Assurances for machine learning trajectory predictors: guaranteed probabilistic bounds with conformal prediction

PhD Student

Aloysio GALVÃO LOPES

Advisors

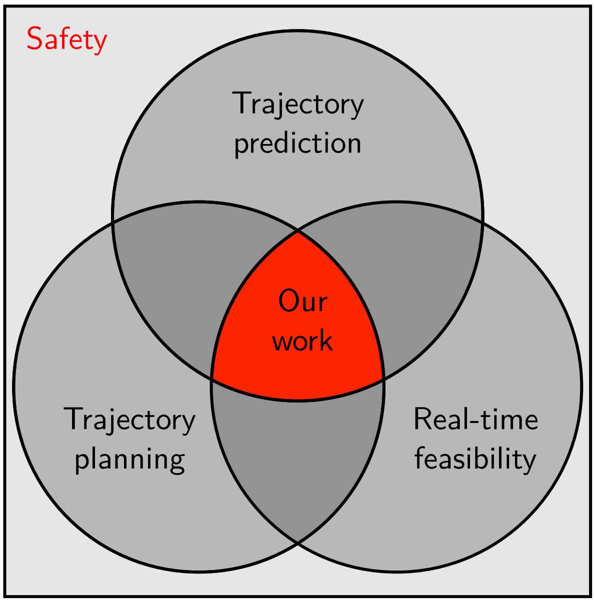
Eric GOUBAULT
Laurent PAUTET
Sylvie PUTOT

LIX - École Polytechnique, LTCI - Télécom Paris

{galvaolopes,goubault,putot}@lix.polytechnique.fr
laurent.pautet@telecom-paris.fr

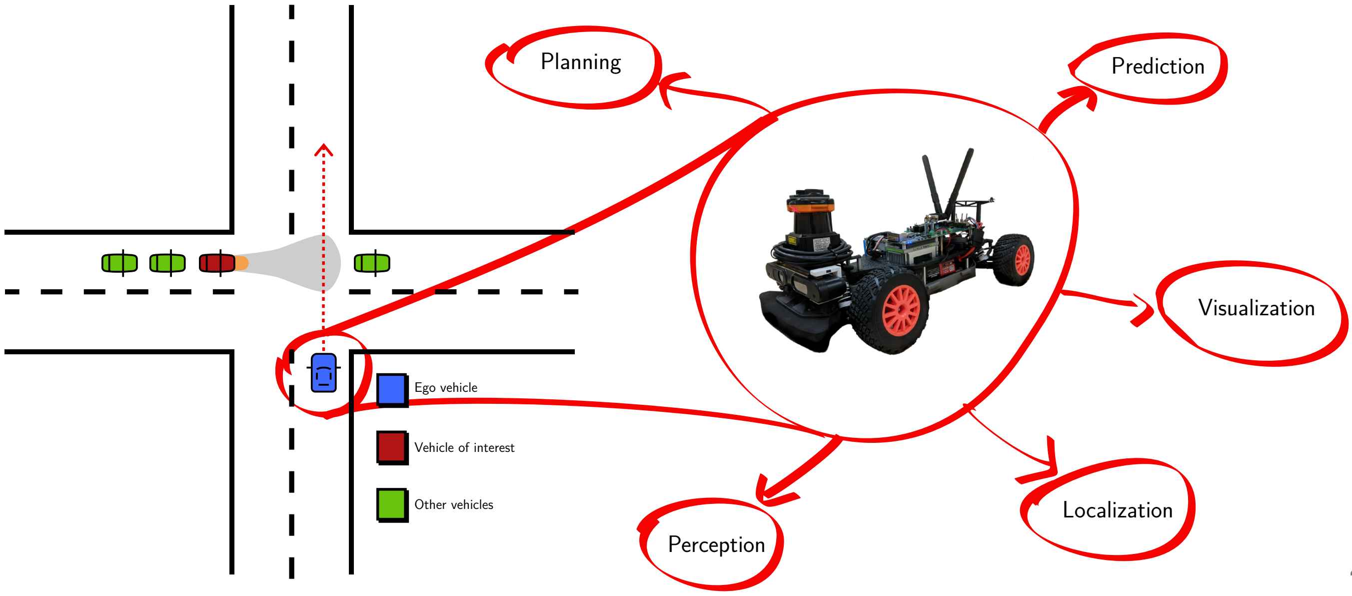
- This is mainly going to be an introductory talk in conformal prediction.
- I will try to show you that it's a very simple yet powerful method.
- I will introduce it in the context of my work.
- I will also give you a glimpse of some results.

What do I do?


Safety
Trajectory prediction
Our work
Trajectory planning
Real-time feasibility

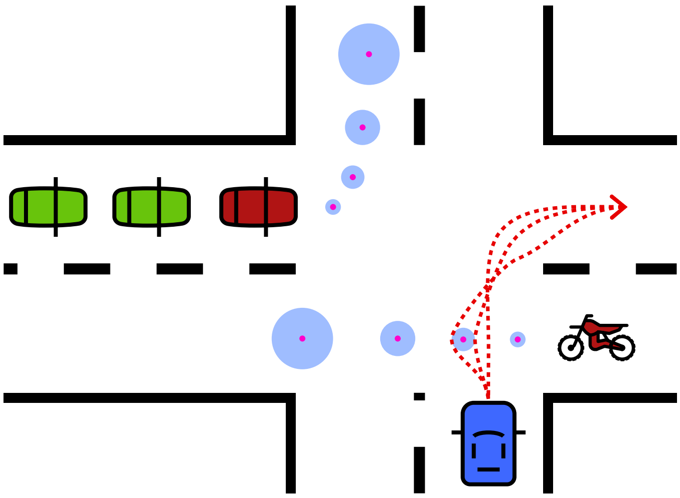
What do I do?

I develop planning algorithms taking **probabilistic motion predictions** of other traffic participants. These algorithms should be able to guarantee **safety**, while being **real-time feasible**.



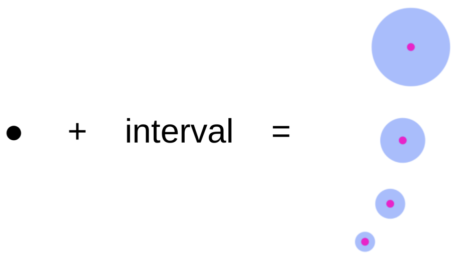
For this talk

Given trajectory predictions in magenta, we want to compute valid prediction regions in blue, for a given desired coverage probability $1-\alpha$ (probability true trajectory is inside the blue regions).

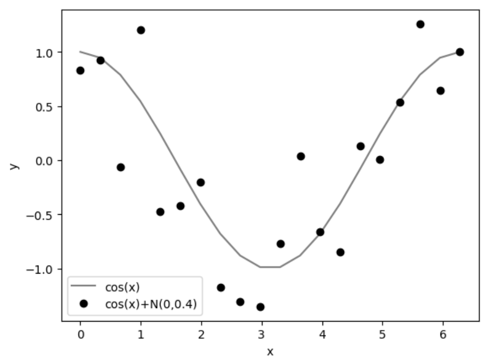


Let's consider a simpler problem

Given a dataset of 1-D points $\mathcal{D} = \{(x_1, y_1), \ldots, (x_n, y_n)\}$, we want to compute valid prediction regions for a new point $\hat{y}_{n+1} = f(x_{n+1})$, for a given desired coverage probability $1 - \alpha = 90\%$.



$$y = \cos(x) + \mathcal{N}(0, 0.4)$$



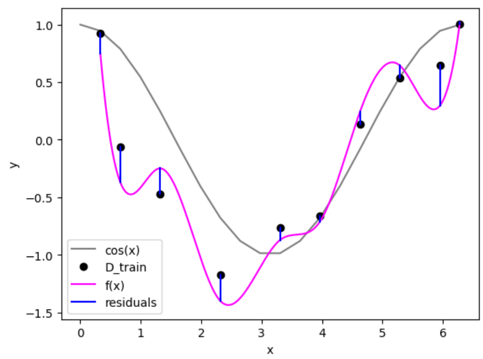
Let's consider a simpler problem

Given a dataset of 1-D points $\mathcal{D} = \{(x_1, y_1), \ldots, (x_n, y_n)\}$, we want to compute valid prediction regions for a new point $\hat{y}_{n+1} = f(x_{n+1})$, for a given desired coverage probability $1 - \alpha = 90\%$.

We can train f using a subset of $\mathcal{D}$, $\mathcal{D}_{train} \subset \mathcal{D}$.

How can we choose a band q, such that:

$$\mathbb{P}(y_{n+1} \in [f(x_{n+1}) - q, f(x_{n+1}) + q]) \geq 90\%$$



Let's consider a simpler problem

Given a dataset of 1-D points $\mathcal{D} = \{(x_1, y_1), \ldots, (x_n, y_n)\}$, we want to compute valid prediction regions for a new point $\hat{y}_{n+1} = f(x_{n+1})$, for a given desired coverage probability $1 - \alpha = 90\%$.

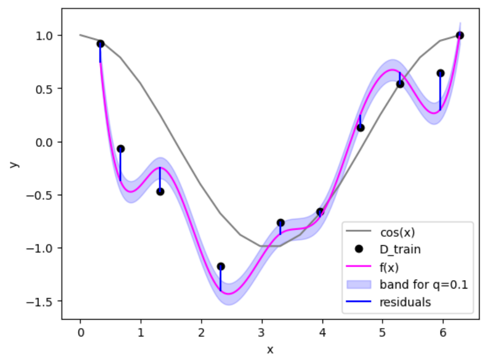
Given the residuals defined as:

$$r_i = |y_i - f(x_i)|$$

The problem is equivalent to finding a band q, such that:

$$\mathbb{P}(r_{n+1} \leq q) \geq 90\%$$

One possible solution is to use the 90% empirical quantile of the residuals.



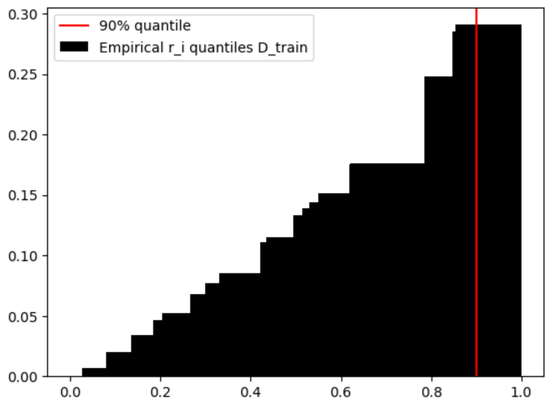
Brief recap on quantiles

Given a distribution F the level β quantile is defined as follows, for $Z \sim F$:

$$Quantile(\beta, F) = \inf\{z : \mathbb{P}(Z \leq z) \geq \beta\}$$

For an empirical distribution X (such as the residuals r_i on the dataset D_{train}) it can be defined as:

$$Quantile(\beta, X) = Quantile(\beta, \frac{1}{n}\sum_{i=1}^{n} \delta_{x_i})$$



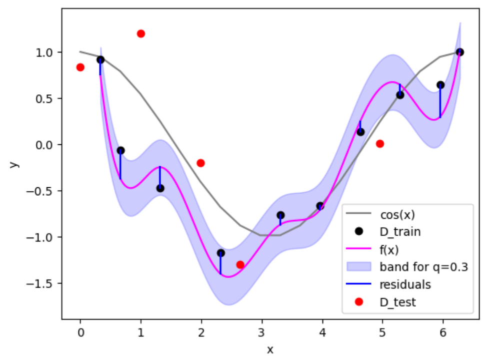
Let's consider a simpler problem

Given a dataset of 1-D points $\mathcal{D} = \{(x_1, y_1), \ldots, (x_n, y_n)\}$, we want to compute valid prediction regions for a new point $\hat{y}_{n+1} = f(x_{n+1})$, for a given desired coverage probability $1 - \alpha = 90\%$.

Given the residuals defined as:

$$r_i = |y_i - f(x_i)|$$

Compute the $q = 90\%$ empirical quantile of the residuals, for the points in $\mathcal{D}_{train}$, and take $[f(x_{n+1}) - q, f(x_{n+1}) + q]$.



Let's consider a simpler problem

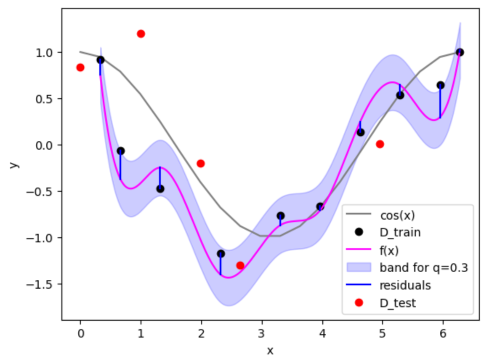
Given a dataset of 1-D points $\mathcal{D} = \{(x_1, y_1), \ldots, (x_n, y_n)\}$, we want to compute valid prediction regions for a new point $\hat{y}_{n+1} = f(x_{n+1})$, for a given desired coverage probability $1 - \alpha = 90\%$.

Given the residuals defined as:

$$r_i = |y_i - f(x_i)|$$

Compute the $q = 90\%$ empirical quantile of the residuals, for the points in $\mathcal{D}_{train}$, and take $[f(x_{n+1}) - q, f(x_{n+1}) + q]$.

Only 40% coverage on a test dataset $\mathcal{D}_{test}$ disjoint with $\mathcal{D}_{train}$(only a sample of the test points is shown in the image)!

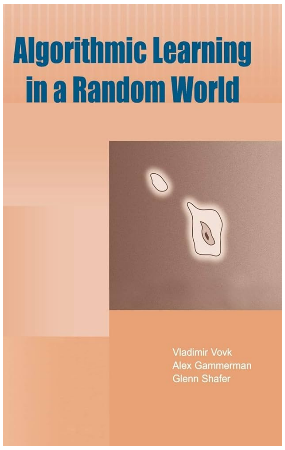


A little bit of history

Vladimir Vovk

A little bit of history

Vladimir Vovk

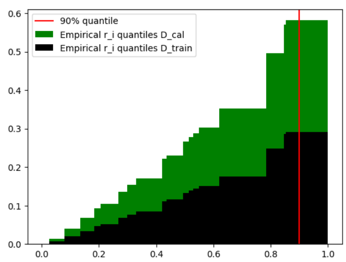


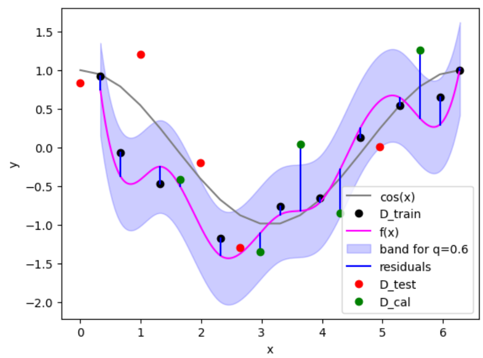
Algorithmic learning in a Random World

Let's consider a simpler problem

Given a dataset of 1-D points $\mathcal{D} = \{(x_1, y_1), \ldots, (x_n, y_n)\}$, we want to compute valid prediction regions for a new point $\hat{y}_{n+1} = f(x_{n+1})$, for a given desired coverage probability $1 - \alpha = 90\%$.

Compute the $q = 90\%$ empirical quantile of the residuals, for the points in $\mathcal{D}_{cal}$, **not used for the training of f!** Then take $[f(x_{n+1}) - q, f(x_{n+1}) + q]$.

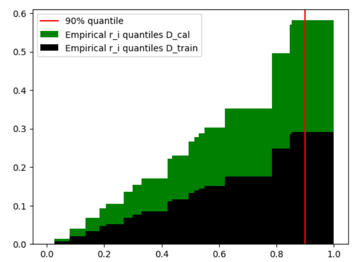


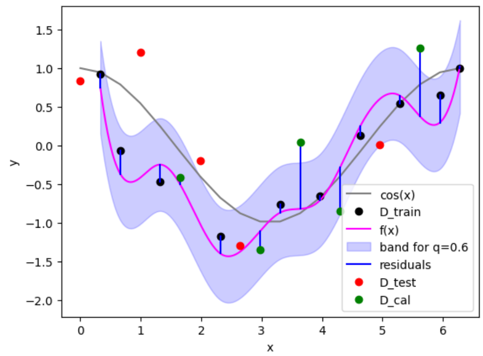


Let's consider a simpler problem

Given a dataset of 1-D points $\mathcal{D} = \{(x_1, y_1), \ldots, (x_n, y_n)\}$, we want to compute valid prediction regions for a new point $\hat{y}_{n+1} = f(x_{n+1})$, for a given desired coverage probability $1 - \alpha = 90\%$.

Compute the $q = 90\%$ empirical quantile of the residuals, for the points in $\mathcal{D}_{cal}$, **not used for the training of** f**!** Then take $[f(x_{n+1}) - q, f(x_{n+1}) + q]$.

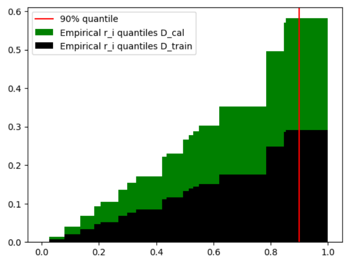


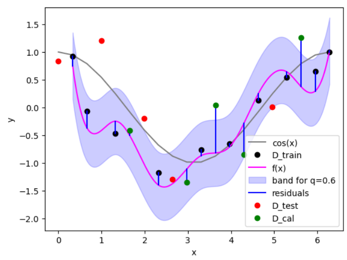


It works!

Split conformal prediction

- Compute the residuals $r_i = |f(x_i) - y_i|$ using the pairs (x_i, y_i) in the calibration dataset $\mathcal{D}_{cal}$.
- Sort the residuals in ascending order: $r_{(1)} \leq r_{(2)} \leq \cdots \leq r_{(n)}$.
- Given a max error probability of α, select the $q_{1-\alpha} = \lceil (n+1)(1-\alpha) \rceil$-th residual in ascending order.
- The prediction set is the set of labels y such that $|f(x) - y| \leq q_{1-\alpha}$. More explicitly $[f(x) - q_{1-\alpha}, f(x) + q_{1-\alpha}]$.





Split conformal prediction

- Take any measurable score function $s(x, y)$ (residual was $r = |f(x) - y|$, with f the predictor trained in $\mathcal{D}_{train}$).
- Compute the $1 - \alpha$ quantile of the scores on the calibration dataset ($Quantile(1 - \alpha, \mathcal{D}_{cal})$).

$$\hat{C}(x) = \{y \text{ s.t } s(x, y) \leq Quantile(1 - \alpha, \mathcal{D}_{cal})\}$$

Theorem [Vovk, Gammerman, Shafer 2005]

$$\mathbb{P}\left\{Y \in \hat{C}(X)\right\} \geq 1 - \alpha$$

Holds as long as the new data (X, Y) is exchangeable with the calibration dataset $\mathcal{D}_{cal}$.

Split conformal prediction - Proof

Given a sequence of random variables:

$$R_1, R_2, \ldots, R_n, R_{n+1}, \ldots$$

Suppose that any permutation is equally likely. That is the sequence is exchangeable, more formally:

$$\mathbb{P}\left\{R_1 \leq r_1, R_{n+1} \leq r_{n+1}, \ldots\right\} = \mathbb{P}\left\{R_{\pi(1)} \leq r_1, R_{\pi(n+1)} \leq r_{n+1}, \ldots\right\}$$

For all permutations, π and all r_i.

Split conformal prediction - Proof

This means that R_{n+1} is equally likely to be among the k smallest values among $R_1, \dots, R_{n+1}$. Suppose that R_i are different almost surely this translates to:

$$\mathbb{P}\left\{R_{n+1} \text{ is among the } k \text{ smallest in } R_1, \dots, R_{n+1}\right\} = \frac{k}{n+1}$$

Which is equivalent to:

$$\mathbb{P}\left\{R_{n+1} \text{ is among the } k \text{ smallest in } R_1, \dots, R_n\right\} = \frac{k}{n+1}$$

Taking $k = \lceil (n+1)(1-\alpha) \rceil$ we have:

$$\mathbb{P}\left\{R_{n+1} \text{ is among the } k \text{ smallest in } R_1, \dots, R_n\right\} = \frac{\lceil (n+1)(1-\alpha) \rceil}{n+1}$$

Split conformal prediction - Proof

We have:

$$\mathbb{P}\left\{R_{n+1} \text{ is among the } k \text{ smallest in } R_1, \ldots, R_{n+1}\right\} \in [1-\alpha, 1-\alpha+1/(n+1))$$

We can translate what is inside $\mathbb{P}$ to:

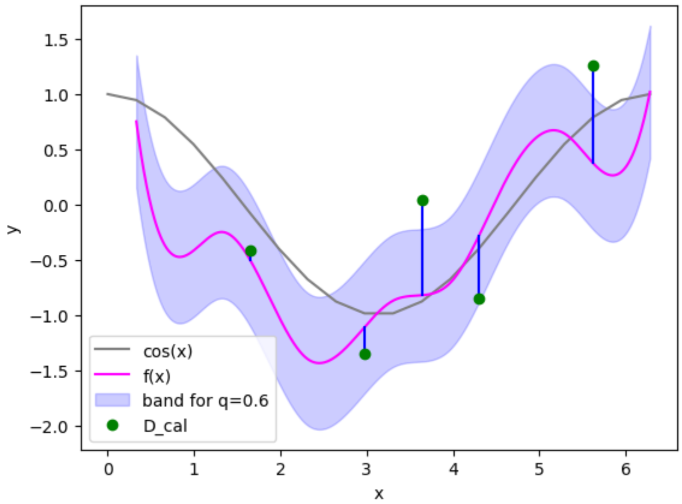
$$q = Quantile\left(\frac{\lceil (n+1)(1-\alpha) \rceil}{n}, \frac{1}{n}\sum_{i=1}^{n} \delta_{R_i}\right)$$

Finally, we can write:

$$1-\alpha \leq \mathbb{P}\left\{R_{n+1} \leq q\right\} < 1-\alpha+\frac{1}{n+1}$$

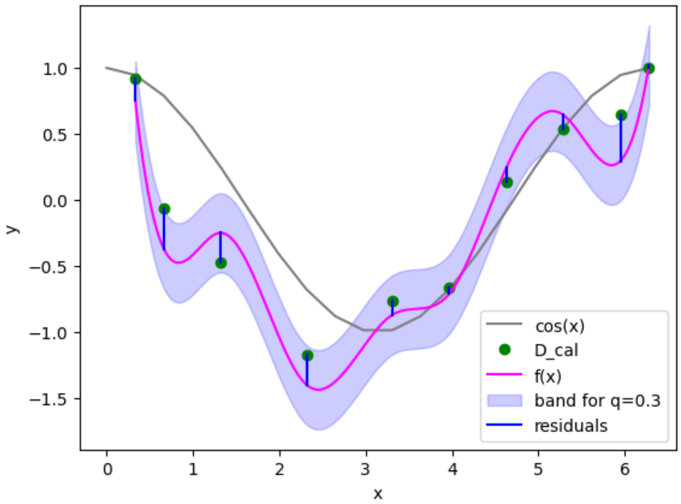
How to choose the calibration dataset size?

Does that mean that we can use any calibration dataset size? What happens if we use a very small calibration dataset?



How to choose the calibration dataset size?

Does that mean that we can use any calibration dataset size? What happens if we use a very small calibration dataset?

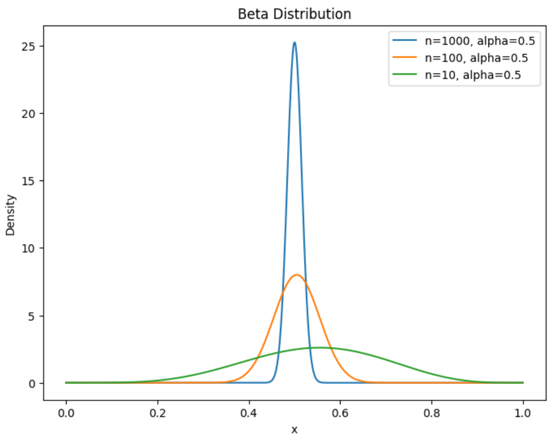


How to choose the calibration dataset size?

Conditional on the data of the calibration dataset, the coverage is distributed as:

$$\mathbb{P}\left\{ Y \in \hat{C}(X) \mid (X_i, Y_i) \in \mathcal{D}_{cal} \right\} \sim Beta(k, n-k+1)$$

With $k = \lceil (n+1)(1-\alpha) \rceil$.



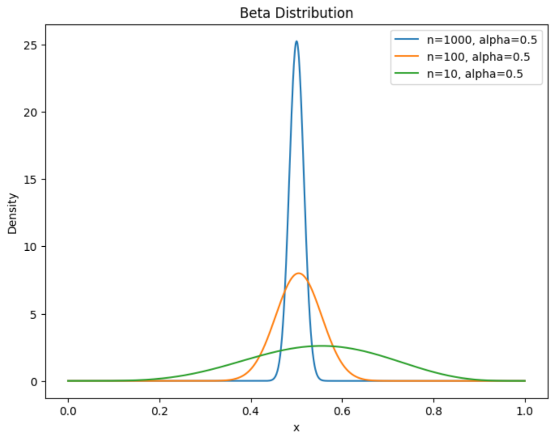
How to choose the calibration dataset size?

Conditional on the data of the calibration dataset, the coverage is distributed as:

$$\mathbb{P}\left\{Y \in \hat{C}(X) \mid (X_i, Y_i) \in \mathcal{D}_{cal}\right\} \sim Beta(k, n-k+1)$$

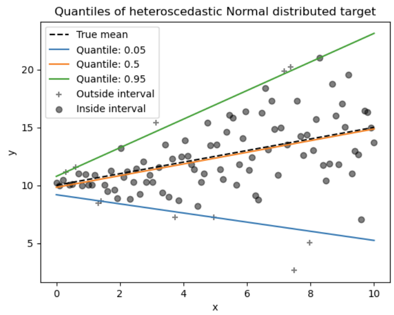
With $k = \lceil (n+1)(1-\alpha) \rceil$.

n (size of D_{cal})	coverage correct +/-5%
10	0.24
100	0.68
1000	0.99



Conformalized quantile regression

In the same spirit, we can try to find adaptive bounds. One possibility is to train our predictor f to output quantiles as it is done with the quantile regression:

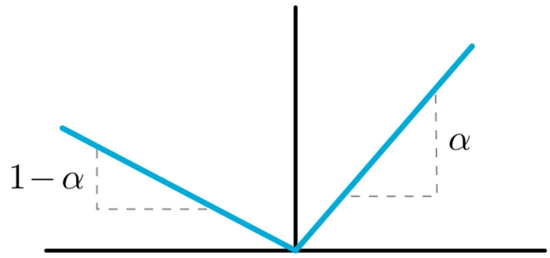


In this case, the outputs of our predictor are given by $f(x) = \{f_{\frac{\alpha}{2}}(x), f_{1-\frac{\alpha}{2}}(x)\}$

Conformalized quantile regression

This can be easily achieved for any learning based predictor by just using the pinball loss:

$$\mathcal{L}_\alpha(y, f(x)) = \begin{cases} \alpha(y - f(x)) & \text{if } y > f(x) \\ (1-\alpha)(f(x) - y) & \text{otherwise} \end{cases}$$



Using the following conformity score:

$$s(x, y) = \max\left\{y - f_{\frac{\alpha}{2}}(x), f_{1-\frac{\alpha}{2}}(x) - y\right\}$$

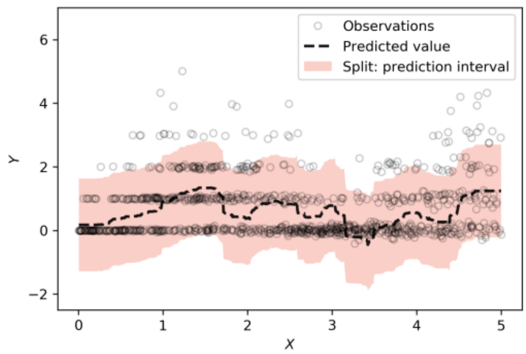
We can, now compute the quantile q and build our conformal predictor.

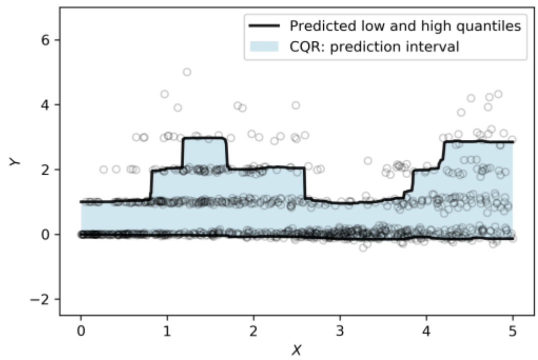
Conformalized quantile regression

Our conformal bands will be given by:

$$\hat{C}(x) = \left[f_{\frac{\alpha}{2}}(x) - q, f_{1-\frac{\alpha}{2}}(x) + q\right]$$

This method is introduced by *Romano et al. (2019)*. Here are some of their showcase results:





How to deal with time series?

Conformal prediction for time series was introduced by *Stankeviciute et al. (2021)*. Some other work has followed the same idea as well.

Given a discrete time series of length n, for simplicity, of real values:

$$(y_1, \ldots, y_n)$$

We want to predict the values from y_{m+1} to y_n, given the values from y_1 to y_m. This is done via a neural network with m inputs and $n - m$ outputs:

$$f(y_1, \ldots, y_m) = (\hat{y}_{m+1}, \ldots, \hat{y}_n)$$

How to deal with time series?

The idea is to build one conformal predictor for each output, of the neural network, independently. Our conformal predictor will look like:

$$\hat{C}(y_1, \dots, y_m) = \left(\hat{C}_{m+1}(y_1, \dots, y_m), \dots, \hat{C}_n(y_1, \dots, y_m)\right)$$

For a coverage of $1 - \beta$ for each individual predictor, each will have an error probability of β. Therefore the probability of at least one error obeys the following given Boole's inequality:

$$\mathbb{P}\left\{\bigcup_{i=m+1}^{n} \hat{y}_i \notin \hat{C}_i(y_1, \dots, y_m)\right\} \leq \sum_{i=m+1}^{n} \mathbb{P}\left\{\hat{y}_i \notin \hat{C}_i(y_1, \dots, y_m)\right\}$$

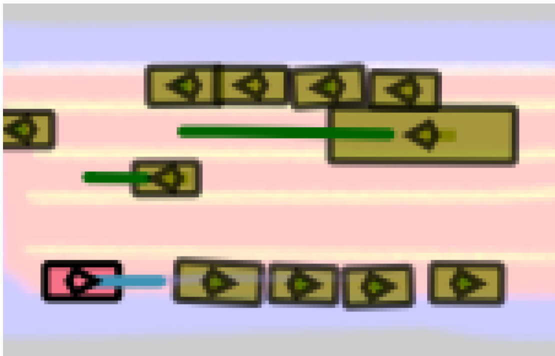
The probability of at least one error is at least $\beta(n - m)$. So if we want all predictions to be valid (tube around our prediction) we want to choose $\beta = \frac{\alpha}{(n-m)}$.

How to deal with time series?

Applying this idea to *Trajectron++* with a time step of 0.5s, we have:

Table 1: Prediction set sizes for a $1 - \alpha = 90\%$

x(m)	y(m)	t(s)
0.8886	0.1881	0.5
1.5965	0.3520	1.0
2.2246	0.5275	1.5
3.0881	0.7505	2.0
4.2229	1.0411	2.5

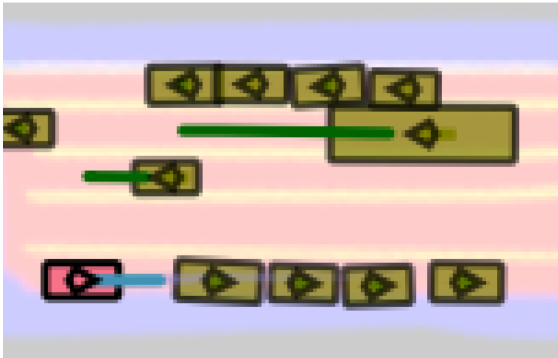


How to deal with time series?

If full coverage is not required *Trajectron++* with a time step of 0.5s, we have:

Table 2: Prediction set sizes for a $1 - \alpha = 90\%$

x(m)	y(m)	t(s)
0.5345	0.0711	0.5
0.7267	0.1453	1.0
1.0552	0.2060	1.5
1.5524	0.2999	2.0
2.1262	0.3672	2.5



Some stuff I did not cover

- Conformal prediction for classification : *Adaptive prediction sets*.
- Use part of calibration data for training: *Full conformal prediction, Cross-Conformal Prediction, CV+, and Jackknife+*.
- Online updates: *Rolling RC and adaptive conformal prediction*.
- Conformal prediction when we face distribution shifts: *Conformal prediction under the covariate shift*.

Conclusions

Takeaways:

- Easy to implement and efficient.
- Provides valid guarantees with finite samples.
- Active area of research, lots of new papers per year.
- Used in world scenarios.

Be attentive to:

- Distribution shifts or anything that breaks the exchangeability assumption.
- Conditional validity could be a problem.

References I

Algorithmic Learning in a Random World. New York: Springer-Verlag, 2005. ISBN: 978-0-387-00152-4. DOI: 10.1007/b106715. URL: http://link.springer.com/10.1007/b106715 (visited on 02/13/2023).

"Conformal Time-series Forecasting". In: *Advances in Neural Information Processing Systems*. Vol. 34. Curran Associates, Inc., 2021, pp. 6216–6228. URL: https://proceedings.neurips.cc/paper/2021/hash/312f1ba2a72318edaaa995a67835fad5-Abstract.html (visited on 07/26/2023).

"Conformal Prediction Under Covariate Shift". In: Neural Information Processing Systems. Apr. 1, 2019. URL: https://www.semanticscholar.org/paper/Conformal-Prediction-Under-Covariate-Shift-Tibshirani-Barber/f08e13d65cb17856427b429d79f01922584a6f01 (visited on 07/04/2023).

Conditional Validity of Inductive Conformal Predictors. Sept. 24, 2012. arXiv: 1209.2673 [cs]. URL: http://arxiv.org/abs/1209.2673 (visited on 07/03/2023). preprint.

References II

"Conformalized Quantile Regression". In: Neural Information Processing Systems. May 8, 2019. URL: https://www.semanticscholar.org/paper/Conformalized-Quantile-Regression-Romano-Patterson/6f9dc6f8519e927d948a13aa7ae0df336f443eb9 (visited on 07/24/2023).

"Adaptive Conformal Prediction for Motion Planning among Dynamic Agents". Version 1. In: (2022). DOI: 10.48550/ARXIV.2212.00278. URL: https://arxiv.org/abs/2212.00278 (visited on 02/08/2023).

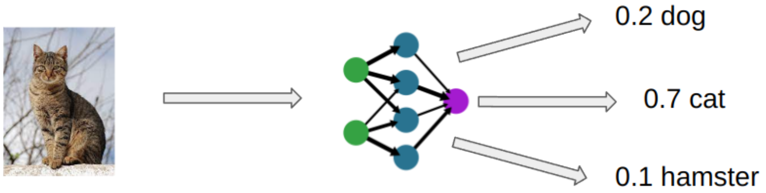
"Trajectron++: Multi-Agent Generative Trajectory Forecasting With Heterogeneous Data for Control". In: *ArXiv* (Jan. 9, 2020). URL: https://www.semanticscholar.org/paper/Trajectron%2B%2B%3A-Multi-Agent-Generative-Trajectory-for-Salzmann-Ivanovic/0e61c3aab3aad963feacc915a23cb1965b152667 (visited on 01/19/2023).

References III

- *A Gentle Introduction to Conformal Prediction and Distribution-Free Uncertainty Quantification*. Dec. 7, 2022. DOI: 10.48550/arXiv.2107.07511. arXiv: 2107.07511 [cs, math, stat]. URL: http://arxiv.org/abs/2107.07511 (visited on 11/13/2023). preprint.
- *Ryantibs/Statlearn-S23: Course Materials for Advanced Topics in Statistical Learning, Spring 2023*. URL: https://github.com/ryantibs/statlearn-s23/tree/main (visited on 11/13/2023).

What about classification problems?

Given a set of labels $\mathcal{Y} = \{cat, dog, hamster\}$, neural networks are able to output estimates of their likelihoods $f(x) = \{\hat{p}_{cat}, \hat{p}_{dog}, \hat{p}_{hamster}\}$:



How to provide safe prediction sets in such scenarios?

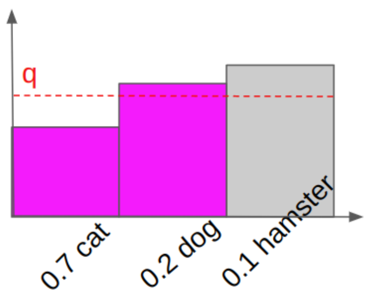
Adaptive prediction sets

APS - adaptive prediction sets (*Romano et al. (2020) Angelopoulos et al. (2021)*) solves this issue.

We define π to be a permutation which sorts the outputs of the predictor $f(x) = \{\hat{p}_1, \ldots, \hat{p}_K\}$ in decreasing order. Our conformal predictor will be:

$$\hat{C}(x) = \left\{\hat{p}_{\pi(1)}, \ldots, \hat{p}_{\pi(k)}\right\}$$

Where k is chosen such that we have a cumulative sum until we reach the quantile q over the conformity scores corresponding to $1 - \alpha$ coverage:



Adaptive prediction sets

The quantile q is chosen as previously:

$$q = Quantile\left(\frac{\lceil (n+1)(1-\alpha) \rceil}{n}, \frac{1}{n}\sum_{i=1}^{n} \delta_{s_i}\right)$$

The conformity scores s_i are defined as:

$$s(x, y) = \sum_{j=1}^{k} \hat{p}_{\pi(j)} \text{ where } y = \pi(k)$$

How to make the coverage correct per class

We wish that, in the classification setting, we could have the coverage guarantees per class, more formally:

$$\mathbb{P}\left\{Y \in \hat{C}(X) \mid Y = y\right\} \geq 1 - \alpha$$

If $\mathcal{Y} = \{\textit{Sick}, \textit{Healthy}\}$, we would like to have prediction sets valid independent of the true label.

	Sick	Healthy
Test Positive	60%	40%
Test Negative	10%	90%

How to make the coverage correct per class

If we define the quantiles per class as:

$$q^k = Quantile\left(\frac{\lceil (n^k+1)(1-\alpha)\rceil}{n^k}, \frac{1}{n}\sum_{i=1}^{n^k}\delta_{s_i^k}\right)$$

Where the superscript k denotes restricts the samples in $\mathcal{D}_{cal}$ to the class k. We can define a class conditional valid conformal predictor as:

$$\hat{C}(x) = \{y \text{ s.t } s(x,y) \leq q^y\}$$

What else could we want?

Instance conditional validity:

$$\mathbb{P}\left\{Y \in \hat{C}(X) \mid X = x\right\} \geq 1 - \alpha$$

Unfortunately, that's impossible :(

Lei and Wasserman (2014) ...any prediction band which claims to cover at almost every point, for every joint distribution, must be infinite in size ...

But not everything is lost

Group conditional validity:

$$\mathbb{P}\left\{Y \in \hat{C}(X) \mid X \in \mathcal{G}_i\right\} \geq 1 - \alpha$$

Given a partition of the input space $\mathcal{G}_1, \ldots, \mathcal{G}_k$. We can define a group conditional valid conformal predictor as:

$$\hat{C}(x) = \{y \text{ s.t } s(x, y) \leq q^g\}$$