$, we want guarantee of the form:

$$\mathbb{P}(\alpha(\mathcal{D}_n) \leq \alpha + \cdots) \geq 1 - \delta \ . \tag{8}$$

Training-conditional coverage in CP

Definition

The conditional miscoverage rate is:

$$\alpha(\mathcal{D}_n) = \mathbb{P}(Y \notin \widehat{C}(X) \mid \mathcal{D}_n) \tag{7}$$

Remark: $\mathbb{E}(1 - \alpha(\mathcal{D}_n)) = \mathbb{P}(Y \in \widehat{C}(X)) \geq 1 - \alpha$

Marginal guarantees control only the expectation and **not** the variance

For given α and δ in $(0, 1)$, we want guarantee of the form:

$$\mathbb{P}(\alpha(\mathcal{D}_n) \leq \alpha + \cdots) \geq 1 - \delta\ . \tag{8}$$

Training-conditional coverage in CP

Definition

The conditional miscoverage rate is:

$$\alpha(\mathcal{D}_n) = \mathbb{P}(Y \notin \widehat{C}(X) \mid \mathcal{D}_n) \tag{7}$$

Remark: $\mathbb{E}(1 - \alpha(\mathcal{D}_n)) = \mathbb{P}(Y \in \widehat{C}(X)) \geq 1 - \alpha$

Marginal guarantees control only the expectation and **not** the variance

For given α and δ in $(0, 1)$, we want guarantee of the form:

$$\mathbb{P}(\alpha(\mathcal{D}_n) \leq \alpha + \cdots) \geq 1 - \delta\ . \tag{8}$$

Training-conditional coverage in CP

Definition

The conditional miscoverage rate is:

$$\alpha(\mathcal{D}_n) = \mathbb{P}(Y \notin \widehat{C}(X) \mid \mathcal{D}_n) \tag{7}$$

Remark: $\mathbb{E}(1 - \alpha(\mathcal{D}_n)) = \mathbb{P}(Y \in \widehat{C}(X)) \geq 1 - \alpha$

Marginal guarantees control only the expectation and **not** the variance

For given α and δ in $(0, 1)$, we want guarantee of the form:

$$\mathbb{P}(\alpha(\mathcal{D}_n) \leq \alpha + \cdots) \geq 1 - \delta\ . \tag{8}$$

Training-conditional coverage in centralized CP

Theorem

(Vovk, 2012)

In the i.i.d. setting, for any distribution P and any $\delta \in [0, 0.5)$,

$$\mathbb{P}\left(\alpha(\mathcal{D}_n) \leq \alpha + \sqrt{\frac{\log(1/\delta)}{2|\mathcal{I}_2|}}\right) \geq 1 - \delta \ , \tag{9}$$

where $\widehat{C}(X)$ is returned by the split conformal method.

And in Federated Learning?

Training-conditional coverage in centralized CP

Theorem

(Vovk, 2012)

In the i.i.d. setting, for any distribution P and any $\delta \in [0, 0.5)$,

$$\mathbb{P}\left(\alpha(\mathcal{D}_n) \leq \alpha + \sqrt{\frac{\log(1/\delta)}{2|\mathcal{I}_2|}}\right) \geq 1 - \delta \ , \tag{9}$$

where $\widehat{C}(X)$ is returned by the split conformal method.

And in Federated Learning?

A result on training-conditional FL

Definition

The (FL) conditional miscoverage rate is

$$\alpha_{\ell,k}(\mathcal{D}_{mn}) = \mathbb{P}\left(Y \notin \widehat{C}_{\ell,k}(X) \mid \mathcal{D}_{mn}\right) \; ,$$

Theorem

If $\delta \in (0, 0.5]$ and $\ell \cdot k \geq (1-\alpha) \cdot mn$, then the conditional miscoverage rate is controlled as follows:

$$\mathbb{P}\left(\alpha_{\ell,k}(\mathcal{D}_{mn}) \leq \alpha + \sqrt{\frac{\log(1/\delta)}{2mn}}\right) \geq 1 - \delta \; . \qquad (10)$$

Remark 1: No lower bound...

Remark 2: $\ell = n, k = \lceil(1-\alpha)m\rceil$ works but too large

Remark 3: Condition not necessarily verified by (ℓ^*, k^*)

A result on training-conditional FL

Definition

The (FL) conditional miscoverage rate is

$$\alpha_{\ell,k}(\mathcal{D}_{mn}) = \mathbb{P}\left(Y \notin \widehat{C}_{\ell,k}(X) \mid \mathcal{D}_{mn}\right) ,$$

Theorem

If $\delta \in (0, 0.5]$ and $\ell \cdot k \geq (1-\alpha) \cdot mn$, then the conditional miscoverage rate is controlled as follows:

$$\mathbb{P}\left(\alpha_{\ell,k}(\mathcal{D}_{mn}) \leq \alpha + \sqrt{\frac{\log(1/\delta)}{2mn}}\right) \geq 1 - \delta . \tag{10}$$

Remark 1: No lower bound...

Remark 2: $\ell = n, k = \lceil (1-\alpha)m \rceil$ works but too large

Remark 3: Condition not necessarily verified by (ℓ^*, k^*)

A result on training-conditional FL

Definition

The (FL) conditional miscoverage rate is

$$\alpha_{\ell,k}(\mathcal{D}_{mn}) = \mathbb{P}\left(Y \notin \widehat{C}_{\ell,k}(X) \mid \mathcal{D}_{mn}\right) \ ,$$

Theorem

If $\delta \in (0, 0.5]$ *and* $\ell \cdot k \geq (1-\alpha) \cdot mn$*, then the conditional miscoverage rate is controlled as follows:*

$$\mathbb{P}\left(\alpha_{\ell,k}(\mathcal{D}_{mn}) \leq \alpha + \sqrt{\frac{\log(1/\delta)}{2mn}}\right) \geq 1 - \delta \ . \tag{10}$$

Remark 1: No lower bound...

Remark 2: $\ell = n, k = \lceil (1-\alpha)m \rceil$ works but too large

Remark 3: Condition not necessarily verified by (ℓ^*, k^*)

A result on training-conditional FL

Definition

The (FL) conditional miscoverage rate is

$$\alpha_{\ell,k}(\mathcal{D}_{mn}) = \mathbb{P}\left(Y \notin \widehat{C}_{\ell,k}(X) \mid \mathcal{D}_{mn}\right) \ ,$$

Theorem

If $\delta \in (0, 0.5]$ and $\ell \cdot k \geq (1-\alpha) \cdot mn$, then the conditional miscoverage rate is controlled as follows:

$$\mathbb{P}\left(\alpha_{\ell,k}(\mathcal{D}_{mn}) \leq \alpha + \sqrt{\frac{\log(1/\delta)}{2mn}}\right) \geq 1 - \delta \ . \tag{10}$$

Remark 1: No lower bound...

Remark 2: $\ell = n, k = \lceil (1-\alpha)m \rceil$ works but too large

Remark 3: Condition not necessarily verified by (ℓ^*, k^*)

A result on training-conditional FL

Definition

The (FL) conditional miscoverage rate is

$$\alpha_{\ell,k}(\mathcal{D}_{mn}) = \mathbb{P}\left(Y \notin \widehat{C}_{\ell,k}(X) \mid \mathcal{D}_{mn}\right) \ ,$$

Theorem

If $\delta \in (0, 0.5]$ and $\ell \cdot k \geq (1-\alpha) \cdot mn$, then the conditional miscoverage rate is controlled as follows:

$$\mathbb{P}\left(\alpha_{\ell,k}(\mathcal{D}_{mn}) \leq \alpha + \sqrt{\frac{\log(1/\delta)}{2mn}}\right) \geq 1 - \delta \ . \qquad (10)$$

Remark 1: No lower bound...

Remark 2: $\ell = n$, $k = \lceil (1-\alpha)m \rceil$ works but too large

Remark 3: Condition not necessarily verified by (ℓ^*, k^*)

A result on training-conditional FL

Definition

The (FL) conditional miscoverage rate is

$$\alpha_{\ell,k}(\mathcal{D}_{mn}) = \mathbb{P}\left(Y \notin \widehat{C}_{\ell,k}(X) \mid \mathcal{D}_{mn}\right) \ ,$$

Theorem

If $\delta \in (0, 0.5]$ and $\ell \cdot k \geq (1-\alpha) \cdot mn$, then the conditional miscoverage rate is controlled as follows:

$$\mathbb{P}\left(\alpha_{\ell,k}(\mathcal{D}_{mn}) \leq \alpha + \sqrt{\frac{\log(1/\delta)}{2mn}}\right) \geq 1 - \delta \ . \tag{10}$$

Remark 1: No lower bound...

Remark 2: $\ell = n$, $k = \lceil (1-\alpha)m \rceil$ works but too large

Remark 3: Condition not necessarily verified by (ℓ^*, k^*)

Why difficult ?

In the standard case, proof of Bian and Barber (2022, Theorem 1) based on:

$$\{Y \in \widehat{C}(X)\} = \{S \leq S_{(\ell)}\} \overset{Rank()}{=} \left\{\sum_{i=1}^{n} 1\{S_i < S\} < \ell\right\}$$

In our case,

$$\begin{aligned}\{Y \in \widehat{C}_{\ell,k}(X)\} &= \left\{S \leq \widehat{Q}_{(\ell,k)}\right\} \\ &\overset{Rank()}{=} \left\{\sum_{j=1}^{m}\sum_{i=1}^{n} 1\{S_i^{(j)} < S\} < \sum_{j=1}^{m}\sum_{i=1}^{n} 1\{S_i^{(j)} \leq \widehat{Q}_{(\ell,k)}\}\right\} \\ &\supseteq \left\{\sum_{j=1}^{m}\sum_{i=1}^{n} 1\{S_i^{(j)} < S\} < \ell \cdot k\right\}\end{aligned}$$

Problem: Taking the bound on the rank is too strong.

Summary of theoretical results

Our theoretical results shows that

- Marginal coverage is possible in one-shot
- Training-conditional coverage is also possible
- Guarantees are closed to the one obtained when data are centralized

And empirically ?

Summary of theoretical results

Our theoretical results shows that

- Marginal coverage is possible in one-shot
- Training-conditional coverage is also possible
- Guarantees are closed to the one obtained when data are centralized

And empirically ?

Summary of theoretical results

Our theoretical results shows that

- Marginal coverage is possible in one-shot
- Training-conditional coverage is also possible
- Guarantees are closed to the one obtained when data are centralized

And empirically ?

Summary of theoretical results

Our theoretical results shows that

- Marginal coverage is possible in one-shot
- Training-conditional coverage is also possible
- Guarantees are closed to the one obtained when data are centralized

And empirically ?

Summary of theoretical results

Our theoretical results shows that

- Marginal coverage is possible in one-shot
- Training-conditional coverage is also possible
- Guarantees are closed to the one obtained when data are centralized

And empirically ?

Summary of theoretical results

Our theoretical results shows that

- Marginal coverage is possible in one-shot
- Training-conditional coverage is also possible
- Guarantees are closed to the one obtained when data are centralized

And empirically ?

One result on a real regression problem

Comparison of FedCP-QQ with

- Central (standard case):
  Split CP when data are **centralized**

- FedCP-Avg (Li et al., 2020):
  Each agent returns a quantile and the server takes the **average** of these quantiles (no theoretical guarantee)

Remark: There was no method with CP guarantees in one-shot FL

One result on a real regression problem

Comparison of FedCP-QQ with

- Central (standard case):
  Split CP when data are **centralized**

- FedCP-Avg (Li et al., 2020):
  Each agent returns a quantile and the server takes the **average** of these quantiles (no theoretical guarantee)

Remark: There was no method with CP guarantees in one-shot FL

One result on a real regression problem

Comparison of FedCP-QQ with

- Central (standard case):
  Split CP when data are **centralized**
- FedCP-Avg (Li et al., 2020):
  Each agent returns a quantile and the server takes the **average** of these quantiles (no theoretical guarantee)

Remark: There was no method with CP guarantees in one-shot FL

One result on a real regression problem

Comparison of FedCP-QQ with

- Central (standard case):
  Split CP when data are **centralized**
- FedCP-Avg (Li et al., 2020):
  Each agent returns a quantile and the server takes the **average** of these quantiles (no theoretical guarantee)

Remark: There was no method with CP guarantees in one-shot FL

One result on a real regression problem

Metrics

On 20 random training-test splits we compute:

- Coverage (on the test set)
- Length of the returned set

▷ $1 - \alpha = 0.9$

▷ $s(x, y) = |\widehat{f}(x) - y|$

One result on a real regression problem

Metrics

On 20 random training-test splits we compute:

- Coverage (on the test set)
- Length of the returned set

▷ $1 - \alpha = 0.9$

▷ $s(x, y) = |\widehat{f}(x) - y|$

One result on a real regression problem

Metrics

On 20 random training-test splits we compute:

- Coverage (on the test set)
- Length of the returned set

▷ $1 - \alpha = 0.9$

▷ $s(x, y) = |\widehat{f}(x) - y|$

One result on a real regression problem

Metrics

On 20 random training-test splits we compute:

- Coverage (on the test set)
- Length of the returned set

▷ $1 - \alpha = 0.9$

▷ $s(x, y) = |\widehat{f}(x) - y|$

One result on a real regression problem

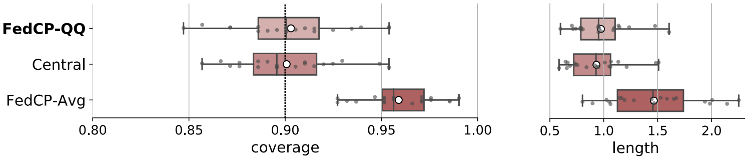


Figure: Coverage (left) and average length (right) of prediction intervals for 20 random training-calibration-test splits. The miscoverage is $\alpha = 0.1$. The white circle represents the mean.

⟶ FedCP-QQ gives prediction sets with coverage and length very similar to those obtained in a centralized setting

One result on a real regression problem

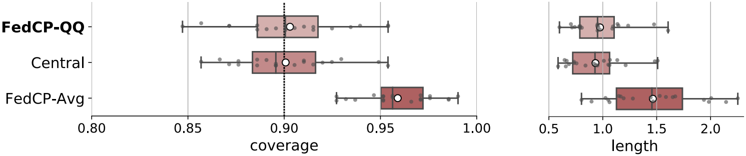


Figure: Coverage (left) and average length (right) of prediction intervals for 20 random training-calibration-test splits. The miscoverage is $\alpha = 0.1$. The white circle represents the mean.

$\longrightarrow$ FedCP-QQ gives prediction sets with coverage and length very similar to those obtained in a centralized setting

In the paper: Real experiments

Evaluation on 5 regression data sets

1. Physicochemical properties of protein tertiary structure (bio)
2. Bike sharing (bike)
3. Communities and crimes (community)
4. Tennessee's student teacher achievement ratio (star)
5. Concrete compressive strength (concrete)

Used methods

1. Split-CP with ridge regression
2. CQR with quantile Regression Forests (RF)
3. CQR with Neural Networks (NN)

Code available at: `https://github.com/yromano/cqr`

In the paper: results on all the data sets

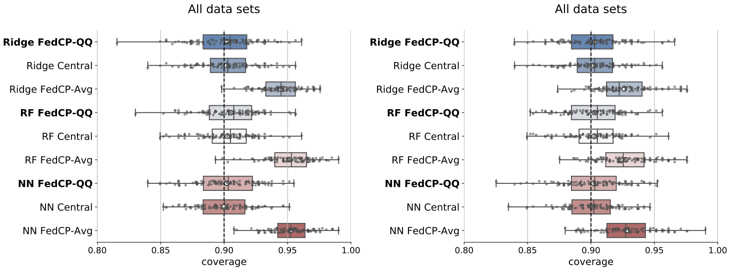


Figure: Empirical coverages of prediction intervals ($\alpha = 0.1$) constructed by various methods. On the left, when $m \gg n$. On the right, when $m \ll n$. Our method is shown in bold font. The white circle represents the mean.

$\longrightarrow$ Same conclusions

Summary

1. We propose an efficient method based on the quantile-of-quantiles to return a valid set in a one-shot federated learning setting
2. An analysis of the method for conditional training coverage ($\approx$ When the dataset is fixed)
3. We show that our method returns prediction sets very similar to those obtained in a centralized setting

$\longrightarrow$ FedCP-QQ is well-suited to perform CP in a one-shot FL setting

Summary

1. We propose an efficient method based on the quantile-of-quantiles to return a valid set in a one-shot federated learning setting
2. An analysis of the method for conditional training coverage (≈ When the dataset is fixed)
3. We show that our method returns prediction sets very similar to those obtained in a centralized setting

⟶ FedCP-QQ is well-suited to perform CP in a one-shot FL setting

Summary

1. We propose an efficient method based on the quantile-of-quantiles to return a valid set in a one-shot federated learning setting

2. An analysis of the method for conditional training coverage ($\approx$ When the dataset is fixed)

3. We show that our method returns prediction sets very similar to those obtained in a centralized setting

$\longrightarrow$ FedCP-QQ is well-suited to perform CP in a one-shot FL setting

Summary

1. We propose an efficient method based on the quantile-of-quantiles to return a valid set in a one-shot federated learning setting
2. An analysis of the method for conditional training coverage (≈ When the dataset is fixed)
3. We show that our method returns prediction sets very similar to those obtained in a centralized setting

$\longrightarrow$ FedCP-QQ is well-suited to perform CP in a one-shot FL setting

Summary

1. We propose an efficient method based on the quantile-of-quantiles to return a valid set in a one-shot federated learning setting
2. An analysis of the method for conditional training coverage ($\approx$ When the dataset is fixed)
3. We show that our method returns prediction sets very similar to those obtained in a centralized setting

$\longrightarrow$ FedCP-QQ is well-suited to perform CP in a one-shot FL setting

Additional results in the paper

In the paper, we also provide

1. A result when data are heterogeneous
   (When agent does not have data from the same P)
2. A private version of the algorithm based on Differential Privacy
3. An extension when the agents have not the same number of data

Additional results in the paper

In the paper, we also provide

1. A result when data are heterogeneous
   (When agent does not have data from the same P)
2. A private version of the algorithm based on Differential Privacy
3. An extension when the agents have not the same number of data

Additional results in the paper

In the paper, we also provide

1. A result when data are heterogeneous
   (When agent does not have data from the same P)
2. A private version of the algorithm based on Differential Privacy
3. An extension when the agents have not the same number of data

Additional results in the paper

In the paper, we also provide

1. A result when data are heterogeneous
   (When agent does not have data from the same P)
2. A private version of the algorithm based on Differential Privacy
3. An extension when the agents have not the same number of data

Future directions

1. Better theoretical guarantees (in particular for training-conditional)

2. Consider the heterogeneous case

3. Robustness to outliers, Byzantine nodes

Future directions

1. Better theoretical guarantees (in particular for training-conditional)

2. Consider the heterogeneous case

3. Robustness to outliers, Byzantine nodes

Future directions

1. Better theoretical guarantees (in particular for training-conditional)

2. Consider the heterogeneous case

3. Robustness to outliers, Byzantine nodes

Future directions

1. Better theoretical guarantees (in particular for training-conditional)

2. Consider the heterogeneous case

3. Robustness to outliers, Byzantine nodes

For more information

One-Shot Federated Conformal Prediction, P. Humbert, B. Le Bars, A. Bellet, and S. Arlot. ICML 2023.

Code is available at: `https://github.com/pierreHmbt/FedCP-QQ`

Thanks!

References

Bian, M. and Barber, R. F. (2022). Training-conditional coverage for distribution-free predictive inference. *arXiv preprint arXiv:2205.03647*.

Lebrun, R. (2013). Efficient time/space algorithm to compute rectangular probabilities of multinomial, multivariate hypergeometric and multivariate pólya distributions. *Statistics and Computing*, 23(5):615–623.

Lei, J., G'Sell, M., Rinaldo, A., Tibshirani, R. J., and Wasserman, L. (2018). Distribution-free predictive inference for regression. *Journal of the American Statistical Association*, 113(523):1094–1111.

Li, T., Sahu, A. K., Zaheer, M., Sanjabi, M., Talwalkar, A., and Smith, V. (2020). Federated optimization in heterogeneous networks. *Proceedings of Machine Learning and Systems*, 2:429–450.

Papadopoulos, H., Proedrou, K., Vovk, V., and Gammerman, A. (2002). Inductive confidence machines for regression. In *European Conference on Machine Learning*, pages 345–356. Springer.

Vovk, V. (2012). Conditional validity of inductive conformal predictors. In *Asian conference on machine learning*, pages 475–490. PMLR.

Vovk, V., Gammerman, A., and Shafer, G. (2005). *Algorithmic learning in a random world.* Springer Science & Business Media.