Adaptive Conformal Predictions for Time Series

An application to forecasting French electricity Spot prices

Margaux Zaffran[1,2,3] Aymeric Dieuleveut[3] Olivier Féron[1,4]
Yannig Goude[1] Julie Josse[2,5]

03/03/2022

International Seminar on Distribution-Free Statistics

[1]EDF R&D [2]INRIA [3]CMAP, Ecole Polytechnique [4]FiME [5]IDESP

Forecasting French electricity Spot prices

Electricity Spot prices

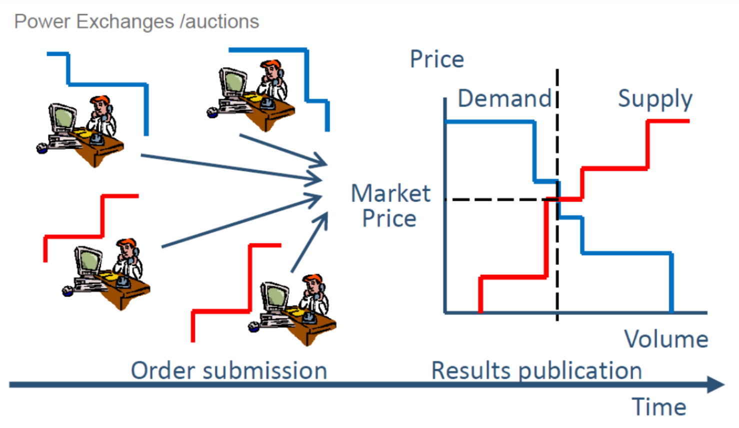


Figure 1: Drawing of spot auctions mechanism

French Electricity Spot prices data set: visualisation

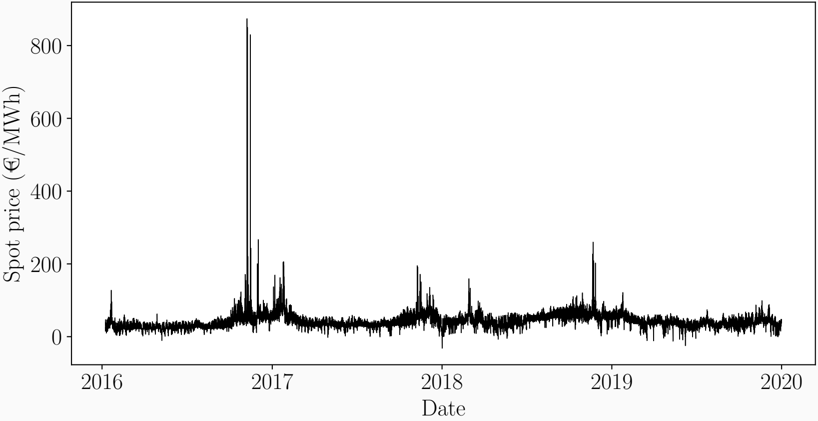


Figure 2: Representation of the French electricity spot price, from 2016 to 2019.

French Electricity Spot prices data set: extract

Date and time	Price	Price D-1	Price D-7	For. cons.	DOW
11/01/16 0PM	21.95	15.58	13.78	58800	Monday
11/01/16 1PM	20.04	19.05	13.44	57600	Monday
⋮	⋮	⋮	⋮	⋮	⋮
12/01/16 0PM	21.51	21.95	25.03	61600	Tuesday
12/01/16 1PM	19.81	20.04	24.42	59800	Tuesday
⋮	⋮	⋮	⋮	⋮	⋮
18/01/16 0PM	38.14	37.86	21.95	70400	Monday
18/01/16 1PM	35.66	34.60	20.04	69500	Monday
⋮	⋮	⋮	⋮	⋮	⋮

Table 1: Extract of the built data set, for French electricity spot price forecasting.

French Electricity Spot prices data set: extract

Date and time	Price	Price D-1	Price D-7	For. cons.	DOW
11/01/16 0PM	21.95	15.58	13.78	58800	Monday
11/01/16 1PM	20.04	19.05	13.44	57600	Monday
⋮	⋮	⋮	⋮	⋮	⋮
12/01/16 0PM	21.51	21.95	25.03	61600	Tuesday
12/01/16 1PM	19.81	20.04	24.42	59800	Tuesday
⋮	⋮	⋮	⋮	⋮	⋮
18/01/16 0PM	38.14	37.86	21.95	70400	Monday
18/01/16 1PM	35.66	34.60	20.04	69500	Monday
⋮	⋮	⋮	⋮	⋮	⋮

Table 1: Extract of the built data set, for French electricity spot price forecasting.

- $y_t \in \mathbb{R}$

French Electricity Spot prices data set: extract

Date and time	Price	Price D-1	Price D-7	For. cons.	DOW
11/01/16 0PM	21.95	15.58	13.78	58800	Monday
11/01/16 1PM	20.04	19.05	13.44	57600	Monday
⋮	⋮	⋮	⋮	⋮	⋮
12/01/16 0PM	21.51	21.95	25.03	61600	Tuesday
12/01/16 1PM	19.81	20.04	24.42	59800	Tuesday
⋮	⋮	⋮	⋮	⋮	⋮
18/01/16 0PM	38.14	37.86	21.95	70400	Monday
18/01/16 1PM	35.66	34.60	20.04	69500	Monday
⋮	⋮	⋮	⋮	⋮	⋮

Table 1: Extract of the built data set, for French electricity spot price forecasting.

- $y_t \in \mathbb{R}$
- $x_t \in \mathbb{R}^d$

Forecasting French electricity Spot prices

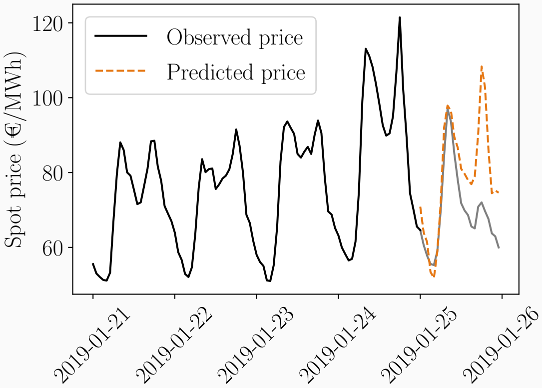


Figure 3: French electricity spot price and its prediction with random forest.

$\hookrightarrow$ $(x_t, y_t) \in \mathbb{R}^d \times \mathbb{R}$ ($d = 56$, details later)

$\hookrightarrow$ 3 years for training

$\hookrightarrow$ 1 year to forecast

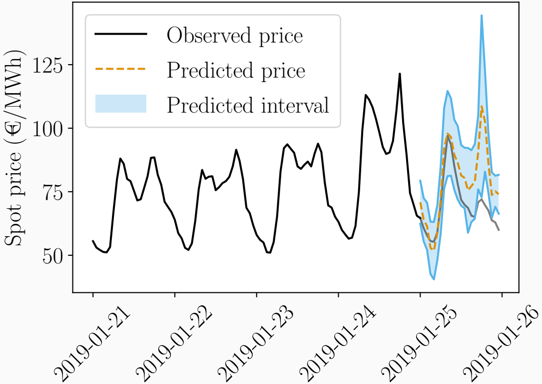


Figure 4: French electricity spot price, its prediction and its uncertainty with AgACI (proposed algorithm).

- Target coverage: 90%
- Empirical coverage: 91.68%

Conformal prediction and time series, what's the issue?

Framework and notations

- Data: T_0 observations $(x_1, y_1), \ldots, (x_{T_0}, y_{T_0})$ in $\mathbb{R}^d \times \mathbb{R}$
- Aim: predict the response values as well as predictive intervals for T_1 subsequent observations $x_{T_0+1}, \ldots, x_{T_0+T_1}$

$\hookrightarrow$ Build the smallest interval $\mathcal{C}_\alpha^t$ such that:

$$\mathbb{P}\left\{Y_t \in \mathcal{C}_\alpha^t(X_t)\right\} \geq 1 - \alpha, \text{ for } t \in [\![T_0 + 1, T_0 + T_1]\!].$$

Split Conformal Prediction

Split conformal prediction is simple to compute and works:

- any regression algorithm (neural nets, random forest...);
- distribution-free as long as the data is exchangeable;

- finite sample.

Split Conformal Prediction

Split conformal prediction is simple to compute and works:

- any regression algorithm (neural nets, random forest...);
- distribution-free as long as the data is exchangeable;
  - $\hookrightarrow$ the scores need to be exchangeable (but then it would not work with any regression algorithm)
- finite sample.

Time series are not exchangeable

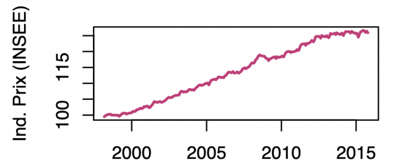


Figure 5: Trend[1]

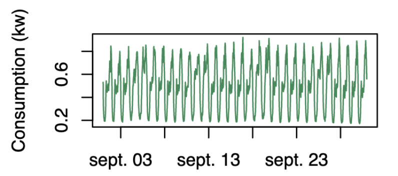


Figure 6: Seasonality[2]

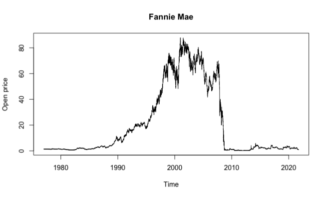


Figure 7: Shift

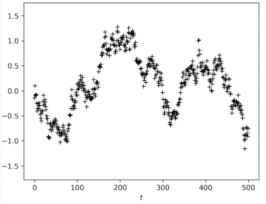


Figure 8: Time dependence

[1]Images from Yannig Goude class material.

Non-exchangeable even if the noise is exchangeable

Assume the following model:

$$Y_t = f_t(X_t) + \varepsilon_t, \text{ for } t \in \mathbb{N}^*,$$

for some function f_t, and some noise ε_t.

Even if the noise ε_t is exchangeable, we can produce dependent residuals.

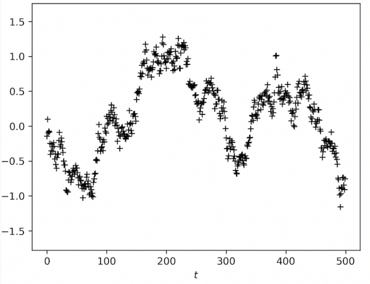


Figure 9: Auto-Regressive residuals

Available methods for non-exchangeable data, in the context of time series

How to adapt to time series?

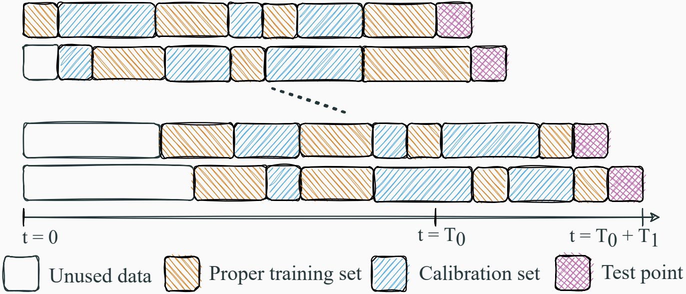
Usual ideas from the time series literature:

- Consider an online procedure (for each new data, re-train and re-calibrate)

How to adapt to time series?

Usual ideas from the time series literature:

- Consider an online procedure (for each new data, re-train and re-calibrate)



How to adapt to time series?

Usual ideas from the time series literature:

- Consider an online procedure (for each new data, re-train and re-calibrate)
  - $\hookrightarrow$ update to recent observations (trend impact, period of the seasonality, dependence...)

How to adapt to time series?

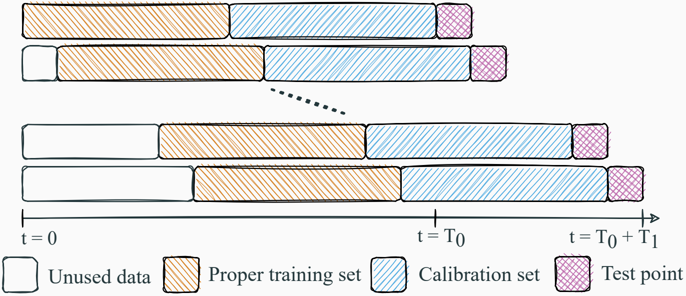
Usual ideas from the time series literature:

- Consider an online procedure (for each new data, re-train and re-calibrate)
  - $\hookrightarrow$ update to recent observations (trend impact, period of the seasonality, dependence...)
- Use a sequential split

How to adapt to time series?

Usual ideas from the time series literature:

- Consider an online procedure (for each new data, re-train and re-calibrate)
  - ↪ update to recent observations (trend impact, period of the seasonality, dependence...)
- Use a sequential split



How to adapt to time series?

Usual ideas from the time series literature:

- Consider an online procedure (for each new data, re-train and re-calibrate)
  - $\hookrightarrow$ update to recent observations (trend impact, period of the seasonality, dependence...)
- Use a sequential split
  - $\hookrightarrow$ use only the past so as to correctly estimate the variance of the residuals (using the future leads to optimistic residuals and underestimation of their variance)

Online sequential split conformal prediction (OSSCP)

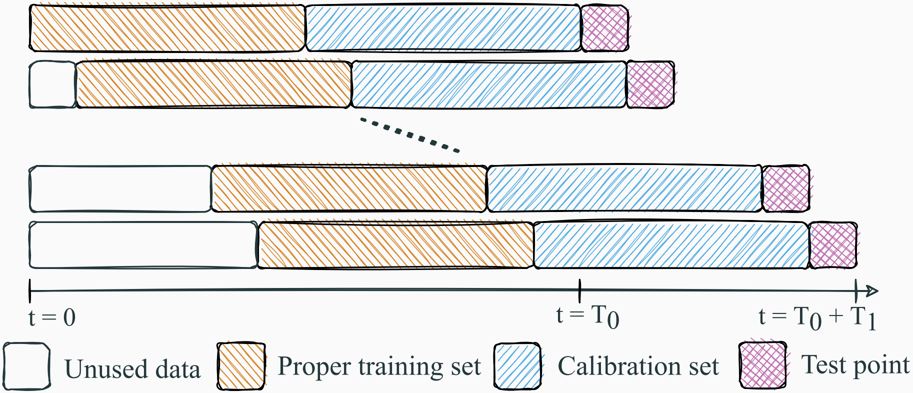


Figure 10: Diagram describing the online sequential split conformal prediction.

Online sequential split conformal prediction (OSSCP)

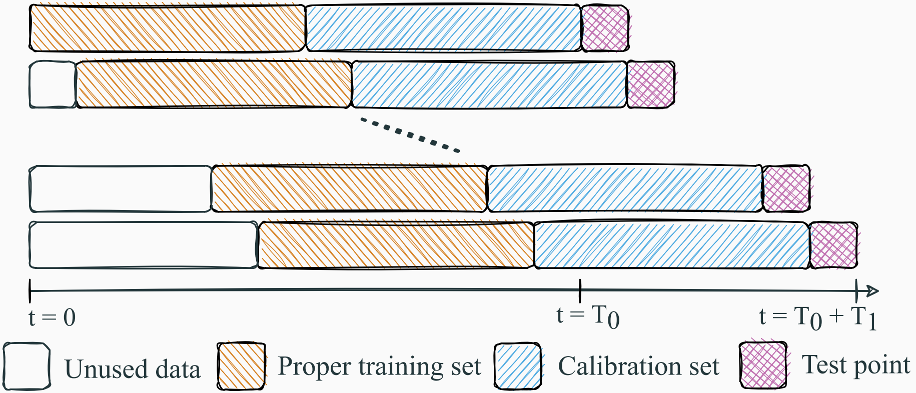


Figure 10: Diagram describing the online sequential split conformal prediction.

Wisniewski et al. (2020); Kath and Ziel (2021); and our study

Online sequential split conformal prediction (OSSCP)

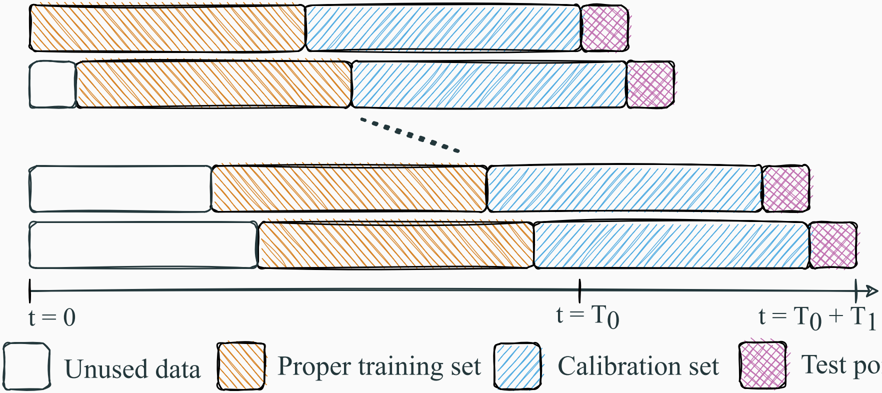


Figure 10: Diagram describing the online sequential split conformal prediction.

Wisniewski et al. (2020); Kath and Ziel (2021); and our study

$\hookrightarrow$ tested on real time series

EnbPI, Xu and Xie (2021)

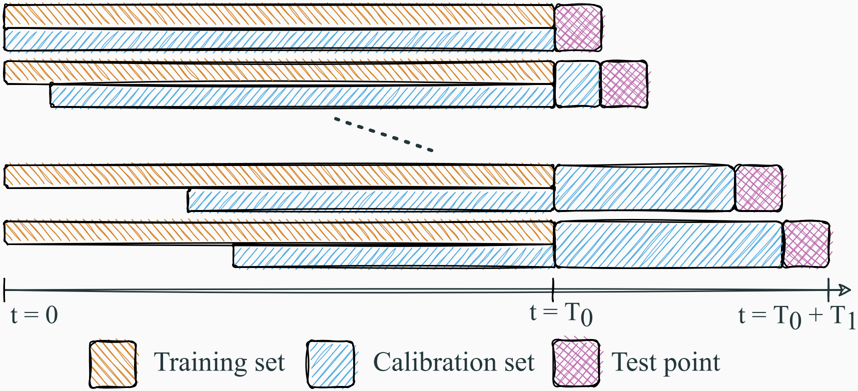


Figure 11: Diagram describing the EnbPI algorithm.

EnbPI, Xu and Xie (2021)

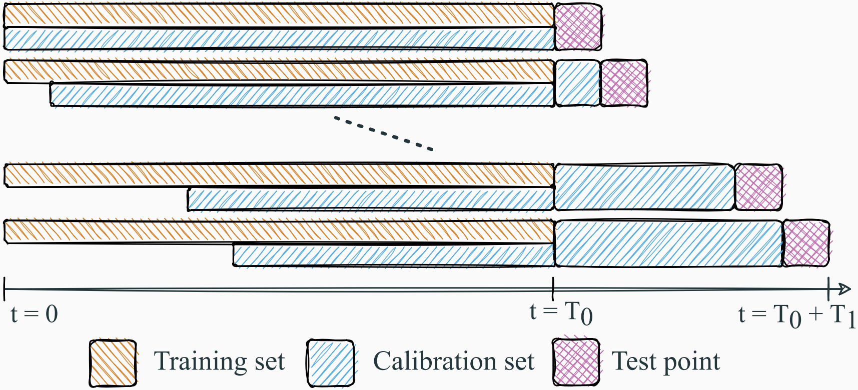


Figure 11: Diagram describing the EnbPI algorithm.

↪ tested on other real time series

↪ compared to offline methods

Adaptive Conformal Inference (ACI), Gibbs and Candès (2021)

Refitting the model may be insufficient $\Rightarrow$ adapt the quantile level used on the calibration's scores.

Adaptive Conformal Inference (ACI), Gibbs and Candès (2021)

Refitting the model may be insufficient $\Rightarrow$ adapt the quantile level used on the calibration's scores. **(Distribution shift)**

Adaptive Conformal Inference (ACI), Gibbs and Candès (2021)

Refitting the model may be insufficient $\Rightarrow$ adapt the quantile level used on the calibration's scores. **(Distribution shift)**

The proposed update scheme is the following:

$$\alpha_{t+1} := \alpha_t + \gamma \left(\alpha - \mathrm{err}_t \right) \tag{1}$$

with:

$$\mathrm{err}_t := \begin{cases} 1 \text{ if } y_t \notin \hat{\mathcal{C}}_{\alpha_t}\left(x_t\right), \\ 0 \text{ otherwise }, \end{cases}$$

and $\alpha_1 = \alpha$, $\gamma \geq 0$.

Adaptive Conformal Inference (ACI), Gibbs and Candès (2021)

Refitting the model may be insufficient $\Rightarrow$ adapt the quantile level used on the calibration's scores. **(Distribution shift)**

The proposed update scheme is the following:

$$\alpha_{t+1} := \alpha_t + \gamma \left(\alpha - \mathrm{err}_t \right) \tag{1}$$

with:

$$\mathrm{err}_t := \begin{cases} 1 \text{ if } y_t \notin \hat{\mathcal{C}}_{\alpha_t}\left(x_t\right), \\ 0 \text{ otherwise }, \end{cases}$$

and $\alpha_1 = \alpha$, $\gamma \geq 0$.

Intuition: if we did make an error, the interval was too small so we want to increase its length by taking a higher quantile (a smaller α_t). Reversely if we included the point.

Adaptive Conformal Inference (ACI), Gibbs and Candès (2021)

Refitting the model may be insufficient $\Rightarrow$ adapt the quantile level used on the calibration's scores. **(Distribution shift)**

The proposed update scheme is the following:

$$\alpha_{t+1} := \alpha_t + \gamma \left(\alpha - \mathrm{err}_t \right) \tag{1}$$

with:

$$\mathrm{err}_t := \begin{cases} 1 \text{ if } y_t \notin \hat{\mathcal{C}}_{\alpha_t}\left(x_t\right), \\ 0 \text{ otherwise }, \end{cases}$$

and $\alpha_1 = \alpha$, $\gamma \geq 0$.

Intuition: if we did make an error, the interval was too small so we want to increase its length by taking a higher quantile (a smaller α_t). Reversely if we included the point.

Gibbs and Candès (2021) provide asymptotic validity result for any distribution.

Visualisation of the procedure

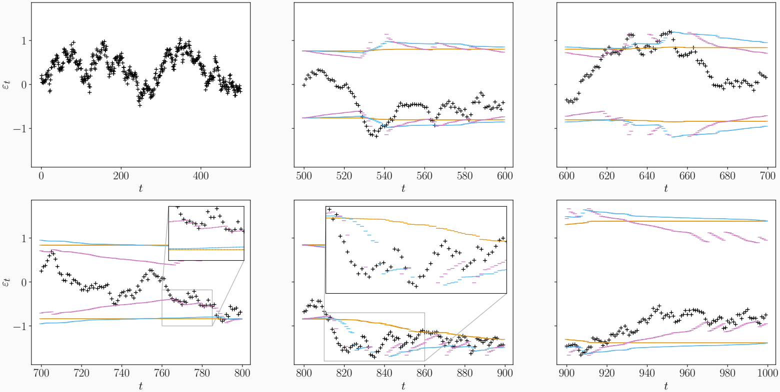


Figure 12: Visualisation of ACI with different values of γ ($\gamma = 0$, $\gamma = 0.01$, $\gamma = 0.05$)

Theoretical analysis of ACI's length

Approach

Aim: derive theoretical results on the **average length** of ACI depending on γ

$\hookrightarrow$ Guideline for choosing γ

Approach

Aim: derive theoretical results on the **average length** of ACI depending on γ

$\hookrightarrow$ Guideline for choosing γ

Approach: consider extreme cases (useful in an adversarial context) with simple theoretical distributions (additional assumptions)

1. exchangeable
2. Auto-Regressive case (AR(1))

Theoretical analysis of ACI's length: exchangeable case

Define $L(\alpha_t) = 2Q(1-\alpha_t)$ the length of the interval predicted by the adaptive algorithm at time t, and $L_0 = 2Q(1-\alpha)$ the length of the interval predicted by the non-adaptive algorithm ($\gamma = 0$).

Theorem

Assume the scores are exchangeable with quantile function Q perfectly estimated at each time, and other assumptions.

Then, for all $\gamma > 0$, $(\alpha_t)_{t>0}$ forms a Markov Chain, that admits a stationary distribution π_γ, and

$$\frac{1}{T}\sum_{t=1}^{T} L(\alpha_t) \xrightarrow[T\to+\infty]{a.s.} \mathbb{E}_{\pi_\gamma}[L] \overset{not.}{=} \mathbb{E}_{\tilde{\alpha}\sim\pi_\gamma}[L(\tilde{\alpha})].$$

Moreover, as $\gamma \to 0$,

$$\mathbb{E}_{\pi_\gamma}[L] = L_0 + Q''(1-\alpha)\frac{\gamma}{2}\alpha(1-\alpha) + O(\gamma^{3/2}).$$

Theoretical analysis of ACI's length: exchangeable case

Define $L(\alpha_t) = 2Q(1-\alpha_t)$ the length of the interval predicted by the adaptive algorithm at time t, and $L_0 = 2Q(1-\alpha)$ the length of the interval predicted by the non-adaptive algorithm ($\gamma = 0$).

Theorem

Assume the scores are exchangeable with quantile function Q perfectly estimated at each time, and other assumptions.

Then, for all $\gamma > 0$, $(\alpha_t)_{t>0}$ forms a Markov Chain, that admits a stationary distribution π_γ, and

$$\frac{1}{T}\sum_{t=1}^{T} L(\alpha_t) \xrightarrow[T\to+\infty]{a.s.} \mathbb{E}_{\pi_\gamma}[L] \overset{not.}{=} \mathbb{E}_{\tilde{\alpha}\sim\pi_\gamma}[L(\tilde{\alpha})].$$

Moreover, as $\gamma \to 0$,

$$\mathbb{E}_{\pi_\gamma}[L] = L_0 + Q''(1-\alpha)\frac{\gamma}{2}\alpha(1-\alpha) + O(\gamma^{3/2}).$$

Theoretical analysis of ACI's length: exchangeable case

Define $L(\alpha_t) = 2Q(1-\alpha_t)$ the length of the interval predicted by the adaptive algorithm at time t, and $L_0 = 2Q(1-\alpha)$ the length of the interval predicted by the non-adaptive algorithm ($\gamma = 0$).

Theorem

Assume the scores are exchangeable with quantile function Q perfectly estimated at each time, and other assumptions.

Then, for all $\gamma > 0$, $(\alpha_t)_{t>0}$ forms a Markov Chain, that admits a stationary distribution π_γ, and

$$\frac{1}{T}\sum_{t=1}^{T} L(\alpha_t) \xrightarrow[T\to+\infty]{a.s.} \mathbb{E}_{\pi_\gamma}[L] \overset{not.}{=} \mathbb{E}_{\tilde{\alpha}\sim\pi_\gamma}[L(\tilde{\alpha})].$$

Moreover, as $\gamma \to 0$,

$$\mathbb{E}_{\pi_\gamma}[L] = L_0 + Q''(1-\alpha)\frac{\gamma}{2}\alpha(1-\alpha) + O(\gamma^{3/2}).$$

Theoretical analysis of ACI's length: exchangeable case

Define $L(\alpha_t) = 2Q(1-\alpha_t)$ the length of the interval predicted by the adaptive algorithm at time t, and $L_0 = 2Q(1-\alpha)$ the length of the interval predicted by the non-adaptive algorithm ($\gamma = 0$).

Theorem

Assume the scores are exchangeable with quantile function Q perfectly estimated at each time, and other assumptions.

Then, for all $\gamma > 0$, $(\alpha_t)_{t>0}$ forms a Markov Chain, that admits a stationary distribution π_γ, and

$$\frac{1}{T}\sum_{t=1}^{T} L(\alpha_t) \xrightarrow[T\to+\infty]{a.s.} \mathbb{E}_{\pi_\gamma}[L] \overset{not.}{=} \mathbb{E}_{\tilde{\alpha}\sim\pi_\gamma}[L(\tilde{\alpha})].$$

Moreover, as $\gamma \to 0$,

$$\mathbb{E}_{\pi_\gamma}[L] = L_0 + Q''(1-\alpha)\frac{\gamma}{2}\alpha(1-\alpha) + O(\gamma^{3/2}).$$

Theoretical analysis of ACI's length: exchangeable case

Define $L(\alpha_t) = 2Q(1-\alpha_t)$ the length of the interval predicted by the adaptive algorithm at time t, and $L_0 = 2Q(1-\alpha)$ the length of the interval predicted by the non-adaptive algorithm ($\gamma = 0$).

Theorem

Assume the scores are exchangeable with quantile function Q perfectly estimated at each time, and other assumptions.

Then, for all $\gamma > 0$, $(\alpha_t)_{t>0}$ forms a Markov Chain, that admits a stationary distribution π_γ, and

$$\frac{1}{T}\sum_{t=1}^{T} L(\alpha_t) \xrightarrow[T\to+\infty]{a.s.} \mathbb{E}_{\pi_\gamma}[L] \overset{not.}{=} \mathbb{E}_{\tilde{\alpha}\sim\pi_\gamma}[L(\tilde{\alpha})].$$

Moreover, as $\gamma \to 0$,

$$\mathbb{E}_{\pi_\gamma}[L] = L_0 + Q''(1-\alpha)\frac{\gamma}{2}\alpha(1-\alpha) + O(\gamma^{3/2}).$$

Numerical analysis of ACI's length: AR(1) case

Theorem

Assume the residuals follow an AR(1) process: $\hat{\varepsilon}_{t+1} = \varphi \hat{\varepsilon}_t + \xi_{t+1}$ *with* $(\xi_t)_t$ *i.i.d. random variables and other assumptions, we have:*

$$\frac{1}{T} \sum_{t=1}^{T} L(\alpha_t) \underset{T \to +\infty}{\overset{a.s.}{\longrightarrow}} \mathbb{E}_{\pi_{\gamma,\varphi}}[L].$$

Numerical analysis of ACI's length: AR(1) case

Theorem

Assume the residuals follow an AR(1) process: $\hat{\varepsilon}_{t+1} = \varphi\hat{\varepsilon}_t + \xi_{t+1}$ with $(\xi_t)_t$ i.i.d. random variables and other assumptions, we have:

$$\frac{1}{T}\sum_{t=1}^{T} L(\alpha_t) \xrightarrow[T\to+\infty]{a.s.} \mathbb{E}_{\pi_{\gamma,\varphi}}[L].$$

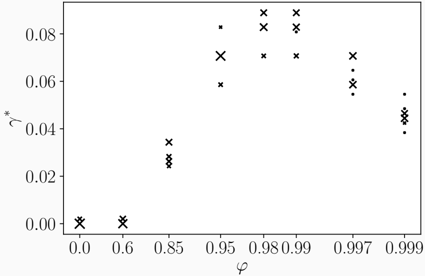


Figure 13: γ^* minimizing the average length for each φ.

AgACI

AgACI: adaptive wrapper around ACI

Online aggregation under expert advice (Cesa-Bianchi and Lugosi, 2006) computes an optimal weighted mean of experts. The weights assigned to each expert depend on all experts losses/performances at previous time steps.

AgACI: adaptive wrapper around ACI

Online aggregation under expert advice (Cesa-Bianchi and Lugosi, 2006) computes an optimal weighted mean of experts. The weights assigned to each expert depend on all experts losses/performances at previous time steps.

`AgACI` performs 2 independent aggregations: one for each bound (the upper and lower ones).

AgACI: adaptive wrapper around ACI

Online aggregation under expert advice (Cesa-Bianchi and Lugosi, 2006) computes an optimal weighted mean of experts. The weights assigned to each expert depend on all experts losses/performances at previous time steps.

`AgACI` performs 2 independent aggregations: one for each bound (the upper and lower ones).

The experts are K versions of ACI with a different γ in $\{(\gamma_k)_{k \leq K}\}$.

AgACI: adaptive wrapper around ACI

Online aggregation under expert advice (Cesa-Bianchi and Lugosi, 2006) computes an optimal weighted mean of experts. The weights assigned to each expert depend on all experts losses/performances at previous time steps.

`AgACI` performs 2 independent aggregations: one for each bound (the upper and lower ones).

The experts are K versions of ACI with a different γ in $\{(\gamma_k)_{k \leq K}\}$.

$$\tilde{b}_{t+1}^{(u)}(x) = \frac{\sum_{k=1}^{K} \omega_{t,k}^{(u)} \hat{b}_{t,k}^{(u)}(x)}{\sum_{l=1}^{K} \omega_{t,l}^{(u)}}$$

AgACI: adaptive wrapper around ACI

Online aggregation under expert advice (Cesa-Bianchi and Lugosi, 2006) computes an optimal weighted mean of experts. The weights assigned to each expert depend on all experts losses/performances at previous time steps.

`AgACI` performs 2 independent aggregations: one for each bound (the upper and lower ones).

The experts are K versions of ACI with a different γ in $\{(\gamma_k)_{k \leq K}\}$.

$$\tilde{b}_{t+1}^{(u)}(x) = \frac{\sum_{k=1}^{K} \omega_{t,k}^{(u)} \hat{b}_{t,k}^{(u)}(x)}{\sum_{l=1}^{K} \omega_{t,l}^{(u)}}$$

The pinball loss (usual in quantile regression) is used to assess the performances of each expert.

AgACI: adaptive wrapper around ACI

Online aggregation under expert advice (Cesa-Bianchi and Lugosi, 2006) computes an optimal weighted mean of experts. The weights assigned to each expert depend on all experts losses/performances at previous time steps.

`AgACI` performs 2 independent aggregations: one for each bound (the upper and lower ones).

The experts are K versions of ACI with a different γ in $\{(\gamma_k)_{k \leq K}\}$.

$$\tilde{b}_{t+1}^{(u)}(x) = \frac{\sum_{k=1}^{K} \omega_{t,k}^{(u)} \hat{b}_{t,k}^{(u)}(x)}{\sum_{l=1}^{K} \omega_{t,l}^{(u)}}$$

The pinball loss (usual in quantile regression) is used to assess the performances of each expert. The pinball loss of level $1 - \alpha/2$ for the upper bound

AgACI: adaptive wrapper around ACI

Online aggregation under expert advice (Cesa-Bianchi and Lugosi, 2006) computes an optimal weighted mean of experts. The weights assigned to each expert depend on all experts losses/performances at previous time steps.

`AgACI` performs 2 independent aggregations: one for each bound (the upper and lower ones).

The experts are K versions of ACI with a different γ in $\{(\gamma_k)_{k \leq K}\}$.

$$\tilde{b}^{(u)}_{t+1}(x) = \frac{\sum_{k=1}^{K} \omega^{(u)}_{t,k} \hat{b}^{(u)}_{t,k}(x)}{\sum_{l=1}^{K} \omega^{(u)}_{t,l}}$$

The pinball loss (usual in quantile regression) is used to assess the performances of each expert. The pinball loss of level $1 - \alpha/2$ for the upper bound, and of level $\alpha/2$ for the lower bound.

Comparison on simulated data

Data generation and simulation settings

$$Y_t = 10\sin\left(\pi X_{t,1} X_{t,2}\right) + 20\left(X_{t,3} - 0.5\right)^2 + 10X_{t,4} + 5X_{t,5} + \varepsilon_t$$

Data generation and simulation settings

$$Y_t = 10 \sin(\pi X_{t,1} X_{t,2}) + 20 (X_{t,3} - 0.5)^2 + 10 X_{t,4} + 5 X_{t,5} + \varepsilon_t$$

where the $X_{t,\cdot} \sim \mathcal{U}([0,1])$

Data generation and simulation settings

$$Y_t = 10\sin\left(\pi X_{t,1} X_{t,2}\right) + 20\left(X_{t,3} - 0.5\right)^2 + 10X_{t,4} + 5X_{t,5} + \varepsilon_t$$

where the $X_{t,\cdot} \sim \mathcal{U}([0,1])$ and ε_t is an ARMA(1,1) process:

Data generation and simulation settings

$$Y_t = 10\sin(\pi X_{t,1}X_{t,2}) + 20(X_{t,3} - 0.5)^2 + 10X_{t,4} + 5X_{t,5} + \varepsilon_t$$

where the $X_{t,\cdot} \sim \mathcal{U}([0,1])$ and ε_t is an ARMA(1,1) process:

$$\varepsilon_{t+1} = \varphi\varepsilon_t + \xi_{t+1} + \theta\xi_t,$$

with ξ_t is a white noise of variance σ^2.

Data generation and simulation settings

$$Y_t = 10\sin(\pi X_{t,1} X_{t,2}) + 20(X_{t,3} - 0.5)^2 + 10X_{t,4} + 5X_{t,5} + \varepsilon_t$$

where the $X_{t,\cdot} \sim \mathcal{U}([0,1])$ and ε_t is an ARMA(1,1) process:

$$\varepsilon_{t+1} = \varphi\varepsilon_t + \xi_{t+1} + \theta\xi_t,$$

with ξ_t is a white noise of variance σ^2.

- $\varphi = \theta$ range in $[0.1, 0.8, 0.9, 0.95, 0.99]$.

Data generation and simulation settings

$$Y_t = 10\sin\left(\pi X_{t,1}X_{t,2}\right) + 20\left(X_{t,3} - 0.5\right)^2 + 10X_{t,4} + 5X_{t,5} + \varepsilon_t$$

where the $X_{t,\cdot} \sim \mathcal{U}([0,1])$ and ε_t is an ARMA(1,1) process:

$$\varepsilon_{t+1} = \varphi\varepsilon_t + \xi_{t+1} + \theta\xi_t,$$

with ξ_t is a white noise of variance σ^2.

- $\varphi = \theta$ range in $[0.1, 0.8, 0.9, 0.95, 0.99]$.
- We fix σ to keep the variance Var(ε_t) constant to 10 (or 1).

Data generation and simulation settings

$$Y_t = 10\sin(\pi X_{t,1} X_{t,2}) + 20(X_{t,3} - 0.5)^2 + 10X_{t,4} + 5X_{t,5} + \varepsilon_t$$

where the $X_{t,\cdot} \sim \mathcal{U}([0,1])$ and ε_t is an ARMA(1,1) process:

$$\varepsilon_{t+1} = \varphi\varepsilon_t + \xi_{t+1} + \theta\xi_t,$$

with ξ_t is a white noise of variance σ^2.

- $\varphi = \theta$ range in $[0.1, 0.8, 0.9, 0.95, 0.99]$.
- We fix σ to keep the variance $\text{Var}(\varepsilon_t)$ constant to 10 (or 1).
- We use random forest as regressor.

Data generation and simulation settings

$$Y_t = 10\sin(\pi X_{t,1}X_{t,2}) + 20(X_{t,3}-0.5)^2 + 10X_{t,4} + 5X_{t,5} + \varepsilon_t$$

where the $X_{t,\cdot} \sim \mathcal{U}([0,1])$ and ε_t is an ARMA(1,1) process:

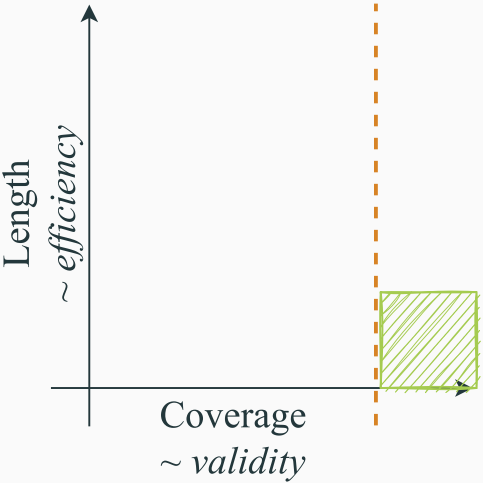
$$\varepsilon_{t+1} = \varphi\varepsilon_t + \xi_{t+1} + \theta\xi_t,$$

with ξ_t is a white noise of variance σ^2.

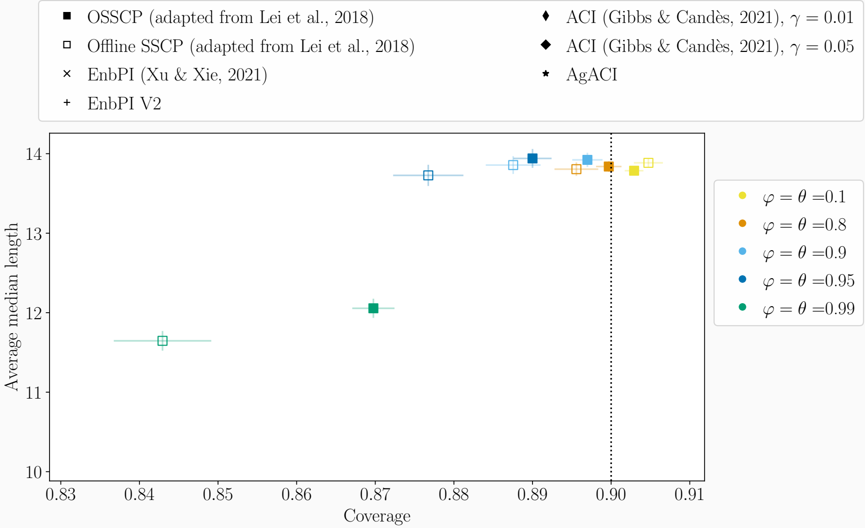
- $\varphi = \theta$ range in $[0.1, 0.8, 0.9, 0.95, 0.99]$.
- We fix σ to keep the variance Var(ε_t) constant to 10 (or 1).
- We use random forest as regressor.
- For each setting (pair variance and φ,θ):
  - 300 points, the last 100 kept for prediction and evaluation,
  - 500 repetitions,

  $\Rightarrow$ in total, $100 \times 500 = 50000$ predictions are evaluated.

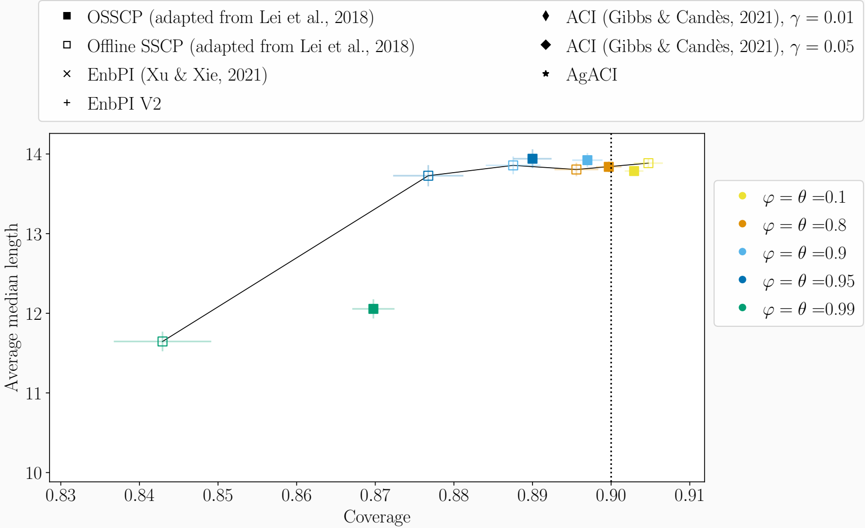
Visualisation of the results



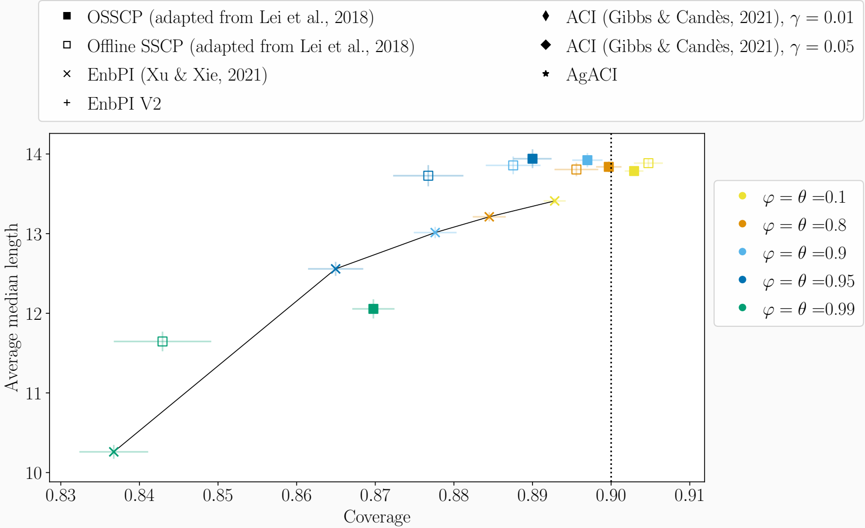
Results: impact of the temporal dependence, ARMA(1,1), variance 10



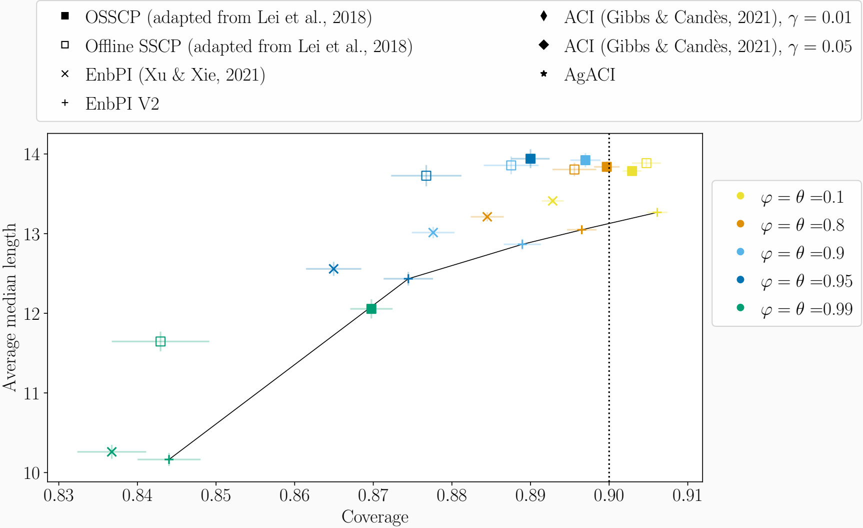
Results: impact of the temporal dependence, ARMA(1,1), variance 10



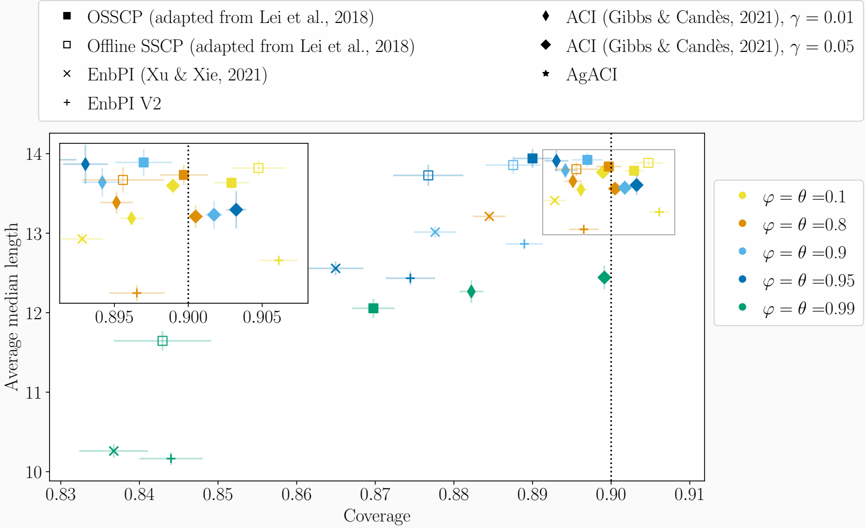
Results: impact of the temporal dependence, ARMA(1,1), variance 10



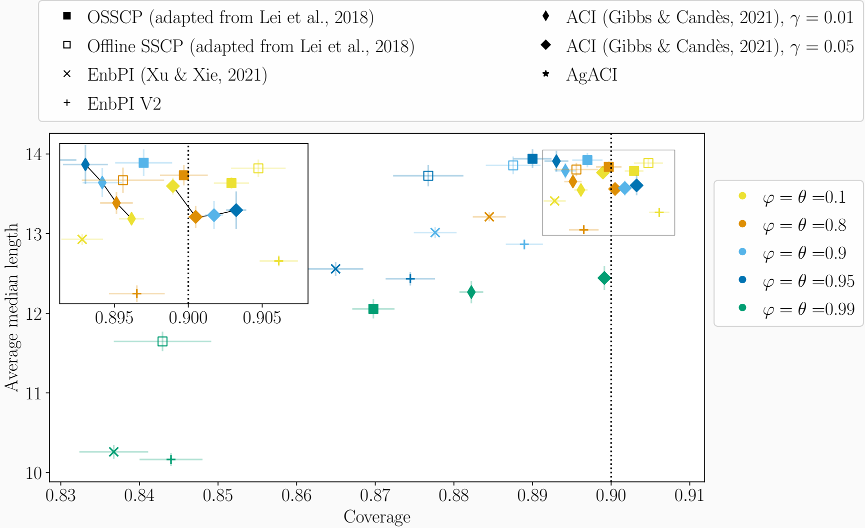
Results: impact of the temporal dependence, ARMA(1,1), variance 10



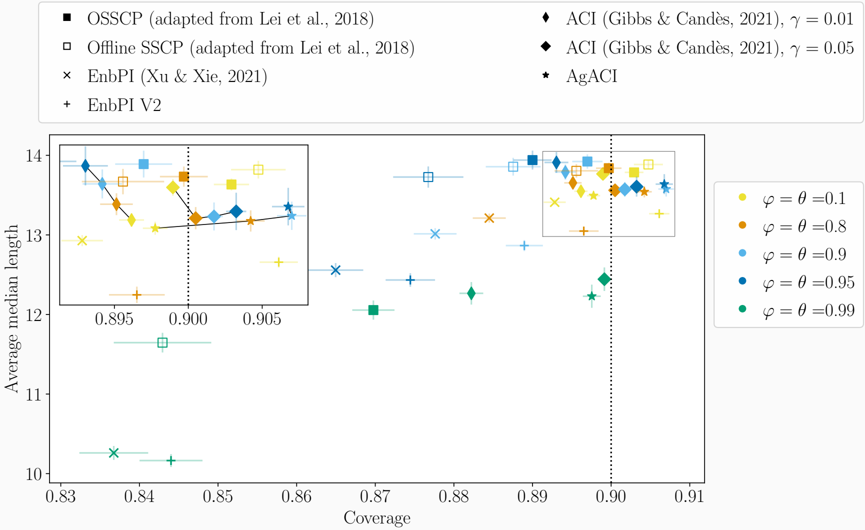
Results: impact of the temporal dependence, ARMA(1,1), variance 10



Results: impact of the temporal dependence, ARMA(1,1), variance 10



Results: impact of the temporal dependence, ARMA(1,1), variance 10



Summary

1. The temporal dependence impacts the *validity*.

Summary

1. The temporal dependence impacts the *validity*.
2. Online is significantly better than offline.

Summary

1. The temporal dependence impacts the *validity*.
2. Online is significantly better than offline.
3. **OSSCP.** Achieves *valid* coverage for φ and θ smaller than 0.9, but is not robust to the increasing dependence.

Summary

1. The temporal dependence impacts the *validity*.
2. Online is significantly better than offline.
3. **OSSCP.** Achieves *valid* coverage for φ and θ smaller than 0.9, but is not robust to the increasing dependence.
4. **EnbPI.** Its *validity* strongly depends on the data distribution. When the method is *valid*, it produces the smallest intervals. EnbPI V2 method should be preferred.

Summary

1. The temporal dependence impacts the *validity*.
2. Online is significantly better than offline.
3. **OSSCP.** Achieves *valid* coverage for φ and θ smaller than 0.9, but is not robust to the increasing dependence.
4. **EnbPI.** Its *validity* strongly depends on the data distribution. When the method is *valid*, it produces the smallest intervals. EnbPI V2 method should be preferred.
5. **ACI.** Achieves *valid* coverage for every simulation settings with a well chosen γ, or for dependence such that $\varphi < 0.95$. It is robust to the strength of the dependence.

Summary

1. The temporal dependence impacts the *validity*.
2. Online is significantly better than offline.
3. **OSSCP.** Achieves *valid* coverage for φ and θ smaller than 0.9, but is not robust to the increasing dependence.
4. **EnbPI.** Its *validity* strongly depends on the data distribution. When the method is *valid*, it produces the smallest intervals. EnbPI V2 method should be preferred.
5. **ACI.** Achieves *valid* coverage for every simulation settings with a well chosen γ, or for dependence such that $\varphi < 0.95$. It is robust to the strength of the dependence.
6. **AgACI.** Achieves *valid* coverage for every simulation settings, with good *efficiency*.

Price prediction with confidence in 2019

Settings

- Forecast for the year 2019.
- Random forest regressor.
- One model per hour, we concatenate the predictions afterwards.

Settings

- Forecast for the year 2019.
- Random forest regressor.
- One model per hour, we concatenate the predictions afterwards.

$\hookrightarrow$ 24 models

Settings

- Forecast for the year 2019.
- Random forest regressor.
- One model per hour, we concatenate the predictions afterwards.

$\hookrightarrow$ 24 models

  - $y_t \in \mathbb{R}$

Settings

- Forecast for the year 2019.
- Random forest regressor.
- One model per hour, we concatenate the predictions afterwards.

$\hookrightarrow$ 24 models

  - $y_t \in \mathbb{R}$
  - $x_t \in \mathbb{R}^d$, with $d = 24 + 24 + 1 + 7 = 56$

24 prices of the day before

Settings

- Forecast for the year 2019.
- Random forest regressor.
- One model per hour, we concatenate the predictions afterwards.

$\hookrightarrow$ 24 models

  - $y_t \in \mathbb{R}$
  - $x_t \in \mathbb{R}^d$, with $d = 24 + 24 + 1 + 7 = 56$

24 prices of the day before

24 prices of the 7 days before

Settings

- Forecast for the year 2019.
- Random forest regressor.
- One model per hour, we concatenate the predictions afterwards.

$\hookrightarrow$ 24 models

- $y_t \in \mathbb{R}$
- $x_t \in \mathbb{R}^d$, with $d = 24 + 24 + 1 + 7 = 56$

24 prices of the day before

24 prices of the 7 days before

Forecasted consumption

Settings

- Forecast for the year 2019.
- Random forest regressor.
- One model per hour, we concatenate the predictions afterwards.

$\hookrightarrow$ 24 models

  - $y_t \in \mathbb{R}$
  - $x_t \in \mathbb{R}^d$, with $d = 24 + 24 + 1 + 7 = 56$

24 prices of the day before

24 prices of the 7 days before

Forecasted consumption

Encoded day of the week

Settings

- Forecast for the year 2019.
- Random forest regressor.
- One model per hour, we concatenate the predictions afterwards.

$\hookrightarrow$ 24 models

  - $y_t \in \mathbb{R}$
  - $x_t \in \mathbb{R}^d$, with $d = 24 + 24 + 1 + 7 = 56$
  - 3 years for training/calibration, i.e. $T_0 = 1096$ observations

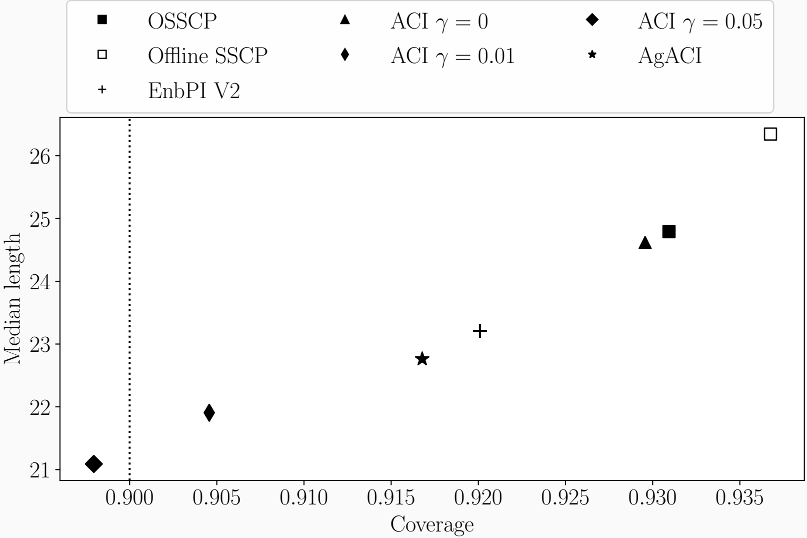
Settings

- Forecast for the year 2019.
- Random forest regressor.
- One model per hour, we concatenate the predictions afterwards.

$\hookrightarrow$ 24 models

  - $y_t \in \mathbb{R}$
  - $x_t \in \mathbb{R}^d$, with $d = 24 + 24 + 1 + 7 = 56$
  - 3 years for training/calibration, i.e. $T_0 = 1096$ observations
  - 1 year to forecast, i.e. $T_1 = 365$ observations

Performance on predicted French electricity Spot price for the year 2019



Performance on predicted French electricity Spot price: visualisation of a day

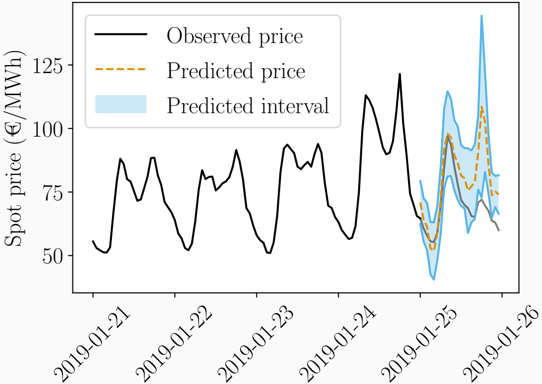


Figure 14: French electricity spot price, its prediction and its uncertainty with AgACI.

Concluding remarks

Contributions and messages

- Pipeline of analysis for simulation of increasing difficulty and real data analysis (code in python) for reproducible work and benchmarking conformal predictions in the framework of time series: `GitHub`

Contributions and messages

- Pipeline of analysis for simulation of increasing difficulty and real data analysis (code in python) for reproducible work and benchmarking conformal predictions in the framework of time series: `GitHub`
- Theoretical results on ACI's length depending on γ

Contributions and messages

- Pipeline of analysis for simulation of increasing difficulty and real data analysis (code in python) for reproducible work and benchmarking conformal predictions in the framework of time series: `GitHub`
- Theoretical results on ACI's length depending on γ
- ACI useful for general time series

Contributions and messages

- Pipeline of analysis for simulation of increasing difficulty and real data analysis (code in python) for reproducible work and benchmarking conformal predictions in the framework of time series: `GitHub`
- Theoretical results on ACI's length depending on γ
- ACI useful for general time series
- Empirical proposition of an adaptive choice of γ: `AgACI`

Contributions and messages

- Pipeline of analysis for simulation of increasing difficulty and real data analysis (code in python) for reproducible work and benchmarking conformal predictions in the framework of time series: `GitHub`
- Theoretical results on ACI's length depending on γ
- ACI useful for general time series
- Empirical proposition of an adaptive choice of γ: `AgACI`

$\hookrightarrow$ **Perspective:** refined analysis of `AgACI` and expert aggregation

Contributions and messages

- Pipeline of analysis for simulation of increasing difficulty and real data analysis (code in python) for reproducible work and benchmarking conformal predictions in the framework of time series: `GitHub`
- Theoretical results on ACI's length depending on γ
- ACI useful for general time series
- Empirical proposition of an adaptive choice of γ: `AgACI`

$\hookrightarrow$ **Perspective:** refined analysis of `AgACI` and expert aggregation

  - Theoretical guarantees about validity: *what happens to the asymptotic result when aggregated?*
  - Analysis of the obtained efficiency
  - More data sets

Thank you!

Cesa-Bianchi, N. and Lugosi, G. (2006). *Prediction, learning, and games*. Cambridge University Press.

Gibbs, I. and Candès, E. (2021). Adaptive Conformal Inference Under Distribution Shift. *arXiv:2106.00170 [stat]*. arXiv: 2106.00170.

Kath, C. and Ziel, F. (2021). Conformal prediction interval estimation and applications to day-ahead and intraday power markets. *International Journal of Forecasting*, 37(2):777–799.

Wisniewski, W., Lindsay, D., and Lindsay, S. (2020). Application of conformal prediction interval estimations to market makers' net positions. In Gammerman, A., Vovk, V., Luo, Z., Smirnov, E., and Cherubin, G., editors, *Proceedings of the Ninth Symposium on Conformal and Probabilistic Prediction and Applications*, volume 128 of *Proceedings of Machine Learning Research*, pages 285–301. PMLR.

Xu, C. and Xie, Y. (2021). Conformal prediction interval for dynamic time-series. In Meila, M. and Zhang, T., editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 11559–11569. PMLR.

Examples of non-exchangeable scores with exchangeable noise

Endogenous and not perfectly estimated

Assume $X_t = Y_{t-1} \in \mathbb{R}$ and that

$$Y_t = aY_{t-1} + \varepsilon_t,$$

where ε_t is a white noise.

Endogenous and not perfectly estimated

Assume $X_t = Y_{t-1} \in \mathbb{R}$ and that

$$Y_t = aY_{t-1} + \varepsilon_t,$$

where ε_t is a white noise.

Assume that the fitted model is $\hat{f}_t(x) = \hat{a}x$, with $\hat{a} \neq a$.

Endogenous and not perfectly estimated

Assume $X_t = Y_{t-1} \in \mathbb{R}$ and that

$$Y_t = aY_{t-1} + \varepsilon_t,$$

where ε_t is a white noise.

Assume that the fitted model is $\hat{f}_t(x) = \hat{a}x$, with $\hat{a} \neq a$.

Then, for any t, we have that:

$$\hat{\varepsilon}_t = Y_t - \hat{Y}_t = (a - \hat{a})\, Y_{t-1} + \varepsilon_t$$

$$\hat{\varepsilon}_t = a\hat{\varepsilon}_{t-1} + \xi_t$$

with $\xi_t = \varepsilon_t - \hat{a}\varepsilon_{t-1}$.

Endogenous and not perfectly estimated

Assume $X_t = Y_{t-1} \in \mathbb{R}$ and that

$$Y_t = aY_{t-1} + \varepsilon_t,$$

where ε_t is a white noise.

Assume that the fitted model is $\hat{f}_t(x) = \hat{a}x$, with $\hat{a} \neq a$.

Then, for any t, we have that:

$$\hat{\varepsilon}_t = Y_t - \hat{Y}_t = (a - \hat{a})\, Y_{t-1} + \varepsilon_t$$

$$\hat{\varepsilon}_t = a\hat{\varepsilon}_{t-1} + \xi_t$$

with $\xi_t = \varepsilon_t - \hat{a}\varepsilon_{t-1}$.

$\hat{\varepsilon}_t$ is an ARMA process of parameters $\varphi = a$ and $\theta = -\hat{a}$.

Endogenous and not perfectly estimated

Assume $X_t = Y_{t-1} \in \mathbb{R}$ and that

$$Y_t = aY_{t-1} + \varepsilon_t,$$

where ε_t is a white noise.

Assume that the fitted model is $\hat{f}_t(x) = \hat{a}x$, with $\hat{a} \neq a$.

Then, for any t, we have that:

$$\hat{\varepsilon}_t = Y_t - \hat{Y}_t = (a - \hat{a})\, Y_{t-1} + \varepsilon_t$$

$$\hat{\varepsilon}_t = a\hat{\varepsilon}_{t-1} + \xi_t$$

with $\xi_t = \varepsilon_t - \hat{a}\varepsilon_{t-1}$.

$\hat{\varepsilon}_t$ is an ARMA process of parameters $\varphi = a$ and $\theta = -\hat{a}$.

Thus, we have generated dependent residuals (ARMA residuals) even if the underlying model only had white noise.

Exogenous and misspecified

Assume $X_t \in \mathbb{R}^2$ and that:

$$Y_t = aX_{1,t} + bX_{2,t} + \varepsilon_t,$$

with $\varepsilon_t \underset{\text{i.i.d.}}{\sim} \mathcal{N}(0,1)$, $X_{2,t+1} = \varphi X_{2,t} + \xi_t,\;\; \xi_t \underset{\text{i.i.d.}}{\sim} \mathcal{N}(0,1)$ and $X_{1,t}$ can be any random variable.

Exogenous and misspecified

Assume $X_t \in \mathbb{R}^2$ and that:

$$Y_t = aX_{1,t} + bX_{2,t} + \varepsilon_t,$$

with $\varepsilon_t \underset{\text{i.i.d.}}{\sim} \mathcal{N}(0,1)$, $X_{2,t+1} = \varphi X_{2,t} + \xi_t, \; \xi_t \underset{\text{i.i.d.}}{\sim} \mathcal{N}(0,1)$ and $X_{1,t}$ can be any random variable.

Assume that we misspecify the model such that the fitted model is $\hat{f}_t(x) = ax_1$.

Exogenous and misspecified

Assume $X_t \in \mathbb{R}^2$ and that:

$$Y_t = aX_{1,t} + bX_{2,t} + \varepsilon_t,$$

with $\varepsilon_t \underset{\text{i.i.d.}}{\sim} \mathcal{N}(0,1)$, $X_{2,t+1} = \varphi X_{2,t} + \xi_t$, $\xi_t \underset{\text{i.i.d.}}{\sim} \mathcal{N}(0,1)$ and $X_{1,t}$ can be any random variable.

Assume that we misspecify the model such that the fitted model is $\hat{f}_t(x) = ax_1$.

Then, for any t, we have that

$$\hat{\varepsilon}_t = Y_t - \hat{Y}_t = bX_{2,t} + \varepsilon_t.$$

Exogenous and misspecified

Assume $X_t \in \mathbb{R}^2$ and that:

$$Y_t = aX_{1,t} + bX_{2,t} + \varepsilon_t,$$

with $\varepsilon_t \underset{\text{i.i.d.}}{\sim} \mathcal{N}(0,1)$, $X_{2,t+1} = \varphi X_{2,t} + \xi_t, \;\; \xi_t \underset{\text{i.i.d.}}{\sim} \mathcal{N}(0,1)$ and $X_{1,t}$ can be any random variable.

Assume that we misspecify the model such that the fitted model is $\hat{f}_t(x) = ax_1$.

Then, for any t, we have that

$$\hat{\varepsilon}_t = Y_t - \hat{Y}_t = bX_{2,t} + \varepsilon_t.$$

Thus, we have generated dependent residuals (auto-regressive residuals) even if the underlying model only had i.i.d. Gaussian noise.

Analysis of ACI's efficiency depending on γ

Numerical analysis of ACI's length: AR(1) case

Assume the residuals follow an AR(1) process: $\hat{\varepsilon}_{t+1} = \varphi \hat{\varepsilon}_t + \xi_{t+1}$ with $(\xi_t)_t$ i.i.d. random variables and other assumptions, we have:

$$\frac{1}{T}\sum_{t=1}^{T} L(\alpha_t) \xrightarrow[T\to+\infty]{a.s.} \mathbb{E}_{\pi_{\gamma,\varphi}}[L].$$

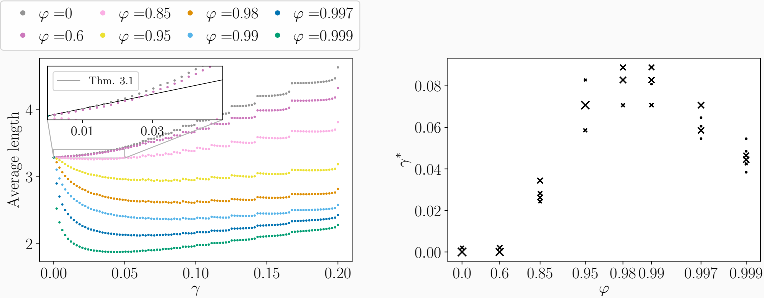


Figure 15: Left: evolution of the mean length depending on γ for various φ. Right: γ^* minimizing the average length for each φ.

EnbPI

EnbPI, Xu and Xie (2021)

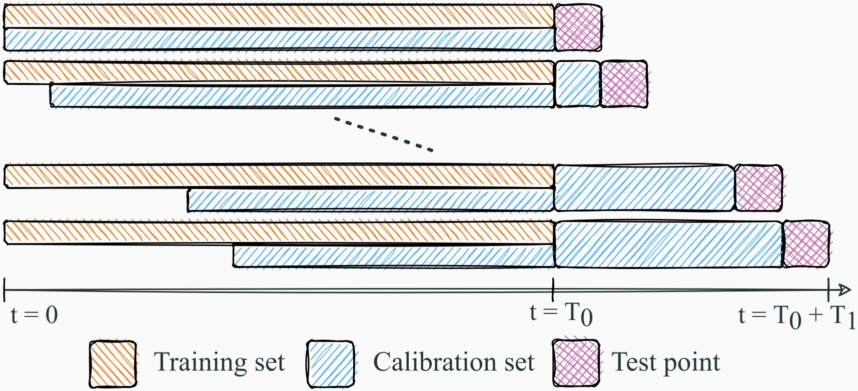
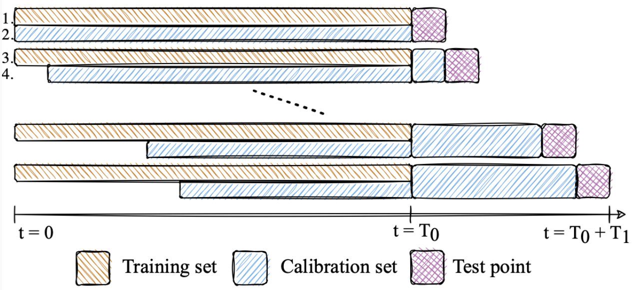


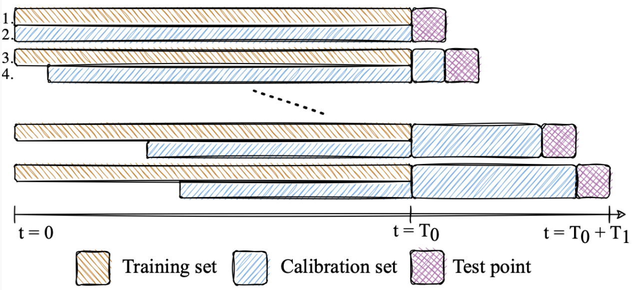
Figure 16: Diagram describing the EnbPI algorithm.

EnbPI, Xu and Xie (2021)



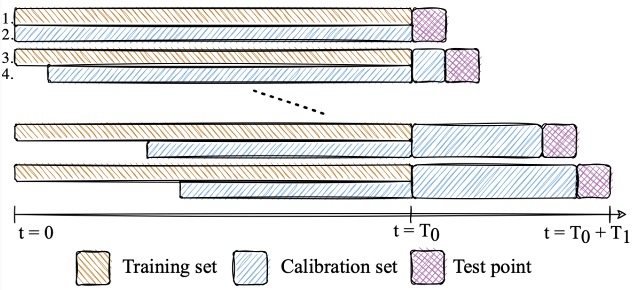
1. Train B bootstrap predictors;

EnbPI, Xu and Xie (2021)



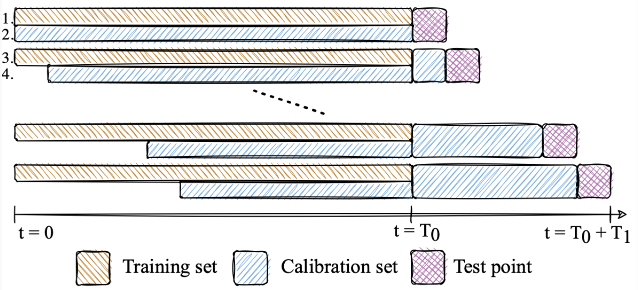
1. Train B bootstrap predictors;
2. Obtain out-of-bootstrap residuals by aggregating the corresponding predictors;

EnbPI, Xu and Xie (2021)



1. Train B bootstrap predictors;
2. Obtain out-of-bootstrap residuals by aggregating the corresponding predictors;
3. Do not re-train the B bootstrap predictors;

EnbPI, Xu and Xie (2021)



1. Train B bootstrap predictors;
2. Obtain out-of-bootstrap residuals by aggregating the corresponding predictors;
3. Do not re-train the B bootstrap predictors;
4. Obtain new residual by aggregating all the predictors. Forget the first residuals.

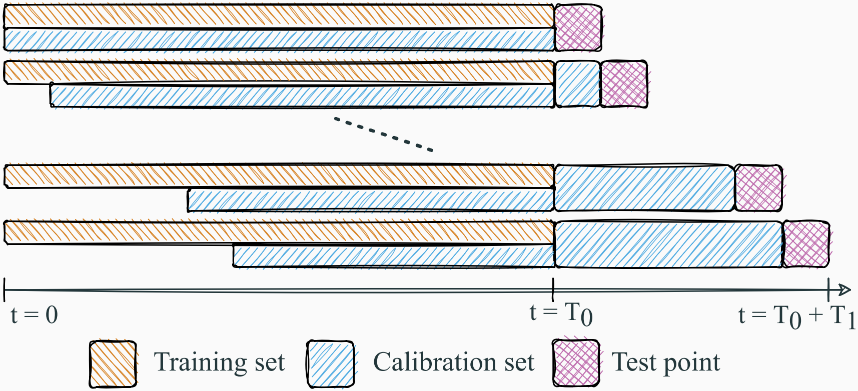


Figure 16: Diagram describing the EnbPI algorithm.

EnbPI (ICML, Xu and Xie, 2021) aggregates with 2 different functions.

EnbPI, Xu and Xie (2021)

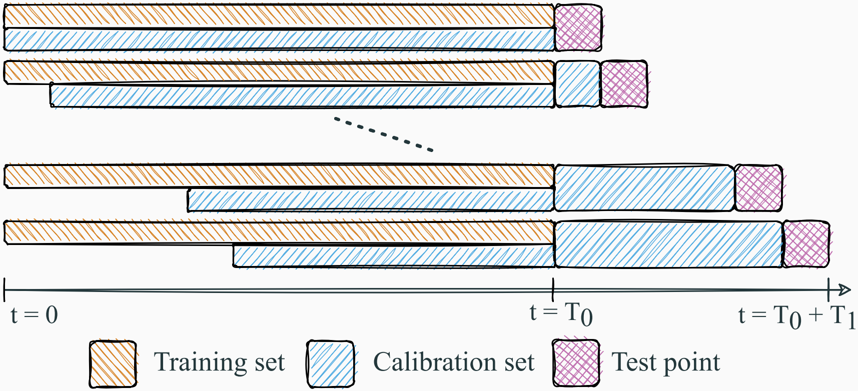


Figure 16: Diagram describing the EnbPI algorithm.

EnbPI (ICML, Xu and Xie, 2021) aggregates with 2 different functions.

$\Rightarrow$ We propose `EnbPI V2` with the same aggregation function all along (similar to EnbPI on last ArXiV version from Xu and Xie).

EnbPI, Xu and Xie (2021)

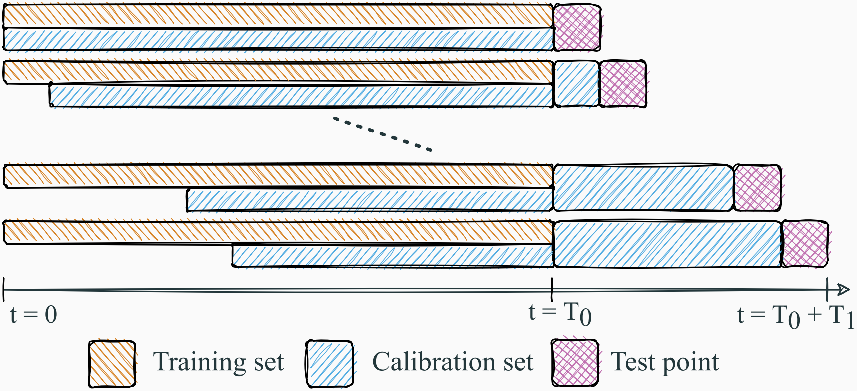


Figure 16: Diagram describing the EnbPI algorithm.

↪ tested on other real time series

↪ compared to offline methods

Details on the simulation set up

Data generation

$$Y_t = 10\sin\left(\pi X_{t,1} X_{t,2}\right) + 20\left(X_{t,3} - 0.5\right)^2 + 10X_{t,4} + 5X_{t,5} + \varepsilon_t$$

where the X_t are multivariate uniformly distributed on $[0, 1]$ and ε_t are generated from an ARMA(1,1) process.

Data generation

$$Y_t = 10\sin\left(\pi X_{t,1} X_{t,2}\right) + 20\left(X_{t,3} - 0.5\right)^2 + 10 X_{t,4} + 5 X_{t,5} + \varepsilon_t$$

where the X_t are multivariate uniformly distributed on $[0, 1]$ and ε_t are generated from an ARMA(1,1) process.

$\Rightarrow$ dependence structure in the noise in order to:

- control the strength of the scores dependence,
- evaluate the impact of this temporal dependence structure of the results.

Auto-Regressive Moving Average

Definition (ARMA(1,1) process)

We say that ε_t is an ARMA(1,1) process if for any t:

$$\varepsilon_{t+1} = \varphi\varepsilon_t + \xi_{t+1} + \theta\xi_t,$$

with:

- $\theta + \varphi \neq 0$, $|\varphi| < 1$ and $|\theta| < 1$;
- ξ_t is a white noise of variance σ^2, called the **innovation**.

Auto-Regressive Moving Average

Definition (ARMA(1,1) process)

We say that ε_t is an ARMA(1,1) process if for any t:

$$\varepsilon_{t+1} = \varphi\varepsilon_t + \xi_{t+1} + \theta\xi_t,$$

with:

- $\theta + \varphi \neq 0$, $|\varphi| < 1$ and $|\theta| < 1$;
- ξ_t is a white noise of variance σ^2, called the **innovation**.

- The higher φ and θ, the stronger the dependence.
- The asymptotic variance of this process is:

$$\text{Var}(\varepsilon_t) = \sigma^2 \frac{1 - 2\varphi\theta + \theta^2}{1 - \varphi^2}.$$

- If $\theta = 0$, only the auto-regressive part, it is an AR(1).
- If $\varphi = 0$, only the moving-average part, it is an MA(1).

Simulation settings

- φ and θ range in $[0.1, 0.8, 0.9, 0.95, 0.99]$.
- We fix σ so as to keep the variance $\mathrm{Var}(\varepsilon_t)$ constant to 1 or 10.
- We use random forest as regressor.

Simulation settings

- φ and θ range in $[0.1, 0.8, 0.9, 0.95, 0.99]$.
- We fix σ so as to keep the variance $\text{Var}(\varepsilon_t)$ constant to 1 or 10.
- We use random forest as regressor.

For each setting:

- 300 points, the last 100 kept for prediction and evaluation,
- 500 repetitions,

$\Rightarrow$ in total, $100 \times 500 = 50000$ predictions are evaluated.

Simulation settings

- φ and θ range in $[0.1, 0.8, 0.9, 0.95, 0.99]$.
- We fix σ so as to keep the variance $\text{Var}(\varepsilon_t)$ constant to 1 or 10.
- We use random forest as regressor.

For each setting:

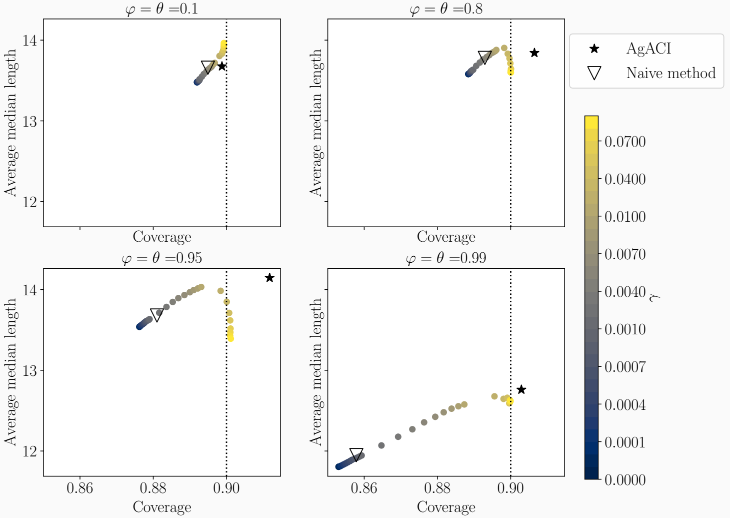
- 300 points, the last 100 kept for prediction and evaluation,
- 500 repetitions,

$\Rightarrow$ in total, $100 \times 500 = 50000$ predictions are evaluated.

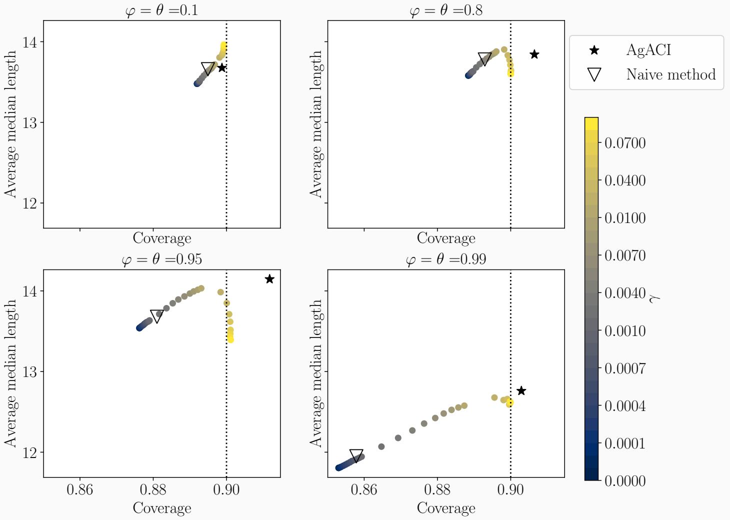
We present the results in the ARMA(1,1) case, but we also have them for AR(1) and MA(1) processes.

Additional results on the synthetic data sets

Empirical evaluation of ACI sensitivity to γ and adaptive choice

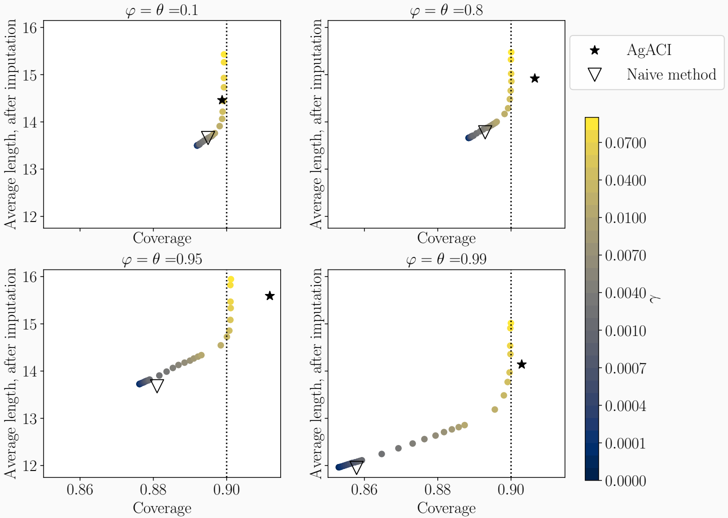


Empirical evaluation of ACI sensitivity to γ and adaptive choice



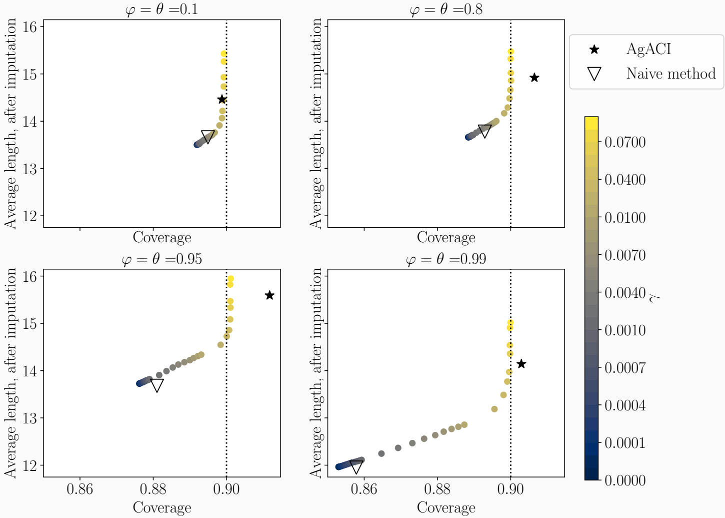
$\Rightarrow$ The more the dependence, the more sensitive to γ is ACI.

Empirical evaluation of ACI sensitivity to γ and adaptive choice



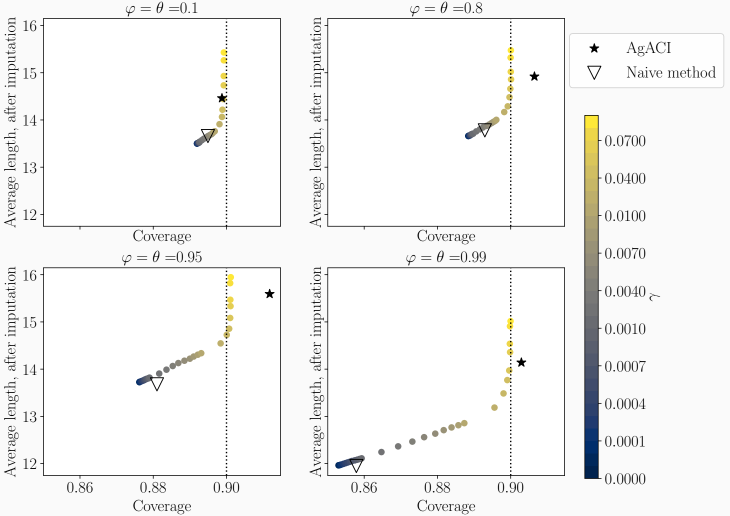
$\Rightarrow$ The more the dependence, the more sensitive to γ is ACI.

Empirical evaluation of ACI sensitivity to γ and adaptive choice



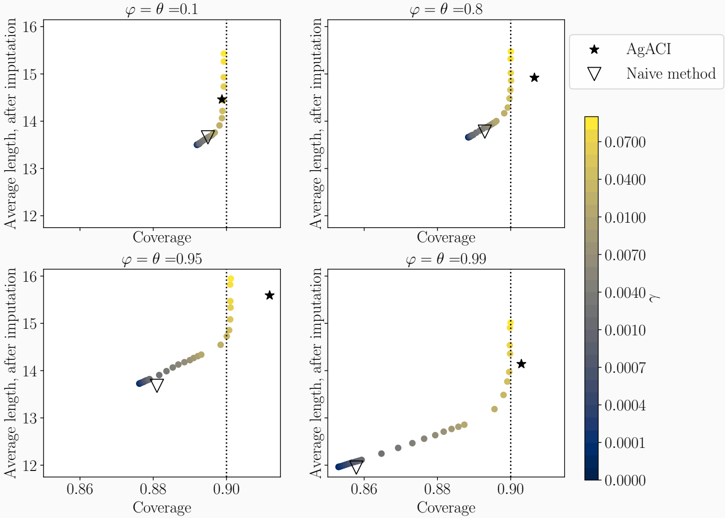
Naive method ($\triangledown$): smallest among valid ones in the past

Empirical evaluation of ACI sensitivity to γ and adaptive choice



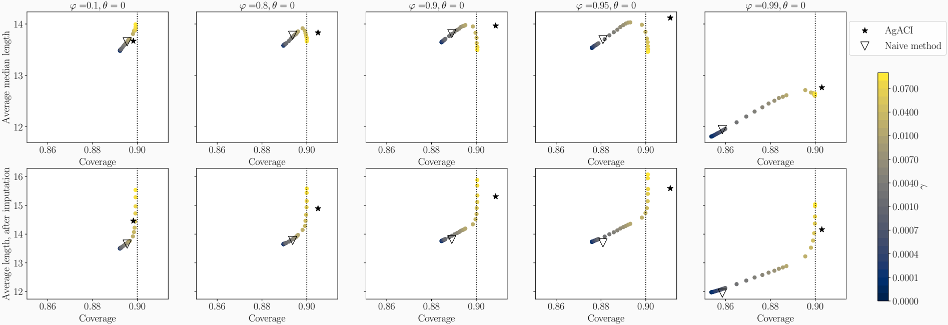
Naive method (∇): smallest among valid ones in the past $\Rightarrow$ accumulates error of the different ACI's versions.

Empirical evaluation of ACI sensitivity to γ and adaptive choice

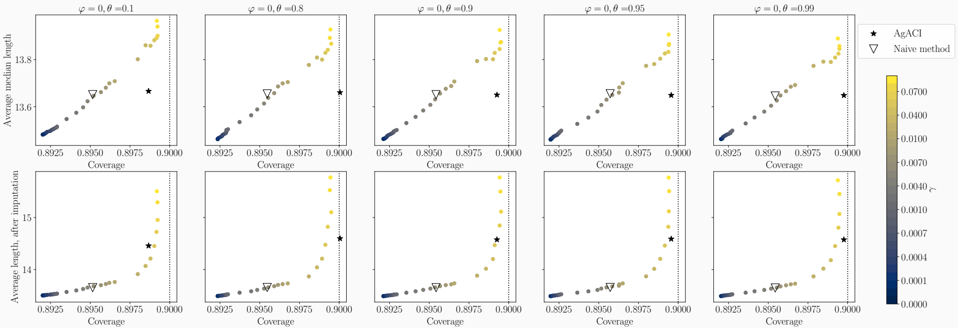


AgACI (★): encouraging preliminary results.

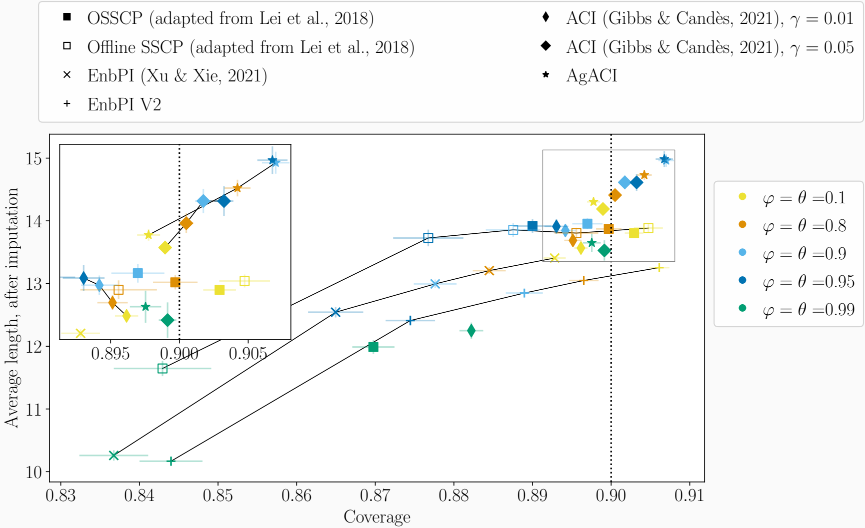
Empirical evaluation of ACI sensitivity to γ and adaptive choice, AR(1)



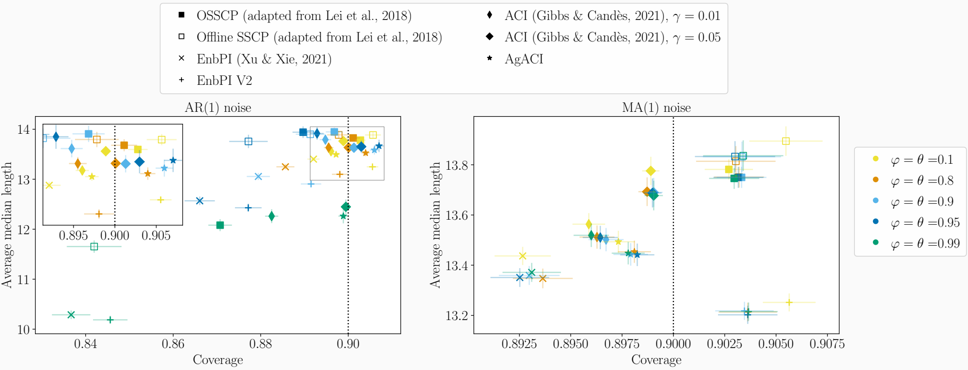
Empirical evaluation of ACI sensitivity to γ and adaptive choice, MA(1)



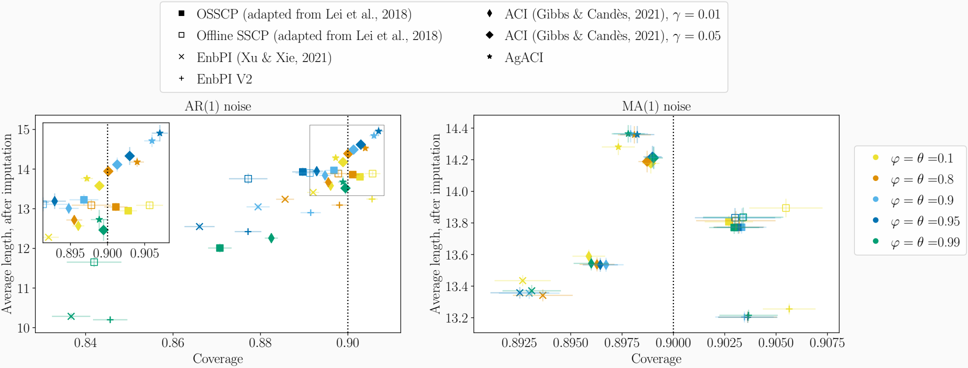
Results: impact of the temporal dependence, ARMA(1), variance 10, average length after imputation



Results: impact of the temporal dependence, AR(1) and MA(1), variance 10



Results: impact of the temporal dependence, AR(1) and MA(1), variance 10, average length after imputation



Additional results on the French electricity spot prices

Forecasting French electricity Spot prices with confidence: results

- Target coverage: 90%
- Empirical coverage: 91.68%
- Median length: 22.76€/MWh

Performance on predicted French electricity Spot price: visualisation of a day

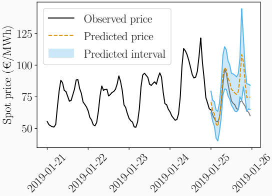


Figure 17: OSSCP

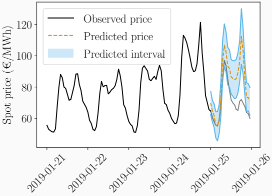


Figure 18: EnbPI V2

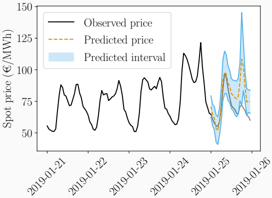


Figure 19: ACI with $\gamma = 0.01$

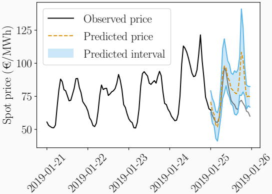


Figure 20: ACI with $\gamma = 0.05$