Conformal Inference Methods in Deep Learning

Matteo Sesia

Department of Data Sciences and Operations
Marshall School of Business
University of Southern California

June 4, 2023

Introduction:
Why conformal inference?

General setup for supervised learning

Assumption:

$$(X_i, Y_i) \stackrel{\text{iid}}{\sim} P_{X,Y}$$

The joint distribution $P_{X,Y}$ is unknown.

- $X \in \mathbb{R}^p$ explanatory variables
- $Y \in \mathbb{R}$ response variable

Data: $\{(X_i, Y_i)\}_{i=1}^n$.

Goal: fit a model that can predict Y_{n+1} given X_{n+1}.

This class: how to **account for predictive uncertainty**?

Example (classification)

X: Image, Y: label

Test point: 

What digit is this? Probably 5 or 6.

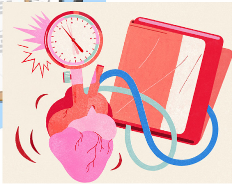
Example (regression)

X: Facebook page features, Y: number of comments

Test: X_{n+1}. What could Y_{n+1} be?

Example (regression)

X: Clinical history, genetic data, demographics. Y: blood pressure



Test: X_{n+1}. What could Y_{n+1} be?

Quantifying uncertainty via prediction sets

Instead of a point prediction, output a set of likely outcomes. E.g.,

- The digit is either a 5 or a 6.
- The blood pressure will be between 120-129 mmHg.

Quantifying uncertainty via prediction sets

Instead of a point prediction, output a set of likely outcomes. E.g.,

- The digit is either a 5 or a 6.
- The blood pressure will be between 120-129 mmHg.

Fix $\alpha \in (0, 1)$ and construct a prediction rule $\hat{C}_\alpha$ s.t. the set

$$\hat{C}_\alpha(X) \subseteq \mathbb{R}$$

has *marginal coverage* at level $1 - \alpha$:

$$\mathbb{P}\left[Y_{n+1} \in \hat{C}_\alpha(X_{n+1})\right] \geq 1 - \alpha.$$

Quantifying uncertainty via prediction sets

Instead of a point prediction, output a set of likely outcomes. E.g.,

- The digit is either a 5 or a 6.
- The blood pressure will be between 120-129 mmHg.

Fix $\alpha \in (0, 1)$ and construct a prediction rule $\hat{C}_\alpha$ s.t. the set

$$\hat{C}_\alpha(X) \subseteq \mathbb{R}$$

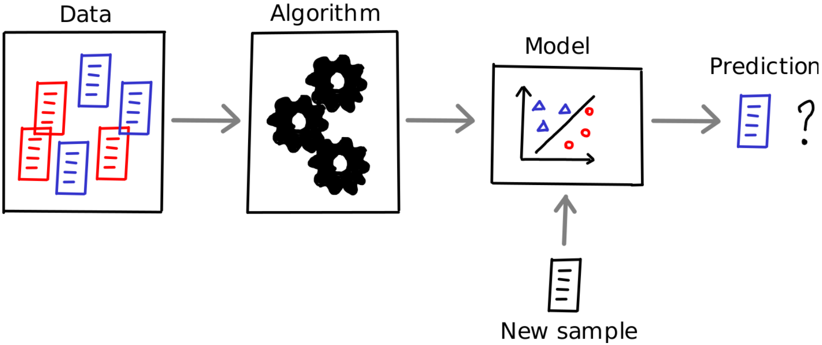
has *marginal coverage* at level $1 - \alpha$:

$$\mathbb{P}\left[Y_{n+1} \in \hat{C}_\alpha(X_{n+1})\right] \geq 1 - \alpha.$$

In regression problems, we may want $\hat{C}_\alpha(X)$ to be an interval.
In classification problems, it will be a discrete set.

The model-free predictive inference framework

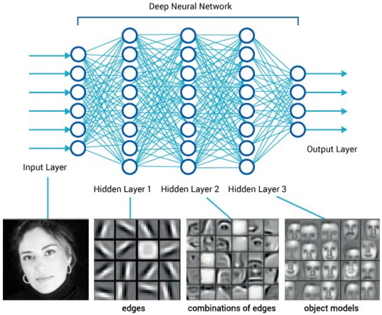
Data-generating model: $(X_i, Y_i) \stackrel{\text{iid}}{\sim} P_{X,Y}$.

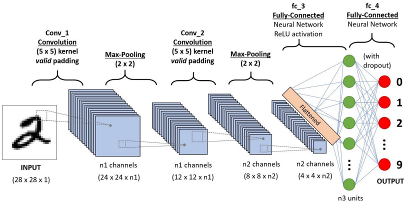


Challenges:

1. $P(Y \mid X)$ could be anything (completely unknown)
2. The prediction model may be a machine learning black box (e.g., neural network, random forests, Bayesian trees, ...)

Successes of deep learning models





Machine learning in healthcare

Machine-Learning Platform Can Accurately Predict Surgical Complications

Article | Open Access | Published: 11 August 2022

Testing the applicability and performance of Auto ML for potential applications in diagnostic neuroradiology

Manfred Musigmann, Burak Han Akkurt, Hermann Krähling, Nabila Gala Nacul, Luca Remonda, Thomas Sartoretti, Dylan Henssen, Benjamin Brokinkel, Walter Stummer, Walter Heindel & Manoj Mannil

Scientific Reports **12**, Article number: 13648 (2022) | Cite this article

Metrics

Article | Open Access | Published: 12 August 2022

Deep learning-based diagnosis from endobronchial ultrasonography images of pulmonary lesions

Takamasa Hotta, Noriaki Kurimoto, Yohei Shiratsuki, Yoshihiro Amano, Megumi Hamaguchi, Akari Tanino, Yukari Tsubata & Takeshi Isobe

Scientific Reports **12**, Article number: 13710 (2022) | Cite this article

Metrics

Article | Open Access | Published: 12 August 2022

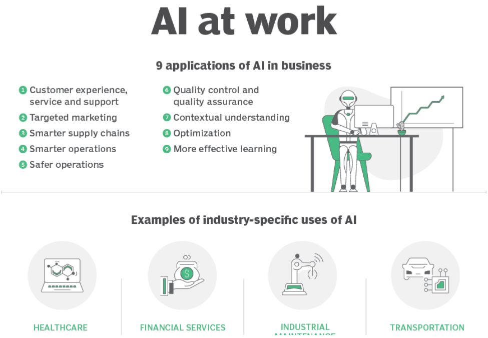
Development of deep learning chest X-ray model for cardiac dose prediction in left-sided breast cancer radiotherapy

Yutaro Koide, Takahiro Aoyama, Hidetoshi Shimizu, Tomoki Kitagawa, Risei Miyauchi, Hiroyuki Tachibana & Takeshi Kodaira

Scientific Reports **12**, Article number: 13706 (2022) | Cite this article

2 Altmetric | Metrics

Machine learning in business



forbes.com

AI Models To Boost Lending And How Your Business Could Benefit From Them

Peter Shubenok

6-8 minutes

Peter Shubenok is the Founder and CEO at RNDpoint, a leading provider of lending platforms for Banks and MFIs.

What could possibly go wrong with machine learning?

HEALTH IT ANALYTICS
xtelligent HEALTHCARE MEDIA

Home News Features Interviews P

Population Health | Precision Medicine | Quality & Governance | Tools & Strategies | Focus on AI | Care Coo

AI May Be More Prone to Errors in Image-Based Diagnoses Than Clinicians

New research indicates that AI may be more prone to making mistakes than humans in image-based medical diagnoses because of the features they use for analysis.

theguardian.com

UK data watchdog investigates whether AI systems show racial bias

NATIONAL

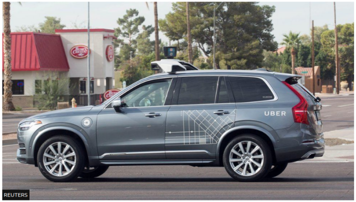
Nearly 400 car crashes in 11 months involved automated tech, companies tell regulators

June 15, 2022 · 1:26 PM ET

THE ASSOCIATED PRESS

Uber's self-driving operator charged over fatal crash

16 September 2020



The self-driving Volvo hit a pedestrian at 39mph, despite the presence of a safety driver

Uncertainty and confidence in machine learning

Published in Towards Data Science

Michel Kana, Ph.D

Apr 26, 2020 · 9 min read · Member-only ·

Uncertainty in Deep Learning. How To Measure?

A hands-on tutorial on Bayesian estimation of epistemic and aleatoric uncertainty with Keras. Towards a social acceptance of AI.

> My Deep Learning Model Says: “sorry, I don’t know the answer”. That can be absolutely OK.

nature > npj digital medicine > perspectives > article

Perspective | Open Access | Published: 05 January 2021

Second opinion needed: communicating uncertainty in medical machine learning

Benjamin Kompa, Jasper Snoek & Andrew L. Beam

npj Digital Medicine **4**, Article number: 4 (2021) | Cite this article

9471 Accesses | **39** Citations | **100** Altmetric | Metrics

MIT News

ON CAMPUS AND AROUND THE WORLD

SUBSCRIBE

A neural network learns when it should not be trusted

A faster way to estimate uncertainty in AI-assisted decision-making could lead to safer outcomes.

Daniel Ackerman | MIT News Office

November 20, 2020

This is a hot topic in machine learning . . .

Stanford University

Stanford | Data Science

Search this site

About | People | Programs | Research Centers | Affiliates | Education | Subscribe >

In the News

Emmanuel Candès Keynotes NeurIPS 2022

Faculty Director, Emmanuel Candès, is giving a keynote address at the upcoming Neural Information Processing System (NeurIPS) conference on Tuesday, November 29, 2022

NEURAL INFORMATION PROCESSING SYSTEMS

Getting Started Schedule Tutorials Main Conference

Invited Talk

Conformal Prediction in 2022

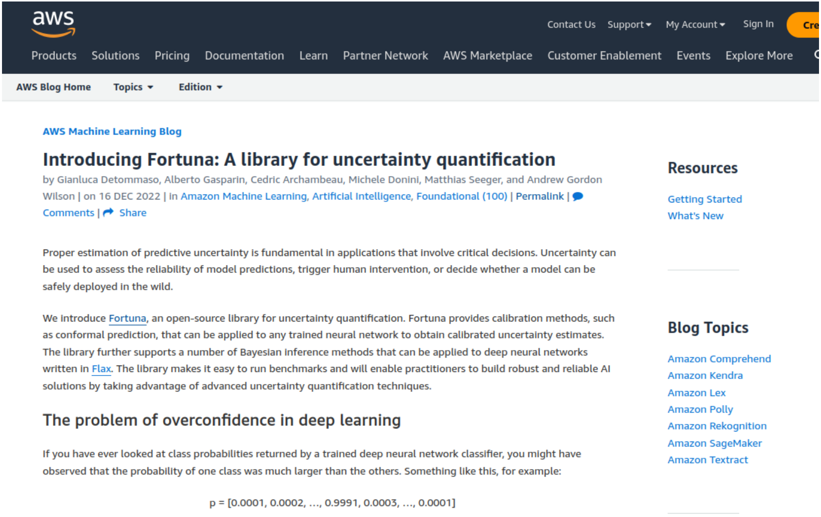
Emmanuel Candes

Moderator: Alekh Agarwal

Hall H

[Abstract]

. . . both within and beyond academia



Contact Us Support My Account Sign In

Products Solutions Pricing Documentation Learn Partner Network AWS Marketplace Customer Enablement Events Explore More

AWS Blog Home Topics Edition

AWS Machine Learning Blog

Introducing Fortuna: A library for uncertainty quantification

by Gianluca Detommaso, Alberto Gasparin, Cedric Archambeau, Michele Donini, Matthias Seeger, and Andrew Gordon Wilson | on 16 DEC 2022 | in Amazon Machine Learning, Artificial Intelligence, Foundational (100) | Permalink | Comments | Share

Proper estimation of predictive uncertainty is fundamental in applications that involve critical decisions. Uncertainty can be used to assess the reliability of model predictions, trigger human intervention, or decide whether a model can be safely deployed in the wild.

We introduce Fortuna, an open-source library for uncertainty quantification. Fortuna provides calibration methods, such as conformal prediction, that can be applied to any trained neural network to obtain calibrated uncertainty estimates. The library further supports a number of Bayesian inference methods that can be applied to deep neural networks written in Flax. The library makes it easy to run benchmarks and will enable practitioners to build robust and reliable AI solutions by taking advantage of advanced uncertainty quantification techniques.

The problem of overconfidence in deep learning

If you have ever looked at class probabilities returned by a trained deep neural network classifier, you might have observed that the probability of one class was much larger than the others. Something like this, for example:

p = [0.0001, 0.0002, …, 0.9991, 0.0003, …, 0.0001]

Resources

Getting Started
What's New

Blog Topics

Amazon Comprehend
Amazon Kendra
Amazon Lex
Amazon Polly
Amazon Rekognition
Amazon SageMaker
Amazon Textract

Chapter 1:
Review of linear regression

Review of classical linear regression

Linear regression model:

- X_i are fixed,
- $Y_i = X_i^\top \beta + \varepsilon_i$, where $\varepsilon_i \sim \mathcal{N}(0, \sigma^2)$.

Review of classical linear regression

Linear regression model:

- X_i are fixed,
- $Y_i = X_i^\top \beta + \varepsilon_i$, where $\varepsilon_i \sim \mathcal{N}(0, \sigma^2)$.

Data: $\mathbb{X} \in \mathbb{R}^{n \times p}, \mathbb{Y} \in \mathbb{R}^n$. Least-squares estimate of β:

$$\hat{\beta} = (\mathbb{X}^\top \mathbb{X})^{-1} \mathbb{X}^\top \mathbb{Y} \sim \mathcal{N}(\beta, \sigma^2 (\mathbb{X}^\top \mathbb{X})^{-1})$$

Review of classical linear regression

Linear regression model:

- X_i are fixed,
- $Y_i = X_i^\top \beta + \varepsilon_i$, where $\varepsilon_i \sim \mathcal{N}(0, \sigma^2)$.

Data: $\mathbb{X} \in \mathbb{R}^{n \times p}, \mathbb{Y} \in \mathbb{R}^n$. Least-squares estimate of β:

$$\hat{\beta} = (\mathbb{X}^\top \mathbb{X})^{-1} \mathbb{X}^\top \mathbb{Y} \sim \mathcal{N}(\beta, \sigma^2 (\mathbb{X}^\top \mathbb{X})^{-1})$$

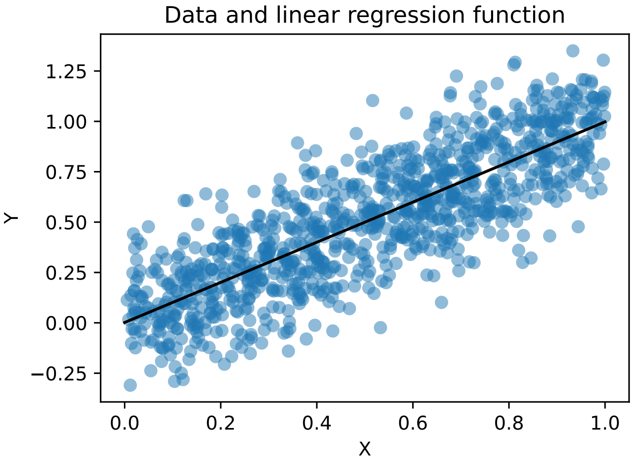
Predictions:

$$\hat{Y}_{n+1} = X_{n+1}^\top \hat{\beta} \sim \mathcal{N}(X_{n+1}^\top \beta, \sigma^2 X_{n+1}^\top (\mathbb{X}^\top \mathbb{X})^{-1} X_{n+1})$$

Review of classical linear regression (continued)

Predictions:

$$\hat{Y}_{n+1} = X_{n+1}^\top \hat{\beta} \sim \mathcal{N}(X_{n+1}^\top \beta, \sigma^2 X_{n+1}^\top (\mathbb{X}^\top \mathbb{X})^{-1} X_{n+1})$$



Review of classical linear regression (continued)

Predictions:

$$\hat{Y}_{n+1} = X_{n+1}^\top \hat{\beta}$$

Review of classical linear regression (continued)

Predictions:

$$\begin{aligned}\hat{Y}_{n+1} &= X_{n+1}^\top \hat{\beta} \\ &= X_{n+1}^\top \beta + \sigma \sqrt{X_{n+1}^\top (\mathbb{X}^\top \mathbb{X})^{-1} X_{n+1}} \cdot \mathcal{N}(0,1)\end{aligned}$$

Review of classical linear regression (continued)

Predictions:

$$
\begin{aligned}
\hat{Y}_{n+1} &= X_{n+1}^\top \hat{\beta} \\
&= X_{n+1}^\top \beta + \sigma \sqrt{X_{n+1}^\top (\mathbb{X}^\top \mathbb{X})^{-1} X_{n+1}} \cdot \mathcal{N}(0,1) \\
&= Y_{n+1} - \sigma \cdot \mathcal{N}(0,1) + \sigma \sqrt{X_{n+1}^\top (\mathbb{X}^\top \mathbb{X})^{-1} X_{n+1}} \cdot \mathcal{N}(0,1)
\end{aligned}
$$

Review of classical linear regression (continued)

Predictions:

$$\begin{aligned}
\hat{Y}_{n+1} &= X_{n+1}^\top \hat{\beta} \\
&= X_{n+1}^\top \beta + \sigma \sqrt{X_{n+1}^\top (\mathbb{X}^\top \mathbb{X})^{-1} X_{n+1}} \cdot \mathcal{N}(0,1) \\
&= Y_{n+1} - \sigma \cdot \mathcal{N}(0,1) + \sigma \sqrt{X_{n+1}^\top (\mathbb{X}^\top \mathbb{X})^{-1} X_{n+1}} \cdot \mathcal{N}(0,1) \\
&= Y_{n+1} + \sigma \sqrt{1 + X_{n+1}^\top (\mathbb{X}^\top \mathbb{X})^{-1} X_{n+1}} \cdot \mathcal{N}(0,1).
\end{aligned}$$

Review of classical linear regression (continued)

Predictions:

$$
\begin{aligned}
\hat{Y}_{n+1} &= X_{n+1}^\top \hat{\beta} \\
&= X_{n+1}^\top \beta + \sigma \sqrt{X_{n+1}^\top (\mathbb{X}^\top \mathbb{X})^{-1} X_{n+1}} \cdot \mathcal{N}(0,1) \\
&= Y_{n+1} - \sigma \cdot \mathcal{N}(0,1) + \sigma \sqrt{X_{n+1}^\top (\mathbb{X}^\top \mathbb{X})^{-1} X_{n+1}} \cdot \mathcal{N}(0,1) \\
&= Y_{n+1} + \sigma \sqrt{1 + X_{n+1}^\top (\mathbb{X}^\top \mathbb{X})^{-1} X_{n+1}} \cdot \mathcal{N}(0,1).
\end{aligned}
$$

Recall that

$$
\hat{\sigma}^2 = \frac{\text{RSS}}{(n-p-1)} \sim \sigma^2 \cdot \frac{\chi^2_{n-p-1}}{n-p-1},
$$

Review of classical linear regression (continued)

Predictions:

$$\begin{aligned}
\hat{Y}_{n+1} &= X_{n+1}^\top \hat{\beta} \\
&= X_{n+1}^\top \beta + \sigma \sqrt{X_{n+1}^\top (\mathbb{X}^\top \mathbb{X})^{-1} X_{n+1}} \cdot \mathcal{N}(0,1) \\
&= Y_{n+1} - \sigma \cdot \mathcal{N}(0,1) + \sigma \sqrt{X_{n+1}^\top (\mathbb{X}^\top \mathbb{X})^{-1} X_{n+1}} \cdot \mathcal{N}(0,1) \\
&= Y_{n+1} + \sigma \sqrt{1 + X_{n+1}^\top (\mathbb{X}^\top \mathbb{X})^{-1} X_{n+1}} \cdot \mathcal{N}(0,1).
\end{aligned}$$

Recall that

$$\hat{\sigma}^2 = \frac{\text{RSS}}{(n-p-1)} \sim \sigma^2 \cdot \frac{\chi^2_{n-p-1}}{n-p-1},$$

Therefore,

$$\sigma = \frac{\hat{\sigma}}{\sqrt{\chi^2_{n-p-1}/(n-p-1)}}.$$

Review of classical linear regression (continued)

Replace σ with $\hat{\sigma}$ into formula for prediction:

$$\begin{aligned}
\hat{Y}_{n+1} &= Y_{n+1} + \sigma\sqrt{1 + X_{n+1}^\top(\mathbb{X}^\top\mathbb{X})^{-1}X_{n+1}} \cdot \mathcal{N}(0,1) \\
&= Y_{n+1} + \hat{\sigma}\sqrt{1 + X_{n+1}^\top(\mathbb{X}^\top\mathbb{X})^{-1}X_{n+1}} \cdot \frac{\mathcal{N}(0,1)}{\sqrt{\chi^2_{n-p-1}/(n-p-1)}} \\
&= Y_{n+1} + \hat{\sigma}\sqrt{1 + X_{n+1}^\top(\mathbb{X}^\top\mathbb{X})^{-1}X_{n+1}} \cdot t_{n-p-1}
\end{aligned}$$

Review of classical linear regression (continued)

Replace σ with $\hat{\sigma}$ into formula for prediction:

$$\begin{aligned}
\hat{Y}_{n+1} &= Y_{n+1} + \sigma\sqrt{1 + X_{n+1}^\top(\mathbb{X}^\top\mathbb{X})^{-1}X_{n+1}} \cdot \mathcal{N}(0,1) \\
&= Y_{n+1} + \hat{\sigma}\sqrt{1 + X_{n+1}^\top(\mathbb{X}^\top\mathbb{X})^{-1}X_{n+1}} \cdot \frac{\mathcal{N}(0,1)}{\sqrt{\chi^2_{n-p-1}/(n-p-1)}} \\
&= Y_{n+1} + \hat{\sigma}\sqrt{1 + X_{n+1}^\top(\mathbb{X}^\top\mathbb{X})^{-1}X_{n+1}} \cdot t_{n-p-1}
\end{aligned}$$

Prediction interval $(1-\alpha)$ for Y_{n+1}:

$$\hat{C}_\alpha(X_{n+1}) = \hat{Y}_{n+1} \pm \hat{\sigma}\sqrt{1 + X_{n+1}^\top(\mathbb{X}^\top\mathbb{X})^{-1}X_{n+1}} \cdot t_{n-p-1}^{(\alpha/2)}.$$

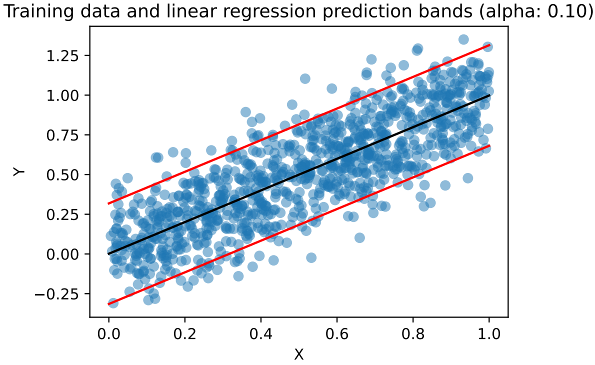
Review of classical linear regression (continued)

The prediction interval

$$\hat{C}_\alpha(X_{n+1}) = \hat{Y}_{n+1} \pm \hat{\sigma}\sqrt{1 + X_{n+1}^\top(\mathbb{X}^\top\mathbb{X})^{-1}X_{n+1}} \cdot t_{n-p-1}^{(\alpha/2)}.$$

satisfies *conditional coverage*:

$$\mathbb{P}\left[Y_{n+1} \in \hat{C}_\alpha(x) \mid \mathbb{X}, X_{n+1} = x\right] = 1 - \alpha.$$



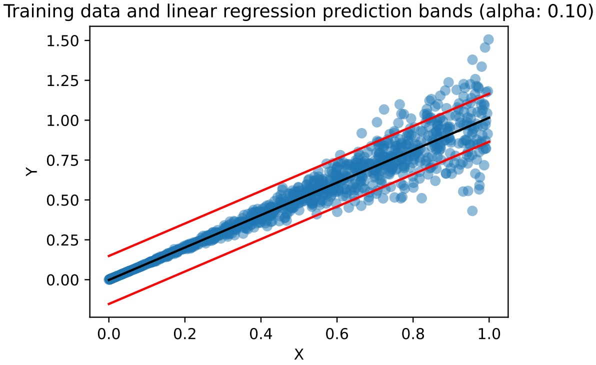
Review of classical linear regression (continued)

The prediction interval

$$\hat{C}_\alpha(X_{n+1}) = \hat{Y}_{n+1} \pm \hat{\sigma}\sqrt{1 + X_{n+1}^\top(\mathbb{X}^\top\mathbb{X})^{-1}X_{n+1}} \cdot t_{n-p-1}^{(\alpha/2)}.$$

satisfies *conditional coverage*:

$$\mathbb{P}\left[Y_{n+1} \in \hat{C}_\alpha(x) \mid \mathbb{X}, X_{n+1} = x\right] = 1 - \alpha.$$



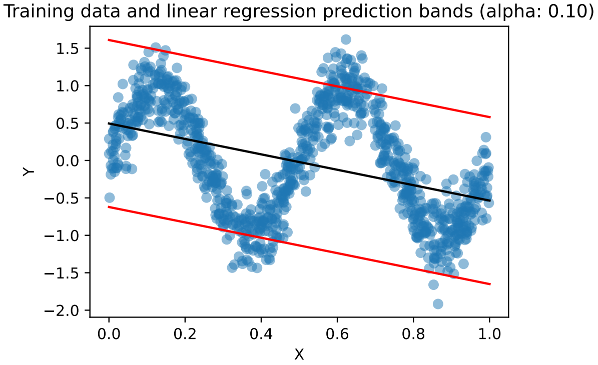
Review of classical linear regression (continued)

The prediction interval

$$\hat{C}_\alpha(X_{n+1}) = \hat{Y}_{n+1} \pm \hat{\sigma}\sqrt{1 + X_{n+1}^\top (\mathbb{X}^\top \mathbb{X})^{-1} X_{n+1}} \cdot t_{n-p-1}^{(\alpha/2)}.$$

satisfies *conditional coverage*:

$$\mathbb{P}\left[Y_{n+1} \in \hat{C}_\alpha(x) \mid \mathbb{X}, X_{n+1} = x\right] = 1 - \alpha.$$



Model-free predictive inference

$$(X_i, Y_i) \stackrel{\text{iid}}{\sim} P_{X,Y}$$

Much more general problem:

- $P(Y \mid X)$ could be anything (completely unknown)
- Prediction rule $\hat{Y}$ is a machine learning black box
  (e.g., neural network, random forests, Bayesian trees, ...)

Model-free predictive inference

$$(X_i, Y_i) \overset{\text{iid}}{\sim} P_{X,Y}$$

Much more general problem:

- $P(Y \mid X)$ could be anything (completely unknown)
- Prediction rule $\hat{Y}$ is a machine learning black box (e.g., neural network, random forests, Bayesian trees, ...)

We need some leverage:

- The data are random
- The test point is random
- Coverage will only be *marginal* (not conditional on X_{n+1}):

$$\mathbb{P}\left[Y_{n+1} \in \hat{C}_\alpha(X_{n+1})\right] \geq 1 - \alpha.$$

Chapter 2:
Exchangeability

Exchangeable random variables

We say that $Z_1, Z_2, \ldots, Z_n$ are exchangeable if and only if, for any permutation σ of $\{1, \ldots, n\}$,

$$p(Z_1, Z_2, \ldots, Z_n) = p(Z_{\sigma(1)}, Z_{\sigma(2)}, \ldots, Z_{\sigma(n)}).$$

For example, $Z_1, Z_2, \ldots, Z_n$ are exchangeable if they are i.i.d.

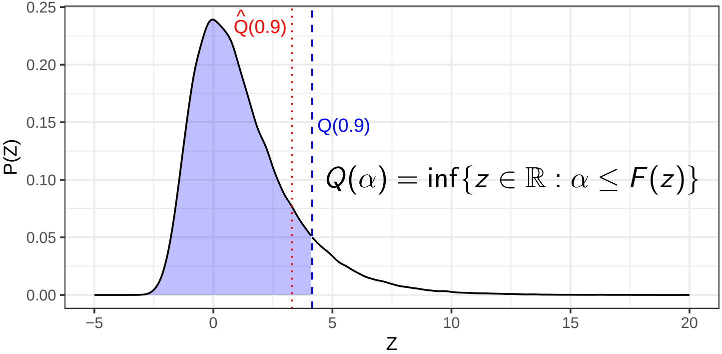
Prediction without covariates

Suppose we have

$$Z_i \overset{\text{exch.}}{\sim} P_Z, \qquad Z \in \mathbb{R}$$

and we want to use the first n data points to construct a one-sided prediction interval $\hat{C}_\alpha = (-\infty, \hat{U}_{1-\alpha}]$ such that

$$\mathbb{P}\left[Z_{n+1} \leq \hat{U}_{1-\alpha}\right] \geq 1 - \alpha.$$



Empirical quantiles

Empirical CDF and quantile function:

$$\hat{F}_n(z) = \frac{1}{n} \sum_{i=1}^{n} \mathbb{1}\left[Z_i \leq z\right], \qquad \hat{Q}_n(\alpha) = Z_{(\lceil \alpha n \rceil)}$$

Lemma (1)

Suppose $Z_1, \ldots, Z_n$ are exchangeable random variables. For any $\alpha \in \{0, 1\}$,

$$\mathbb{P}\left[Z_n \leq \hat{Q}_n(\alpha)\right] \geq \alpha.$$

Moreover, if $Z_1, \ldots, Z_n$ are a.s. distinct,

$$\mathbb{P}\left[Z_n \leq \hat{Q}_n(\alpha)\right] \leq \alpha + \frac{1}{n}.$$

Proof

See whiteboard.

Ref: [Romano et al., 2019b]

Inflation of quantiles

Lemma (2)

Suppose $Z_1, \dots, Z_{n+1}$ are exchangeable random variables.
For any $\alpha \in \{0, 1\}$, define α_n as:

$$\alpha_n = \left(1 + \frac{1}{n}\right) \alpha.$$

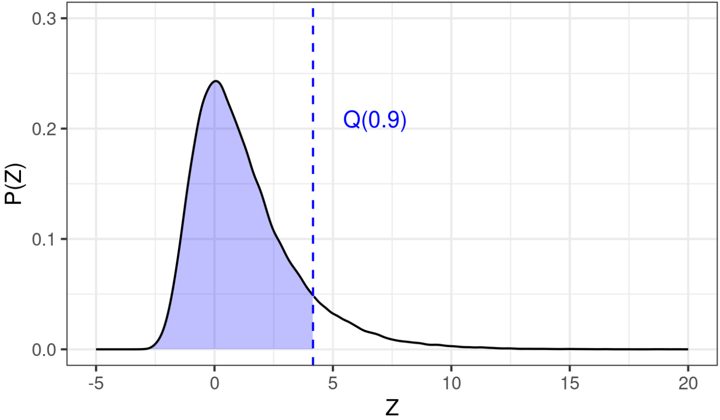
Then,

$$\mathbb{P}\left[Z_{n+1} \leq \hat{Q}_n(\alpha_n)\right] \geq \alpha.$$

Moreover, if $Z_1, \dots, Z_{n+1}$ are a.s. distinct,

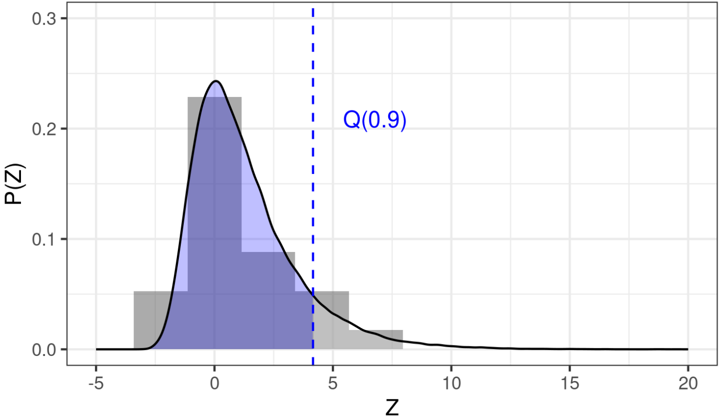
$$\mathbb{P}\left[Z_{n+1} \leq \hat{Q}_n(\alpha_n)\right] \leq \alpha + \frac{1}{n}.$$

Inflation of quantiles (intuition)



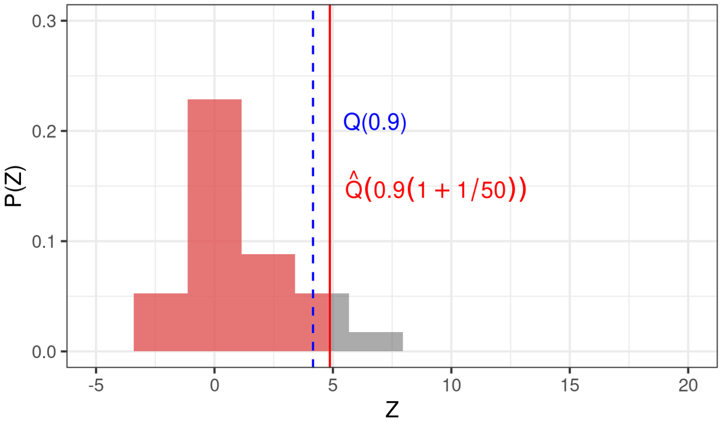
Inflation of quantiles (intuition)

Data set of size 50



Inflation of quantiles (intuition)

Data set of size 50



Proof

See whiteboard.

Ref: [Romano et al., 2019b]

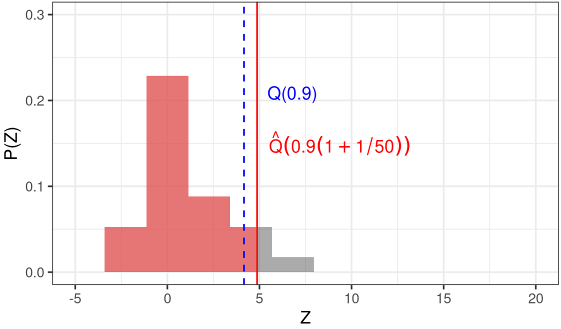
One-sided prediction interval without covariates

Suppose $Z_1, \ldots, Z_{n+1}$ are exchangeable random variables. For any $\alpha \in \{0, 1\}$, define $\hat{C}_\alpha$ as

$$\hat{C}_\alpha = (-\infty, \hat{Q}_n(\alpha_n)].$$

Then,

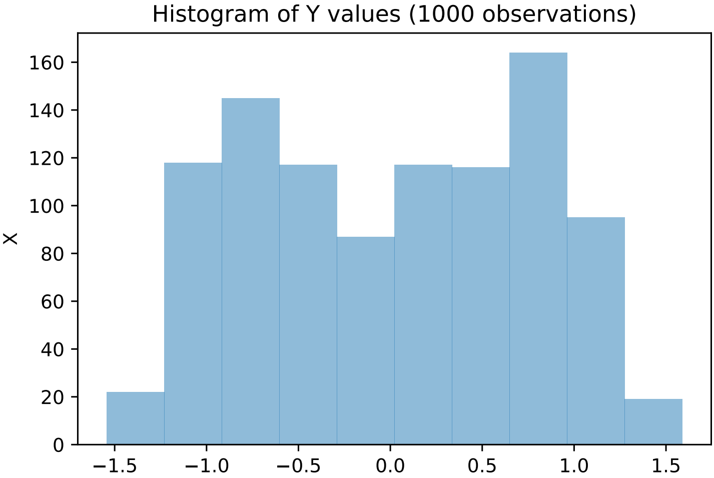
$$\alpha \leq \mathbb{P}\left[Z_{n+1} \in \hat{C}_\alpha\right] \leq \alpha + \frac{1}{n}.$$



Chapter 3:
Split Conformal Prediction

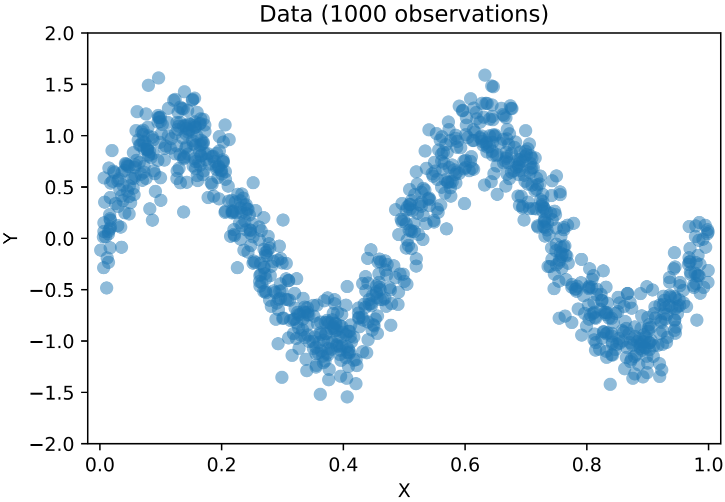
Prediction with covariates

We would like to predict a variable Y...



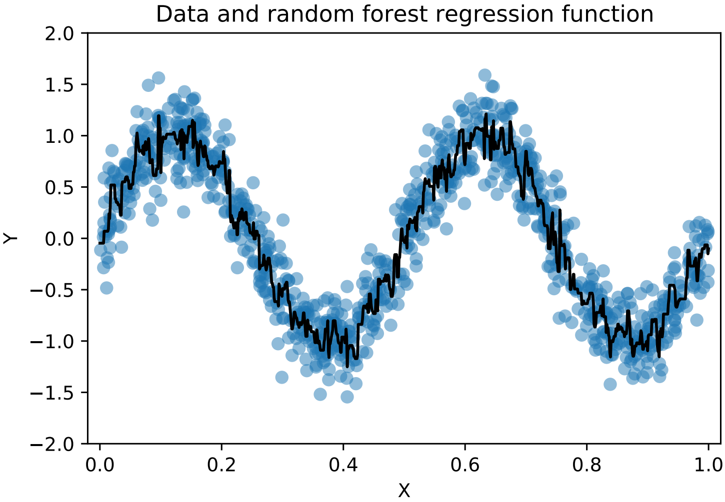
Prediction with covariates

We would like to predict a variable Y... **using some covariates** X.



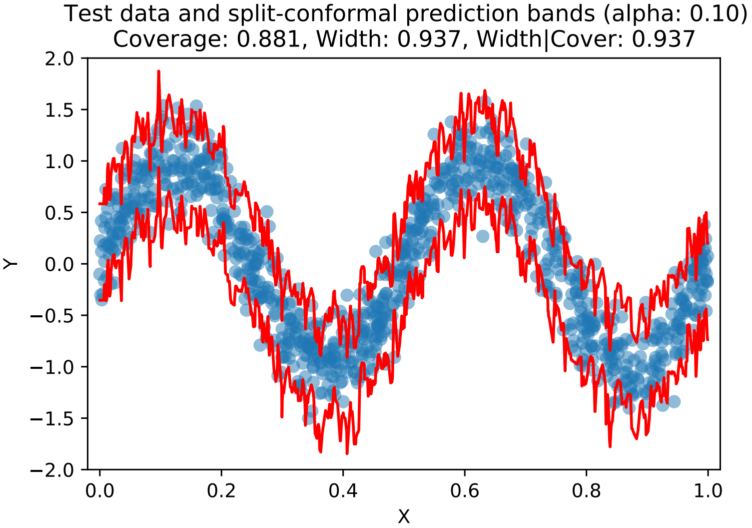
Machine-learning prediction

Lots of machine-learning algorithms. But how confident are we?



Machine-learning prediction

Lots of machine-learning algorithms. But how confident are we?



Conformal prediction

Key ideas:

1. Use ML to project project the problem into 1 dimension (evaluate residuals).
2. Apply the empirical quantile lemmas presented earlier to predict the (absolute) residual of the test point.
3. Some kind of data hold-out is needed to ensure exchangeability with the test data (avoid overfitting).

This is a general recipe, many different variations are possible.

Split-conformal prediction

Algorithm 1: Split-conformal prediction

1: **Input**: Data $\{(X_i, Y_i)\}_{i=1}^{n}$, test point X_{n+1}, $\alpha \in (0, 1)$

2: black-box model $\mathcal{B}$, level $\alpha \in (0, 1)$

Split-conformal prediction

Algorithm 1: Split-conformal prediction

Input: Data $\{(X_i, Y_i)\}_{i=1}^n$, test point X_{n+1}, $\alpha \in (0, 1)$
black-box model $\mathcal{B}$, level $\alpha \in (0, 1)$

Split the data: $\mathcal{I}_1 = \{1, \dots, n/2\}, \mathcal{I}_2 = \{n/2 + 1, \dots, n\}$

Split-conformal prediction

Algorithm 1: Split-conformal prediction

Input: Data $\{(X_i, Y_i)\}_{i=1}^n$, test point X_{n+1}, $\alpha \in (0,1)$
black-box model $\mathcal{B}$, level $\alpha \in (0,1)$

Split the data: $\mathcal{I}_1 = \{1, \ldots, n/2\}, \mathcal{I}_2 = \{n/2+1, \ldots, n\}$
Train $\mathcal{B}$ on $\mathcal{I}_1 : \mathcal{B}(\{(X_i, Y_i)\}_{i \in \mathcal{I}_1}) \to \hat{f}$

Split-conformal prediction

Algorithm 1: Split-conformal prediction

Input: Data $\{(X_i, Y_i)\}_{i=1}^n$, test point X_{n+1}, $\alpha \in (0,1)$
black-box model $\mathcal{B}$, level $\alpha \in (0,1)$

Split the data: $\mathcal{I}_1 = \{1, \ldots, n/2\}, \mathcal{I}_2 = \{n/2+1, \ldots, n\}$
Train $\mathcal{B}$ on $\mathcal{I}_1 : \mathcal{B}(\{(X_i, Y_i)\}_{i \in \mathcal{I}_1}) \to \hat{f}$
Evaluate residuals on $\mathcal{I}_2 : Z_i = |Y_i - \hat{f}(X_i)|$, for all $i \in \mathcal{I}_2$

Split-conformal prediction

Algorithm 1: Split-conformal prediction

Input: Data $\{(X_i, Y_i)\}_{i=1}^{n}$, test point X_{n+1}, $\alpha \in (0,1)$
black-box model $\mathcal{B}$, level $\alpha \in (0,1)$

Split the data: $\mathcal{I}_1 = \{1, \ldots, n/2\}, \mathcal{I}_2 = \{n/2+1, \ldots, n\}$
Train $\mathcal{B}$ on $\mathcal{I}_1 : \mathcal{B}(\{(X_i, Y_i)\}_{i \in \mathcal{I}_1}) \to \hat{f}$
Evaluate residuals on $\mathcal{I}_2 : Z_i = |Y_i - \hat{f}(X_i)|$, for all $i \in \mathcal{I}_2$
Compute $\hat{Q}_n(Z_{\mathcal{I}_2}, \beta_n) = Z_{(\lceil n\beta_n \rceil)}$, where
$\beta_n = (1-\alpha)(1+1/n)$, to predict the next residual.

Split-conformal prediction

Algorithm 1: Split-conformal prediction

1: **Input**: Data $\{(X_i, Y_i)\}_{i=1}^n$, test point X_{n+1}, $\alpha \in (0,1)$
2: black-box model $\mathcal{B}$, level $\alpha \in (0,1)$

3: Split the data: $\mathcal{I}_1 = \{1, \ldots, n/2\}, \mathcal{I}_2 = \{n/2+1, \ldots, n\}$
4: Train $\mathcal{B}$ on $\mathcal{I}_1 : \mathcal{B}(\{(X_i, Y_i)\}_{i \in \mathcal{I}_1}) \to \hat{f}$
5: Evaluate residuals on $\mathcal{I}_2 : Z_i = |Y_i - \hat{f}(X_i)|$, for all $i \in \mathcal{I}_2$
6: Compute $\hat{Q}_n(Z_{\mathcal{I}_2}, \beta_n) = Z_{(\lceil n\beta_n \rceil)}$, where $\beta_n = (1-\alpha)(1+1/n)$, to predict the next residual.

7: **Output**:
$\hat{C}_\alpha(X_{n+1}) = [\hat{f}(X_{n+1}) - \hat{Q}_n(Z_{\mathcal{I}_2}, \beta_n), \hat{f}(X_{n+1}) + \hat{Q}_n(Z_{\mathcal{I}_2}, \beta_n)]$

Split-conformal prediction

Algorithm 1: Split-conformal prediction

1: **Input**: Data $\{(X_i, Y_i)\}_{i=1}^{n}$, test point X_{n+1}, $\alpha \in (0,1)$
2: black-box model $\mathcal{B}$, level $\alpha \in (0,1)$

3: Split the data: $\mathcal{I}_1 = \{1, \ldots, n/2\}, \mathcal{I}_2 = \{n/2+1, \ldots, n\}$
4: Train $\mathcal{B}$ on $\mathcal{I}_1 : \mathcal{B}(\{(X_i, Y_i)\}_{i \in \mathcal{I}_1}) \to \hat{f}$
5: Evaluate residuals on $\mathcal{I}_2 : Z_i = |Y_i - \hat{f}(X_i)|$, for all $i \in \mathcal{I}_2$
6: Compute $\hat{Q}_n(Z_{\mathcal{I}_2}, \beta_n) = Z_{(\lceil n\beta_n \rceil)}$, where
$\beta_n = (1-\alpha)(1+1/n)$, to predict the next residual.

7: **Output**:
$\hat{C}_\alpha(X_{n+1}) = [\hat{f}(X_{n+1}) - \hat{Q}_n(Z_{\mathcal{I}_2}, \beta_n), \hat{f}(X_{n+1}) + \hat{Q}_n(Z_{\mathcal{I}_2}, \beta_n)]$

Why does this work?

$$Y_{n+1} \in \hat{C}_\alpha(X_{n+1}) \quad \Longleftrightarrow \quad Z_{n+1} \leq \hat{Q}_n(Z_{\mathcal{I}_2}, \beta_n).$$

Marginal coverage of split-conformal prediction

Theorem ([Vovk et al., 2005, Lei et al., 2018])

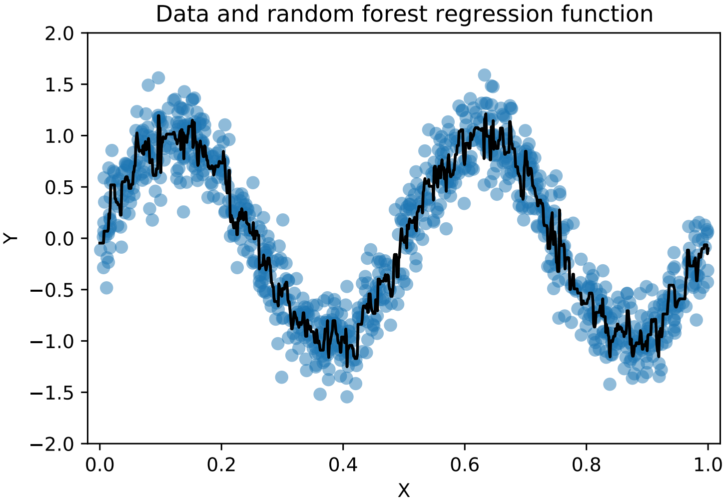
Suppose $(X_1, Y_1), (X_2, Y_2), \dots, (X_{n+1}, Y_{n+1})$ *are exchangeable. Then, the split-conformal prediction intervals* $\hat{C}_\alpha$ *satisfy*

$$\mathbb{P}\left[Y_{n+1} \in \hat{C}_\alpha(X_{n+1})\right] \geq 1 - \alpha.$$

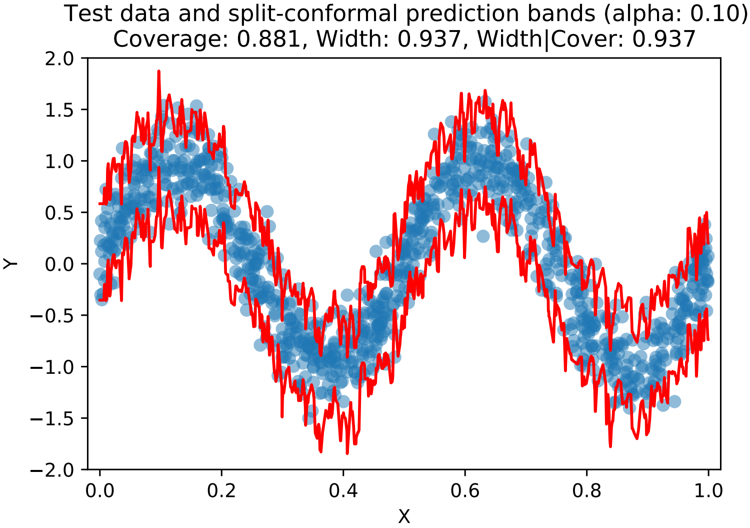
Moreover, if the residuals $\{Z_{n/2+1}, \dots, Z_{n+1}\}$ *are a.s. distinct,*

$$\mathbb{P}\left[Y_{n+1} \in \hat{C}_\alpha(X_{n+1})\right] \leq 1 - \alpha + \frac{1}{n}.$$

Split-conformal prediction bands



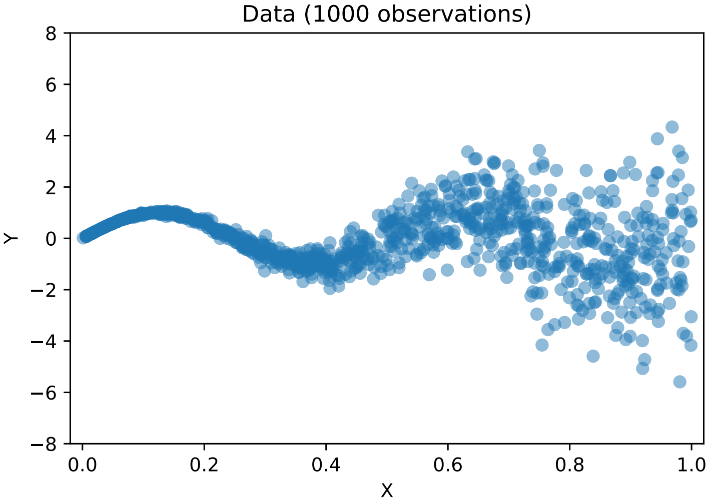
Split-conformal prediction bands



Chapter 4:
Conformalized Quantile Regression

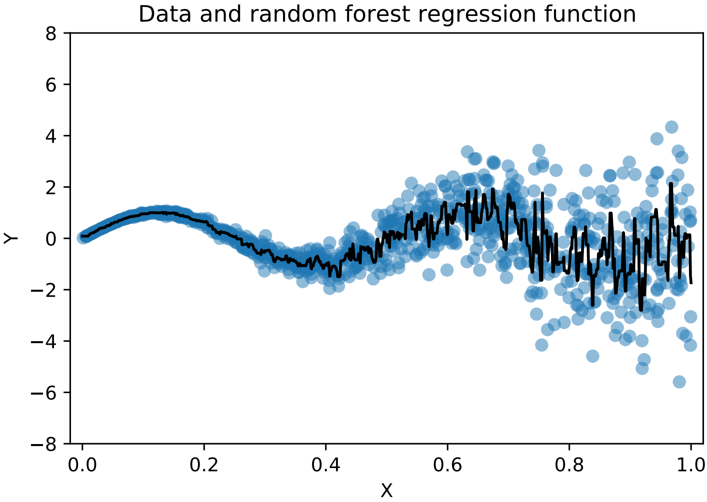
Heteroscedasticity

Suppose now Y heteroscedastic.



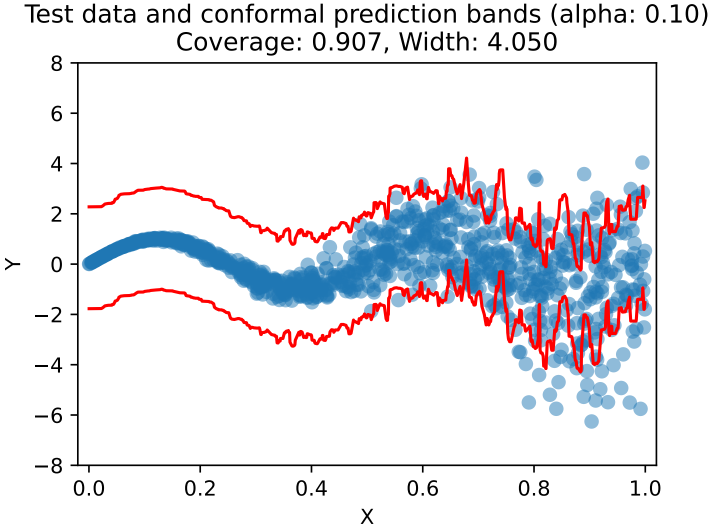
Heteroscedasticity

Suppose now Y heteroscedastic.



Heteroscedasticity

Suppose now Y heteroscedastic.



Conditional quantiles

The goal of quantile regression is to estimate conditional quantiles of $Y \mid X$ instead of the conditional mean, $\mathbb{E}[Y \mid X]$.

$$q_\alpha(x) = \inf\{y \in \mathbb{R} : F(y \mid X = x) \geq \alpha\}$$

Conditional quantiles

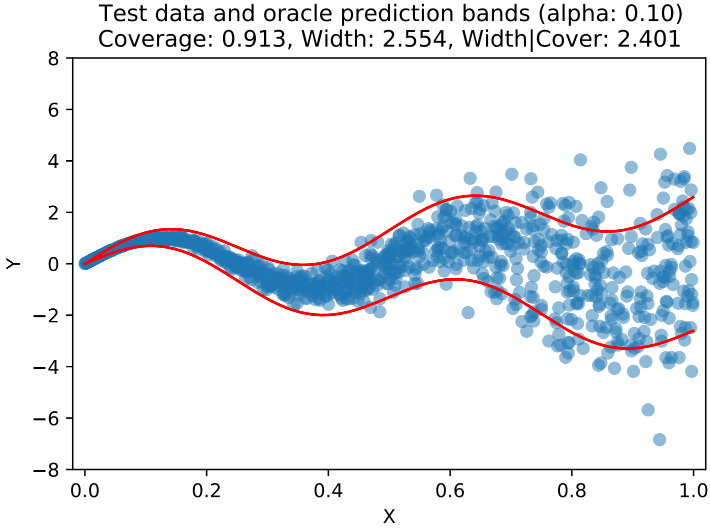
The goal of quantile regression is to estimate conditional quantiles of $Y \mid X$ instead of the conditional mean, $\mathbb{E}[Y \mid X]$.

$$q_\alpha(x) = \inf\{y \in \mathbb{R} : F(y \mid X = x) \geq \alpha\}$$

An oracle that knows $P(Y \mid X)$ would predict as follows:

$$C_\alpha^{\text{oracle}}(Y_{n+1} \mid X_{n+1} = x) = [q_{\alpha/2}(x), q_{1-\alpha/2}(x)].$$

Oracle predictions



Quantile regression

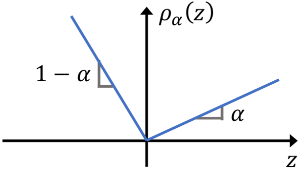
Goal: estimate a function $\hat{q}_\alpha(x)$,

$$\hat{q}_\alpha(x) \approx q_\alpha(x) = \inf \{y \in \mathbb{R} : F(y \mid X = x) \geq \alpha\}$$

Fit the parameters θ_α by minimizing the "pinball" loss:

$$\hat{\theta}_\alpha = \arg\min_\theta \frac{1}{n} \sum_{i=1}^{n} \rho_\alpha \left(Y_i, f_\theta(X_i) \right)$$

$$\rho_\alpha(y, \hat{y}) := \begin{cases} \alpha(y - \hat{y}) & \text{if } y - \hat{y} > 0, \\ (1 - \alpha)(\hat{y} - y) & \text{otherwise} \end{cases}$$



This loss function can be used in a variety of machine-learning models. E.g., linear models, random forests, neural networks, ...

Quantile regression

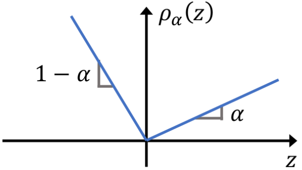
Goal: estimate a function $\hat{q}_\alpha(x)$,

$$\hat{q}_\alpha(x) \approx q_\alpha(x) = \inf\{y \in \mathbb{R} : F(y \mid X = x) \geq \alpha\}$$

Fit the parameters θ_α by minimizing the "pinball" loss:

$$\hat{\theta}_\alpha = \arg\min_\theta \frac{1}{n} \sum_{i=1}^{n} \rho_\alpha\left(Y_i, f_\theta(X_i)\right)$$

$$\rho_\alpha(y, \hat{y}) := \begin{cases} \alpha(y - \hat{y}) & \text{if } y - \hat{y} > 0, \\ (1-\alpha)(\hat{y} - y) & \text{otherwise} \end{cases}$$



This loss function can be used in a variety of machine-learning models. E.g., linear models, random forests, neural networks, ...

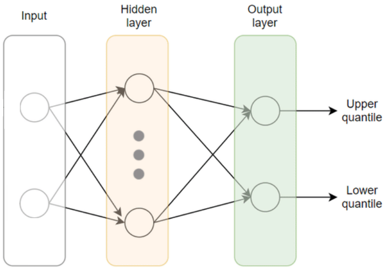
Key idea: Leibniz integral rule

$$q_\alpha(x) = \arg\min_{f(x)} \mathbb{E}\left[\rho_\alpha(Y, f(x))\right]$$

Multiple deep quantile regression models

Goal: estimate two functions, $\hat{q}_{\alpha_{\text{lower}}}(x)$ and $\hat{q}_{\alpha_{\text{upper}}}(x)$,

$$\hat{q}_\alpha(x) \approx q_\alpha(x) = \inf\left\{y \in \mathbb{R} : F(y \mid X = x) \geq \alpha\right\}.$$



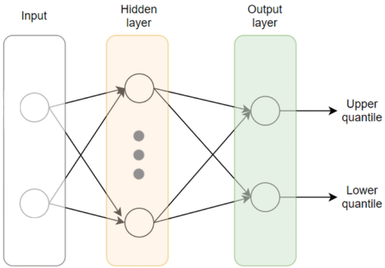
Fit the parameters θ_α by minimizing the "pinball" loss:

$$\hat{\theta}_\alpha = \arg\min_\theta \frac{1}{n} \sum_{i=1}^{n} \rho_\alpha\left(Y_i, f_\theta(X_i)\right).$$

Multiple deep quantile regression models

Goal: estimate two functions, $\hat{q}_{\alpha_{\text{lower}}}(x)$ and $\hat{q}_{\alpha_{\text{upper}}}(x)$,

$$\hat{q}_{\alpha}(x) \approx q_{\alpha}(x) = \inf\left\{y \in \mathbb{R} : F(y \mid X = x) \geq \alpha\right\}.$$



Fit the parameters θ_{α} by minimizing the "pinball" loss:

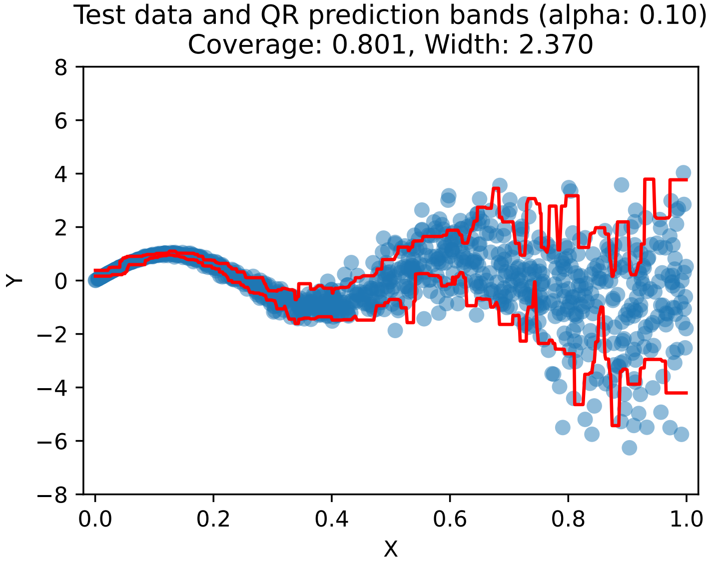
$$\hat{\theta}_{\alpha} = \arg\min_{\theta} \frac{1}{n} \sum_{i=1}^{n} \rho_{\alpha}\left(Y_i, f_{\theta}(X_i)\right).$$

Beware of *quantile crossing*.

E.g., swap $\hat{q}_{\alpha_{\text{lower}}}(x)$ and $\hat{q}_{\alpha_{\text{upper}}}(x)$ if necessary.

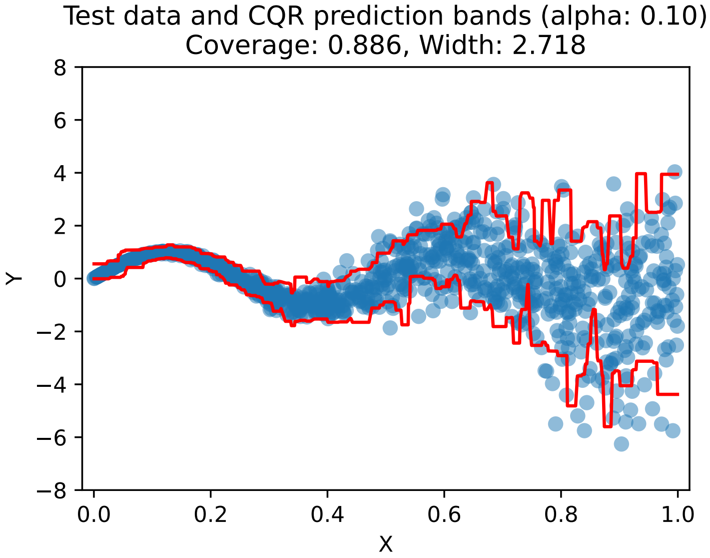
Quantile regression in action

We can fit conditional quantiles, but without guarantees.



Quantile regression in action

We can fit conditional quantiles, but without guarantees.



Generalized residuals for quantile regression

Instead of defining the residuals as

$$Z_i = |Y_i - \hat{f}(X_i)|$$

Generalized residuals for quantile regression

Instead of defining the residuals as

$$Z_i = |Y_i - \hat{f}(X_i)|$$

we are going to define them as:

$$Z_i = \begin{cases} \leq 0 & \text{if } Y_i \in [\hat{q}_{\alpha/2}(X_i), \hat{q}_{1-\alpha/2}(X_i)], \\ > 0 & \text{otherwise,} \end{cases}$$

Generalized residuals for quantile regression

Instead of defining the residuals as

$$Z_i = |Y_i - \hat{f}(X_i)|$$

we are going to define them as:

$$Z_i = \begin{cases} \leq 0 & \text{if } Y_i \in [\hat{q}_{\alpha/2}(X_i), \hat{q}_{1-\alpha/2}(X_i)], \\ > 0 & \text{otherwise}, \end{cases}$$

$$= \begin{cases} Y_i - \hat{q}_{1-\alpha/2}(X_i) & \text{if } Y_i > \hat{q}_{1-\alpha/2}(X_i), \\ \hat{q}_{\alpha/2}(X_i) - Y_i & \text{if } Y_i < \hat{q}_{\alpha/2}(X_i), \\ \max\left\{ Y_i - \hat{q}_{1-\alpha/2}(X_i), \hat{q}_{\alpha/2}(X_i) - Y_i \right\} & \text{otherwise.} \end{cases}$$

Generalized residuals for quantile regression

Instead of defining the residuals as

$$Z_i = |Y_i - \hat{f}(X_i)|$$

we are going to define them as:

$$
\begin{aligned}
Z_i &= \begin{cases} \leq 0 & \text{if } Y_i \in [\hat{q}_{\alpha/2}(X_i), \hat{q}_{1-\alpha/2}(X_i)], \\ > 0 & \text{otherwise,} \end{cases} \\
&= \begin{cases} Y_i - \hat{q}_{1-\alpha/2}(X_i) & \text{if } Y_i > \hat{q}_{1-\alpha/2}(X_i), \\ \hat{q}_{\alpha/2}(X_i) - Y_i & \text{if } Y_i < \hat{q}_{\alpha/2}(X_i), \\ \max\left\{ Y_i - \hat{q}_{1-\alpha/2}(X_i), \hat{q}_{\alpha/2}(X_i) - Y_i \right\} & \text{otherwise.} \end{cases}
\end{aligned}
$$

Compact notation (equivalent):

$$Z_i = \max\left\{ Y_i - \hat{q}_{1-\alpha/2}(X_i), \hat{q}_{\alpha/2}(X_i) - Y_i \right\}.$$

Split-conformal + quantile regression

Algorithm 2: Split-conformal quantile regression

Input: Data $\{(X_i, Y_i)\}_{i=1}^{n}$, test point X_{n+1}, $\alpha \in (0, 1)$
black-box QR model $\mathcal{B}$, level $\alpha \in (0, 1)$

Split-conformal + quantile regression

Algorithm 2: Split-conformal quantile regression

1: **Input**: Data $\{(X_i, Y_i)\}_{i=1}^n$, test point X_{n+1}, $\alpha \in (0, 1)$
2: black-box QR model $\mathcal{B}$, level $\alpha \in (0, 1)$

3: Split the data: $\mathcal{I}_1 = \{1, \ldots, n/2\}, \mathcal{I}_2 = \{n/2 + 1, \ldots, n\}$

Split-conformal + quantile regression

Algorithm 2: Split-conformal quantile regression

Input: Data $\{(X_i, Y_i)\}_{i=1}^{n}$, test point X_{n+1}, $\alpha \in (0, 1)$
black-box QR model $\mathcal{B}$, level $\alpha \in (0, 1)$

Split the data: $\mathcal{I}_1 = \{1, \ldots, n/2\}, \mathcal{I}_2 = \{n/2 + 1, \ldots, n\}$
Train $\mathcal{B}$ on $\mathcal{I}_1$: $\mathcal{B}\left(\{(X_i, Y_i)\}_{i \in \mathcal{I}_1}\right) \rightarrow (\hat{q}_{\alpha/2}, \hat{q}_{1-\alpha/2})$

Split-conformal + quantile regression

Algorithm 2: Split-conformal quantile regression

1: **Input**: Data $\{(X_i, Y_i)\}_{i=1}^n$, test point X_{n+1}, $\alpha \in (0,1)$

2: black-box QR model $\mathcal{B}$, level $\alpha \in (0,1)$

3: Split the data: $\mathcal{I}_1 = \{1, \ldots, n/2\}, \mathcal{I}_2 = \{n/2+1, \ldots, n\}$

4: Train $\mathcal{B}$ on $\mathcal{I}_1$: $\mathcal{B}(\{(X_i, Y_i)\}_{i \in \mathcal{I}_1}) \to (\hat{q}_{\alpha/2}, \hat{q}_{1-\alpha/2})$

5: Evaluate residuals (*conformity scores*) on $\mathcal{I}_2$:

$$Z_i = \max\left\{ Y_i - \hat{q}_{1-\alpha/2}(X_i), \hat{q}_{\alpha/2}(X_i) - Y_i \right\}$$

Split-conformal + quantile regression

Algorithm 2: Split-conformal quantile regression

Input: Data $\{(X_i, Y_i)\}_{i=1}^n$, test point X_{n+1}, $\alpha \in (0,1)$
black-box QR model $\mathcal{B}$, level $\alpha \in (0,1)$

Split the data: $\mathcal{I}_1 = \{1, \dots, n/2\}, \mathcal{I}_2 = \{n/2+1, \dots, n\}$
Train $\mathcal{B}$ on $\mathcal{I}_1$: $\mathcal{B}\left(\{(X_i, Y_i)\}_{i \in \mathcal{I}_1}\right) \to (\hat{q}_{\alpha/2}, \hat{q}_{1-\alpha/2})$
Evaluate residuals (*conformity scores*) on $\mathcal{I}_2$:

$$Z_i = \max\left\{Y_i - \hat{q}_{1-\alpha/2}(X_i), \hat{q}_{\alpha/2}(X_i) - Y_i\right\}$$

Compute $\hat{Q}_n(Z_{\mathcal{I}_2}, \beta_n) = Z_{(\lceil n\beta_n \rceil)}$, where $\beta_n = (1-\alpha)(1+1/n)$

Split-conformal + quantile regression

Algorithm 2: Split-conformal quantile regression

Input: Data $\{(X_i, Y_i)\}_{i=1}^{n}$, test point X_{n+1}, $\alpha \in (0,1)$
black-box QR model $\mathcal{B}$, level $\alpha \in (0,1)$

Split the data: $\mathcal{I}_1 = \{1, \ldots, n/2\}, \mathcal{I}_2 = \{n/2+1, \ldots, n\}$
Train $\mathcal{B}$ on $\mathcal{I}_1 : \mathcal{B}\left(\{(X_i, Y_i)\}_{i \in \mathcal{I}_1}\right) \to (\hat{q}_{\alpha/2}, \hat{q}_{1-\alpha/2})$
Evaluate residuals (*conformity scores*) on $\mathcal{I}_2$:

$$Z_i = \max\left\{ Y_i - \hat{q}_{1-\alpha/2}(X_i), \hat{q}_{\alpha/2}(X_i) - Y_i \right\}$$

Compute $\hat{Q}_n(Z_{\mathcal{I}_2}, \beta_n) = Z_{(\lceil n\beta_n \rceil)}$, where $\beta_n = (1-\alpha)(1+1/n)$

Output: $\hat{C}_\alpha(X_{n+1}) =$
$[\hat{q}_{\alpha/2}(X_{n+1}) - \hat{Q}_n(Z_{\mathcal{I}_2}, \beta_n), \hat{q}_{1-\alpha/2}(X_{n+1}) + \hat{Q}_n(Z_{\mathcal{I}_2}, \beta_n)]$

Split-conformal + quantile regression

Algorithm 2: Split-conformal quantile regression

1: **Input**: Data $\{(X_i, Y_i)\}_{i=1}^n$, test point X_{n+1}, $\alpha \in (0,1)$
2: black-box QR model $\mathcal{B}$, level $\alpha \in (0,1)$
3: Split the data: $\mathcal{I}_1 = \{1, \dots, n/2\}, \mathcal{I}_2 = \{n/2+1, \dots, n\}$
4: Train $\mathcal{B}$ on $\mathcal{I}_1$: $\mathcal{B}(\{(X_i, Y_i)\}_{i \in \mathcal{I}_1}) \to (\hat{q}_{\alpha/2}, \hat{q}_{1-\alpha/2})$
5: Evaluate residuals (*conformity scores*) on $\mathcal{I}_2$:

$$Z_i = \max\left\{ Y_i - \hat{q}_{1-\alpha/2}(X_i), \hat{q}_{\alpha/2}(X_i) - Y_i \right\}$$

6: Compute $\hat{Q}_n(Z_{\mathcal{I}_2}, \beta_n) = Z_{(\lceil n\beta_n \rceil)}$, where $\beta_n = (1-\alpha)(1+1/n)$
7: **Output**: $\hat{C}_\alpha(X_{n+1}) =$
$[\hat{q}_{\alpha/2}(X_{n+1}) - \hat{Q}_n(Z_{\mathcal{I}_2}, \beta_n), \hat{q}_{1-\alpha/2}(X_{n+1}) + \hat{Q}_n(Z_{\mathcal{I}_2}, \beta_n)]$

Why does this work? Same story as before.

$$Y_{n+1} \in \hat{C}_\alpha(X_{n+1}) \quad \Longleftrightarrow \quad Z_{n+1} \le \hat{Q}_n(Z_{\mathcal{I}_2}, \beta_n).$$

Marginal coverage of split-conformal prediction

Theorem ([Romano et al., 2019b])

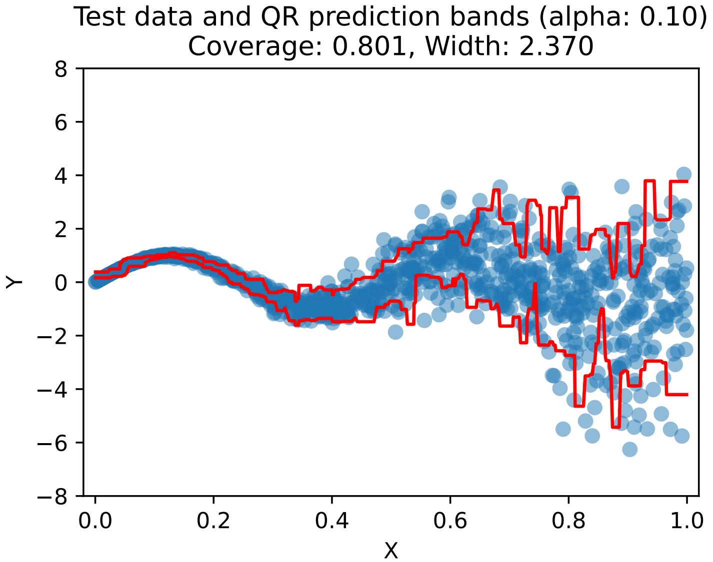
Suppose $(X_1, Y_1), (X_2, Y_2), \ldots, (X_{n+1}, Y_{n+1})$ *are exchangeable. Then, the split-conformal QR prediction intervals* $\hat{C}_\alpha$ *satisfy*

$$\mathbb{P}\left[Y_{n+1} \in \hat{C}_\alpha(X_{n+1})\right] \geq 1 - \alpha.$$

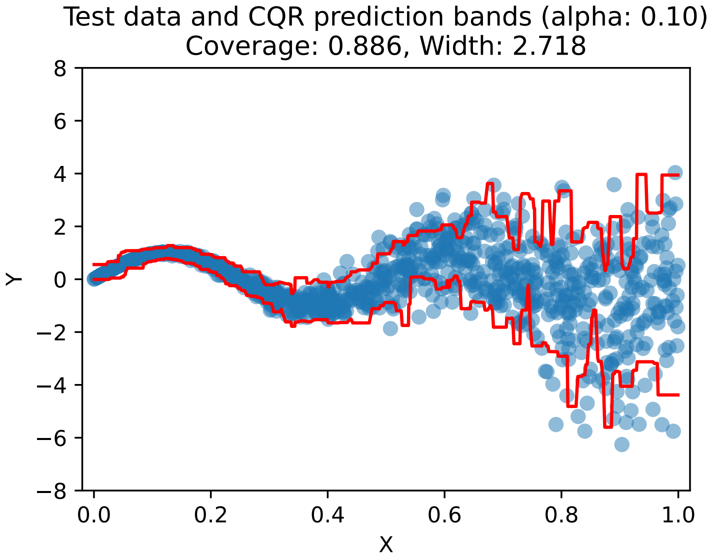
Moreover, if the residuals $\{Z_{n/2+1}, \ldots, Z_{n+1}\}$ *are a.s. distinct,*

$$\mathbb{P}\left[Y_{n+1} \in \hat{C}_\alpha(X_{n+1})\right] \leq 1 - \alpha + \frac{1}{n}.$$

Conformal quantile regression



Conformal quantile regression



Computer session I

Chapter 5:
Beyond marginal coverage

Efficiency of conformal quantile regression

Theorem ([Sesia and Candès, 2020])

(A1) Assume $(X_1, Y_1), (X_2, Y_2), \ldots, (X_{n+1}, Y_{n+1})$ *are i.i.d.*

Efficiency of conformal quantile regression

Theorem ([Sesia and Candès, 2020])

(A1) Assume $(X_1, Y_1), (X_2, Y_2), \ldots, (X_{n+1}, Y_{n+1})$ *are i.i.d.*

(A2) Assume that

$$\mathbb{P}\left[\mathbb{E}\left[\left(\hat{q}_{\alpha/2}(X) - q_{\alpha/2}(X)\right)^2 \mid \hat{q}_{\alpha/2}, \hat{q}_{1-\alpha/2}\right] \leq \eta_n\right] \geq 1 - \rho_n,$$

$$\mathbb{P}\left[\mathbb{E}\left[\left(\hat{q}_{1-\alpha/2}(X) - q_{1-\alpha/2}(X)\right)^2 \mid \hat{q}_{\alpha/2}, \hat{q}_{1-\alpha/2}\right] \leq \eta_n\right] \geq 1 - \rho_n,$$

for some sequences $\eta_n = o(1)$ *and* $\rho_n = o(1)$*, as* $n \to \infty$*.*

Efficiency of conformal quantile regression

Theorem ([Sesia and Candès, 2020])

(A1) Assume $(X_1, Y_1), (X_2, Y_2), \ldots, (X_{n+1}, Y_{n+1})$ *are i.i.d.*

(A2) Assume that

$$\mathbb{P}\left[\mathbb{E}\left[\left(\hat{q}_{\alpha/2}(X) - q_{\alpha/2}(X)\right)^2 \mid \hat{q}_{\alpha/2}, \hat{q}_{1-\alpha/2}\right] \leq \eta_n\right] \geq 1 - \rho_n,$$
$$\mathbb{P}\left[\mathbb{E}\left[\left(\hat{q}_{1-\alpha/2}(X) - q_{1-\alpha/2}(X)\right)^2 \mid \hat{q}_{\alpha/2}, \hat{q}_{1-\alpha/2}\right] \leq \eta_n\right] \geq 1 - \rho_n,$$

for some sequences $\eta_n = o(1)$ *and* $\rho_n = o(1)$*, as* $n \to \infty$*.*

(A3) Assume that the probability density of the conformity scores is bounded away from zero in an open neighborhood of zero.

Efficiency of conformal quantile regression

Theorem ([Sesia and Candès, 2020])

(A1) Assume $(X_1, Y_1), (X_2, Y_2), \ldots, (X_{n+1}, Y_{n+1})$ *are i.i.d.*

(A2) Assume that

$$\mathbb{P}\left[\mathbb{E}\left[\left(\hat{q}_{\alpha/2}(X) - q_{\alpha/2}(X)\right)^2 \mid \hat{q}_{\alpha/2}, \hat{q}_{1-\alpha/2}\right] \leq \eta_n\right] \geq 1 - \rho_n,$$

$$\mathbb{P}\left[\mathbb{E}\left[\left(\hat{q}_{1-\alpha/2}(X) - q_{1-\alpha/2}(X)\right)^2 \mid \hat{q}_{\alpha/2}, \hat{q}_{1-\alpha/2}\right] \leq \eta_n\right] \geq 1 - \rho_n,$$

for some sequences $\eta_n = o(1)$ *and* $\rho_n = o(1)$*, as* $n \to \infty$.

(A3) Assume that the probability density of the conformity scores is bounded away from zero in an open neighborhood of zero.

Then,

$$\mathcal{L}\left(\hat{C}_\alpha(X_{n+1}) \triangle C_\alpha^{\text{oracle}}(X_{n+1})\right) = o_{\mathbb{P}}(1),$$

where $\mathcal{L}$ *is the Lebesgue measure and* $A \triangle B = (A \setminus B) \cup (B \setminus A)$.

Asymptotic conditional coverage

Definition (Asymptotic conditional coverage)

We say that a sequence $\hat{C}_n$ of random prediction bands has asymptotic conditional coverage at the level $1-\alpha$ if there exists a sequence of random sets $\Lambda_n \subseteq \mathbb{R}^d$ such that

$$\mathbb{P}\left[X \in \Lambda_n\right] = 1 - o_{\mathbb{P}}(1)$$

and

$$\sup_{x \in \Lambda_n} \left| \mathbb{P}\left[Y \in \hat{C}_n(x) \mid X = x \right] - (1-\alpha) \right| = o_{\mathbb{P}}(1).$$

Asymptotic conditional coverage for CQR (under consistency and regularity assumptions) follows immediately from previous theorem.

Approximate finite-sample conditional coverage?

Is it possible to achieve finite-sample conditional coverage?

$$\mathbb{P}\left[Y_{n+1} \in \hat{C}_n^?(x) \mid X_{n+1} = x\right] \geq 1 - \alpha, \qquad \forall x$$

Approximate finite-sample conditional coverage?

Is it possible to achieve finite-sample conditional coverage?

$$\mathbb{P}\left[Y_{n+1} \in \hat{C}_n^{?}(x) \mid X_{n+1} = x\right] \geq 1 - \alpha, \qquad \forall x$$

No.

Proposition ([Vovk, 2012, Lei et al., 2013])

Suppose $\hat{C}_n$ satisfies conditional coverage at level α. Then,

$$\mathbb{E}\left[\mathcal{L}(\hat{C}_n(X_{n+1})\right] = +\infty$$

unless

$$\mathbb{P}\left[X_{n+1} = x\right] > 0.$$

Finite-sample conditional coverage?

Is approximate finite-sample conditional coverage possible?
Fix $\delta \in (0,1)$. Can we obtain the following in a non-trivial way?

$$\mathbb{P}\left[Y_{n+1} \in \hat{C}_n^{?}(x) \mid X_{n+1} \in \mathcal{X}\right] \geq 1-\alpha,$$
$$\forall \mathcal{X} \subseteq \mathcal{R}^d : \mathbb{P}\left[X_{n+1} \in \mathcal{X}\right] \geq \delta$$

Finite-sample conditional coverage?

Is approximate finite-sample conditional coverage possible?
Fix $\delta \in (0,1)$. Can we obtain the following in a non-trivial way?

$$\mathbb{P}\left[Y_{n+1} \in \hat{C}_n^?(x) \mid X_{n+1} \in \mathcal{X}\right] \geq 1-\alpha,$$
$$\forall \mathcal{X} \subseteq \mathcal{R}^d : \mathbb{P}\left[X_{n+1} \in \mathcal{X}\right] \geq \delta$$

An easy way to achieve this is to seek marginal coverage at level

$$1-\alpha\delta$$

However, this is extremely conservative.

Finite-sample conditional coverage?

Is approximate finite-sample conditional coverage possible?
Fix $\delta \in (0, 1)$. Can we obtain the following in a non-trivial way?

$$\mathbb{P}\left[Y_{n+1} \in \hat{C}_n^?(x) \mid X_{n+1} \in \mathcal{X}\right] \geq 1 - \alpha,$$
$$\forall \mathcal{X} \subseteq \mathcal{R}^d : \mathbb{P}\left[X_{n+1} \in \mathcal{X}\right] \geq \delta$$

An easy way to achieve this is to seek marginal coverage at level

$$1 - \alpha\delta$$

However, this is extremely conservative.

Sadly, [Foygel Barber et al., 2020] prove this is also the best way.

Coverage conditional on a discrete variable [Romano et al., 2019a]

Suppose $X_i = (X_{i,1}, X_{i,2}) \in \mathbb{R} \times \{0, 1\}$.
It's easy to obtain coverage conditional on the discrete variable.

$$\mathbb{P}\left[Y_{n+1} \in \hat{C}_n^?(x) \mid X_{n+1} \in \mathbb{R} \times \{k\}\right] \geq 1 - \alpha, \qquad \forall k \in \{0, 1\}$$

Coverage conditional on a discrete variable [Romano et al., 2019a]

Suppose $X_i = (X_{i,1}, X_{i,2}) \in \mathbb{R} \times \{0, 1\}$.
It's easy to obtain coverage conditional on the discrete variable.

$$\mathbb{P}\left[Y_{n+1} \in \hat{C}_n^?(x) \mid X_{n+1} \in \mathbb{R} \times \{k\}\right] \geq 1 - \alpha, \qquad \forall k \in \{0, 1\}$$

Compute quantiles of conformity scores separately for each class.
For $k \in \{0, 1\}$, use only the data in

$$\mathcal{I}_{2,k} = \{i \in \mathcal{I}_2 : X_{i,2} = k\},$$

The predictions will use the $\hat{Q}$ corresponding to the k of $X_{n+1,2}$.

Relaxed conditional coverage [Foygel Barber et al., 2020]

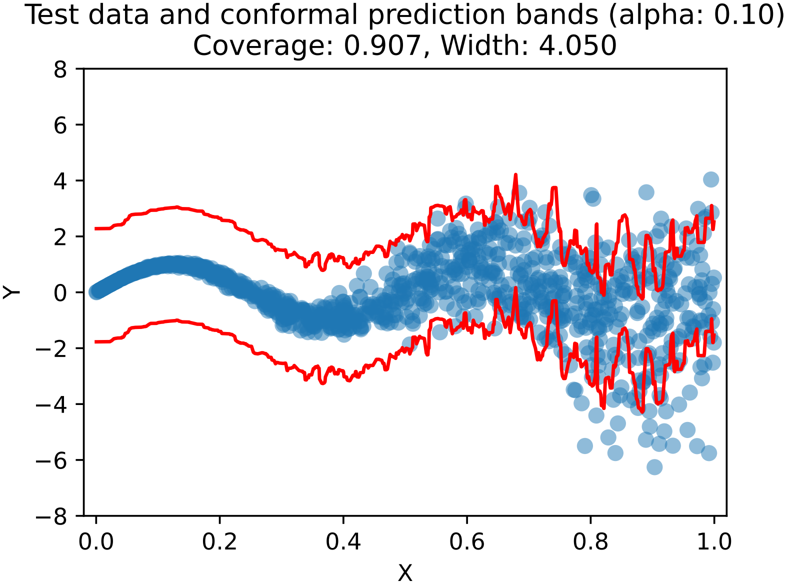
Similar idea can also be used with continuous variables,
conditioning on a ball around a certain point.
However, this will greatly reduce the effective sample size.

Relaxed conditional coverage [Foygel Barber et al., 2020]

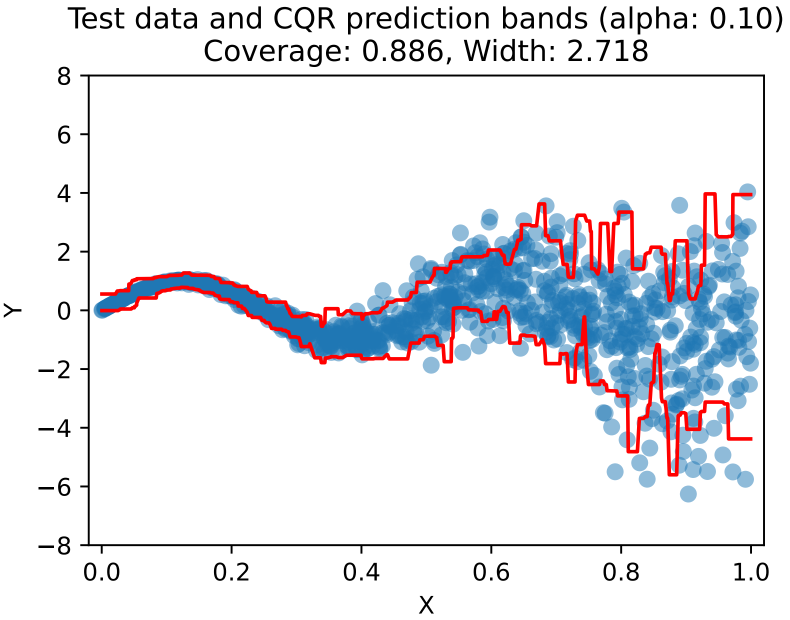
Similar idea can also be used with continuous variables, conditioning on a ball around a certain point.
However, this will greatly reduce the effective sample size.

In the end, we typically settle for marginal coverage in theory, but we can design the algorithm carefully to seek good conditional coverage in practice.

CQR can improve conditional coverage in practice



CQR can improve conditional coverage in practice



Worst-slab coverage [Cauchois et al., 2020]

How can we measure conditional coverage?

Fix a vector $v \in \mathbb{R}^p$ and two scalars $a < b$. Then, define

$$S_{v,a,b} = \{x \in \mathbb{R}^p : a \leq v^T x \leq b\}$$

For any fixed prediction set $\hat{\mathcal{C}}$ and $\delta \in (0,1)$, define

$$\mathrm{WSC}(\hat{\mathcal{C}}; \delta) = \inf_{v \in \mathbb{R}^p,\ a < b \in \mathbb{R}} \left\{ \mathbb{P}[Y \in \hat{\mathcal{C}}(X) \mid X \in S_{v,a,b}] \text{ s.t. } \mathbb{P}[X \in S_{v,a,b}] \geq 1 - \delta \right\}.$$

Worst-slab coverage [Cauchois et al., 2020]

How can we measure conditional coverage?

Fix a vector $v \in \mathbb{R}^p$ and two scalars $a < b$. Then, define

$$S_{v,a,b} = \{x \in \mathbb{R}^p : a \leq v^T x \leq b\}$$

For any fixed prediction set $\hat{\mathcal{C}}$ and $\delta \in (0,1)$, define

$$\mathrm{WSC}(\hat{\mathcal{C}}; \delta) = \inf_{v \in \mathbb{R}^p,\, a < b \in \mathbb{R}} \left\{ \mathbb{P}[Y \in \hat{\mathcal{C}}(X) \mid X \in S_{v,a,b}] \text{ s.t. } \mathbb{P}[X \in S_{v,a,b}] \geq 1 - \delta \right\}.$$

Can be approximated by estimating v^*, a^*, b^* on hold-out data.
[Romano et al., 2020]

Chapter 6:
Split Conformal Classification

The classification problem

Suppose $Y_i \in \{1, 2, \ldots, C\}$ is a *categorical* variable.

$$\mathbb{P}\left[Y_{n+1} \in \hat{\mathcal{C}}_n(X_{n+1})\right] \geq 1 - \alpha.$$

The previous residuals (or conformity scores) no longer make sense.

The classification problem

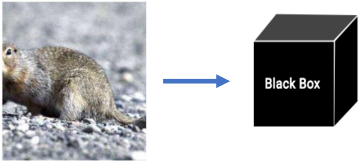
Suppose $Y_i \in \{1, 2, \ldots, C\}$ is a *categorical* variable.

$$\mathbb{P}\left[Y_{n+1} \in \hat{\mathcal{C}}_n(X_{n+1})\right] \geq 1 - \alpha.$$

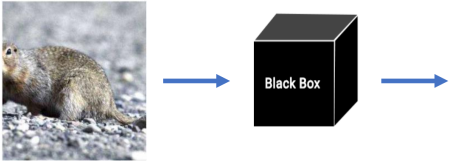
The previous residuals (or conformity scores) no longer make sense.

$C($$) = \{5, 6\}$

Uncertainty estimation via calibrated prediction sets

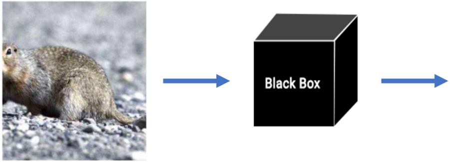


Uncertainty estimation via calibrated prediction sets



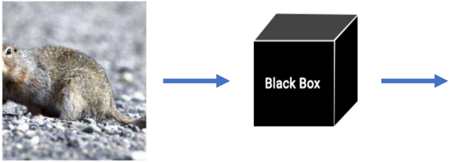
{Marmot, Fox Squirrel, Mink, Weasel}

Uncertainty estimation via calibrated prediction sets



{Marmot, Fox Squirrel, Mink, Weasel}
with 90% coverage

Uncertainty estimation via calibrated prediction sets



{Marmot, Fox Squirrel, Mink, Weasel}
with 90% coverage

Ideal goal: prediction sets with conditional coverage.

$$\mathbb{P}\left[Y_{n+1} \in \hat{C}_n(X_{n+1}) \mid X_{n+1} = x\right] \geq 1 - \alpha, \qquad \forall x$$

Uncertainty estimation via calibrated prediction sets

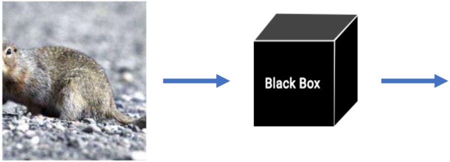
{Marmot, Fox Squirrel, Mink, Weasel}
with 90% coverage

Ideal goal: prediction sets with <u>conditional</u> coverage.

$$\mathbb{P}\left[Y_{n+1} \in \hat{C}_n(X_{n+1}) \mid X_{n+1} = x\right] \geq 1 - \alpha, \qquad \forall x$$

Unfortunately, this is impossible to achieve in finite samples without very strong assumptions. [Barber et al. (2021)]

Uncertainty estimation via calibrated prediction sets



{Marmot, Fox Squirrel, Mink, Weasel}
with 90% coverage

Ideal goal: prediction sets with conditional coverage.

$$\mathbb{P}\left[Y_{n+1} \in \hat{\mathcal{C}}_n(X_{n+1}) \mid X_{n+1} = x\right] \geq 1 - \alpha, \qquad \forall x$$

Unfortunately, this is impossible to achieve in finite samples without very strong assumptions. [Barber et al. (2021)]

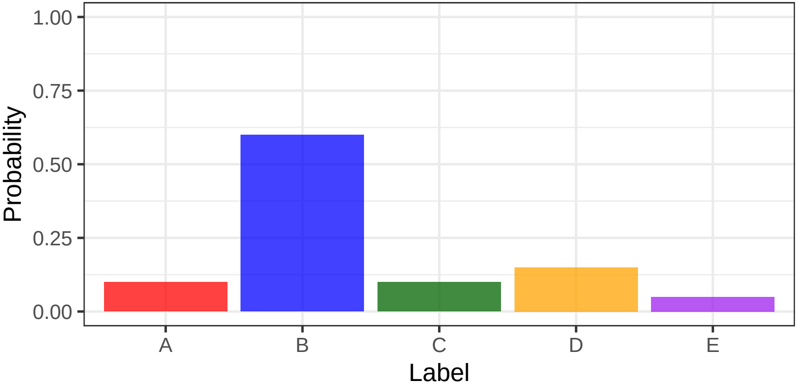
Practical goal: prediction sets with marginal coverage.

$$\mathbb{P}\left[Y_{n+1} \in \hat{\mathcal{C}}_n(X_{n+1}) \mid X_{n+1}\right] \geq 1 - \alpha \quad (\text{e.g.,} = 90\%)$$

Prediction sets for classification: the ideal approach

Conditional class probabilities (oracle):

$$\pi_y(x) = \mathbb{P}[Y = y \mid X = x]$$

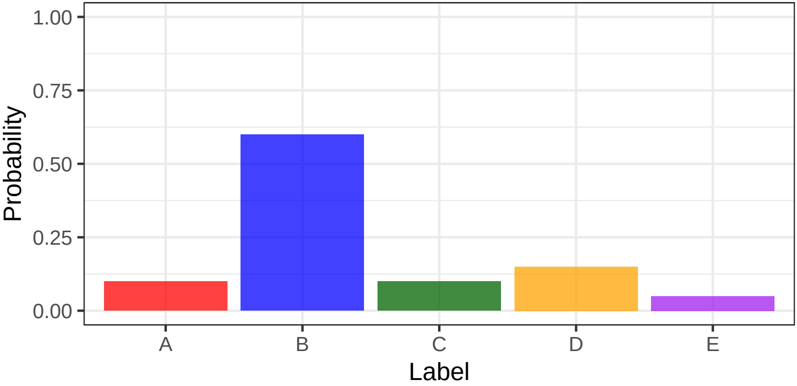


Prediction sets for classification: the ideal approach

Conditional class probabilities (oracle):

$$\pi_y(x) = \mathbb{P}[Y = y \mid X = x]$$

Suppose $\alpha = 0.1$. (We want 90% coverage)

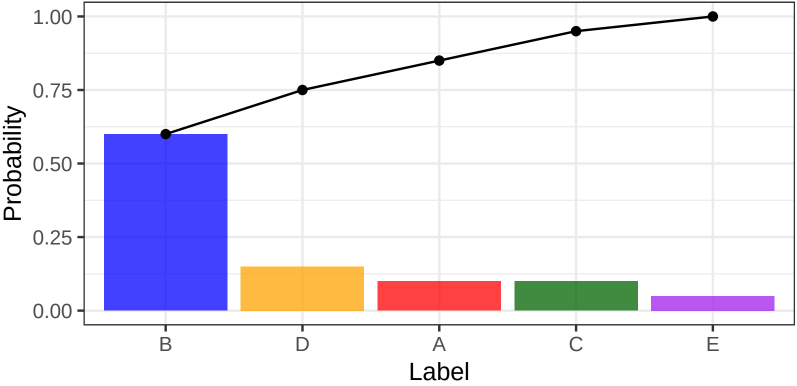


Prediction sets for classification: the ideal approach

Conditional class probabilities (oracle):

$$\pi_y(x) = \mathbb{P}[Y = y \mid X = x]$$

Suppose $\alpha = 0.1$. (We want 90% coverage)



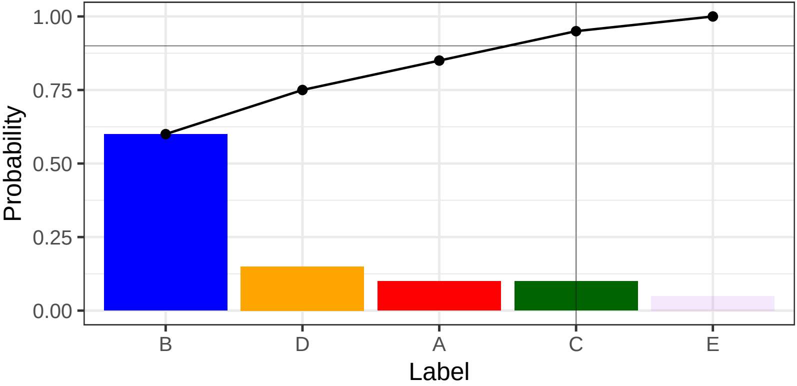
Step 1: sort the classes by their probability and compute the CDF.

Prediction sets for classification: the ideal approach

Conditional class probabilities (oracle):

$$\pi_y(x) = \mathbb{P}[Y = y \mid X = x]$$

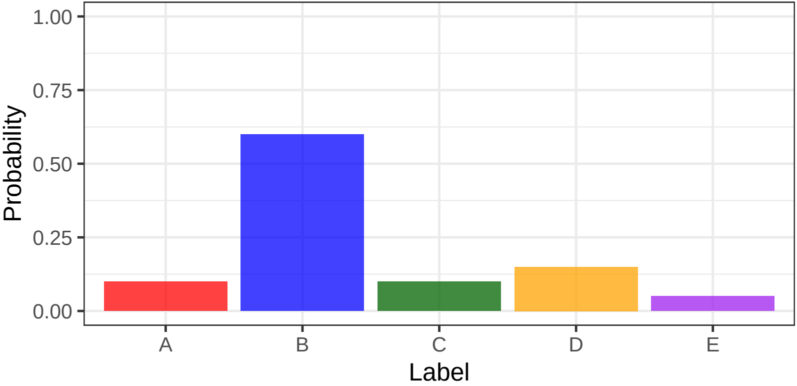
Suppose $\alpha = 0.1$. (We want 90% coverage)



Step 2: find where the CDF crosses above the $1 - \alpha$ level.

The classification oracle [Romano et al., 2020]

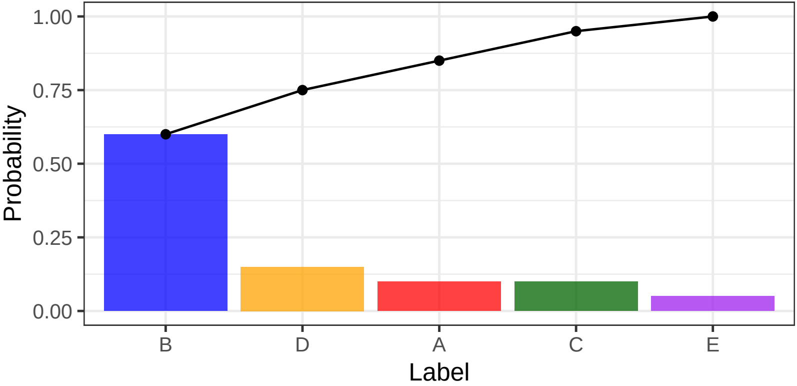
For any $x \in \mathbb{R}^p$, set $\pi_y(x) = \mathbb{P}[Y = y \mid X = x]$ for each $y \in \mathcal{Y}$.



The classification oracle [Romano et al., 2020]

For any $x \in \mathbb{R}^p$, set $\pi_y(x) = \mathbb{P}[Y = y \mid X = x]$ for each $y \in \mathcal{Y}$.

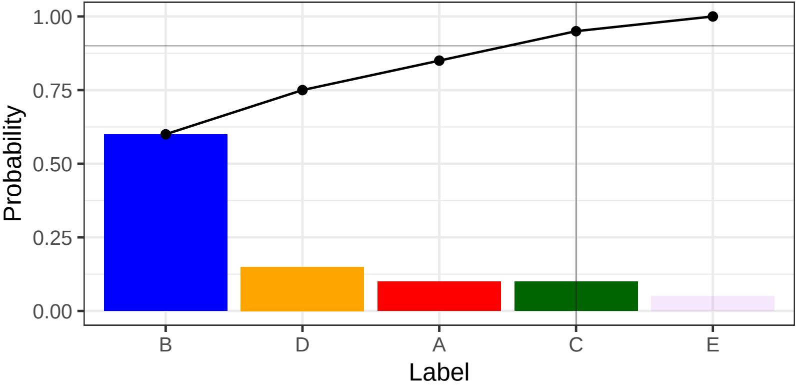
Suppose $\alpha = 0.1$.



The classification oracle [Romano et al., 2020]

For any $x \in \mathbb{R}^p$, set $\pi_y(x) = \mathbb{P}[Y = y \mid X = x]$ for each $y \in \mathcal{Y}$.

Suppose $\alpha = 0.1$.



The conservative classification oracle [Romano et al., 2020]

For any $x \in \mathbb{R}^p$, set $\pi_y(x) = \mathbb{P}[Y = y \mid X = x]$ for each $y \in \mathcal{Y}$.

For $\tau \in [0, 1]$, define the *generalized conditional quantile* function

$$L(x; \pi, \tau) = \\ \min\{c \in \{1, \ldots, C\} \ : \ \pi_{(1)}(x) + \pi_{(2)}(x) + \ldots + \pi_{(c)}(x) \geq \tau\},$$

The conservative classification oracle [Romano et al., 2020]

For any $x \in \mathbb{R}^p$, set $\pi_y(x) = \mathbb{P}[Y = y \mid X = x]$ for each $y \in \mathcal{Y}$.

For $\tau \in [0, 1]$, define the *generalized conditional quantile* function

$$L(x; \pi, \tau) = \min\{c \in \{1, \ldots, C\} \ : \ \pi_{(1)}(x) + \pi_{(2)}(x) + \ldots + \pi_{(c)}(x) \geq \tau\},$$

The (conservative) oracle prediction set is:

$$C^{\text{oracle}+}(x) = \{\text{`}y\text{' indices of the } L(x; \pi, 1-\alpha) \text{ largest } \pi_y(x)\}.$$

The classification oracle

Define a function $\mathcal{S}$ with input x, $u \in [0,1]$, π, and τ:

$$\mathcal{S}(x, u; \pi, \tau) =$$
$$\begin{cases} \text{`}y\text{' indices of the } L(x;\pi,\tau) - 1 \text{ largest } \pi_y(x), & \text{if } u \leq V(x;\pi,\tau), \\ \text{`}y\text{' indices of the } L(x;\pi,\tau) \text{ largest } \pi_y(x), & \text{otherwise,} \end{cases}$$

where

$$V(x;\pi,\tau) = \frac{1}{\pi_{(L(x;\pi,\tau))}(x)} \left[\sum_{c=1}^{L(x;\pi,\tau)} \pi_{(c)}(x) - \tau \right].$$

The classification oracle

Define a function $\mathcal{S}$ with input x, $u \in [0,1]$, π, and τ:

$$\mathcal{S}(x,u;\pi,\tau) =$$
$$\begin{cases} \text{`}y\text{' indices of the } L(x;\pi,\tau) - 1 \text{ largest } \pi_y(x), & \text{if } u \leq V(x;\pi,\tau), \\ \text{`}y\text{' indices of the } L(x;\pi,\tau) \text{ largest } \pi_y(x), & \text{otherwise,} \end{cases}$$

where

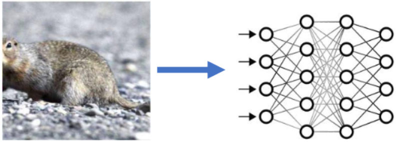
$$V(x;\pi,\tau) = \frac{1}{\pi_{(L(x;\pi,\tau))}(x)} \left[\sum_{c=1}^{L(x;\pi,\tau)} \pi_{(c)}(x) - \tau \right].$$

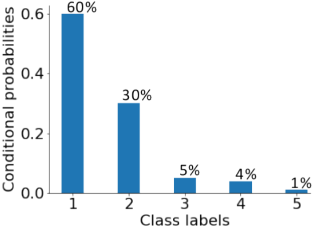
Then, the (tight) oracle would draw $U \sim \mathrm{Unif}(0,1)$ and predict:

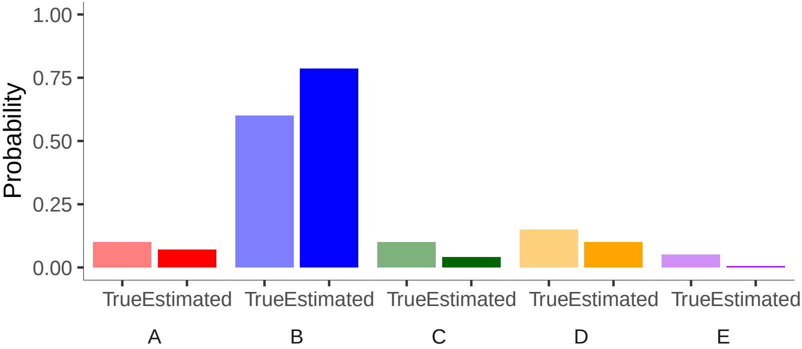
$$C^{\text{oracle}}(x) = \mathcal{S}(x,U;\pi,1-\alpha).$$

Machine learning classification models

In practice, we use probability estimates that may not be accurate.

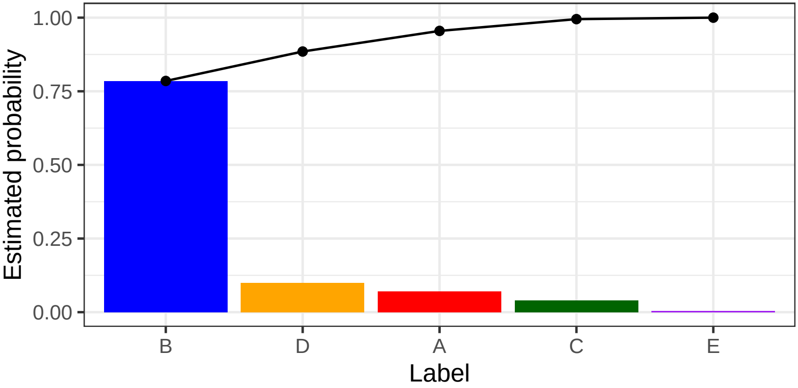






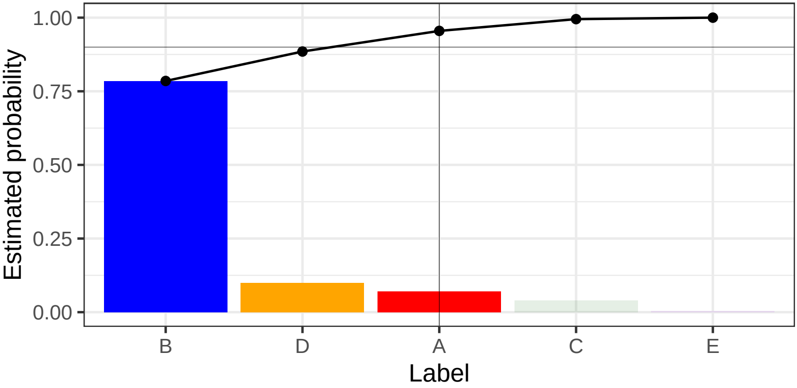
Plug-in prediction sets will often fail

Plug the estimated probabilities into the oracle prediction rule.



Plug-in prediction sets will often fail

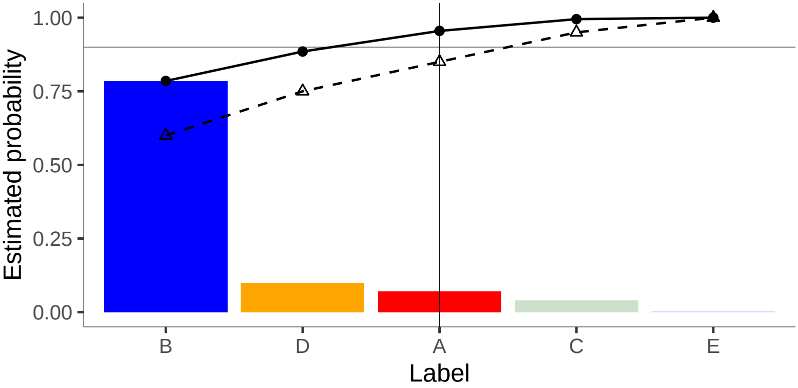
Plug the estimated probabilities into the oracle prediction rule.



Plug-in prediction sets will often fail

Plug the estimated probabilities into the oracle prediction rule.

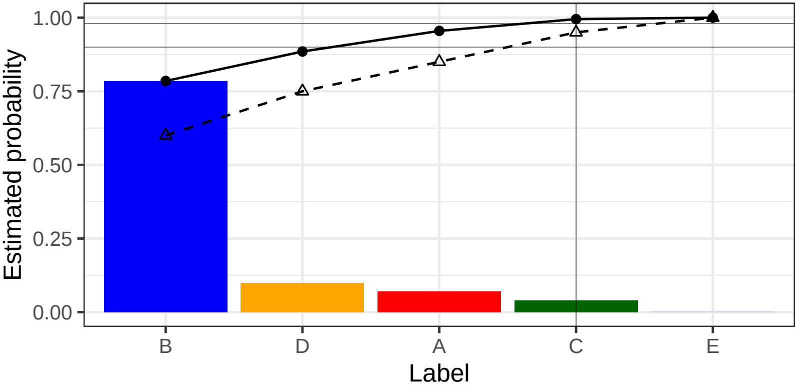
The probability estimates are often overconfident.



Plug-in prediction sets will often fail

Plug the estimated probabilities into the oracle prediction rule.

The probability estimates are often overconfident.
This tends to lead to under-coverage.

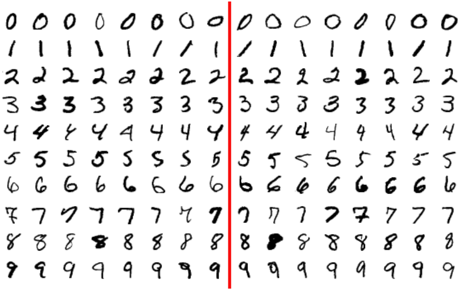


Split-conformal inference

Full data set

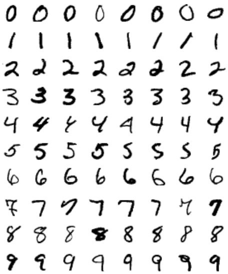
Split-conformal inference

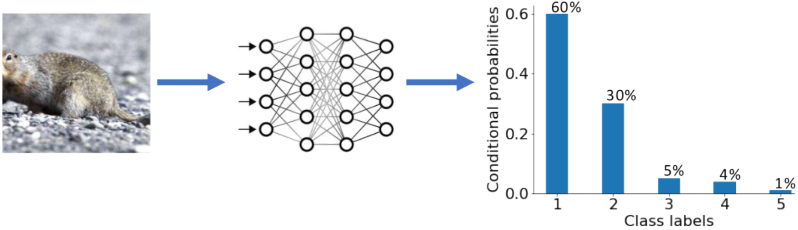
(1) Randomly split the data into two subsets



Split-conformal inference

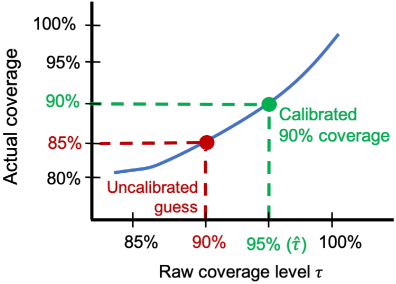
(2) Learn a model using training data





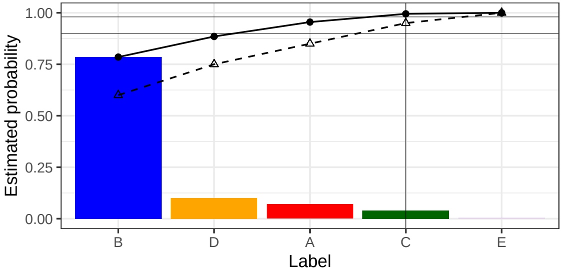
Split-conformal inference

(3) Make prediction sets and evaluate coverage on calibration data



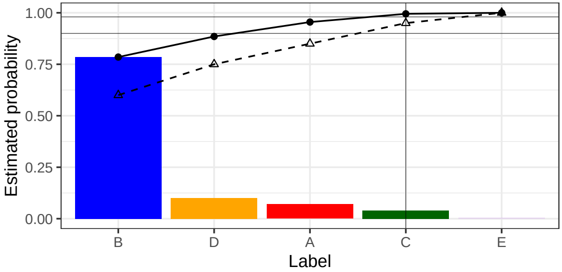
Calibration via conformity scores

How far in τ (above $1-\alpha$) before Y_i is classified correctly?



Calibration via conformity scores

How far in τ (above $1-\alpha$) before Y_i is classified correctly?

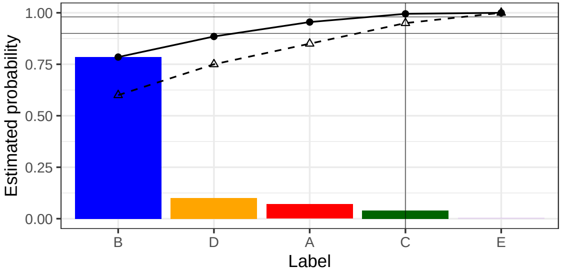


Evaluate conformity scores for all n calibration data points:

$$W_i = \min \left\{ \tau \in [0,1] : Y_i \in \mathcal{S}(X_i; \hat{\pi}, \tau) \right\},$$

Calibration via conformity scores

How far in τ (above $1-\alpha$) before Y_i is classified correctly?



Evaluate conformity scores for all n calibration data points:

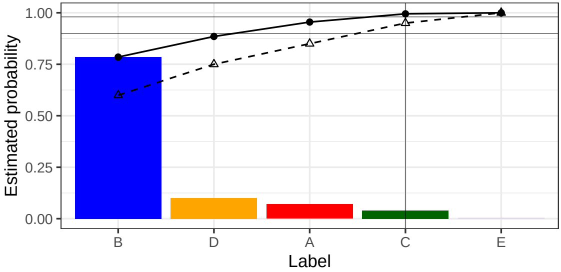
$$W_i = \min \left\{ \tau \in [0,1] : Y_i \in \mathcal{S}(X_i; \hat{\pi}, \tau) \right\},$$

Compute

$$\hat{Q}_n(W_{\mathcal{I}_{\mathrm{cal}}}, \beta_n) = W_{(\lceil n\beta_n \rceil)}, \qquad \beta_n = (1-\alpha)(1+1/n)$$

Calibration via conformity scores

How far in τ (above $1-\alpha$) before Y_i is classified correctly?



Evaluate conformity scores for all n calibration data points:

$$W_i = \min\left\{\tau \in [0,1] : Y_i \in \mathcal{S}(X_i; \hat{\pi}, \tau)\right\},$$

Compute

$$\hat{Q}_n(W_{\mathcal{I}_{\text{cal}}}, \beta_n) = W_{(\lceil n\beta_n \rceil)}, \qquad \beta_n = (1-\alpha)(1+1/n)$$

Predict

$$\hat{C}_\alpha(X_{n+1}) = \mathcal{S}(X_{n+1}, U_{n+1}; \hat{\pi}, \hat{\tau}), \qquad \hat{\tau} = \hat{Q}_n(W_{\mathcal{I}_2}, \beta_n)$$

Split-conformal classification

Algorithm 3: Split-conformal classification

1: **Input**: Data $\{(X_i, Y_i)\}_{i=1}^n$, test point X_{n+1}, $\alpha \in (0, 1)$
2: black-box model $\mathcal{B}$, level $\alpha \in (0, 1)$

Split-conformal classification

Algorithm 3: Split-conformal classification

Input: Data $\{(X_i, Y_i)\}_{i=1}^{n}$, test point X_{n+1}, $\alpha \in (0, 1)$
black-box model $\mathcal{B}$, level $\alpha \in (0, 1)$

Sample $U_i \sim \text{Uniform}(0, 1)$ for each $i \in \{1, \ldots, n + 1\}$

Split-conformal classification

Algorithm 3: Split-conformal classification

1: **Input**: Data $\{(X_i, Y_i)\}_{i=1}^n$, test point X_{n+1}, $\alpha \in (0,1)$
2: black-box model $\mathcal{B}$, level $\alpha \in (0,1)$

3: Sample $U_i \sim \text{Uniform}(0,1)$ for each $i \in \{1, \ldots, n+1\}$
4: Split the data: $\mathcal{I}_1 = \{1, \ldots, n/2\}, \mathcal{I}_2 = \{n/2+1, \ldots, n\}$

Split-conformal classification

Algorithm 3: Split-conformal classification

Input: Data $\{(X_i, Y_i)\}_{i=1}^n$, test point X_{n+1}, $\alpha \in (0,1)$
black-box model $\mathcal{B}$, level $\alpha \in (0,1)$

Sample $U_i \sim \text{Uniform}(0,1)$ for each $i \in \{1, \ldots, n+1\}$
Split the data: $\mathcal{I}_1 = \{1, \ldots, n/2\}, \mathcal{I}_2 = \{n/2+1, \ldots, n\}$
Train $\mathcal{B}$ on $\mathcal{I}_1$: $\mathcal{B}\left(\{(X_i, Y_i)\}_{i \in \mathcal{I}_1}\right) \to \hat{\pi}$

Split-conformal classification

Algorithm 3: Split-conformal classification

Input: Data $\{(X_i, Y_i)\}_{i=1}^n$, test point X_{n+1}, $\alpha \in (0,1)$
black-box model $\mathcal{B}$, level $\alpha \in (0,1)$

Sample $U_i \sim \text{Uniform}(0,1)$ for each $i \in \{1, \ldots, n+1\}$
Split the data: $\mathcal{I}_1 = \{1, \ldots, n/2\}, \mathcal{I}_2 = \{n/2+1, \ldots, n\}$
Train $\mathcal{B}$ on $\mathcal{I}_1$: $\mathcal{B}\left(\{(X_i, Y_i)\}_{i \in \mathcal{I}_1}\right) \to \hat{\pi}$
Evaluate $Z_i = \mathcal{Z}(X_i, Y_i, U_i; \hat{\pi})$ for all $i \in \mathcal{I}_2$

Split-conformal classification

Algorithm 3: Split-conformal classification

Input: Data $\{(X_i, Y_i)\}_{i=1}^{n}$, test point X_{n+1}, $\alpha \in (0,1)$

black-box model $\mathcal{B}$, level $\alpha \in (0,1)$

Sample $U_i \sim \text{Uniform}(0,1)$ for each $i \in \{1, \ldots, n+1\}$

Split the data: $\mathcal{I}_1 = \{1, \ldots, n/2\}, \mathcal{I}_2 = \{n/2+1, \ldots, n\}$

Train $\mathcal{B}$ on $\mathcal{I}_1 : \mathcal{B}\left(\{(X_i, Y_i)\}_{i \in \mathcal{I}_1}\right) \to \hat{\pi}$

Evaluate $Z_i = \mathcal{Z}(X_i, Y_i, U_i; \hat{\pi})$ for all $i \in \mathcal{I}_2$

Compute $\hat{Q}_n(Z_{\mathcal{I}_2}, \beta_n) = Z_{(\lceil n\beta_n \rceil)}$, where $\beta_n = (1-\alpha)(1+1/n)$

Split-conformal classification

Algorithm 3: Split-conformal classification

Input: Data $\{(X_i, Y_i)\}_{i=1}^{n}$, test point X_{n+1}, $\alpha \in (0,1)$
black-box model $\mathcal{B}$, level $\alpha \in (0,1)$

Sample $U_i \sim \text{Uniform}(0,1)$ for each $i \in \{1, \ldots, n+1\}$
Split the data: $\mathcal{I}_1 = \{1, \ldots, n/2\}, \mathcal{I}_2 = \{n/2+1, \ldots, n\}$
Train $\mathcal{B}$ on $\mathcal{I}_1$: $\mathcal{B}(\{(X_i, Y_i)\}_{i \in \mathcal{I}_1}) \to \hat{\pi}$
Evaluate $Z_i = \mathcal{Z}(X_i, Y_i, U_i; \hat{\pi})$ for all $i \in \mathcal{I}_2$
Compute $\hat{Q}_n(Z_{\mathcal{I}_2}, \beta_n) = Z_{(\lceil n\beta_n \rceil)}$, where $\beta_n = (1-\alpha)(1+1/n)$

Output: $\hat{C}_\alpha(X_{n+1}) = \mathcal{S}(X_{n+1}, U_{n+1}; \hat{\pi}, \hat{Q}_n(Z_{\mathcal{I}_2}, \beta_n))$

Split-conformal classification

Algorithm 3: Split-conformal classification

1: **Input**: Data $\{(X_i, Y_i)\}_{i=1}^n$, test point X_{n+1}, $\alpha \in (0,1)$
2: black-box model $\mathcal{B}$, level $\alpha \in (0,1)$

3: Sample $U_i \sim \text{Uniform}(0,1)$ for each $i \in \{1, \ldots, n+1\}$
4: Split the data: $\mathcal{I}_1 = \{1, \ldots, n/2\}, \mathcal{I}_2 = \{n/2+1, \ldots, n\}$
5: Train $\mathcal{B}$ on $\mathcal{I}_1$: $\mathcal{B}\left(\{(X_i, Y_i)\}_{i \in \mathcal{I}_1}\right) \to \hat{\pi}$
6: Evaluate $Z_i = \mathcal{Z}(X_i, Y_i, U_i; \hat{\pi})$ for all $i \in \mathcal{I}_2$
7: Compute $\hat{Q}_n(Z_{\mathcal{I}_2}, \beta_n) = Z_{(\lceil n\beta_n \rceil)}$, where $\beta_n = (1-\alpha)(1+1/n)$

8: **Output**: $\hat{C}_\alpha(X_{n+1}) = \mathcal{S}(X_{n+1}, U_{n+1}; \hat{\pi}, \hat{Q}_n(Z_{\mathcal{I}_2}, \beta_n))$

Why does this work?

$$Y_{n+1} \in \hat{C}_\alpha(X_{n+1}) \quad \Longleftrightarrow \quad Z_{n+1} \leq \hat{Q}_n(Z_{\mathcal{I}_2}, \beta_n).$$

Marginal coverage of split-conformal classification

Theorem (Romano, S., and Candès, 2020)

Suppose $(X_1, Y_1), (X_2, Y_2), \ldots, (X_{n+1}, Y_{n+1})$ *are exchangeable. Then, the split-conformal classification sets* $\hat{C}_\alpha$ *satisfy*

$$\mathbb{P}\left[Y_{n+1} \in \hat{C}_\alpha(X_{n+1})\right] \geq 1 - \alpha.$$

Moreover, under some additional smoothness assumption,

$$\mathbb{P}\left[Y_{n+1} \in \hat{C}_\alpha(X_{n+1})\right] \leq 1 - \alpha + \frac{1}{n}.$$

Computer session II

Chapter 7:
Full conformal inference

Full conformal

Split conformal uses only $n/2$ samples to fit the black-box model. Can we do better?

Full conformal

Split conformal uses only $n/2$ samples to fit the black-box model.
Can we do better?

1. For each possible value y of Y, define an augmented data set:

$$\mathcal{D}_y = \{(X_1, Y_1), (X_2, Y_2), \ldots, (X_{n+1}, y)\}$$

2. Fit the black-box model on the new data. $\mathcal{B} : \mathcal{D}_y \to \hat{f}_y$

Full conformal

Split conformal uses only $n/2$ samples to fit the black-box model. Can we do better?

1. For each possible value y of Y, define an augmented data set:

$$\mathcal{D}_y = \{(X_1, Y_1), (X_2, Y_2), \ldots, (X_{n+1}, y)\}$$

2. Fit the black-box model on the new data. $\mathcal{B} : \mathcal{D}_y \to \hat{f}_y$
3. Compute residuals (or conformity scores) on all points in $\mathcal{D}_y$:

$$\begin{aligned} Z_{y,i} &= |Y_i - \hat{f}_y(X_i)|, \quad i \in \{1, \ldots, n\}, \\ Z_{y,n+1} &= |y - \hat{f}_y(X_{n+1})|. \end{aligned}$$

Full conformal

Split conformal uses only $n/2$ samples to fit the black-box model. Can we do better?

1. For each possible value y of Y, define an augmented data set:

$$\mathcal{D}_y = \{(X_1, Y_1), (X_2, Y_2), \ldots, (X_{n+1}, y)\}$$

2. Fit the black-box model on the new data. $\mathcal{B} : \mathcal{D}_y \to \hat{f}_y$
3. Compute residuals (or conformity scores) on all points in $\mathcal{D}_y$:

$$\begin{aligned} Z_{y,i} &= |Y_i - \hat{f}_y(X_i)|, \quad i \in \{1, \ldots, n\}, \\ Z_{y,n+1} &= |y - \hat{f}_y(X_{n+1})|. \end{aligned}$$

4. Rank $Z_{y,n+1}$ among $\{Z_{y,i}\}_{i=1}^{n+1}$.

Full conformal (continued)

4. Rank $Z_{y,n+1}$ among $\{Z_{y,i}\}_{i=1}^{n+1}$

$$R_y = \sum_{i=1}^{n+1} \mathbb{1}\left[Z_{y,i} \leq Z_{y,n+1}\right]$$

Full conformal (continued)

4. Rank $Z_{y,n+1}$ among $\{Z_{y,i}\}_{i=1}^{n+1}$

$$R_y = \sum_{i=1}^{n+1} \mathbb{1}\left[Z_{y,i} \leq Z_{y,n+1}\right] = 1 + \sum_{i=1}^{n} \mathbb{1}\left[Z_{y,i} \leq Z_{y,n+1}\right]$$

Full conformal (continued)

4. Rank $Z_{y,n+1}$ among $\{Z_{y,i}\}_{i=1}^{n+1}$

$$R_y = \sum_{i=1}^{n+1} \mathbb{1}\left[Z_{y,i} \leq Z_{y,n+1}\right] = 1 + \sum_{i=1}^{n} \mathbb{1}\left[Z_{y,i} \leq Z_{y,n+1}\right]$$

5. Include y in $\hat{C}_\alpha^{\text{full}}$ if $R_y \leq \lceil (1-\alpha)(n+1) \rceil$.

Full conformal (continued)

4. Rank $Z_{y,n+1}$ among $\{Z_{y,i}\}_{i=1}^{n+1}$

$$R_y = \sum_{i=1}^{n+1} \mathbb{1}\left[Z_{y,i} \leq Z_{y,n+1}\right] = 1 + \sum_{i=1}^{n} \mathbb{1}\left[Z_{y,i} \leq Z_{y,n+1}\right]$$

5. Include y in $\hat{C}_\alpha^{\text{full}}$ if $R_y \leq \lceil (1-\alpha)(n+1) \rceil$.

Finally, the prediction set is

$$\hat{C}_\alpha^{\text{full}}(X_{n+1}) = \left\{ y \in \mathbb{R} : R_y \leq \lceil (1-\alpha)(n+1) \rceil \right\}.$$

Full conformal (continued)

4. Rank $Z_{y,n+1}$ among $\{Z_{y,i}\}_{i=1}^{n+1}$

$$R_y = \sum_{i=1}^{n+1} \mathbb{1}\left[Z_{y,i} \leq Z_{y,n+1}\right] = 1 + \sum_{i=1}^{n} \mathbb{1}\left[Z_{y,i} \leq Z_{y,n+1}\right]$$

5. Include y in $\hat{C}_\alpha^{\text{full}}$ if $R_y \leq \lceil (1-\alpha)(n+1) \rceil$.

Finally, the prediction set is

$$\hat{C}_\alpha^{\text{full}}(X_{n+1}) = \left\{ y \in \mathbb{R} : R_y \leq \lceil (1-\alpha)(n+1) \rceil \right\}.$$

Of course, we could also use full-conformal with different scores (e.g., quantile regression or classification).

Marginal coverage of full-conformal prediction

Theorem ([Vovk et al., 2005, Lei et al., 2018])

Suppose $(X_1, Y_1), (X_2, Y_2), \ldots, (X_{n+1}, Y_{n+1})$ *are exchangeable. Then, the full-conformal prediction intervals* $\hat{C}_\alpha$ *satisfy*

$$\mathbb{P}\left[Y_{n+1} \in \hat{C}_\alpha(X_{n+1})\right] \geq 1 - \alpha.$$

Moreover, if the residuals $\{Z_{n/2+1}, \ldots, Z_{n+1}\}$ *are a.s. distinct,*

$$\mathbb{P}\left[Y_{n+1} \in \hat{C}_\alpha(X_{n+1})\right] \leq 1 - \alpha + \frac{1}{n+1}.$$

Limitations of full-conformal inference

1. Overfitting may reduce power, since all in-sample residuals may become equal to zero.

Limitations of full-conformal inference

1. Overfitting may reduce power, since all in-sample residuals may become equal to zero.
2. Re-training the model for each new test point (and possible value of Y_{n+1}) is often prohibitively expensive.

Can we do something in between full and split conformal?

Chapter 8:
Cross-validation+

Cross-validation+ [Barber et al., 2019]

Or perhaps we could call this *CV-conformal*.

Very similar to cross-conformal inference. [Vovk, 2015]

1. Divide the data points into K folds

$$\mathcal{I}_1 = \left\{1, \dots, \frac{n}{K}\right\},$$
$$\mathcal{I}_2 = \left\{\frac{n}{K} + 1, \dots, 2\frac{n}{K}\right\},$$
$$\dots$$
$$\mathcal{I}_K = \left\{(K-1)\frac{n}{K} + 1, \dots, n\right\},$$

Cross-validation+ [Barber et al., 2019]

Or perhaps we could call this *CV-conformal*.

Very similar to cross-conformal inference. [Vovk, 2015]

1. Divide the data points into K folds

$$\mathcal{I}_1 = \left\{1, \ldots, \frac{n}{K}\right\},$$
$$\mathcal{I}_2 = \left\{\frac{n}{K} + 1, \ldots, 2\frac{n}{K}\right\},$$
$$\ldots$$
$$\mathcal{I}_K = \left\{(K-1)\frac{n}{K} + 1, \ldots, n\right\},$$

2. Train the black-box model on each $\mathcal{I}_k$ and evaluate the conformity scores on $\{1, \ldots, n\} \setminus \mathcal{I}_k$.

Cross-validation+ (continued)

Define a *conformity score function*:

$$\mathcal{Z}(x, y, \hat{f}) = |y - \hat{f}(x)|$$

Denote by $\hat{f}_k$ the black-box model trained on $\mathcal{I}_k$.

Denote by $k(i)$ the fold to which point i belongs, $\forall i \in \{1, \ldots, n\}$. Then, we will compute

$$Z_i = \mathcal{Z}(X_i, Y_i, \hat{f}_{k(i)}).$$

Cross-validation+ (continued)

Define a *conformity score function*:

$$\mathcal{Z}(x, y, \hat{f}) = |y - \hat{f}(x)|$$

Denote by $\hat{f}_k$ the black-box model trained on $\mathcal{I}_k$.

Denote by $k(i)$ the fold to which point i belongs, $\forall i \in \{1, \dots, n\}$. Then, we will compute

$$Z_i = \mathcal{Z}(X_i, Y_i, \hat{f}_{k(i)}).$$

The prediction set at level α for X_{n+1} will be:

$$\hat{C}_\alpha^{\text{cv+}} = \left\{ y : \sum_{i=1}^{n} \mathbb{1}\left[Z_i < \mathcal{Z}(X_{n+1}, y, \hat{f}_{k(i)}) \right] \leq (1 - \alpha)(n + 1) \right\}$$

Closed-form cross-validation+

The prediction set at level α for X_{n+1} will be:

$$\hat{C}_{\alpha}^{\text{cv+}} = \left\{ y : \sum_{i=1}^{n} \mathbb{1}\left[Z_i < \mathcal{Z}(X_{n+1}, y, \hat{f}_{k(i)}) \right] \leq (1-\alpha)(n+1) \right\}$$

is equivalent to

$$\hat{C}_{\alpha}^{\text{cv+}} = \left[\hat{Q}_{\alpha,n}^{-} \left(\hat{f}_{k(i)}(X_{n+1}) - Z_i \right), \hat{Q}_{\alpha,n}^{+} \left(\hat{f}_{k(i)}(X_{n+1}) + Z_i \right) \right],$$

where

$$\hat{Q}_{\alpha,n}^{-}(\tilde{Z}) = \tilde{Z}_{\lfloor \alpha(n+1) \rfloor}, \qquad \hat{Q}_{\alpha,n}^{+}(\tilde{Z}) = \tilde{Z}_{\lceil (1-\alpha)(n+1) \rceil}.$$

Closed-form cross-validation+ (continued)

Suppose

$$Y_{n+1} \notin \left\{ y : \sum_{i=1}^{n} \mathbb{1}\left[Z_i < \mathcal{Z}(X_{n+1}, y, \hat{f}_{k(i)}) \right] \leq (1-\alpha)(n+1) \right\}.$$

That means, for at least $(1-\alpha)(n+1)$ values of i,

$$\mathcal{Z}(X_{n+1}, Y_{n+1}, \hat{f}_{k(i)}) > Z_i$$
$$|Y_{n+1} - \hat{f}_{k(i)}(X_{n+1})| > |Y_i - \hat{f}_{k(i)}(X_i)|$$

Closed-form cross-validation+ (continued)

Suppose

$$Y_{n+1} \notin \left\{ y : \sum_{i=1}^{n} \mathbb{1}\left[Z_i < \mathcal{Z}(X_{n+1}, y, \hat{f}_{k(i)}) \right] \leq (1-\alpha)(n+1) \right\}.$$

That means, for at least $(1-\alpha)(n+1)$ values of i,

$$\mathcal{Z}(X_{n+1}, Y_{n+1}, \hat{f}_{k(i)}) > Z_i$$
$$|Y_{n+1} - \hat{f}_{k(i)}(X_{n+1})| > |Y_i - \hat{f}_{k(i)}(X_i)|$$

So, for at least $(1-\alpha)(n+1)$ values of i, either

$$Y_{n+1} > \hat{f}_{k(i)}(X_{n+1}) + |Y_i - \hat{f}_{k(i)}(X_i)|$$

or

$$Y_{n+1} < \hat{f}_{k(i)}(X_{n+1}) - |Y_i - \hat{f}_{k(i)}(X_i)|$$

Closed-form cross-validation+ (continued)

Suppose

$$Y_{n+1} \notin \left\{ y : \sum_{i=1}^{n} \mathbb{1}\left[Z_i < \mathcal{Z}(X_{n+1}, y, \hat{f}_{k(i)}) \right] \leq (1-\alpha)(n+1) \right\}.$$

That means, for at least $(1-\alpha)(n+1)$ values of i,

$$\mathcal{Z}(X_{n+1}, Y_{n+1}, \hat{f}_{k(i)}) > Z_i$$
$$|Y_{n+1} - \hat{f}_{k(i)}(X_{n+1})| > |Y_i - \hat{f}_{k(i)}(X_i)|$$

So, for at least $(1-\alpha)(n+1)$ values of i, either

$$Y_{n+1} > \hat{f}_{k(i)}(X_{n+1}) + |Y_i - \hat{f}_{k(i)}(X_i)| \Rightarrow Y_{n+1} > \hat{Q}^{+}_{\alpha,n}\left(\hat{f}_{k(i)}(X_{n+1}) + Z_i\right)$$

or

$$Y_{n+1} < \hat{f}_{k(i)}(X_{n+1}) - |Y_i - \hat{f}_{k(i)}(X_i)|$$

Closed-form cross-validation+ (continued)

Suppose

$$Y_{n+1} \notin \left\{ y : \sum_{i=1}^{n} \mathbb{1}\left[Z_i < \mathcal{Z}(X_{n+1}, y, \hat{f}_{k(i)}) \right] \le (1-\alpha)(n+1) \right\}.$$

That means, for at least $(1-\alpha)(n+1)$ values of i,

$$\begin{aligned} \mathcal{Z}(X_{n+1}, Y_{n+1}, \hat{f}_{k(i)}) &> Z_i \\ |Y_{n+1} - \hat{f}_{k(i)}(X_{n+1})| &> |Y_i - \hat{f}_{k(i)}(X_i)| \end{aligned}$$

So, for at least $(1-\alpha)(n+1)$ values of i, either

$$Y_{n+1} > \hat{f}_{k(i)}(X_{n+1}) + |Y_i - \hat{f}_{k(i)}(X_i)| \Rightarrow Y_{n+1} > \hat{Q}^{+}_{\alpha,n}\left(\hat{f}_{k(i)}(X_{n+1}) + Z_i \right)$$

or

$$Y_{n+1} < \hat{f}_{k(i)}(X_{n+1}) - |Y_i - \hat{f}_{k(i)}(X_i)| \Rightarrow Y_{n+1} < \hat{Q}^{-}_{\alpha,n}\left(\hat{f}_{k(i)}(X_{n+1}) + Z_i \right)$$

Marginal coverage of CV+

Theorem ([Barber et al., 2019])

Suppose $(X_1, Y_1), (X_2, Y_2), \ldots, (X_{n+1}, Y_{n+1})$ *are exchangeable. Then, the CV+ prediction intervals* $\hat{C}_\alpha^{cv+}$ *satisfy*

$$\mathbb{P}\left[Y_{n+1} \in \hat{C}_\alpha^{cv+}(X_{n+1})\right] \geq 1 - 2\alpha - \frac{1 - K/n}{K + 1}.$$

Marginal coverage of CV+

Theorem ([Barber et al., 2019])

Suppose $(X_1, Y_1), (X_2, Y_2), \ldots, (X_{n+1}, Y_{n+1})$ *are exchangeable. Then, the CV+ prediction intervals* $\hat{C}_\alpha^{cv+}$ *satisfy*

$$\mathbb{P}\left[Y_{n+1} \in \hat{C}_\alpha^{cv+}(X_{n+1})\right] \geq 1 - 2\alpha - \frac{1 - K/n}{K + 1}.$$

Why 2α? It's pessimistic. Coverage often above $1 - \alpha$ in practice.

Coverage is almost exact if the base algorithm is "stable" [Barber et al., 2019].

Marginal coverage of CV+

Theorem ([Barber et al., 2019])

Suppose $(X_1, Y_1), (X_2, Y_2), \ldots, (X_{n+1}, Y_{n+1})$ *are exchangeable. Then, the CV+ prediction intervals* $\hat{C}_\alpha^{cv+}$ *satisfy*

$$\mathbb{P}\left[Y_{n+1} \in \hat{C}_\alpha^{cv+}(X_{n+1})\right] \geq 1 - 2\alpha - \frac{1 - K/n}{K+1}.$$

Why 2α? It's pessimistic. Coverage often above $1 - \alpha$ in practice.

Coverage is almost exact if the base algorithm is "stable" [Barber et al., 2019].

A more conservative version has provable coverage above $1 - \alpha$.

Marginal coverage of CV+

Theorem ([Barber et al., 2019])

Suppose $(X_1, Y_1), (X_2, Y_2), \ldots, (X_{n+1}, Y_{n+1})$ *are exchangeable. Then, the CV+ prediction intervals* $\hat{C}_\alpha^{cv+}$ *satisfy*

$$\mathbb{P}\left[Y_{n+1} \in \hat{C}_\alpha^{cv+}(X_{n+1})\right] \geq 1 - 2\alpha - \frac{1 - K/n}{K + 1}.$$

Why 2α? It's pessimistic. Coverage often above $1 - \alpha$ in practice.

Coverage is almost exact if the base algorithm is "stable" [Barber et al., 2019].

A more conservative version has provable coverage above $1 - \alpha$.

[Steinberger and Leeb, 2018] proves a related method is "valid conditional on data set" if the base algorithm is "stable".

Proof for CV+ (setup)

Augmented data: imagine we have access to $m = n/K$ test points

$$(X_{n+1}, Y_{n+1}, U_{n+1}), \ldots, (X_{n+m}, Y_{n+m}, U_{n+m}),$$

which we put in the extra fold $\mathcal{I}_{K+1}$.

Proof for CV+ (setup)

Augmented data: imagine we have access to $m = n/K$ test points

$$(X_{n+1}, Y_{n+1}, U_{n+1}), \ldots, (X_{n+m}, Y_{n+m}, U_{n+m}),$$

which we put in the extra fold $\mathcal{I}_{K+1}$.

For any $k \neq k' \in \{1, \ldots, K+1\}$, define $\tilde{f}_{k,k'}$ as the black-box model fit on all data points except those in $(\mathcal{I}_k \cup \mathcal{I}_{k'})$.

Proof for CV+ (setup)

Augmented data: imagine we have access to $m = n/K$ test points

$$(X_{n+1}, Y_{n+1}, U_{n+1}), \ldots, (X_{n+m}, Y_{n+m}, U_{n+m}),$$

which we put in the extra fold $\mathcal{I}_{K+1}$.

For any $k \neq k' \in \{1, \ldots, K+1\}$, define $\tilde{f}_{k,k'}$ as the black-box model fit on all data points except those in $(\mathcal{I}_k \cup \mathcal{I}_{k'})$.

Note that $\tilde{f}_{k,K+1} = \hat{f}_k$.

Proof for CV+ (setup)

Augmented data: imagine we have access to $m = n/K$ test points

$$(X_{n+1}, Y_{n+1}, U_{n+1}), \ldots, (X_{n+m}, Y_{n+m}, U_{n+m}),$$

which we put in the extra fold $\mathcal{I}_{K+1}$.

For any $k \neq k' \in \{1, \ldots, K+1\}$, define $\tilde{f}_{k,k'}$ as the black-box model fit on all data points except those in $(\mathcal{I}_k \cup \mathcal{I}_{k'})$.

Note that $\tilde{f}_{k,K+1} = \hat{f}_k$.

Define the matrix $A \in \{0,1\}^{(n+m)\times(n+m)}$ as:

$$A_{ij} = \begin{cases} 0, & \text{if } k(i) = k(j), \\ \mathbb{1}\left[\mathcal{Z}(X_j, Y_j, \tilde{f}_{k(i),k(j)}) < \mathcal{Z}(X_i, Y_i, \tilde{f}_{k(i),k(j)})\right], & \text{if } k(i) \neq k(j), \end{cases}$$

Tournament (with teams) interpretation: i "won" against j.

Proof for CV+ (setup)

We will show that that $Y_{n+1} \in \hat{C}_{\alpha}^{\text{cv+}}$ if and only if

$$\sum_{i=1}^{n+m} A_{n+1,i} > (1-\alpha)(n+1)$$

Proof for CV+ (setup)

We will show that that $Y_{n+1} \in \hat{C}_\alpha^{\text{cv+}}$ if and only if

$$\sum_{i=1}^{n+m} A_{n+1,i} > (1-\alpha)(n+1)$$

Recall that $Y_{n+1} \notin \hat{C}_\alpha^{\text{cv+}}$ if and only if

$$\begin{aligned}\beta_n &= (1-\alpha)(n+1) \\ &\geq \sum_{i=1}^{n} \mathbb{1}\left[Z_i < \mathcal{Z}(X_{n+1}, Y_{n+1}, \hat{f}_{k(i)})\right]\end{aligned}$$

Proof for CV+ (setup)

We will show that that $Y_{n+1} \in \hat{C}_\alpha^{\text{cv+}}$ if and only if

$$\sum_{i=1}^{n+m} A_{n+1,i} > (1-\alpha)(n+1)$$

Recall that $Y_{n+1} \notin \hat{C}_\alpha^{\text{cv+}}$ if and only if

$$\begin{aligned}
\beta_n &= (1-\alpha)(n+1) \\
&\geq \sum_{i=1}^{n} \mathbb{1}\left[Z_i < \mathcal{Z}(X_{n+1}, Y_{n+1}, \hat{f}_{k(i)})\right] \\
&= \sum_{i=1}^{n} \mathbb{1}\left[\mathcal{Z}(X_i, Y_i, \tilde{f}_{k(i),k(n+1)}) < \mathcal{Z}(X_{n+1}, Y_{n+1}, \tilde{f}_{k(i),k(n+1)})\right]
\end{aligned}$$

Proof for CV+ (setup)

We will show that that $Y_{n+1} \in \hat{C}_\alpha^{\text{cv+}}$ if and only if

$$\sum_{i=1}^{n+m} A_{n+1,i} > (1-\alpha)(n+1)$$

Recall that $Y_{n+1} \notin \hat{C}_\alpha^{\text{cv+}}$ if and only if

$$\begin{aligned}
\beta_n &= (1-\alpha)(n+1) \\
&\geq \sum_{i=1}^{n} \mathbb{1}\left[Z_i < \mathcal{Z}(X_{n+1}, Y_{n+1}, \hat{f}_{k(i)})\right] \\
&= \sum_{i=1}^{n} \mathbb{1}\left[\mathcal{Z}(X_i, Y_i, \tilde{f}_{k(i),k(n+1)}) < \mathcal{Z}(X_{n+1}, Y_{n+1}, \tilde{f}_{k(i),k(n+1)})\right] \\
&= \sum_{i=1}^{n} A_{n+1,i}
\end{aligned}$$

Proof for CV+ (setup)

We will show that that $Y_{n+1} \in \hat{C}_\alpha^{\text{cv+}}$ if and only if

$$\sum_{i=1}^{n+m} A_{n+1,i} > (1-\alpha)(n+1)$$

Recall that $Y_{n+1} \notin \hat{C}_\alpha^{\text{cv+}}$ if and only if

$$\begin{aligned}
\beta_n &= (1-\alpha)(n+1) \\
&\geq \sum_{i=1}^{n} \mathbb{1}\left[Z_i < \mathcal{Z}(X_{n+1}, Y_{n+1}, \hat{f}_{k(i)})\right] \\
&= \sum_{i=1}^{n} \mathbb{1}\left[\mathcal{Z}(X_i, Y_i, \tilde{f}_{k(i),k(n+1)}) < \mathcal{Z}(X_{n+1}, Y_{n+1}, \tilde{f}_{k(i),k(n+1)})\right] \\
&= \sum_{i=1}^{n} A_{n+1,i} = \sum_{i=1}^{n+m} A_{n+1,i}.
\end{aligned}$$

Proof for CV+ (strategy)

Define the set of *outstanding players*

$$\mathcal{S}(A) = \left\{ i \in \{1, \ldots, n+m\} : \sum_{i=1}^{n+m} A_{n+1,i} > (1-\alpha)(n+1) \right\}$$

We know $Y_{n+1} \notin \hat{C}_{\alpha}^{\text{cv+}}$ if and only if $(n+1) \in \mathcal{S}(A)$.

Proof for CV+ (strategy)

Define the set of *outstanding players*

$$\mathcal{S}(A) = \left\{ i \in \{1, \ldots, n+m\} : \sum_{i=1}^{n+m} A_{n+1,i} > (1-\alpha)(n+1) \right\}$$

We know $Y_{n+1} \notin \hat{C}_{\alpha}^{\text{cv+}}$ if and only if $(n+1) \in \mathcal{S}(A)$.

We need to bound

$$\mathbb{P}\left[(n+1) \in \mathcal{S}(A)\right].$$

Proof for CV+ (strategy)

Define the set of *outstanding players*

$$\mathcal{S}(A) = \left\{ i \in \{1, \ldots, n+m\} : \sum_{i=1}^{n+m} A_{n+1,i} > (1-\alpha)(n+1) \right\}$$

We know $Y_{n+1} \notin \hat{C}_{\alpha}^{\text{cv+}}$ if and only if $(n+1) \in \mathcal{S}(A)$.

We need to bound

$$\mathbb{P}\left[(n+1) \in \mathcal{S}(A)\right].$$

Strategy: prove that

- all players equally likely to be outstanding (exchangeability)
- only so many players can be outstanding (basic logic)

Proof for CV+ (exchangeability)

Let Π be a $(n+m) \times (n+m)$ permutation matrix **that does not mix players assigned to different teams**, such that

$$(\Pi A \Pi^\top)_{ij} = A_{i',j'}$$

We can prove that $A \stackrel{d}{=} \Pi A \Pi^\top$.

Proof for CV+ (exchangeability)

Let Π be a $(n+m)\times(n+m)$ permutation matrix **that does not mix players assigned to different teams**, such that

$$(\Pi A \Pi^\top)_{ij} = A_{i',j'}$$

We can prove that $A \overset{d}{=} \Pi A \Pi^\top$.

Assume $k(i) \neq k(j)$. Then,

$A_{\sigma(i)\sigma(j)}$

Proof for CV+ (exchangeability)

Let Π be a $(n+m) \times (n+m)$ permutation matrix **that does not mix players assigned to different teams**, such that

$$(\Pi A \Pi^\top)_{ij} = A_{i',j'}$$

We can prove that $A \overset{d}{=} \Pi A \Pi^\top$.

Assume $k(i) \neq k(j)$. Then,

$$A_{\sigma(i)\sigma(j)} = \mathbb{1}\left[\mathcal{Z}(X_{\sigma(j)}, Y_{\sigma(j)}, \tilde{f}_{k(\sigma(i)),k(\sigma(j))}) < \mathcal{Z}X_{\sigma(i)}, Y_{\sigma(i)}, \tilde{f}_{k(\sigma(i)),k(\sigma(j))})\right]$$

Proof for CV+ (exchangeability)

Let Π be a $(n+m) \times (n+m)$ permutation matrix **that does not mix players assigned to different teams**, such that

$$(\Pi A \Pi^\top)_{ij} = A_{i',j'}$$

We can prove that $A \stackrel{d}{=} \Pi A \Pi^\top$.

Assume $k(i) \neq k(j)$. Then,

$$\begin{aligned} A_{\sigma(i)\sigma(j)} &= \mathbb{1}\left[\mathcal{Z}(X_{\sigma(j)}, Y_{\sigma(j)}, \tilde{f}_{k(\sigma(i)),k(\sigma(j))}) < \mathcal{Z}X_{\sigma(i)}, Y_{\sigma(i)}, \tilde{f}_{k(\sigma(i)),k(\sigma(j))})\right] \\ &= \mathbb{1}\left[\mathcal{Z}(X_{\sigma(j)}, Y_{\sigma(j)}, \tilde{f}_{k(i),k(j)}) < \mathcal{Z}(X_{\sigma(i)}, Y_{\sigma(i)}, \tilde{f}_{k(i),k(j)})\right] \end{aligned}$$

Proof for CV+ (exchangeability)

Let Π be a $(n+m) \times (n+m)$ permutation matrix **that does not mix players assigned to different teams**, such that

$$(\Pi A \Pi^\top)_{ij} = A_{i',j'}$$

We can prove that $A \stackrel{d}{=} \Pi A \Pi^\top$.

Assume $k(i) \neq k(j)$. Then,

$$\begin{aligned}
A_{\sigma(i)\sigma(j)} &= \mathbb{1}\left[\mathcal{Z}(X_{\sigma(j)}, Y_{\sigma(j)}, \tilde{f}_{k(\sigma(i)),k(\sigma(j))}) < \mathcal{Z}X_{\sigma(i)}, Y_{\sigma(i)}, \tilde{f}_{k(\sigma(i)),k(\sigma(j))})\right] \\
&= \mathbb{1}\left[\mathcal{Z}(X_{\sigma(j)}, Y_{\sigma(j)}, \tilde{f}_{k(i),k(j)}) < \mathcal{Z}(X_{\sigma(i)}, Y_{\sigma(i)}, \tilde{f}_{k(i),k(j)})\right] \\
&\stackrel{d}{=} \mathbb{1}\left[\mathcal{Z}(X_j, Y_j, \tilde{f}_{k(i),k(j)}) < \mathcal{Z}(X_i, Y_i, \tilde{f}_{k(i),k(j)})\right] \\
&= A_{i,j}.
\end{aligned}$$

Proof for CV+ (exchangeability)

Let Π be a $(n+m) \times (n+m)$ permutation matrix **that does not mix players assigned to different teams**, such that

$$(\Pi A \Pi^\top)_{ij} = A_{i',j'}$$

We can prove that $A \overset{d}{=} \Pi A \Pi^\top$.

Assume $k(i) \neq k(j)$. Then,

$$\begin{aligned}
A_{\sigma(i)\sigma(j)} &= \mathbb{1}\left[\mathcal{Z}(X_{\sigma(j)}, Y_{\sigma(j)}, \tilde{f}_{k(\sigma(i)),k(\sigma(j))}) < \mathcal{Z}X_{\sigma(i)}, Y_{\sigma(i)}, \tilde{f}_{k(\sigma(i)),k(\sigma(j))})\right] \\
&= \mathbb{1}\left[\mathcal{Z}(X_{\sigma(j)}, Y_{\sigma(j)}, \tilde{f}_{k(i),k(j)}) < \mathcal{Z}(X_{\sigma(i)}, Y_{\sigma(i)}, \tilde{f}_{k(i),k(j)})\right] \\
&\overset{d}{=} \mathbb{1}\left[\mathcal{Z}(X_j, Y_j, \tilde{f}_{k(i),k(j)}) < \mathcal{Z}(X_i, Y_i, \tilde{f}_{k(i),k(j)})\right] \\
&= A_{i,j}.
\end{aligned}$$

Clearly, $A_{\sigma(i)\sigma(j)} = 0 = A_{i,j}$ if $k(i) = k(j)$.

Proof for CV+ (exchangeability)

OK, so we have $A \stackrel{d}{=} \Pi A \Pi^\top$.

Suppose Π is such that $\sigma(n+1) = j$, for any $j \in \{1, \ldots, n+m\}$. Then,

$$(n+1) \in \mathcal{S}(A) \quad \Leftrightarrow \quad j \in \mathcal{S}(\Pi A \Pi^\top).$$

Proof for CV+ (exchangeability)

OK, so we have $A \overset{d}{=} \Pi A \Pi^\top$.

Suppose Π is such that $\sigma(n+1) = j$, for any $j \in \{1, \ldots, n+m\}$. Then,

$$(n+1) \in \mathcal{S}(A) \quad \Leftrightarrow \quad j \in \mathcal{S}(\Pi A \Pi^\top).$$

Therefore,

$$\mathbb{P}\left[(n+1) \in \mathcal{S}(A)\right] = \mathbb{P}\left[j \in \mathcal{S}(\Pi A \Pi^\top)\right] = \mathbb{P}\left[j \in \mathcal{S}(A)\right].$$

All players are equally likely to be outstanding!

Proof for CV+ (exchangeability)

OK, so we have $A \overset{d}{=} \Pi A \Pi^\top$.

Suppose Π is such that $\sigma(n+1) = j$, for any $j \in \{1, \ldots, n+m\}$. Then,

$$(n+1) \in \mathcal{S}(A) \quad \Leftrightarrow \quad j \in \mathcal{S}(\Pi A \Pi^\top).$$

Therefore,

$$\mathbb{P}\left[(n+1) \in \mathcal{S}(A)\right] = \mathbb{P}\left[j \in \mathcal{S}(\Pi A \Pi^\top)\right] = \mathbb{P}\left[j \in \mathcal{S}(A)\right].$$

All players are equally likely to be outstanding!

$$\mathbb{P}\left[(n+1) \in \mathcal{S}(A)\right] = \frac{1}{n+m} \sum_{i=1}^{n+m} \mathbb{P}\left[j \in \mathcal{S}(A)\right] = \frac{\mathbb{E}\left[|\mathcal{S}(A)|\right]}{n+m}.$$

Proof for CV+ (logic)

How large can $|\mathcal{S}(A)|$ be? Remember we defined

$$A_{ij} = \begin{cases} 0, & \text{if } k(i) = k(j), \\ \mathbb{1}\left[\mathcal{Z}(X_j, Y_j, \tilde{f}_{k(i),k(j)}) < \mathcal{Z}(X_i, Y_i, \tilde{f}_{k(i),k(j)})\right], & \text{if } k(i) \neq k(j), \end{cases}$$

Think of A_{ij} as indicating whether i wins a game against j, within a tournament with $n + m$ participant.

Note that i and j do not play each other if $k(i) = k(j)$.

Proof for CV+ (logic)

How large can $|\mathcal{S}(A)|$ be? Remember we defined

$$A_{ij} = \begin{cases} 0, & \text{if } k(i) = k(j), \\ \mathbb{1}\left[\mathcal{Z}(X_j, Y_j, \tilde{f}_{k(i),k(j)}) < \mathcal{Z}(X_i, Y_i, \tilde{f}_{k(i),k(j)})\right], & \text{if } k(i) \neq k(j), \end{cases}$$

Think of A_{ij} as indicating whether i wins a game against j, within a tournament with $n + m$ participant.

Note that i and j do not play each other if $k(i) = k(j)$.

$\mathcal{S}(A)$ is the set of players that win at least $(1 - \alpha)(n + 1)$ games.

Proof for CV+ (logic)

If $i \in \mathcal{S}(A)$, it lost at most $\alpha(n+1)+1$ games.

Let $s = |\mathcal{S}(A)|$ and $s_k = |\mathcal{S}(A) \cap \mathcal{I}_k|$ ($\#$ outstanding players in k).

Proof for CV+ (logic)

If $i \in \mathcal{S}(A)$, it lost at most $\alpha(n+1)+1$ games.

Let $s = |\mathcal{S}(A)|$ and $s_k = |\mathcal{S}(A) \cap \mathcal{I}_k|$ (# outstanding players in k).

The number of games involving two strange players is:

$$\frac{s(s-1)}{2}$$

Proof for CV+ (logic)

If $i \in \mathcal{S}(A)$, it lost at most $\alpha(n+1)+1$ games.

Let $s = |\mathcal{S}(A)|$ and $s_k = |\mathcal{S}(A) \cap \mathcal{I}_k|$ (# outstanding players in k).

The number of games involving two strange players is:

$$\frac{s(s-1)}{2}$$

Each game has one loser.

Proof for CV+ (logic)

If $i \in \mathcal{S}(A)$, it lost at most $\alpha(n+1)+1$ games.

Let $s = |\mathcal{S}(A)|$ and $s_k = |\mathcal{S}(A) \cap \mathcal{I}_k|$ (# outstanding players in k).

The number of games involving two strange players is:

$$\frac{s(s-1)}{2}$$

Each game has one loser.

Each outstanding player lost at most $\alpha(n+1)+1$ games (with other outstanding players).

Proof for CV+ (logic)

If $i \in \mathcal{S}(A)$, it lost at most $\alpha(n+1)+1$ games.

Let $s = |\mathcal{S}(A)|$ and $s_k = |\mathcal{S}(A) \cap \mathcal{I}_k|$ (# outstanding players in k).

The number of games involving two strange players is:

$$\frac{s(s-1)}{2}$$

Each game has one loser.

Each outstanding player lost at most $\alpha(n+1)+1$ games (with other outstanding players).

Outstanding players overall lost at most $s(\alpha(n+1)+1)$ games.

Proof for CV+ (logic)

If $i \in \mathcal{S}(A)$, it lost at most $\alpha(n+1)+1$ games.

Let $s = |\mathcal{S}(A)|$ and $s_k = |\mathcal{S}(A) \cap \mathcal{I}_k|$ (# outstanding players in k).

The number of games involving two strange players is:

$$\frac{s(s-1)}{2}$$

Each game has one loser.

Each outstanding player lost at most $\alpha(n+1)+1$ games (with other outstanding players).

Outstanding players overall lost at most $s(\alpha(n+1)+1)$ games.

$$\frac{s(s-1)}{2} \leq s(\alpha(n+1)+1) + \sum_{k=1}^{k} \frac{s_k(s_k-1)}{2}.$$

Therefore,

$$|\mathcal{S}(A)| = s \leq 2\alpha(n+1) + m - 2.$$

Proof for CV+ (wrapping up)

Putting everything together:

$$\mathbb{P}\left[(n+1) \in \mathcal{S}(A)\right] = \frac{\mathbb{E}\left[|\mathcal{S}(A)|\right]}{n+m}.$$

$$|\mathcal{S}(A)| = s \leq 2\alpha(n+1) + m - 2.$$

Proof for CV+ (wrapping up)

Putting everything together:

$$\mathbb{P}\left[(n+1) \in \mathcal{S}(A)\right] = \frac{\mathbb{E}\left[|\mathcal{S}(A)|\right]}{n+m}.$$

$$|\mathcal{S}(A)| = s \leq 2\alpha(n+1) + m - 2.$$

Therefore,

$$\begin{aligned}
\mathbb{P}\left[(n+1) \in \mathcal{S}(A)\right] &\leq \frac{2\alpha(n+1) + m - 2}{n+m} \\
&= \frac{2\alpha(n+m) + 2\alpha(1-m) + m - 2}{n+m} \\
&= 2\alpha + \frac{(m-1)(1-2\alpha) - 1}{n+m} \\
&\leq 2\alpha + \frac{1 - K/n}{K+1}.
\end{aligned}$$

Computer session III

To learn more

References and resources

github.com/valeman/awesome-conformal-prediction

References and resources

github.com/valeman/awesome-conformal-prediction

Some areas on which I've worked recently.

Skewed data

- Conformal prediction with conditional histograms [Sesia and Romano, 2021]

References and resources

github.com/valeman/awesome-conformal-prediction

Some areas on which I've worked recently.

Skewed data

- Conformal prediction with conditional histograms [Sesia and Romano, 2021]

Conformalized learning.

- Conformal loss for training uncertainty-aware classifiers [Einbinder et al., 2022]
- Conformalized early stopping [Liang et al., 2023]

References and resources

github.com/valeman/awesome-conformal-prediction

Some areas on which I've worked recently.

Skewed data

- Conformal prediction with conditional histograms [Sesia and Romano, 2021]

Conformalized learning.

- Conformal loss for training uncertainty-aware classifiers [Einbinder et al., 2022]
- Conformalized early stopping [Liang et al., 2023]

Testing for outliers with conformal p-values.

- [Bates et al., 2023]
- [Liang et al., 2022]
- [Bashari et al., 2023]

NESS session on Tuesday morning

NESS Symposium 2023

Event Schedule

<	Sat Jun 03	Sun Jun 04	Mon Jun 05	Tue Jun 06	>

sesia

Displaying agenda in event timezone (6:14 PM EDT)

9:50 AM

Recent Developments of Conformal Inference

9:50 AM – 11:30 AM

CDS 1101

Invited Session

This session will consist four speakers from academia who are experts in conformal inference.

Session Chair/Organizer

G

Guanyu Hu

Speakers

Matteo Sesia
Assistant Professor of Data Sciences and Operations
University of Southern California

Yinchu Zhu
Brandeis University

Adam Fisch
MIT

Stephen Bates
Postdoctoral researcher
UC Berkeley

4 Subsessions

Bibliography

Barber, R. F., Candes, E. J., Ramdas, A., and Tibshirani, R. J. (2019).
Predictive inference with the jackknife+.
arXiv preprint arXiv:1905.02928.

Bashari, M., Epstein, A., Romano, Y., and Sesia, M. (2023).
Derandomized novelty detection with fdr control via conformal e-values.
arXiv preprint arXiv:2302.07294.

Bates, S., Candès, E., Lei, L., Romano, Y., and Sesia, M. (2023).
Testing for outliers with conformal p-values.
Ann. Stat., 51(1):149 – 178.

Cauchois, M., Gupta, S., and Duchi, J. (2020).
Knowing what you know: valid confidence sets in multiclass and multilabel prediction.
arXiv preprint arXiv:2004.10181.

Einbinder, B.-S., Romano, Y., Sesia, M., and Zhou, Y. (2022).
Training uncertainty-aware classifiers with conformalized deep learning.
In *Adv. Neural Inf. Process. Syst.*, volume 35.

Foygel Barber, R., Candès, E. J., Ramdas, A., and Tibshirani, R. J. (2020).
The limits of distribution-free conditional predictive inference.
Information and Inference: A Journal of the IMA.

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

Lei, J., Robins, J., and Wasserman, L. (2013).
Distribution-free prediction sets.
Journal of the American Statistical Association, 108(501):278–287.

Liang, Z., Sesia, M., and Sun, W. (2022).
Integrative conformal p-values for powerful out-of-distribution testing with labeled outliers.
arXiv preprint arXiv:2208.11111.

Liang, Z., Zhou, Y., and Sesia, M. (2023).
Conformal inference is (almost) free for neural networks trained with early stopping.
arXiv preprint arXiv:2301.11556.

Romano, Y., Barber, R. F., Sabatti, C., and Candès, E. J. (2019a).
With malice towards none: Assessing uncertainty via equalized coverage.
arXiv preprint arXiv:1908.05428.

Romano, Y., Patterson, E., and Candes, E. (2019b).
Conformalized quantile regression.
In *Advances in Neural Information Processing Systems*, pages 3543–3553.

Romano, Y., Sesia, M., and Candès, E. J. (2020).
Classification with valid and adaptive coverage.
arXiv preprint arXiv:2006.02544.

Sesia, M. and Candès, E. J. (2020).
A comparison of some conformal quantile regression methods.
Stat, 9(1):e261.

Sesia, M. and Romano, Y. (2021).
Conformal prediction using conditional histograms.
Advances in Neural Information Processing Systems, 34:6304–6315.

Steinberger, L. and Leeb, H. (2018).
Conditional predictive inference for high-dimensional stable algorithms.
arXiv preprint arXiv:1809.01412.

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

Vovk, V. (2015).
Cross-conformal predictors.
Annals of Mathematics and Artificial Intelligence, 74(1-2):9–28.

Vovk, V., Gammerman, A., and Shafer, G. (2005).
Algorithmic Learning in a Random World.
Springer-Verlag, Berlin, Heidelberg.