Stable Conformal Prediction Sets

Eugene Ndiaye [1]

Abstract

When one observes a sequence of variables $(x_1, y_1), \dots, (x_n, y_n)$, Conformal Prediction (CP) is a methodology that allows to estimate a confidence set for y_{n+1} given x_{n+1} by merely assuming that the distribution of the data is exchangeable. CP sets have guaranteed coverage for any finite population size n. While appealing, the computation of such a set turns out to be infeasible in general, *e.g.,* when the unknown variable y_{n+1} is continuous. The bottleneck is that it is based on a procedure that readjusts a prediction model on data where we replace the unknown target by all its possible values in order to select the most probable one. This requires computing an infinite number of models, which often makes it intractable. In this paper, we combine CP techniques with classical algorithmic stability bounds to derive a prediction set computable with a single model fit. We demonstrate that our proposed confidence set does not lose any coverage guarantees while avoiding the need for data splitting as currently done in the literature. We provide some numerical experiments to illustrate the tightness of our estimation when the sample size is sufficiently large, on both synthetic and real datasets.

1. Introduction

Modern machine learning algorithms can predict the label of an object based on its observed characteristics with impressive accuracy. They are often trained on historical datasets sampled from the same distribution and it is important to quantify the uncertainty of their predictions. Conformal prediction is a versatile and simple method introduced in (Vovk et al., 2005; Shafer & Vovk, 2008) that provides a finite sample and distribution free $100(1-\alpha)\%$ confidence region on the predicted object based on past observations. The main idea can be subsumed as a hypothesis testing between

$$H_0 : y_{n+1} = z \qquad \text{and} \qquad H_1 : y_{n+1} \neq z \ ,$$

where z is any replacement candidate for the unknown response y_{n+1}. The conformal prediction set will consist of the collection of candidates whose tests are not rejected. The construction of a p-value function is simple. We start by fitting a model with training set $\{(x_1, y_1), \dots, (x_n, y_n), (x_{n+1}, z)\}$ and sort the prediction scores/errors for each instance in ascending order. A candidate z will be considered as conformal or typical if the rank of its score is sufficiently small compared to the others. The key assumption is that the predictive model and the joint probability distribution of the sequence $\{(x_i, y_i)\}_{i=1}^{n+1}$ are invariant *w.r.t.* permutation of the data. As a consequence, the ranks of the scores are equally likely and thus follow a uniform distribution which allow to calibrate a threshold on the rank statistics leading to a valid confidence set. This method has a strong coverage guarantee without any further assumptions on the distribution and is valid for any finite sample size n; see more details in Section 2.

Conformal prediction technique has been applied for designing uncertainty sets in active learning (Ho & Wechsler, 2008), anomaly detection (Laxhammar & Falkman, 2015; Bates et al., 2021), few shot learning (Fisch et al., 2021), time series (Chernozhukov et al., 2018; Xu & Xie, 2021; Chernozhukov et al., 2021), robust optimization (Johnstone & Cox, 2021) or to infer the performance guarantee for statistical learning algorithms (Holland, 2020; Cella & Ryan, 2020). Currently, we are seeing a growing interest in these approaches due to their flexibility and ease of deployment even for very complex problems where classical approaches offer limited performance (Efron, 2021). We refer to (Balasubramanian et al., 2014) for other AI applications.

Despite its nice properties, the computation of conformal prediction sets requires fitting a model on a new augmented dataset where the unknown quantity y_{n+1} is replaced by a set of candidates. In a regression setting where an object can take an uncountable possible value, the set of candidates is infinite. Therefore, computing the conformal prediction is infeasible without additional structural assumptions about the underlying model fit, and even so, the current computational costs remain very high. Hence the prevailing recommendation to use less efficient data splitting methods.

[1]H. Milton Stewart School of Industrial and Systems Engineering, Georgia Institute of Technology, Atlanta, GA, USA. Correspondence to: Eugene Ndiaye <endiaye3@gatech.edu>.

Proceedings of the 39^{th} International Conference on Machine Learning, Baltimore, Maryland, USA, PMLR 162, 2022.

Contribution. We leverage algorithmic stability to bound the variation of the predictive model *w.r.t.* to changes in the input data. This results in a circumvention of the computational bottleneck induced by the necessary readjustment of the model each time we want to assess the typicalness of a candidate replacement of the target variable. As such, we can provide a tight estimation of the confidence sets without loss in the coverage guarantee. Our method is computationally and statistically efficient since it requires only a single model fit and does not involve any data splitting.

Notation. For a nonzero integer n, we denote $[n]$ to be the set $\{1, \cdots, n\}$. The dataset of size n is denoted $\mathcal{D}_n = (x_i, y_i)_{i\in[n]}$, the row-wise input feature matrix $X = [x_1, \cdots, x_n, x_{n+1}]^\top$. Given a set $\{u_1, \cdots, u_n\}$, the rank of u_j for $j \in [n]$ is defined as

$$\mathrm{Rank}(u_j) = \sum_{i=1}^{n} \mathbb{1}_{u_i \le u_j} \ .$$

We denote $u_{(i)}$ the i-th order statistics.

2. Conformal Prediction

Conformal prediction (Vovk et al., 2005) is a framework for constructing online confidence sets, with the remarkable properties of being distribution free, having a finite sample coverage guarantee, and being able to be adapted to any estimator under mild assumptions. We recall the arguments in (Shafer & Vovk, 2008; Lei et al., 2018) to construct a conformity/typicalness function based on rank statistics that yields to distribution-free inference methods. The main tool is that the rank of one variable among an exchangeable and identically distributed sequence follows a (sub)-uniform distribution (Bröcker & Kantz, 2011).

Lemma 2.1. *Let $U_1, \ldots, U_n, U_{n+1}$ be an exchangeable and identically distributed sequence of random variables. Then for any $\alpha \in (0, 1)$, we have*

$$\mathbb{P}^{n+1}(\mathrm{Rank}(U_{n+1}) \le (n+1)(1-\alpha)) \ge 1 - \alpha \ .$$

We remind that y_{n+1} is the *unknown* target variable. We introduce a learning problem with the augmented training data $\mathcal{D}_{n+1}(z) := \mathcal{D}_n \cup \{(x_{n+1}, z)\}$ for $z \in \mathbb{R}$ and with the augmented vector of labels $y(z) = (y_1, \cdots, y_n, z)$:

$$\beta(z) \in \operatorname*{arg\,min}_{\beta\in\mathbb{R}^p} \mathcal{L}(y(z), \Phi(X), \beta)) + \Omega(\beta) \ , \qquad (1)$$

where Φ is a feature map and for any parameter $\beta \in \mathbb{R}^p$

$$\Phi(X, \beta) = [\Phi(x_1, \beta), \ldots, \Phi(x_{n+1}, \beta)] \in \mathbb{R}^{n+1} \ .$$

Given an input feature vector x, the prediction of its output/label adjusted on the augmented data, can be defined as

$$\mu_z(x) := \Phi(x, \beta(z)) \ .$$

For example in case of empirical risk minimization, we have

$$\mathcal{L}(y(z), \Phi(X, \beta)) = \sum_{i=1}^{n} \ell(y_i, \Phi(x_i, \beta)) + \ell(z, \Phi(x_{n+1}, \beta)) \ .$$

There are many examples of cost functions in the literature. A popular example is the power norm regression, where $\ell(a, b) = |a - b|^q$. When $q = 2$, this corresponds to the classical linear regression. The cases where $q = (1, 2)$ are frequent in robust statistics where the case $q = 1$ is known as the least absolute deviation. The loss `logcosh` $\ell(a, b) = \gamma \log(\cosh(a - b)/\gamma)$ is a differentiable alternative to the ℓ_∞ norm (Chebychev approximation). One can also have the loss function `Linex` (Gruber, 2010; Chang & Hung, 2007) which provides an asymmetric loss function $\ell(a, b) = \exp(\gamma(a - b)) - \gamma(a - b) - 1$, for $\gamma \neq 0$. Any convex regularization function Ω *e.g.,* Ridge (Hoerl & Kennard, 1970) or norm inducing sparsity (Bach et al., 2012) can be considered. Also the feature map Φ can be parameterized and learned *à la* neural network. Equation (1) includes many modern formulations of statistical learning estimators. The only requirement on these is to be invariant with respect to the data permutation; this leaves a very large degree of freedom on their choice. For example, $\beta(z)$ can be the output of an iterative model *e.g.,* proximal gradient descent, with early stopping.

Let us define the conformity measure for $\mathcal{D}_{n+1}(z)$ as

$$\forall i \in [n],\ E_i(z) = S(y_i, \mu_z(x_i)) \ , \qquad (2)$$
$$E_{n+1}(z) = S(z, \mu_z(x_{n+1})) \ , \qquad (3)$$

where S is a real-valued function *e.g.,* in a linear regression problem, one can take $s(a, b) = |a - b|$. The main idea for constructing a conformal confidence set is to consider the typicalness/conformity of a candidate point z measured as

$$\pi(z) := 1 - \frac{1}{n+1}\mathrm{Rank}(E_{n+1}(z)) \ . \qquad (4)$$

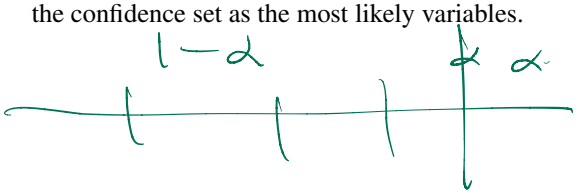
The conformal set gathers all the real values z such that $\pi(z) \ge \alpha$, if and only if, the score $E_{n+1}(z)$ is ranked no higher than $\lceil (n+1)(1-\alpha) \rceil$, among $\{E_i(z)\}_{i\in[n+1]}$ *i.e.,*

$$\Gamma^{(\alpha)}(x_{n+1}) := \{z \in \mathbb{R} : \pi(z) \ge \alpha\} \ . \qquad (5)$$

A direct application of Lemma 2.1 to $U_i = E_i(y_{n+1})$ reads $\mathbb{P}(\pi(y_{n+1}) \le \alpha) \le \alpha$ *i.e.,* the random variable $\pi(y_{n+1})$ takes small values with small probability and it reads the coverage guarantee

$$\mathbb{P}(y_{n+1} \in \Gamma^{(\alpha)}(x_{n+1})) \ge 1 - \alpha \ .$$

Figure 1 illustrates the candidates selected for inclusion in the confidence set as the most likely variables.



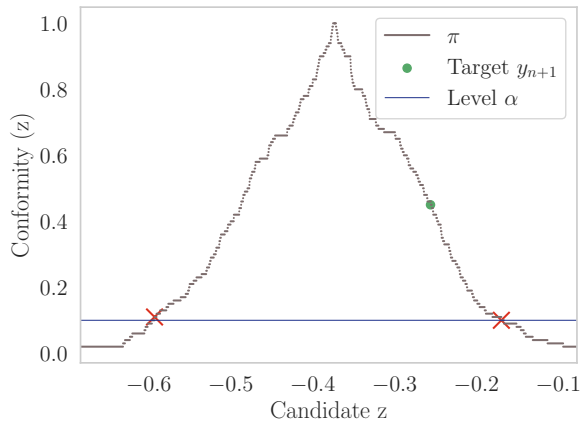


Figure 1. Illustration of a conformal prediction set with a confidence level level 0.9 *i.e.,* $\alpha = 0.1$. The ends of the CP set are indicated by the red cross.

2.1. Computational Limitations and Previous Works

For regression problems where y_{n+1} lies in a subset of $\mathbb{R}$, obtaining the conformal set $\Gamma^{(\alpha)}(x_{n+1})$ in Equation (5) is computationally challenging. It requires re-fitting the prediction model $\beta(z)$ for infinitely many candidates z in order to compute the map of conformity measure such as $z \mapsto E_i(z) = |y_i - x_i^\top \beta(z)|$. Except for a few examples, the computation of a conformal prediction set is infeasible in general. We describe below some successful computational strategies while pointing out their potential shortcomings.

In Ridge regression, for any x in $\mathbb{R}^p$, $z \mapsto x^\top \beta(z)$ is a linear function of z, implying that $E_i(z)$ is piecewise linear. Exploiting this fact, an exact conformal set $\Gamma^{(\alpha)}(x_{n+1})$ for Ridge regression was efficiently constructed in (Nouretdinov et al., 2001). Similarly, using the piecewise linearity *w.r.t.* sparsity level of the Lasso path provided by the `Lars` algorithm (Efron et al., 2004), (Hebiri, 2010) builds a sequence of conformal sets for the Lasso associated to the transition points of the `Lars` with the observed data $\mathcal{D}_n$. Nevertheless, such procedure breaks the proof technique for the coverage guarantee as the exchangeability of the sequence $(E_i(y_{n+1}))_{i\in[n+1]}$ is not necessarily maintained. However, a slight adaptation can fix the previous problem. Indeed using the piecewise linearity in z of the Lasso solution, (Lei, 2019) proposed a piecewise linear homotopy under mild assumptions, when a single input sample point is perturbed. This finally allows to compute the whole solution path $z \mapsto \beta(z)$ and successfully provides a conformal set for the Lasso and Elastic Net. These processes are however limited to quadratic loss function. Later, (Ndiaye & Takeuchi, 2019) proposed an adaptation using approximate solution path (Ndiaye et al., 2019) instead of exact solution. This results in a careful discretization of the set of candidates restricted into a preselected compact $[z_{\min}, z_{\max}]$. Assuming that the optimization problem in Equation (1) is convex and that the loss function is smooth, this leads to a computational complexity of $O(1/\sqrt{\epsilon})$ where $\epsilon > 0$ is a prescribed optimization error. All these previous methods are at best restricted to convex optimization formulations. A different road consists in merely assuming that the conformal set $\Gamma^{(\alpha)}(x_{n+1})$ in Equation (5) itself is a bounded interval. As such, its endpoints can be estimated by approximating the roots of the function $z \mapsto \pi(z) - \alpha$. Under slight additional assumptions, a direct bisection search can then compute a conformal set with a complexity of $O(\log_2(1/\epsilon_r))$ (Ndiaye & Takeuchi, 2021) where $\epsilon_r > 0$ is the tolerance error *w.r.t.* to exact root.

Cross-conformal Predictors was initially introduced in its one split version in (Papadopoulos et al., 2002).The idea is to separate the data into two independent parts, fit the model on one part and rank the scores on the other part where Lemma 2.1 remains applicable and thus preserves the coverage guarantee. Although this approach avoids the computational bottleneck by requiring only one data adjustment, the statistical efficiency of the model may be reduced due to a much smaller sample size available during the training and calibration phases. In general, the proportion of the training set to the calibration set is a hyperparameter that requires appropriate tuning: a small calibration set leads to highly variable conformational scores and a small training set leads to poor model fit. Such trade-off is very recurrent in machine learning and often appears in the debate between bias reduction and variance reduction. It is often decided by the cross-validation method with several folds (Arlot & Celisse, 2010). *Cross-conformal predictors* (Vovk, 2015) follow the same ideas and exploit the full dataset for calibration and significant proportions for training the model. The dataset is partitioned into K folds and one performs a split conformal set by sequentially defining the kth fold as calibration set and the remaining as training set for $k \in \{1, \dots, K\}$. The leave-one-out aka Jackknife CP set, requires $K = n$ model fit which is prohibitive even when n is moderately large. On the other hand, the K-fold version will require K model fit but will come at the cost of fitting on a lower sample size and will leads to an additional excess coverage of $O(\sqrt{2/n})$ and requires a subtle aggregation of the different pi-values obtained; see (Carlsson et al., 2014; Linusson et al., 2017). (Barber et al., 2021) shown that the confidence level attained is $1 - 2\alpha$ instead of $1 - \alpha$ and can only *approximately* reaches the target coverage $1 - \alpha$ under additional stability assumption.

Although these recent advances have drastically improved the tractability of the calculations, in practice multiple re-adjustments of the data are required. This remains very expensive especially for complex models. Imagine having to re-train a neural network from scratch ten or twenty times

to get a reasonable estimate. In this paper, we actually show that a single model fit is enough to tightly approximate the conformal set when the underlying model fitting is stable.

3. Approximation via Algorithmic Stability

The Section 2 guarantees that $\pi(y_{n+1}) \geq \alpha$ with high probability. Therefore, since y_{n+1} is unknown, the conformal set just selects all z that satisfies the same inequality *i.e.,* $\Gamma^{(\alpha)}(x_{n+1}) = \{z : \pi(z) \geq \alpha\}$. This leads to fitting a new model for any z. Here, we take a different strategy. The main remark is that only one element of the dataset changes at a time, then with mild stability assumptions, one can expect that the model prediction will not change drastically. Instead of inverting $\pi(\cdot)$, we will bound it with quantities independent of the model fit μ_z for any z.

Definition 3.1 (Algorithmic Stability)**.** A prediction function $\mu_\cdot$ is stable if for any observed features x_i, $i \in [n+1]$, we have

$$|S(q, \mu_z(x_i)) - S(q, \mu_{z_0}(x_i))| \leq \tau_i \quad \forall z, z_0, q \in \mathbb{R} \ . \quad (6)$$

In the literature, it is common to make assumptions about the stability of a predictive model to obtain upper bounds on its generalization error and thus ensure that it does not overfit the training data *e.g.,* (O. Bousquet & Elisseeff, 2002). Although the conformal prediction framework applies even when the underlying model is not stable, we show that this additional assumption allows for efficient evaluation of confidence sets. We also add that in cases where the generalization capabilities of the model are poor, the size of the confidence intervals can become very large; even unbounded, and not at all informative.

Proposition 3.2. *Assume that the model fit $\mu_\cdot$ is stable as in Definition 3.1. Then, we have:*

$$\forall z, \hat{z}, \quad \pi_{\text{lo}}(z, \hat{z}) \leq \pi(z) \leq \pi_{\text{up}}(z, \hat{z}) \ ,$$

with

$$\pi_{\text{lo}}(z, \hat{z}) := 1 - \frac{1}{n+1} \sum_{i=1}^{n+1} \mathbb{1}_{L_i(z,\hat{z}) \leq U_{n+1}(z,\hat{z})} \ ,$$

$$\pi_{\text{up}}(z, \hat{z}) := 1 - \frac{1}{n+1} \sum_{i=1}^{n+1} \mathbb{1}_{U_i(z,\hat{z}) \leq L_{n+1}(z,\hat{z})} \ ,$$

where, we define, for any index i in $[n]$,

$$\begin{aligned} L_i(z, \hat{z}) &= E_i(\hat{z}) - \tau_i \ , \\ U_i(z, \hat{z}) &= E_i(\hat{z}) + \tau_i \ , \\ L_{n+1}(z, \hat{z}) &= S(z, \mu_{\hat{z}}(x_{n+1})) - \tau_{n+1} \ , \\ U_{n+1}(z, \hat{z}) &= S(z, \mu_{\hat{z}}(x_{n+1})) + \tau_{n+1} \ . \end{aligned}$$

Proof. By stability, for any q, we have:

$$|S(q, \mu_z(x_i)) - S(q, \mu_{\hat{z}}(x_i))| \leq \tau_i \ .$$

Applying the previous inequality to $q = y_i$ for any index i in $[n+1]$, we have $L_i(z, \hat{z}) \leq E_i(z) \leq U_i(z, \hat{z})$ and it holds:

$$\begin{aligned} U_i(z, \hat{z}) \leq L_{n+1}(z, \hat{z}) &\Longrightarrow E_i(z) \leq E_{n+1}(z) \\ &\Longrightarrow L_i(z, \hat{z}) \leq U_{n+1}(z, \hat{z}) \ . \end{aligned}$$

Taking the indicator of the corresponding sets, we obtain the result. □

A direct consequence of Proposition 3.2 is that the exact conformal set can be wrapped as follows.

Corollary 3.3 (Stable Conformal Sets)**.** *Under the assumption of Proposition 3.2, the conformal prediction set is lower and upper approximated as*

$$\Gamma^{(\alpha)}_{\text{lo}}(x_{n+1}) \subset \Gamma^{(\alpha)}(x_{n+1}) \subset \Gamma^{(\alpha)}_{\text{up}}(x_{n+1}) \ ,$$

where

$$\begin{aligned} \Gamma^{(\alpha)}_{\text{lo}}(x_{n+1}) &= \{z : \pi_{\text{lo}}(z, \hat{z}) \geq \alpha\} \ , \\ \Gamma^{(\alpha)}_{\text{up}}(x_{n+1}) &= \{z : \pi_{\text{up}}(z, \hat{z}) \geq \alpha\} \ . \end{aligned}$$

Since our proposal arises from a combination of the conformal prediction sets with a correction from the stability bounds, we call the resulting (upper) confidence set $\Gamma^{(\alpha)}_{\text{up}}(x_{n+1})$ **`stabCP`** for stable conformal set. By construction, it contains the exact confidence set $\Gamma^{(\alpha)}(x_{n+1})$ and therefore enjoys at least the same statistical benefits displayed in the following result.

Proposition 3.4 (Coverage guarantee)**.** *Assume that the model fit $\mu_\cdot$ is stable as in Definition 3.1. Then the `stabCP` set is an upper envelope of the exact conformal prediction set in Equation (5) and is thus valid* i.e.,

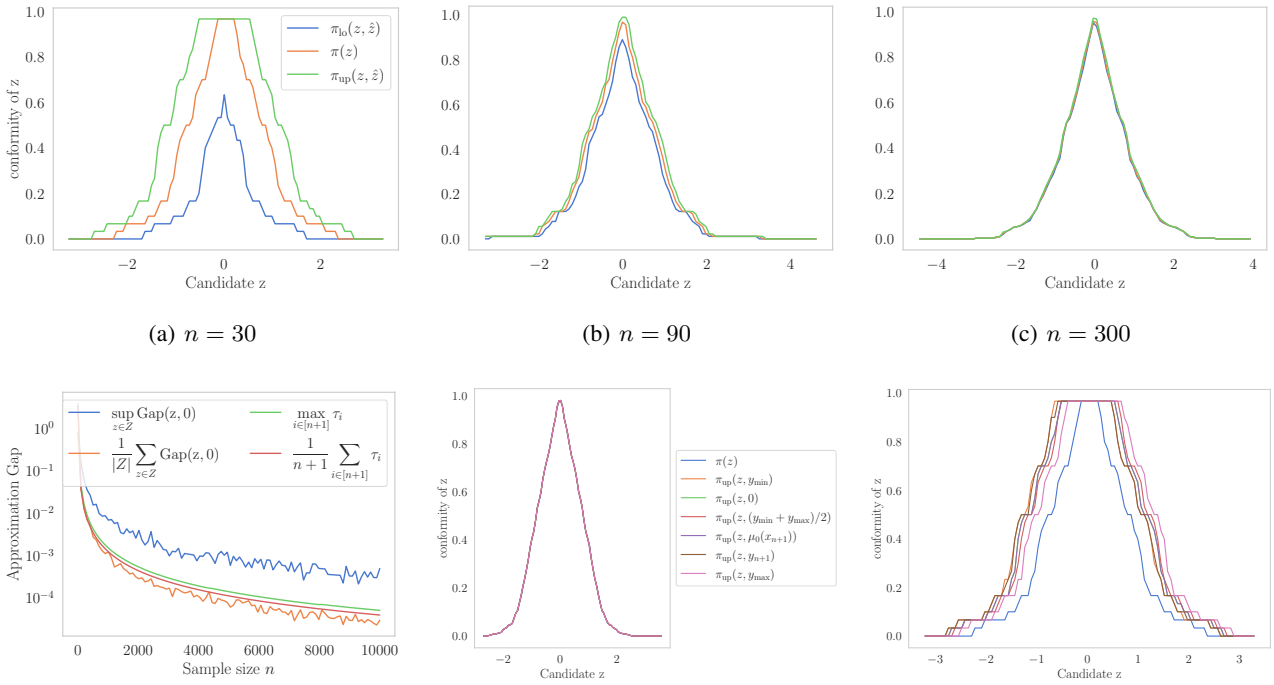
$$\mathbb{P}(y_{n+1} \in \Gamma^{(\alpha)}_{\text{up}}(x_{n+1})) \geq 1 - \alpha \ .$$

As promised in the abstract, our proposed method suffers no loss of statistical coverage, requires only one model adjustment to the data at an arbitrary candidate point $\hat{z}$, and fully uses all the data (no splitting). Thus we can benefit both from statistical efficiency with a smaller confidence interval as in the case of the exact calculation; but also we completely break the computational difficulty as in the case of splitting methods. To our knowledge, there is no equivalent method that can benefit from such a double performance.

3.1. Practical Computation of `stabCP` sets

By construction, the computation of stable conformal sets is equivalent to collecting all z such that $\pi_{\text{up}}(z, \hat{z}) \geq \alpha$. Let's begin by noting that $U_{n+1}(z, \hat{z}) > L_{n+1}(z, \hat{z})$ when $\tau_{n+1} > 0$ which we will assume for simplicity. We have

$$\pi_{\text{up}}(z, \hat{z}) \geq \alpha \Leftrightarrow \sum_{i=1}^{n} \mathbb{1}_{U_i(z,\hat{z}) \leq L_{n+1}(z,\hat{z})} \leq (1-\alpha)(n+1) \ .$$



(a) $n = 30$ (b) $n = 90$ (c) $n = 300$

(d) Convergence of the approximation gap $\text{Gap}(z,\hat{z})$ (e) `Diabetes` $(442, 10)$ (f) `Synthetic` $(30, 100)$

Figure 2. Illustration of the evolution of the conformity function as a function of sample size. The underlying model fit is $\beta(z) \in \arg\min_{\beta\in\mathbb{R}^p} \|y(z) - X\beta\|_1 / (n+1) + \lambda\|\beta\|^2$ where $y(z) = (y_1, \cdots, y_n, z)$ and we use `sklearn` synthetic dataset $\text{make\_regression(n, p} = 100, \text{noise} = 1)$. We fixed $\hat{z} = 0$ and $\lambda = 0.5$. The set Z is a linear grid in the interval $[y_{(1)}, y_{(n)}]$. We also illustrate the batch approximation for different values of $\hat{z}$.

This means that a candidate z is selected, if at most $(1-\alpha)(n+1)$ elements of $\{U_i(z,\hat{z})\}_{i\in[n]}$ are smaller than $L_{n+1}(z,\hat{z})$. Which is equivalent to[1]

$$L_{n+1}(z,\hat{z}) \leq U_{(\lceil(1-\alpha)(n+1)\rceil)}(z,\hat{z}) =: Q_{1-\alpha}(\hat{z}) \ .$$

Hence, we can conclude that

$$\Gamma^{(\alpha)}_{\text{up}}(x_{n+1}) = \{z : S(z, \mu_{\hat{z}}(x_{n+1})) \leq Q_{1-\alpha}(\hat{z}) + \tau_{n+1}\}.$$

For the absolute value score, it reduces to the interval

$$\Gamma^{(\alpha)}_{\text{up}}(x_{n+1}) = [\mu_{\hat{z}}(x_{n+1}) \pm (Q_{1-\alpha}(\hat{z}) + \tau_{n+1})] \ .$$

For the sake of clarity, we summarize the computations for this simplest case in Algorithm 1 and discuss the generalization in the appendix. In general terms, `stabCP` sets are convex sets when the score function $z \mapsto S(z, \mu_{\hat{z}}(x_{n+1}))$ has convex level sets. This presumes that our strategy will also facilitate the calculations in cases where the target y_{n+1} is multi-dimensional.

[1]For $i \in [n]$, $U_i(z,\hat{z})$ and $L_i(z,\hat{z})$ are independent of z.

Algorithm 1 Stable Conformal Prediction Set

Input: data $\{(x_1, y_1), \ldots, (x_n, y_n)\}$ and x_{n+1}
Coverage level $\alpha \in (0, 1)$, any estimate $\hat{z} \in \mathbb{R}$
Stability bounds $\tau_1, \ldots, \tau_{n+1}$ of the learning algorithm
Output: prediction interval at x_{n+1}
Fit a model $\mu_{\hat{z}}$ on the training data $\mathcal{D}_{n+1}(\hat{z})$
Compute the quantile $Q_{1-\alpha}(\hat{z}) = U_{(\lceil(1-\alpha)(n+1)\rceil)}(z,\hat{z})$ where the U_is are defined in Proposition 3.2
Return: $[\mu_{\hat{z}}(x_{n+1}) \pm (Q_{1-\alpha}(\hat{z}) + \tau_{n+1})]$

3.2. Batch Approximation

The stable conformal sets require a single model fit $\mu_{\hat{z}}$ for an arbitrary candidate $\hat{z}$. The approximation gaps are computable as

$$\max\{\pi(z) - \pi_{\text{lo}}(z,\hat{z}), \pi_{\text{up}}(z,\hat{z}) - \pi(z)\} \leq \text{Gap}(z,\hat{z}) \ ,$$

where

$$\text{Gap}(z,\hat{z}) := \pi_{\text{up}}(z,\hat{z}) - \pi_{\text{lo}}(z,\hat{z}) \ .$$

Since the above upper and lower bounds hold for any $\hat{z}$, tighter approximations are obtained with a batch of candi-

dates $\mathcal{Z} = \hat{z}_1, \cdots, \hat{z}_d$ as

$$\pi_{\text{up}}(z, \mathcal{Z}) = \inf_{\hat{z} \in \mathcal{Z}} \pi_{\text{up}}(z, \hat{z}) \text{ and } \pi_{\text{lo}}(z, \mathcal{Z}) = \sup_{\hat{z} \in \mathcal{Z}} \pi_{\text{lo}}(z, \hat{z}) \ .$$

Another possibility is to build an interpolation of $z \mapsto \mu_z(\cdot)$ based on query points $\hat{z}_1, \cdots, \hat{z}_d \in (z_{\min}, z_{\max}) \subset \mathbb{R}$. For example, one can consider as predictive model the following piecewise linear interpolation

$$\tilde{\mu}_z = \begin{cases} \frac{\hat{z}_1 - z}{\hat{z}_1 - z_{\min}} \mu_{z_{\min}} + \frac{z_{\min} - z}{\hat{z}_1 - z_{\min}} \mu_{\hat{z}_1} & \text{if } z \leq z_{\min} \ , \\ \frac{z - \hat{z}_{t+1}}{\hat{z}_t - \hat{z}_{t+1}} \mu_{\hat{z}_t} + \frac{z - \hat{z}_t}{\hat{z}_{t+1} - \hat{z}_t} \mu_{\hat{z}_{t+1}} & \text{if } z \in [\hat{z}_t, \hat{z}_{t+1}] \ , \\ \frac{z - \hat{z}_d}{z_{\max} - \hat{z}_d} \mu_{z_{\max}} + \frac{z_{\max} - z}{z_{\max} - \hat{z}_d} \mu_{\hat{z}_d} & \text{if } z \geq z_{\max} \ , \end{cases}$$

An important point is that, by using the stability bound, the coverage guarantee of the interpolated conformal set is preserved without the need of the expensive symmetrization proposed in (Ndiaye & Takeuchi, 2021). Such techniques are more relevant when the sample size is small or when precise estimates of the stability bounds are not available. The corresponding conformity function is defined in a similar way as the previous versions, where we simply plugin the interpolated model. We refer to the appendix for more details.

Remark 3.5 (Categorical Variables). In this article, we have essentially limited ourselves to regression problems which, in general, pose intractable computational difficulties. However, the methods remain applicable for classification problems where the set of candidates can only take a finite number of values in $\mathcal{C} := c_1, \ldots, c_m$. In this case, an additional precaution of encoding the categories in real numbers is necessary. Considering the leave-one-out score function, our proposal is therefore an alternative to the approximations via influence function used in (Alaa & Schaar, 2020; Abad et al., 2022) when an exact computation (Cherubin et al., 2021) would be unusable or too costly.

3.3. Stability Bounds

In this section, we recall some stability bounds. The proof techniques rely on regularity assumptions on the function to be minimized and are relatively standard in optimization (Shalev-Shwartz & Ben-David, 2014, Chapter 13). Stability is a widely used assumption to provide generalization bounds for machine learning algorithms (O. Bousquet & Elisseeff, 2002; Hardt et al., 2016). We specify that here the notion of stability that we require is related to the variation of the score and not of the loss function in the optimization objective. However, the ideas for establishing the stability bounds are essentially the same and we recall the core strategies here for the sake of completeness.

Let us start with the unregularized model where $\Omega = 0$ *i.e.,*

$$\beta(z) \in \underset{\beta \in \mathbb{R}^p}{\arg\min} \mathcal{L}(y(z), \Phi(X, \beta)) = F_z(\Phi(X, \beta)) \ . \quad (7)$$

Definition 3.6. A function f is λ-strongly convex if for any w_0, w and $\varsigma \in (0, 1)$

$$f(\varsigma w_0 + (1 - \varsigma) w) \leq \varsigma f(w_0) + (1 - \varsigma) f(w) - \frac{\lambda}{2} \varsigma (1 - \varsigma) \left\| w_0 - w \right\|^2 \ .$$

Proposition 3.7. *Assume that for any z, F_z is λ-strongly convex and ρ-Lipschitz. It holds*

$$\left\| \mu_z(X) - \mu_{z_0}(X) \right\| \leq \frac{2\rho}{\lambda} \ .$$

Proof. By optimality of $\beta(z)$, we have

$$F_z(\Phi(X, \beta(z))) \leq F_z(\Phi(X, \beta)) \quad \forall \beta \ . \quad (8)$$

We simply apply the optimality condition and strong convexity of the function F_z to the vectors $w_0 = \Phi(X, \beta(z_0)) = \mu_{z_0}(X)$ and $w = \Phi(X, \beta(z)) = \mu_z(X)$, it holds

$$\begin{aligned} 0 &\overset{(8)}{\leq} \frac{F_z(\varsigma w_0 + (1 - \varsigma) w) - F_z(w)}{\varsigma} \\ &\overset{(3.6)}{\leq} F_z(w_0) - F_z(w) - \frac{\lambda}{2} (1 - \varsigma) \left\| w_0 - w \right\|^2 \ . \end{aligned}$$

Since F_z is ρ-Lipschitz, we have

$$\frac{\lambda}{2} \left\| w_0 - w \right\|^2 \leq F_z(w_0) - F_z(w) \leq \rho \left\| w - w_0 \right\| \ .$$

Therefore, $\frac{\lambda}{2} \left\| w_0 - w \right\| \leq \rho$, hence the result. □

The Proposition 3.7 does not assume that the optimization problem in Equation (7) is convex in the model parameter β. We can now easily deduce a stability bound according to the Definition 3.1.

Corollary 3.8. *If the score function $S(q, \cdot)$ is γ-Lipschitz for any q, then*

$$\tau_i = \frac{2\gamma\rho}{\lambda}, \qquad \forall i \in [n + 1] \ .$$

When the loss function is not strongly convex, it is known that adding a strongly convex regularization can stabilize the algorithm (Shalev-Shwartz & Ben-David, 2014, Chapter 13). The proof technique is similar to the previous one with the difference that now the bound is on the $\arg\min$ of the optimization problem and not the predictions of the model. This requires stronger assumptions.

Proposition 3.9. *Assume the optimization problem Equation (1) is convex, Ω is λ-strongly convex. If the loss $\mathcal{L}$ is convex-ρ-Lipschitz, then*

$$\left\| \beta(z) - \beta(z_0) \right\| \leq \frac{2\rho}{\lambda} \ .$$

When the loss function $\mathcal{L}$ is convex-ν-smooth with $\nu < \lambda$ and $\mathcal{L}(y(z), \mu_z(X)) \leq C$ for any z, then

$$\left\| \beta(z) - \beta(z_0) \right\| \leq \frac{2\sqrt{2\nu C}}{\lambda - \nu} \ .$$

These optimization error bounds also imply the following stability bounds.

Corollary 3.10. *Assume that the score function $S(q, \cdot)$ is γ-Lipschitz for any q, and that the prediction model $\mu_\cdot(x) := \Phi(x, \beta(\cdot))$ satisfies for any $x \in \mathbb{R}^p$, $z, z_0 \in \mathbb{R}$,*

$$|\mu_z(x) - \mu_{z_0}(x)| \leq L_\Phi |x^\top \beta(z) - x^\top \beta(z_0)| \ .$$

If the loss is ρ-Lipschitz, then

$$\tau_i = \frac{2\gamma\rho L_\Phi \|x_i\|}{\lambda}, \qquad \forall i \in [n+1] \ .$$

If the loss is ν-smooth with $\nu < \lambda$ and bounded by C, then

$$\tau_i = \frac{2\gamma L_\Phi \|x_i\| \sqrt{2\nu C}}{\lambda - \nu}, \qquad \forall i \in [n+1] \ .$$

Another way to understand such regularized bounds, is to leverage duality. A smoothness assumption in the primal space will translate into a strongly concave assumption in the dual space (Hiriart-Urruty & Lemaréchal, 1993, Theorem 4.2.2, p. 83). The dual formulation (Rockafellar, 1997, Chapter 31) of Equation (1) reads:

$$\theta(z) \in \operatorname*{arg\,max}_{\theta \in \mathbb{R}^{n+1}} -\mathcal{L}^*(y(z), -\theta) - \Omega^*(X^\top \theta) \ , \qquad (9)$$

where, given a proper, closed and convex function $f : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$, we denoted its Fenchel-Legendre transform as $f^* : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$ defined by $f^*(x^*) = \sup_{x \in \operatorname{dom} f} \langle x^*, x \rangle - f(x)$ with $\operatorname{dom} f = \{x \in \mathbb{R}^n : f(x) < +\infty\}$.

Let P_z and D_z denote the primal and dual objective functions. We have the following classical error bounds for the dual optimization problem. If the loss function $\mathcal{L}$ is ν-smooth, then $\mathcal{L}^*$ is $1/\nu$-strongly convex and we have for $\forall (\beta, \theta) \in \operatorname{dom} P_z \times \operatorname{dom} D_z$

$$\begin{aligned} \frac{1}{2\nu} \|\theta(z) - \theta\|^2 &\leq D_z(\theta(z)) - D_z(\theta) \\ &= P_z(\beta(z)) - D_z(\theta) \\ &\leq \text{Duality\_Gap}_z(\beta, \theta) \ , \end{aligned}$$

where the equality follows from strong duality and we recall from weak duality that the duality gap upper bounds the optimization error as follow:

$$\begin{aligned} \text{Duality\_Gap}_z(\beta, \theta) &:= P_z(\beta) - D_z(\theta) \\ &\geq P_z(\beta) - P_z(\beta(z)) \ . \end{aligned}$$

This readily leads to several possible bounds. If the dual function $D_z(\cdot)$ is ρ^*-Lipschitz for any z, then

$$\|\theta(z) - \theta\| \leq \sqrt{2\nu\rho^*} \ .$$

If the duality gap can be assumed to be bounded by C for any $z \in [z_{\min}, z_{\max}]$, then

$$\|\theta(z) - \theta\| \leq \sqrt{2\nu C} \ .$$

We obtain stability bounds when one uses the dual solution (which is a function of the residual) as a conformity score $S(y(z), \mu_z(X)) = |\theta(z)|$ where the absolute value is taken coordinate wise. For example, these dual based score functions were used in (Ndiaye & Takeuchi, 2019).

Remark 3.11 (Bound on the loss). The assumption of a bounded loss function that we make, is not rigorously feasible and some adaptations are necessary. For simplicity, let us consider that $\Phi(x, 0) = 0$ and $\Omega(0) = 0$. Using the optimality of $\beta(z)$, we obtain for any candidate z

$$\begin{aligned} \mathcal{L}(y(z), \mu_z(X)) &\leq \mathcal{L}(y(z), \mu_z(X)) + \Omega(\beta(z)) \\ &\leq \mathcal{L}(y(z), 0) \ . \end{aligned}$$

Unfortunately, for common examples such as least squares, the right hand side is unbounded. Nevertheless, since the data are assumed to be exchangeable, we have

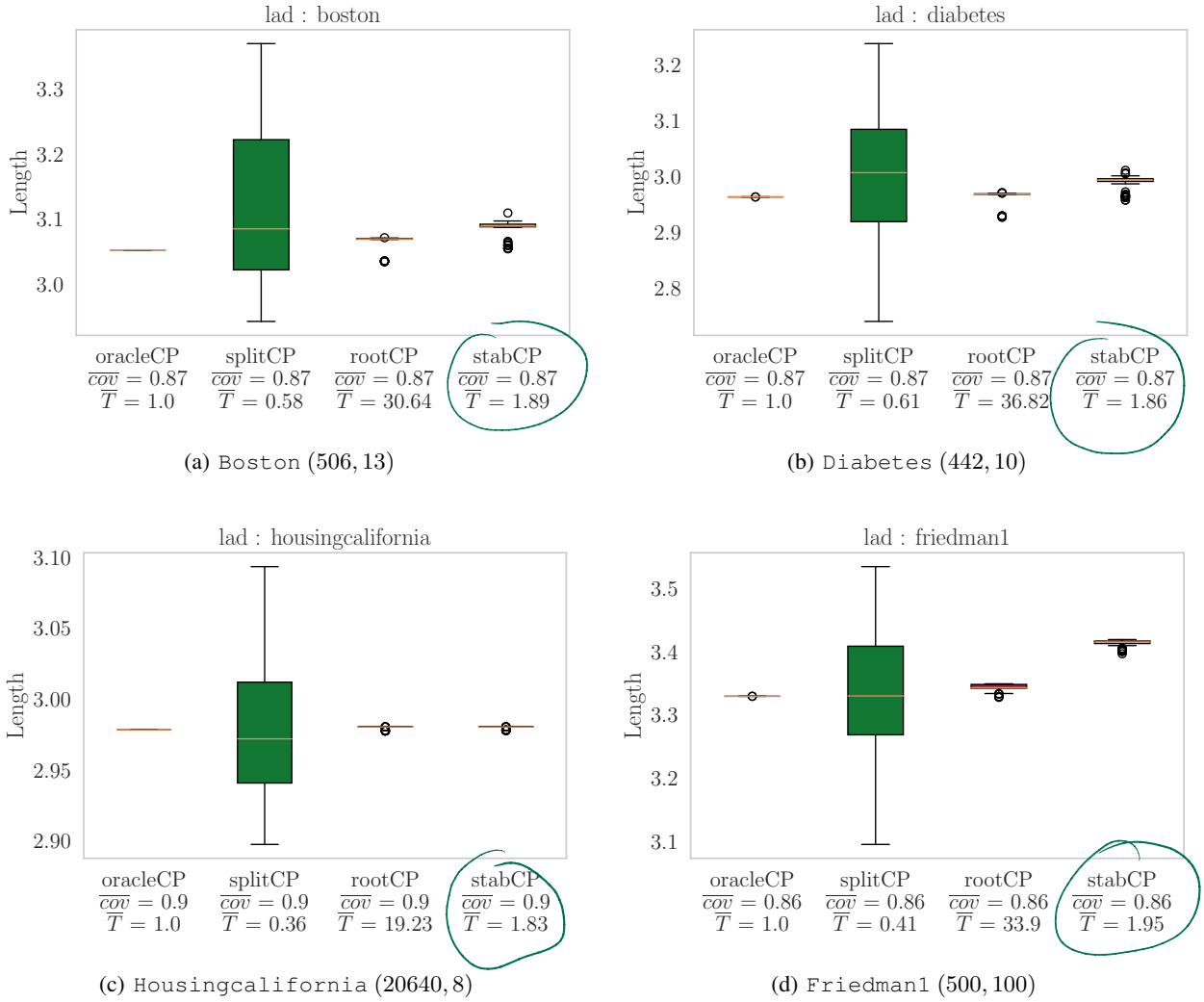
$$\mathbb{P}(y_{n+1} \in [y_{(1)}, y_{(n)}]) \geq 1 - \frac{2}{n+1} \ .$$

Hence it is reasonable to restrict the range of candidates as $z \in [y_{(1)}, y_{(n)}]$, which implies

$$\mathcal{L}(y(z), \mu_z(X)) \leq \sup_{z \in [y_{(1)}, y_{(n)}]} \mathcal{L}(y(z), 0) =: C \ .$$

4. Numerical Experiments

We conduct all the experiments with a coverage level of 0.9 *i.e.,* $\alpha = 0.1$. For comparisons, we run the evaluations on 100 repetitions of examples and display the average of the following performance statistics for different methods: the empirical coverage *i.e.,* the percentage of times the prediction set contains the held-out target y_{n+1}, the length of the confidence intervals, and the execution time. We compare the method we propose `stabCP` with the conformal prediction set computed with an oracle method defined below, with a splitting strategy `splitCP` (Papadopoulos et al., 2002; Lei et al., 2018), and finally with an estimation of the α-level set of the conformity function `rootCP` (Ndiaye & Takeuchi, 2021) by root-finding solvers. Note that, when the conformal set is a bounded interval, `stabCP` approximates `rootCP` as in Figure 2. In all experiments conducted, we observed that the exact conformal prediction set is indeed an interval. Although this is often the case, we recall that it might not be in general. Just for the comparisons, we therefore estimated the `stabCP` sets with a root-finding solver as well, as if a closed form solution was not available. A `python` package with our implementation is available at `https://github.com/EugeneNdiaye/stable_conformal_prediction` where additional numerical experiments (*e.g.,* using large pre-trained neural net) and benchmarks will be provided.



(a) `Boston` $(506, 13)$ (b) `Diabetes` $(442, 10)$

(c) `Housingcalifornia` $(20640, 8)$ (d) `Friedman1` $(500, 100)$

Figure 3. Benchmarking conformal sets for the least absolute deviation regression models with a ridge regularization on real datasets. We display the lengths of the confidence sets over 100 random permutation of the data. We denoted $\overline{cov}$ the average coverage and $\overline{T}$ the average computational time normalized with the average time for computing `oracleCP` which requires a single full data model fit. The full and exact CP set can always be approximated with a fine (costly) grid discretization of the output space and can then be used as a default baseline. Here, it is represented by rootCP since in the examples displayed the full CP set turns out to be an interval and then rootCP is equal to the full CP up to ϵ_r digit precision on the decimals; we used a default value of $\epsilon_r = 10^{-4}$.

`oracleCP`. To define an oracle prediction set as reference, we follow in (Ndiaye & Takeuchi, 2019; 2021) and assume that the unavailable target variable y_{n+1} is observed by the algorithm. Hence, we define the oracle scores

$$\forall i \in [n], \quad E_i^{\text{or}} = S(y_i,\, \mu_{y_{n+1}}(x_i)) \ , \\ E_{n+1}^{\text{or}}(z) = S(z,\, \mu_{y_{n+1}}(x_{n+1})) \ ,$$

and the oracle conformal set as

$$\Gamma^{(\alpha)}_{\text{oracle}}(x_{n+1}) := \{z : \pi_{\text{oracle}}(z) \geq \alpha\} \ , \\ \pi_{\text{oracle}}(z) = 1 - \frac{1}{n+1} \sum_{i=1}^{n+1} \mathbb{1}_{E_i^{\text{or}} \leq E_{n+1}^{\text{or}}(z)} \ .$$

`splitCP`. A popular and classical estimation of conformal prediction sets relies on splitting the dataset. The split conformal prediction set introduced in (Papadopoulos et al., 2002), separates the model fitting and the calibration steps. Let us define

- the training set

$$\mathcal{D}_{\text{tr}} = \{(x_1, y_1), \cdots, (x_m, y_m)\} \text{ with } m < n \ ,$$

- the calibration set

$$\mathcal{D}_{\text{cal}} = \{(x_{m+1}, y_{m+1}), \cdots, (x_n, y_n)\} \ .$$

Then the model is fitted on the training set $\mathcal{D}_{\text{tr}}$ to get $\mu_{\text{tr}}(\cdot)$ and define the score function on the calibration set $\mathcal{D}_{\text{cal}}$:

$$\forall i \in [m+1, n], \quad E_i^{\text{cal}} = S(y_i,\, \mu_{\text{tr}}(x_i)) \ , \\ E_{n+1}^{\text{cal}}(z) = S(z,\, \mu_{\text{tr}}(x_{n+1})) \ .$$

Thus, we obtain the split conformal set as

$$\Gamma^{(\alpha)}_{\text{split}}(x_{n+1}) = \{z : \pi_{\text{split}}(z) \geq \alpha\} \ , \\ \pi_{\text{split}}(z) = 1 - \frac{1}{n-m+1} \sum_{i=m+1}^{n+1} \mathbb{1}_{E_i^{\text{cal}} \leq E_{n+1}^{\text{cal}}(z)} \ .$$

5. Discussion

The data splitting approach does not use all the data in the training phase. It is often less statistically efficient, and its interval length can vary greatly depending on the additional randomness of the split. On the contrary, our approach does not use any splitting, provides an approximation of the exact conformal set that is pretty accurate depending on the stability of the model as can be observed on Figure 3. All this requires one and only one data fitting of the underlying learning model. You will notice that `splitCP` and `stabCP` have the same structure and are simple intervals if the score functions are reasonably simple. The presence of data splitting in the former is replaced by an additional stability term in the latter. So if the predictive model is very stable, `stabCP` benefits from all the data, and very little regularization to get closer to the oracle version that includes the unknown target y_{n+1}. To date, we are not aware of any other method that can obtain a full conformal prediction set with such computational efficiency while ensuring no loss on the coverage guarantee. We observe on the benchmarks with real data Figure 3 that the `stabCP` is often very similar to the `rootCP` which approximates with a very fine precision the exact set (under the assumption that the latter is a bounded interval). Our proposal has the net advantage of being twenty to thirty times faster and can often be computed in closed form.

However, as can be seen in Figure 2, our proposed method loses precision when the sample size is small. This reflects the difficulty of estimating a reliable confidence set in the absence of algorithmic stability. At the same time, it is difficult to have an algorithm that generalizes well with so little training data. Otherwise, when the size of the data is important, the influence of the stability bound is very little felt because they are often of the order of magnitude $O(1/n)$.

Finally, a notorious limitation is that one needs to know explicitly the stability bounds. This can be difficult to estimate for some models. The bounds we presented in Section 3.3 cover a wide range of examples and can be completed by bounds displayed in (Hardt et al., 2016; Bassily et al., 2020; Lei et al., 2021; Klochkov & Zhivotovskiy, 2021) for stochastic gradient descent. Even if the notion of stability required here is slightly different, any error bound on the estimator can be naturally converted into a stability bound for conformal prediction sets. So we don't lose much generality as long as we make the assumption that the score function is sufficiently regular *e.g.,* Lipschitz. This is precisely what allowed us to obtain the bounds presented in this article. Yet, if the parameter of the predictive model is defined iteratively by a gradient descent process on a non-convex objective function, obtaining stability bounds becomes quite delicate. Moreover, the Lipschitz constant of neural network objectives can be poorly estimated. In this case, our approach could not be applied safely or could lead to uninformative confidence intervals. The splitting strategy remains more flexible. It would be interesting to study fine combinations of data splitting and inclusion of stability bounds to reduce the size of the confidence intervals and their variance while being pivotal to explicit stability bounds.

Acknowledgements

We warmly thank the reviewers for their insightful comments and contributions to improve the presentation of this paper. We also thank Elvis Dohmatob and Xiaoming Huo for proofreading and for pointing out mistakes in notations.

References

Abad, J., Bhatt, U., Weller, A., and Cherubin, G. Approximating full conformal prediction at scale via influence functions. *arXiv preprint arXiv:2202.01315*, 2022.

Alaa, A. and Schaar, M. V. D. Discriminative jackknife: Quantifying uncertainty in deep learning via higher-order influence functions. *International Conference on Machine Learning*, 2020.

Arlot, S. and Celisse, A. A survey of cross-validation procedures for model selection. *Statistics surveys*, 2010.

Bach, F., Jenatton, R., Mairal, J., and Obozinski, G. Optimization with sparsity-inducing penalties. *Foundations and Trends in Machine Learning*, 2012.

Balasubramanian, V., Ho, S.-S., and Vovk, V. *Conformal prediction for reliable machine learning: theory, adaptations and applications*. Elsevier, 2014.

Barber, R. F., Candes, E. J., Ramdas, A., and Tibshirani, R. J. Predictive inference with the jackknife+. *The Annals of Statistics*, 2021.

Bassily, R., Feldman, V., Guzmán, C., and Talwar, K. Stability of stochastic gradient descent on nonsmooth convex losses. *Advances in Neural Information Processing Systems*, 2020.

Bates, S., Candès, E., Lei, L., Romano, Y., and Sesia, M. Testing for outliers with conformal p-values. *arXiv preprint arXiv:2104.08279*, 2021.

Bröcker, J. and Kantz, H. The concept of exchangeability in ensemble forecasting. *Nonlinear Processes in Geophysics*, 2011.

Carlsson, L., Eklund, M., and Norinder, U. Aggregated conformal prediction. *IFIP International Conference on Artificial Intelligence Applications and Innovations*, 2014.

Cella, L. and Ryan, R. Valid distribution-free inferential models for prediction. *arXiv preprint arXiv:2001.09225*, 2020.

Chang, Y.-C. and Hung, W.-L. Linex loss functions with applications to determining the optimum process parameters. *Quality & Quantity*, 2007.

Chernozhukov, V., Wüthrich, K., and Zhu, Y. Exact and robust conformal inference methods for predictive machine learning with dependent data. *Conference On Learning Theory*, 2018.

Chernozhukov, V., Wüthrich, K., and Zhu, Y. An exact and robust conformal inference method for counterfactual and synthetic controls. *Journal of the American Statistical Association*, 2021.

Cherubin, G., Chatzikokolakis, K., and Jaggi, M. Exact optimization of conformal predictors via incremental and decremental learning. *International Conference on Machine Learning*, 2021.

Efron, B. Resampling plans and the estimation of prediction error. *Stats*, 2021.

Efron, B., Hastie, T., Johnstone, I. M., and Tibshirani, R. Least angle regression. *The Annals of Statistics*, 2004.

Fisch, A., Schuster, T., Jaakkola, T., and Barzilay, R. Few-shot conformal prediction with auxiliary tasks. *ICML*, 2021.

Gruber, M. *Regression estimators: A comparative study*. JHU Press, 2010.

Hardt, M., Recht, B., and Singer, Y. Train faster, generalize better: Stability of stochastic gradient descent. *International Conference on Machine Learning*, 2016.

Hebiri, M. Sparse conformal predictors. *Statistics and Computing*, 2010.

Hiriart-Urruty, J.-B. and Lemaréchal, C. *Convex analysis and minimization algorithms. II*. Springer-Verlag, 1993.

Ho, S.-S. and Wechsler, H. Query by transduction. *IEEE transactions on pattern analysis and machine intelligence*, 2008.

Hoerl, A. E. and Kennard, R. W. Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics*, 1970.

Holland, M. J. Making learning more transparent using conformalized performance prediction. *arXiv preprint arXiv:2007.04486*, 2020.

Johnstone, C. and Cox, B. Conformal uncertainty sets for robust optimization. *Conformal and Probabilistic Prediction and Applications*, 2021.

Klochkov, Y. and Zhivotovskiy, N. Stability and deviation optimal risk bounds with convergence rate $o(1/n)$. *Advances in Neural Information Processing Systems*, 2021.

Laxhammar, R. and Falkman, G. Inductive conformal anomaly detection for sequential detection of anomalous sub-trajectories. *Annals of Mathematics and Artificial Intelligence*, 2015.

Lei, J. Fast exact conformalization of lasso using piecewise linear homotopy. *Biometrika*, 2019.

Lei, J., GSell, M., Rinaldo, A., Tibshirani, R. J., and Wasserman, L. Distribution-free predictive inference for regression. *Journal of the American Statistical Association*, 2018.

Lei, Y., Yang, Z., Yang, T., and Ying, Y. Stability and generalization of stochastic gradient methods for minimax problems. 2021.

Linusson, H., Norinder, U., Boström, H., Johansson, U., and Löfström, T. On the calibration of aggregated conformal predictors. *Conformal and probabilistic prediction and applications*, 2017.

Ndiaye, E. and Takeuchi, I. Computing full conformal prediction set with approximate homotopy. *NeurIPS*, 2019.

Ndiaye, E. and Takeuchi, I. Root-finding approaches for computing conformal prediction set. *arXiv preprint arXiv:2104.06648*, 2021.

Ndiaye, E., Le, T., Fercoq, O., Salmon, J., and Takeuchi, I. Safe grid search with optimal complexity. *ICML*, 2019.

Nouretdinov, I., Melluish, T., and Vovk, V. Ridge regression confidence machine. *ICML*, 2001.

O. Bousquet, O. and Elisseeff, A. Stability and generalization. *The Journal of Machine Learning Research*, 2002.

Papadopoulos, H., Proedrou, K., Vovk, V., and Gammerman, A. Inductive confidence machines for regression. *European Conference on Machine Learning*, 2002.

Rockafellar, R. T. *Convex analysis*. Princeton University Press, 1997. Reprint of the 1970 original, Princeton Paperbacks.

Shafer, G. and Vovk, V. A tutorial on conformal prediction. *Journal of Machine Learning Research*, 2008.

Shalev-Shwartz, S. and Ben-David, S. *Understanding machine learning: From theory to algorithms*. Cambridge university press, 2014.

Vovk, V. Cross-conformal predictors. *Annals of Mathematics and Artificial Intelligence*, 2015.

Vovk, V., Gammerman, A., and Shafer, G. *Algorithmic learning in a random world*. Springer, 2005.

Xu, C. and Xie, Y. Conformal prediction interval for dynamic time-series. *ICML*, 2021.

6. Appendix

In these supplementary notes, we complete some proofs and bring algorithmic precisions of our approach as well as additional numerical experiments.

6.1. StabCP Set with General Score Function

We explain a simple procedure to approximate the set prediction with an arbitrary pre-defined accuracy. We recall that

$$\begin{aligned}\Gamma^{(\alpha)}_{\mathrm{up}}(x_{n+1}) &= \{z : \pi_{\mathrm{up}}(z, \hat{z}) \geq \alpha\} \\ &= \{z : S(z, \mu_{\hat{z}}(x_{n+1})) \leq Q_{1-\alpha}(\hat{z}) + \tau_{n+1}\} \ ,\end{aligned}$$

which is a convex set when the level-set of the score function is convex. By simplicity, we assume that the score function is such that $\Gamma^{(\alpha)}_{\mathrm{up}}(x_{n+1})$ is a bounded interval. Algorithm 2 summarizes the process.

Algorithm 2 Stable conformal prediction set for score function with convex level-set

Input: data $\{(x_1, y_1), \ldots, (x_n, y_n)\}$ and x_{n+1}
Coverage level $\alpha \in (0, 1)$, any estimate $\hat{z} \in \mathbb{R}$
Stability bounds $\tau_1, \ldots, \tau_{n+1}$ of the learning algorithm
Output: prediction interval at x_{n+1}
Fit a model $\mu_{\hat{z}}$ on the training data $\mathcal{D}_{n+1}(\hat{z})$
Compute the quantile $Q_{1-\alpha}(\hat{z}) = U_{(\lceil (1-\alpha)(n+1) \rceil)}(z, \hat{z})$ where the U_is are defined in Proposition 3.2
Compute $\Gamma^{(\alpha)}_{\mathrm{up}}(x_{n+1}) = [\ell_\alpha(x_{n+1}), u_\alpha(x_{n+1})]$ up to $\epsilon_r > 0$ tolerance error as follow:

1. find $z_{\min} < z_0 < z_{\max}$ such that

$$\pi_{\mathrm{up}}(z_{\min}, \hat{z}) < \alpha < \pi_{\mathrm{up}}(z_0, \hat{z}) \text{ and } \alpha > \pi_{\mathrm{up}}(z_{\max}, \hat{z}) \ . \tag{10}$$

2. Perform a bisection search in $[z_{\min}, z_0]$. It will output a point $\hat{\ell}$ such that $\ell_\alpha(x_{n+1})$ belongs to $[\hat{\ell} \pm \epsilon_r]$ after at most $\log_2(\frac{z_0 - z_{\min}}{\epsilon_r})$ iterations.

3. Perform a bisection search in $[z_0, z_{\max}]$. It will output a point $\hat{u}$ such that $u_\alpha(x_{n+1})$ belongs to $[\hat{u} \pm \epsilon_r]$ after at most $\log_2(\frac{z_{\max} - z_0}{\epsilon_r})$ iterations.

Return: $\Gamma^{(\alpha)}_{\mathrm{up}}(x_{n+1})$

6.2. Stability of the Linear Interpolation

We discussed in Section 3.2 the potential gain in accuracy when approximating the conformity function using not a single point but a batch of points. Here we justify the interpolation approach when the score function S is sufficiently regular.

Proposition 6.1. *Let us assume that the score function $S(q, \cdot)$ is γ-Lipschitz for any q, and consider the interpolated prediction model defined as*

$$\tilde{\mu}_z = \begin{cases} \frac{\hat{z}_1 - z}{\hat{z}_1 - z_{\min}} \mu_{z_{\min}} + \frac{z_{\min} - z}{\hat{z}_1 - z_{\min}} \mu_{\hat{z}_1} & \text{if } z \leq z_{\min} \ , \\ \frac{z - \hat{z}_{t+1}}{\hat{z}_t - \hat{z}_{t+1}} \mu_{\hat{z}_t} + \frac{z - \hat{z}_t}{\hat{z}_{t+1} - \hat{z}_t} \mu_{\hat{z}_{t+1}} & \text{if } z \in [\hat{z}_t, \hat{z}_{t+1}] \ , \\ \frac{z - \hat{z}_d}{z_{\max} - \hat{z}_d} \mu_{z_{\max}} + \frac{z_{\max} - z}{z_{\max} - \hat{z}_d} \mu_{\hat{z}_d} & \text{if } z \geq z_{\max} \ , \end{cases} \tag{11}$$

where $\mu_\cdot$ is stable according to Definition 3.1. It holds

$$|S(q, \tilde{\mu}_z(x_i)) - S(q, \tilde{\mu}_{z_0}(x_i))| \leq 3\gamma\tau_i \ . \tag{12}$$

Proof. Using the triangle inequality, we have

$$\begin{aligned}\tilde{\mathrm{stab}} &:= |S(q, \tilde{\mu}_z(x_i)) - S(q, \tilde{\mu}_{z_0}(x_i))| \\ &\leq |S(q, \tilde{\mu}_z(x_i)) - S(q, \mu_z(x_i))| + |S(q, \mu_z(x_i)) - S(q, \mu_{z_0}(x_i))| + |S(q, \mu_{z_0}(x_i)) - S(q, \tilde{\mu}_{z_0}(x_i))| \ .\end{aligned}$$

If $\mu_.$ is stable, then the second term of the right hand side of the previous inequality is bounded by τ_i. Now, assuming that S is γ-Lipschitz in its second argument, for any q, we have:

$$|S(q, \tilde{\mu}_z(x_i)) - S(q, \mu_z(x_i))| \leq \gamma \mathcal{E}_z^i \ ,$$

where

$$\begin{aligned}
\mathcal{E}_z^i &= |\mu_z(x_i) - \tilde{\mu}_z(x_i)| \\
&\leq |\mu_z(x_i) - \alpha_t \mu_{z_t}(x_i) - (1-\alpha_t)\mu_{z_{t+1}}(x_i)| \\
&\leq \alpha_t |\mu_z(x_i) - \mu_{z_t}(x_i)| + (1-\alpha_t)|\mu_z(x_i) - \mu_{z_{t+1}}(x_i)| \\
&\leq \alpha_t \tau_i + (1-\alpha_t)\tau_i = \tau_i \ ,
\end{aligned}$$

with $\alpha_t \in \left\{ \frac{z_1 - z}{z_1 - z_{\min}}, \frac{z - z_{t+1}}{z_t - z_{t+1}}, \frac{z - z_d}{z_{\max} - z_d} \right\}$ is the scaling of interpolation points. Thus, we obtain

$$\tilde{\text{stab}} \leq \gamma(\mathcal{E}_z^i + \tau_i + \mathcal{E}_{z_0}^i) \leq 3\gamma\tau_i \ .$$

□

The upper and lower approximation of the conformity function obtained with the interpolated model fit along with stability bounds are defined as:

$$\begin{aligned}
\tilde{\pi}_{\text{lo}}(z) &= 1 - \frac{1}{n+1} \sum_{i=1}^{n+1} \mathbb{1}_{\tilde{L}_i(z) \leq \tilde{U}_{n+1}(z)} \ , \\
\tilde{\pi}_{\text{up}}(z) &= 1 - \frac{1}{n+1} \sum_{i=1}^{n+1} \mathbb{1}_{\tilde{U}_i(z) \leq \tilde{L}_{n+1}(z)} \ ,
\end{aligned}$$

where for any index i in $[n+1]$, using the stability bound in Equation (12), we define

$$\begin{aligned}
&\tilde{L}_i(z) = \tilde{E}_i(z) - 3\gamma\tau_i \text{ and } \tilde{U}_i(z) = \tilde{E}_i(z) + 3\gamma\tau_i \ , \\
&\tilde{E}_i(z) = S(y_i, \tilde{\mu}_z(x_i)) \text{ and } \tilde{E}_{n+1}(z) = S(z, \tilde{\mu}_z(x_i)) \ .
\end{aligned}$$

In general, approximating the entire model path with respect to output/label changes using finite grid points is not always safe for calculating the conformal prediction set because it breaks the exchangeability assumptions of the data set. Incorporating the stability bound will regularize the conformity function to restore the validity of the method. However, the procedure proves to be quite robust to wrong estimation of the stability bounds. The experiments in (Ndiaye & Takeuchi, 2021) are conducted with estimates $\tau_i = 0$ and the prediction sets obtained are essentially the same as the exact one. More detailed experiments will be proposed in our github implementation.

6.3. Additional Experiments

In this appendix, we add some numerical experiments to illustrate how `stabCP` can behave when using an estimator that is not defined as an `argmin` but rather as an output of an iterative process. In this case, we use a Multi-Layer Perceptron regressor trained with $T =$ `n_iter` number of gradient descent iterations. Recent analyses (Hardt et al., 2016) have shown that any model trained with the stochastic gradient method in a reasonable amount of time achieves low generalization error. The proof of these results consists in showing that the estimator verifies a stability condition when the input data are slightly perturbed. The bounds on the iterates of stochastic gradient methods are often proportional to $\frac{T}{n}$. They also depend on the Lipschitz regularity constants which unfortunately can be hard to estimate in practice. Here, we will be satisfied with the order of magnitude and evaluate the behavior of the conformity function according to the number of iterations performed. We run the experiments on two different datasets with a sample size of 442 and 20640.

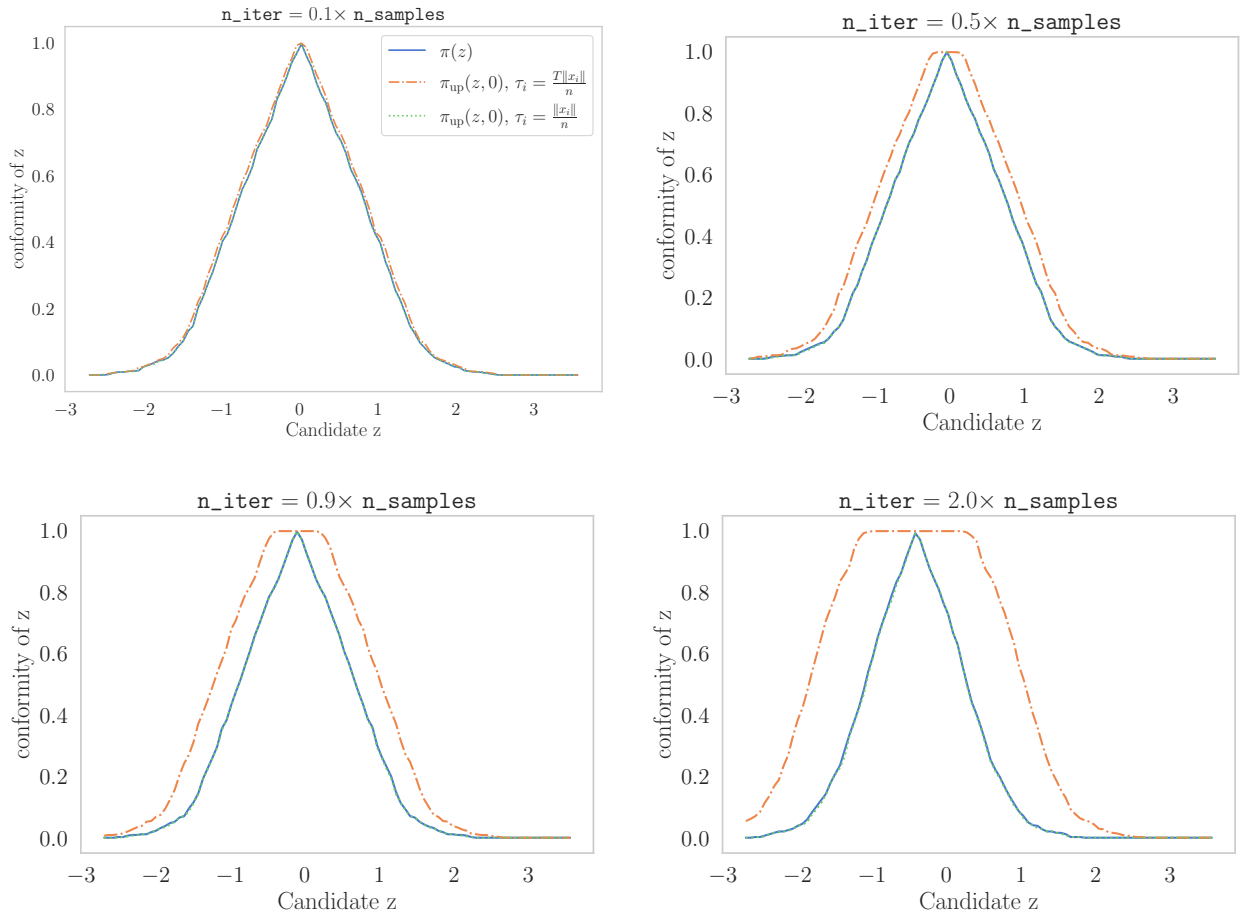


Figure 4. Illustration of different conformity functions with respect to a sequence of stability bounds. We observe that by merely staking an order of magnitude $O(1/n)$ as stability bound, gives a good estimate of the conformal prediction set even if the bound is not safe. These experiments are conducted with a Multi-Layer Perceptron regressor on the `Diabetes` $(442, 10)$ dataset, trained with $T =$ `n_iter` iterations of Stochastic Gradient Descent.

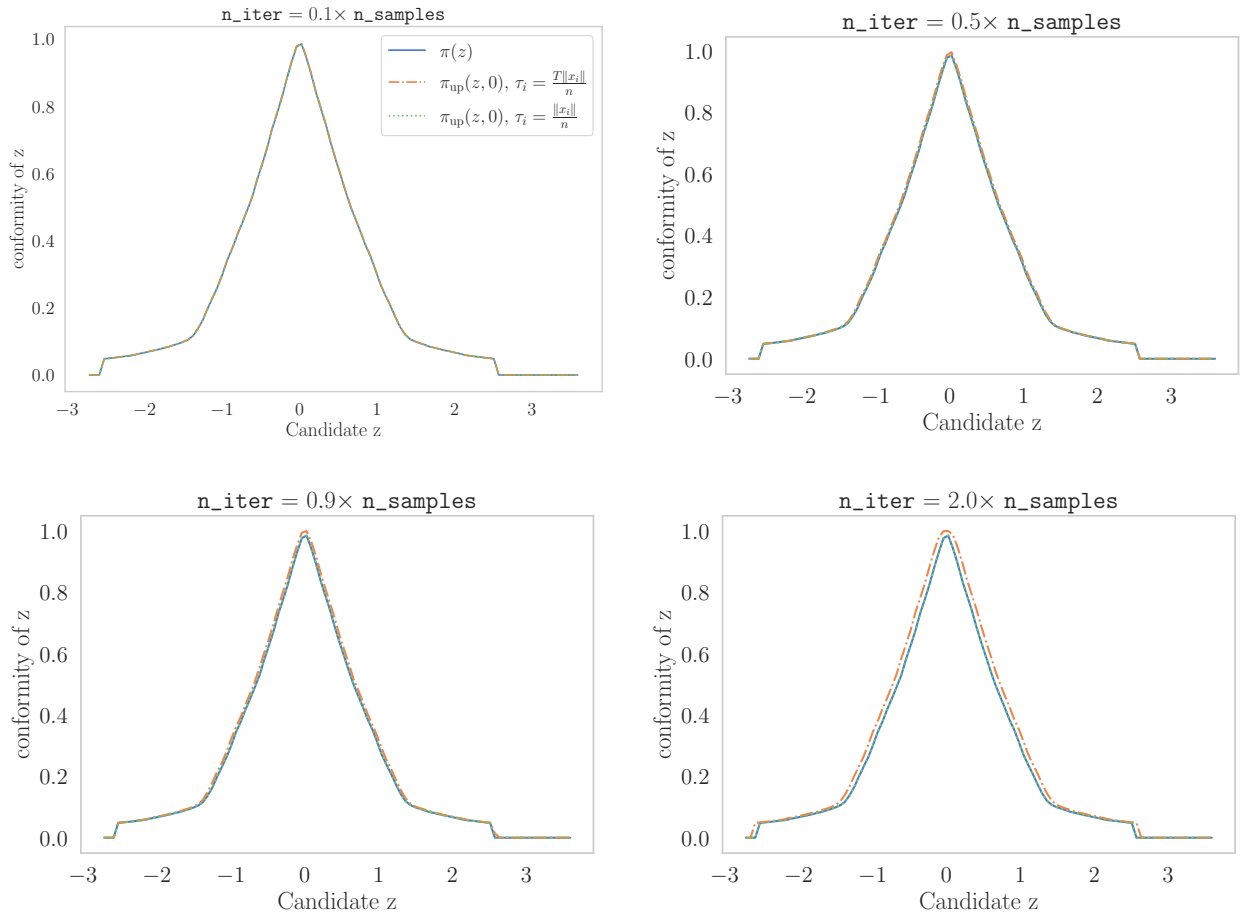
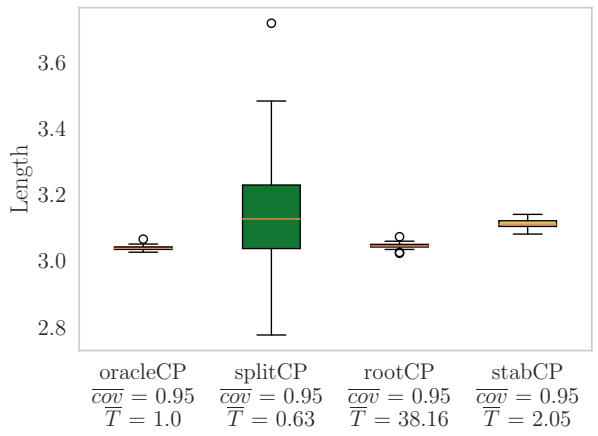
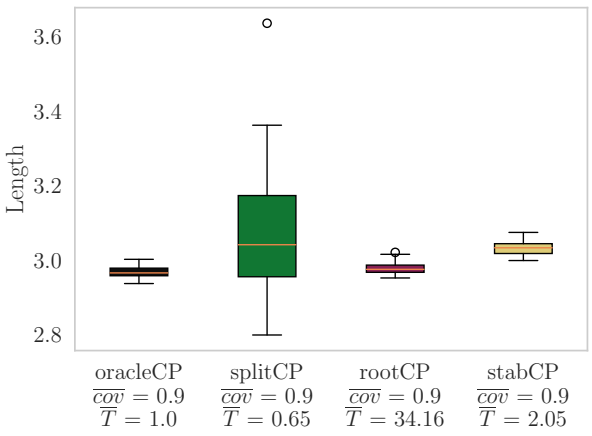


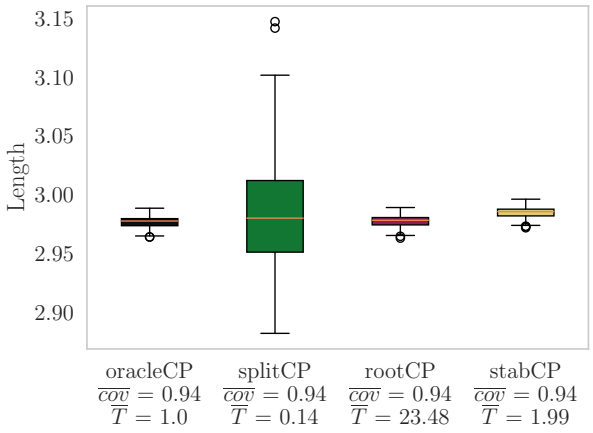
Figure 5. Illustration of different conformity functions with respect to a sequence of stability bounds. We observe that by merely staking an order of magnitude $O(1/n)$ as stability bound, gives a good estimate of the conformal prediction set even if the bound is not safe. These experiments are conducted with a Multi-Layer Perceptron regressor on the `Housingcalifornia` $(20640, 8)$ dataset, trained with $T =$ `n_iter` iterations of Stochastic Gradient Descent.



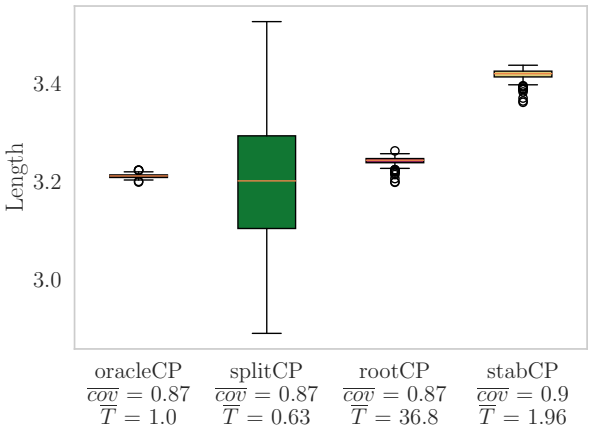
(a) `Boston` $(506, 13)$



(b) `Diabetes` $(442, 10)$

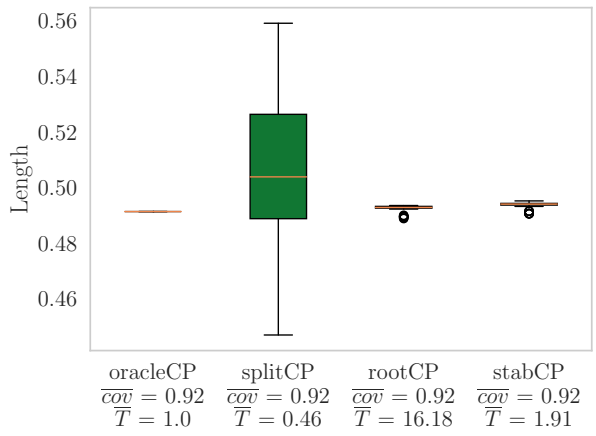


(c) `Housingcalifornia` $(20640, 8)$

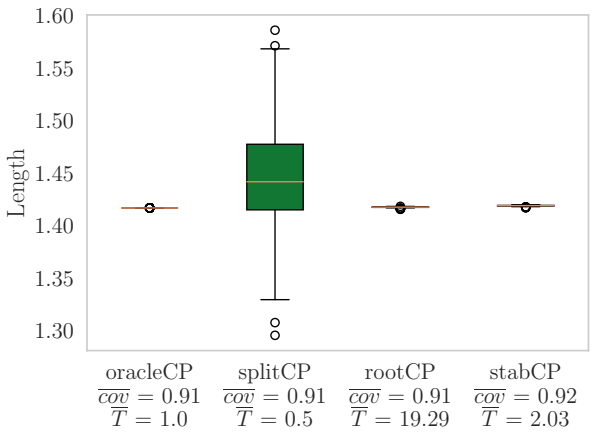


(d) `Friedman1` $(500, 100)$

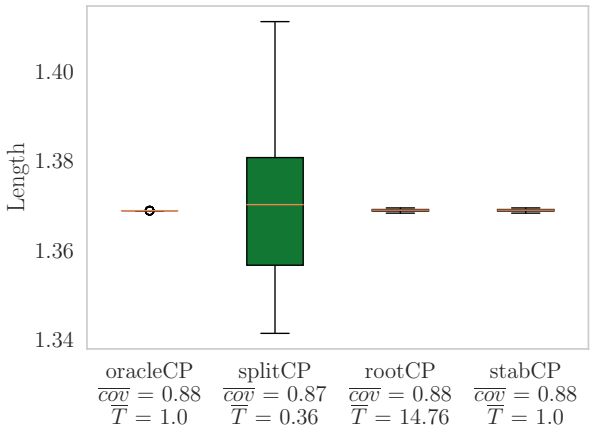
Figure 6. Benchmarking conformal sets for MLP regression models with a ridge regularization on real datasets. The parameter of the model is obtained after $T = n/10$ iterations of stochastic gradient descent. For `stabCP`, we use a stability bound estimate $\tau_i = T \|x_i\| / (n+1)$. We display the lengths of the confidence sets over 100 random permutation of the data. We denoted $\overline{cov}$ the average coverage and $\overline{T}$ the average computational time normalized with the average time for computing `oracleCP` which requires a single full data model fit.



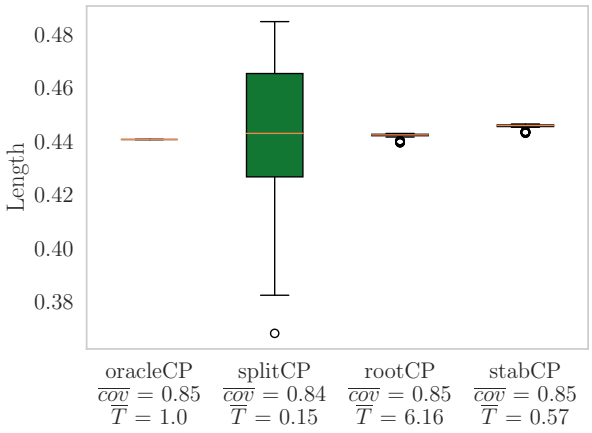
(a) `Boston` $(506, 13)$



(b) `Diabetes` $(442, 10)$

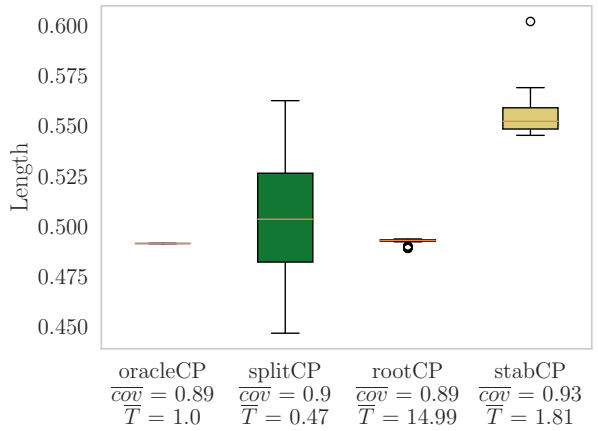


(c) `Housingcalifornia` $(20640, 8)$

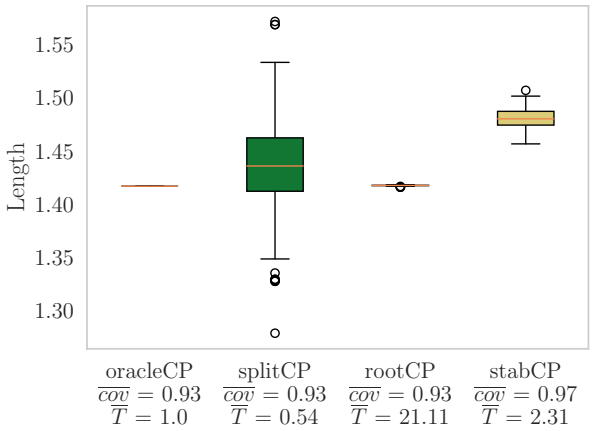


(d) `Friedman1` $(500, 100)$

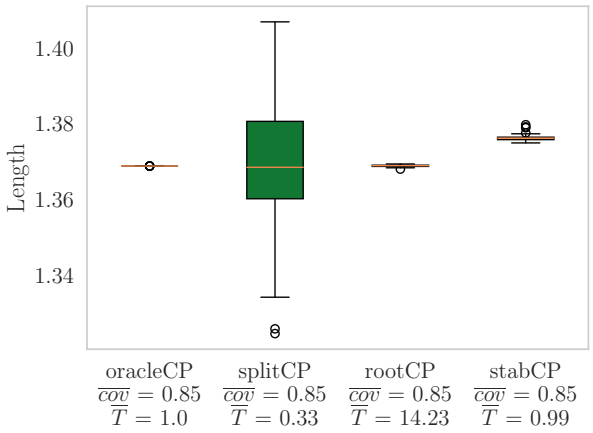
Figure 7. Benchmarking conformal sets for Gradient Boosting regression models with a ridge regularization on real datasets. For `stabCP`, we use a stability bound estimate $\tau_i = \|x_i\| / (n+1)$. We display the lengths of the confidence sets over 100 random permutation of the data. We denoted $\overline{cov}$ the average coverage and $\overline{T}$ the average computational time normalized with the average time for computing `oracleCP` which requires a single full data model fit.



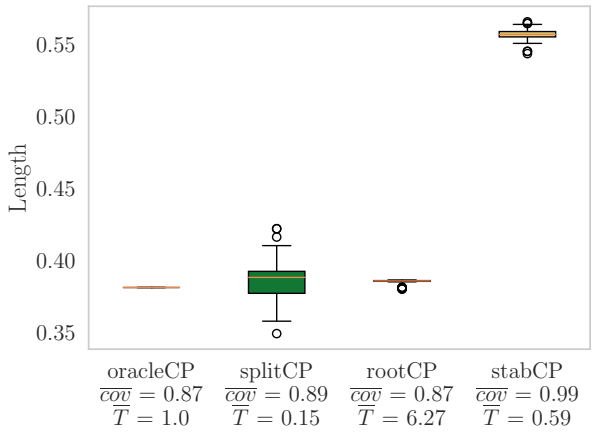
(a) `Boston` $(506, 13)$



(b) `Diabetes` $(442, 10)$



(c) `Housingcalifornia` $(20640, 8)$



(d) `Friedman1` $(500, 100)$

Figure 8. Benchmarking conformal sets for Gradient Boosting regression models with a ridge regularization on real datasets. For `stabCP`, we use a rough stability bound estimate $\tau_i \approx \|x_i\|/10$. We display the lengths of the confidence sets over 100 random permutation of the data. We denoted $\overline{cov}$ the average coverage and $\overline{T}$ the average computational time normalized with the average time for computing `oracleCP` which requires a single full data model fit. This example shows that for unstable models such as decision trees, a coarse estimation of the stability bound can result in an overestimation of the confidence interval, which is a notable limitation of the proposed method.