Conformal prediction and beyond

Uncertainty quantification for regression & time-series problems

Gerard Castro

Universitat de Barcelona

June 16, 2024

Universitat de Barcelona

Facultat de Matemàtiques i Informàtica

Motivation

- We extract n samples from $(X, Y) \in \mathbb{R}^d \times \mathbb{R}$ random variables with unknown marginal & joint distributions.
- Given a new sample X_{n+1} & miscoverage level $\alpha \in [0, 1]$:
  - We want to **estimate** a predictive **interval** $\mathcal{C}_\alpha$ such that the probability of Y_{n+1} falling into $\mathcal{C}_\alpha$ is at least $1 - \alpha$, *i.e.*

$$\mathbb{P}\{Y_{n+1} \in \mathcal{C}_\alpha(X_{n+1})\} \geq 1 - \alpha$$

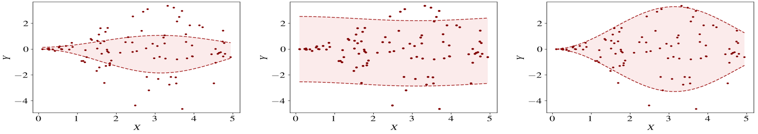
  - The interval should be the **smallest** possible while **keeping coverage**. **Conditional** coverage ideally sought.

Motivation

- We extract n samples from $(X, Y) \in \mathbb{R}^d \times \mathbb{R}$ random variables with unknown marginal & joint distributions.
- Given a new sample X_{n+1} & miscoverage level $\alpha \in [0,1]$:
  - We want to **estimate** a predictive **interval** $\mathcal{C}_\alpha$ such that the probability of Y_{n+1} falling into $\mathcal{C}_\alpha$ is at least $1-\alpha$, *i.e.*

$$\mathbb{P}\{Y_{n+1} \in \mathcal{C}_\alpha(X_{n+1})\} \geq 1-\alpha$$

  - The interval should be the **smallest** possible while **keeping coverage**. **Conditional** coverage ideally sought.



(a) No coverage **(b)** Marginal coverage **(c)** Conditional cov.

Figure 1: Different types of coverage.

Split Conformal Prediction (SCP)

We need to use out-of-training data to understand how errors distribute: we need to "*conformalize*" the predictions to the data using a "*conformity score*". SCP proposes:

1. Split data into **training** Tr & **calibration** Cal.
2. Obtain $\hat{\mu}$ by training it in Tr.
3. Obtain a set $\mathcal{S}$ of conformity scores by using the Cal set: $\mathcal{S}_{\mathrm{Cal}} := \{|Y_i - \hat{\mu}(X_i)|, i \in \mathrm{Cal}\}$.
4. Compute the $1-\alpha$ "*empirical quantile*" of $\mathcal{S}_{\mathrm{Cal}}$: $q_{1-\alpha}(\mathcal{S}_{\mathrm{Cal}})$.

Split Conformal Prediction (SCP)

We need to use out-of-training data to understand how errors distribute: we need to "*conformalize*" the predictions to the data using a "*conformity score*". SCP proposes:

1. Split data into **training** Tr & **calibration** Cal.
2. Obtain $\hat{\mu}$ by training it in Tr.
3. Obtain a set $\mathcal{S}$ of conformity scores by using the Cal set: $\mathcal{S}_{\mathrm{Cal}} := \{|Y_i - \hat{\mu}(X_i)|, i \in \mathrm{Cal}\}$.
4. Compute $(1-\alpha)\left(\frac{1}{\#\mathrm{Cal}}+1\right)$ quantile of $\mathcal{S}_{\mathrm{Cal}}$: $q_{1-\alpha}(\mathcal{S}_{\mathrm{Cal}})$.

Split Conformal Prediction (SCP)

We need to use out-of-training data to understand how errors distribute: we need to "*conformalize*" the predictions to the data using a "*conformity score*". SCP proposes:

1. Split data into **training** Tr & **calibration** Cal.
2. Obtain $\hat{\mu}$ by training it in Tr.
3. Obtain a set $\mathcal{S}$ of conformity scores by using the Cal set: $\mathcal{S}_{\mathrm{Cal}} := \{ |Y_i - \hat{\mu}(X_i)|, i \in \mathrm{Cal} \}$.
4. Compute $(1-\alpha)\left(\frac{1}{\#\mathrm{Cal}} + 1\right)$ quantile of $\mathcal{S}_{\mathrm{Cal}}$: $q_{1-\alpha}(\mathcal{S}_{\mathrm{Cal}})$.
5. For a new sample X_{n+1}, return

$$\hat{\mathcal{C}}_\alpha = [\hat{\mu}(X_{n+1}) - q_{1-\alpha}(\mathcal{S}_{\mathrm{Cal}}),\ \hat{\mu}(X_{n+1}) + q_{1-\alpha}(\mathcal{S}_{\mathrm{Cal}})]$$

Note

The only hypothesis required is **data exchangeability**.

Conformalized Quantile Regression (CQR)

We need to use out-of-training data to understand how errors distribute: we need to "*conformalize*" the predictions to the data using a "*conformity score*". ~~SCP~~ CQR proposes:

1. Split data into **training** Tr & **calibration** Cal.
2. Obtain $\cancel{\hat{\mu}}$ $\hat{\mu}_{\mathrm{down}}$ & $\hat{\mu}_{\mathrm{up}}$ trained in Tr.
3. Obtain a set $\mathcal{S}$ of conformity scores by using the Cal set: $\mathcal{S}_{\mathrm{Cal}} := \{\max(\hat{\mu}_{\mathrm{down}}(X_i) - Y_i,\ Y_i - \hat{\mu}_{\mathrm{up}}(X_i)), i \in \mathrm{Cal}\}$.
4. Compute $(1-\alpha)\left(\frac{1}{\#\mathrm{Cal}} + 1\right)$ quantile of $\mathcal{S}_{\mathrm{Cal}}$: $q_{1-\alpha}(\mathcal{S}_{\mathrm{Cal}})$.
5. For a new sample X_{n+1}, return

$$\hat{C}_\alpha(X_{n+1}) = [\hat{\mu}_{\mathrm{down}}(X_{n+1}) - q_{1-\alpha}(\mathcal{S}_{\mathrm{Cal}}),\ \hat{\mu}_{\mathrm{up}}(X_{n+1}) + q_{1-\alpha}(\mathcal{S}_{\mathrm{Cal}})]$$

Note

The only hypothesis required is **data exchangeability**.

Other flavours

Then, other flavours are proposed to reconcile this trade-off between statistical and computational efficiency, for instance CV+ & J+aB:

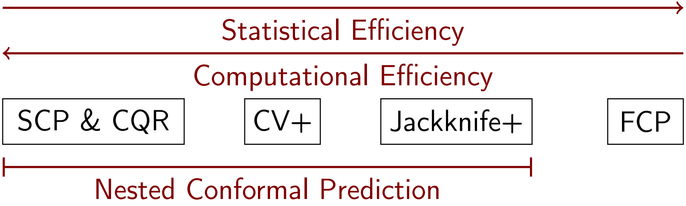


Figure 2: Trade-off between statistical & computational efficiency.

- Both CV+ & J+aB are based on defining multiple folds to apply a similar methodology as SCP: cross-validation & leave-one-out (LOO) folds, respectively.

Covariate shift: changes in features' distribution

- $\{(X_i, Y_i)\}_{i=1}^{n} \overset{\text{exch.}}{\sim} P_X \times P_{Y|X}$
- $(X_{n+1}, Y_{n+1}) \sim \tilde{P}_X \times P_{Y|X}$
- *Tibshirani et al. (2019)* heuristic idea:
  1. Estimate how "close" a sample X_i ($\sim P_X$) is *w.r.t.* to the test point ($\sim \tilde{P}_X$) using the likelihood ratio: $w(X_i) := \frac{d\tilde{P}_X(X_i)}{dP_X(X_i)}$.
  2. Normalize the weights: $\omega_i := \frac{w(X_i)}{\sum_{j=1}^{n+1} w(X_j)}$.
  3. Build the predictive interval $\mathcal{C}_\alpha$ using the weighted calibration samples:

$$\hat{\mathcal{C}}_\alpha(X_{n+1}) = \{Y : \ s_{\hat{\mu}}(X_{n+1}, Y) \leq q_{1-\alpha}(\{\omega_i S_i\}_{i \in \mathrm{Cal}})\}$$

Label shift: changes in target's distribution

- $\{(X_i, Y_i)\}_{i=1}^{n} \overset{\text{exch.}}{\sim} P_{X|Y} \times P_Y$
- $(X_{n+1}, Y_{n+1}) \sim P_{X|Y} \times \tilde{P}_Y$
- *A. Podkopaev & A. Ramdas (2021)* adapts former idea letting weights as function of Y, ω_i^Y:
  1. Estimate how "close" a label Y_i ($\sim P_Y$) is *w.r.t.* to the hypothetical point ($\sim \tilde{P}_Y$) using the likelihood ratio: $w(Y_i) := \frac{d\tilde{P}_Y(Y_i)}{dP_Y(Y_i)}$.
  2. Normalize the weights: $\omega_i^Y := \frac{w(Y_i)}{\sum_{j=1}^{n} w(Y_j) + w(Y)}$.
  3. Build the predictive interval $\mathcal{C}_\alpha$ traversing all the variable output's space and using the weighted calibration samples:

$$\hat{\mathcal{C}}_\alpha(X_{n+1}) = \{Y : \ s_{\hat{\mu}}(X_{n+1}, Y) \leq q_{1-\alpha}\left(\{\omega_i^Y S_i\}_{i \in \mathrm{Cal}}\right)\}$$

Time series data: samples (temporal) auto-correlation

- Assume a setup like $Y_t = \mu(X_t) + \epsilon_t$, where ϵ_t are *i.i.d.* according to a cumulative distribution function F.
- Let the first T sample points $\mathcal{D} := \{(X_t, Y_t)_{t=1}^{T}\}$ be training data: we **want** a **sequence** of $s \geq 1$ **intervals** of α miscoverage level, $\{\mathcal{C}^{\alpha}_{T,T+i}\}_{i=1}^{s}$ (for the unknown labels $\{Y^{\alpha}_{T+i}\}_{i=1}^{s}$).
    - s is the batch size (nº steps to look ahead)
- Also, once **new samples** $\{(X_{T+i}, Y_{T+i})\}_{i=1}^{s}$ become **available**, we would like to also **leverage them**.
    - We want to use the most recent $T+s$ points for the $\{\mathcal{C}^{\alpha}_{T+s,j}\}_{j=T+s+1}^{=T+2s}$ intervals.

Time series data: samples (temporal) auto-correlation

- Assume a setup like $Y_t = \mu(X_t) + \epsilon_t$, where ϵ_t are *i.i.d.* according to a cumulative distribution function F.
- Let the first T sample points $\mathcal{D} := \{(X_t, Y_t)_{t=1}^{T}\}$ be training data: we **want** a **sequence** of $s \geq 1$ **intervals** of α miscoverage level, $\{\mathcal{C}^{\alpha}_{T,T+i}\}_{i=1}^{s}$ (for the unknown labels $\{Y^{\alpha}_{T+i}\}_{i=1}^{s}$).
  - s is the batch size (nº steps to look ahead)
- Also, once **new samples** $\{(X_{T+i}, Y_{T+i})\}_{i=1}^{s}$ become **available**, we would like to also **leverage them**.
  - We want to use the most recent $T+s$ points for the $\{\mathcal{C}^{\alpha}_{T+s,j}\}_{j=T+s+1}^{=T+2s}$ intervals.

$\Longrightarrow$ *C. Xu & Y. Xie (2021)* proposes the "*EnbPI*" methodology:

- It uses no data-splitting but LOO estimators ($\hat{\mu}_{-i}$ model trained with $\mathcal{D} \setminus \{(X_i, Y_i)\}$).
- Models not refitted during test time, but newest samples' residuals used to further *conformalize* predictions.

EnbPI idea

There are T training samples and we build T_1 intervals (indices $T+1,\ldots,T+T_1$):

- Obtain B bootstrapped models μ^b by:
  - Sampling, with replacement, an index set $S_b := (i_1,\ldots,i_T)$
  - Fitting the bootstrapped model with S_b
- For $i = 1,\ldots,T$:
  - Aggregate μ^b with any function ϕ: obtaining $\hat{\mu}^{\phi}_{-i}$.
  - Compute conformity scores: $\epsilon^{\phi}_i := |Y_i - \hat{\mu}^{\phi}_{-i}(X_i)|$.
- For each $t = T+1,\ldots,T+T_1$ timestamps, return in batches of s size:

$$\hat{C}^{\alpha}_{T,t}(X_t) = \left[\hat{\mu}^{\phi}_{-t}(X_t) \pm w^{\phi}_t\right], \text{where } \begin{cases} \hat{\mu}^{\phi}_{-t}(X_t): \ 1-\alpha \text{ quant. } \{\hat{\mu}^{\phi}_{-i}(X_t)\}_{i=1}^{T} \\ w^{\phi}_t: \ 1-\alpha \text{ quantile of } \{\epsilon^{\phi}_i\}_{i=1}^{T} \end{cases}$$

- "*Partial fit*" **step**: for each s returned intervals, conformity score w^{ϕ}_t is re-computed with the most recent observations.

Metrics definition

The following metrics will be used:

- **Coverage level**: *i.e.* fraction of true labels lying within the prediction intervals (the closer to $1-\alpha$, the better)
- **Interval width**: intervals' mean width (the smaller, the better)
- **"*Informativeness*"**: best width-coverage ratio, assessed through CWC score (the higher, the better):

$$\mathrm{CWC} = (1-w) * \exp\left(-\eta(c-(1-\alpha))^2\right), \text{ with } \begin{cases} w & \text{mean width} \\ c & \text{attained coverage} \\ \eta & \text{balancing term} \end{cases}$$

- **Adaptability**: ability of achieving conditional coverage, assessed through SSC score (the closer to $1-\alpha$, the better).
  - Maximum coverage violation along all width groups.
  - Only usable for non-constant width intervals.
- **Computational efficiency**: measured by CPU time.

1. Introduction
2. Conformal prediction
3. Beyond exchangeability
4. Results
   - Assessment
   - Regression problem
   - Time series problem
5. Conclusions

Data & modeling

A tabular regression problem is considered with:

- The `sklearn` built-in California Housing dataset (20,640 samples, 8 features).
- A (light) gradient boosting regressor, `LGBM`, automatically fine-tuned through grid-search.
- A 5-fold cross-validation assessment for $\alpha = 0.20$ miscoverage level.

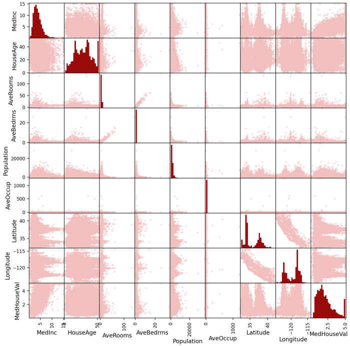


Figure 3: Marginal distributions.

Metrics table

Strategy	Coverage	RMSE	Train. time	Inf. time
SCP	0.806 ± 0.008	0.472 ± 0.007	1.6 ± 0.2	0.07 ± 0.05
CV+	0.853 ± 0.004	0.467 ± 0.009	9 ± 3	8.0 ± 0.3
J+aB	0.734 ± 0.007	0.467 ± 0.009	51 ± 5	9.7 ± 0.4
CQR	0.805 ± 0.010	0.494 ± 0.013	2.6 ± 0.1	0.10 ± 0.04

Strategy	Coverage	Width	CWC	SSC
SCP	0.806 ± 0.008	0.971 ± 0.015	0.798 ± 0.004	—
CV+	0.853 ± 0.004	1.042 ± 0.005	0.784 ± 0.002	0.65 ± 0.01
J+aB	0.734 ± 0.007	0.710 ± 0.003	0.853 ± 0.001	—
CQR	0.805 ± 0.010	1.013 ± 0.013	0.790 ± 0.004	0.75 ± 0.04

Other visualizations

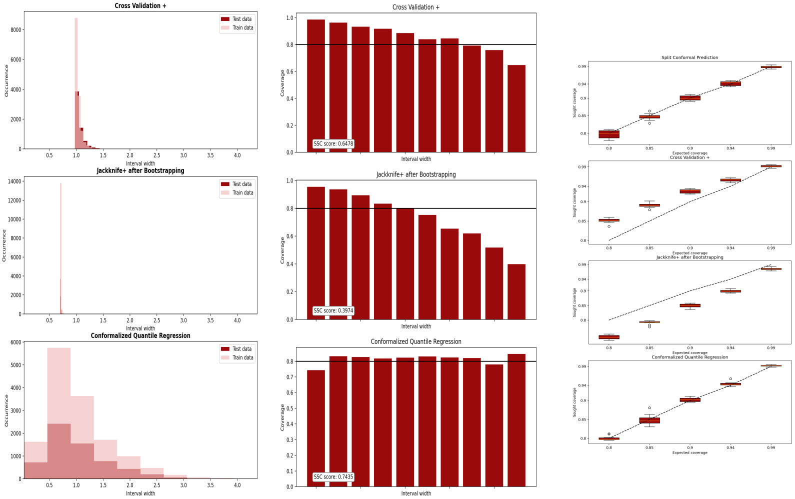


Figure 4: Width histograms & coverage vs. width & α.

Dataset

A time series forecasting problem is considered with:

- Victoria electricity demand dataset (1340 samples, features: time, demand lagged up to 7 days & temperature).
- A `sklearn` random forest regressor automatically fine-tuned through grid-search.
- A 5-fold cross-validation for $\alpha = 0.20$:

Dataset

A time series forecasting problem is considered with:

- Victoria electricity demand dataset (1340 samples, features: time, demand lagged up to 7 days & temperature).
- A `sklearn` random forest regressor automatically fine-tuned through grid-search.
- A 5-fold cross-validation for $\alpha = 0.20$:

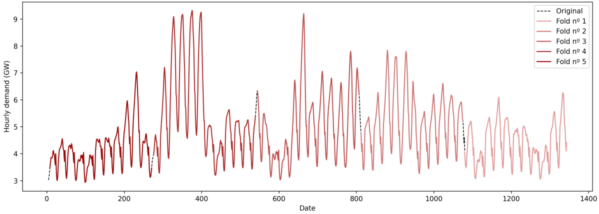


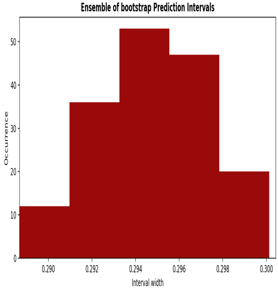
Figure 5: 5-fold CV splits.

Metrics table

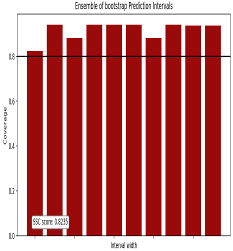
Strategy	Coverage	RMSE	Total time
EnbPI_nP	0.780 ± 0.069	0.165 ± 0.067	6.2 ± 0.3
EnbPI	0.789 ± 0.058	0.165 ± 0.067	528.3 ± 0.4

Strategy	Coverage	Width	CWC	SSC
EnbPI_nP	0.780 ± 0.069	0.293 ± 0.013	0.935 ± 0.018	—
EnbPI	0.789 ± 0.058	0.300 ± 0.007	0.93 ± 0.02	0.5 ± 0.2

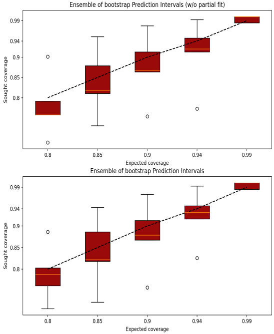
Other visualizations



(a) EnbPI intervals' width histograms



(b) EnbPI coverage in function of intervals' width



(c) EnbPI_nP & EnbPI coverage in function of α

Figure 6: Width histograms & coverage vs. width & α.

Idea

The consistency of former time series may outshine the benefits of the "*partial fit*" EnbPI feature. Thus:

- A change point is added in test to mock off a distribution shift.
- The same random forest regressor will be applied to a 5-fold cross-validation, now for $\alpha = 0.05$:

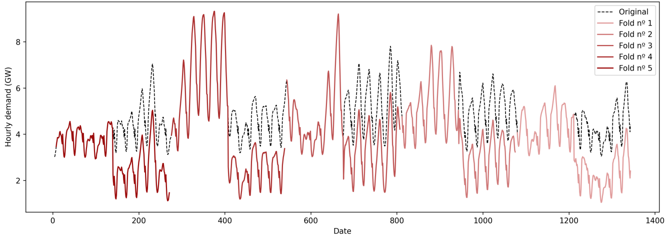
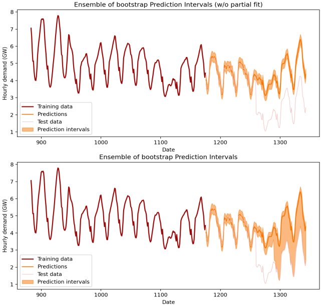


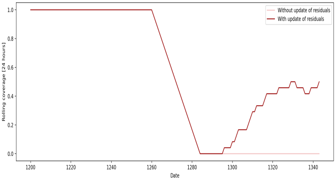
Figure 7: 5-fold CV splits with change points in each test's.

Metrics & plot

Strategy	Coverage	RMSE	Total time
EnbPI_nP	0.439 ± 0.075	1.431 ± 0.024	6.0 ± 0.3
EnbPI	0.696 ± 0.042	1.431 ± 0.024	530 ± 1

Strategy	Coverage	Width	SSC
EnbPI_nP	0.439 ± 0.075	0.569 ± 0.043	—
EnbPI	0.696 ± 0.042	1.300 ± 0.034	0.07 ± 0.12





Regression & time series

The best strategies for exchangeable data are, **decreasingly ordered** by:

- *Statistical efficiency*: CQR, SCP, CV+, J+aB.
  - This is fulfilled independently of α.
- *Computational efficiency*: SCP, CQR, CV+, J+aB.
- *Predictive power* are: CV+ & J+aB, SCP, CQR.
- "*Informativeness*": J+aB, SCP, CQR, CV+.
- *Adaptability*: CQR, CV+, J+aB (slight to none). Contrarily, SCP intervals are not adaptive at all.

Regarding the time series case, **EnbPI** is a **suitable option** to provide valid intervals.

- EnbPI's adjustment using test residuals is necessary.
- This option also allows all the issued **intervals** to be **adaptive**.

Thank you for your attention!

Questions?