FURTHER ADVENTURES IN UNCERTAINTY QUANTIFICATION: CONFORMAL PREDICTIONS

Viviana Acquaviva
CUNY/Columbia University

https://github.com/vacquaviva/IntroConformalPredictions

EAS 2024, Padova

Holy Grail of Uncertainty Quantification:

Generate predictions that are calibrated *for every example*

e.g., guarantee that true value is within quoted “90%” interval 90% of the time for any input value

Conformal predictions

Goal: starting from predictions for any model, generate calibrated sets (classification) or intervals (regression).

- **distribution-free:** the only assumption is that data points are exchangeable (no time series).
- **model-agnostic:** conformal predictions can be applied to any predictive model.
- **marginal coverage guarantee:** the resulting prediction sets (intervals) come with guarantees of covering the true outcome with at least desired probability on a set with the same statistical properties

Process

Divide data in train / calibration / validation (test) set

Build model on train set; generate predictions on calibration set

Use calibration set to build non-conformity (or conformal) scores

- Can be defined in many ways (important: good model ⇔ low scores)
- In practice, they measure how ”off” model predictions are by looking at some statistical property of the scores’ distribution

Use non-conformity scores to generate new prediction sets/intervals C that satisfy coverage guarantee for any chosen probability 1 – α:

$$P(Y_{val}) \in C(X_{val}) \geq (1 - \alpha)$$

Classification example

- Let's say the true class of object i is y_i
- Start from any model that predicts probabilities $f(x_i)$ for each class

  (for example, $f(x_i)$ could be 0.2, 0.7, 0.1 in a three-class model)
- On calibration set, compute scores $s_i = 1 - f(x_i)[y_i]$, i.e.

  1 – (probability assigned to correct class).

  s_i are a simple example of conformal scores (low = good)
- Sort from certain (low s_i) to uncertain
- Select probability threshold of interest q (e.g. 80%)
- Find "q" quantile of s_i, multiply by finite sample size correction ($\hat{q} = q * (n+1)/n$)
- Prediction output is the set of all classes that had a probability score > $1 - \hat{q}$
- Marginal coverage: on statistically equivalent (i.i.d.) data, sets contain the true class 80% of the time

	$p(y_1)$	$p(y_2)$	$p(y_3)$
#1	0.2	0.7	0.1
#2	0.4	0.4	0.2
#3	0.9	.05	.05
#4	0.1	0.1	0.8
#5	0.2	0.6	0.2

s_i
0.3
0.6
0.1
0.2
0.8

$s_{i\ (sorted)}$
0.1
0.2
0.3
0.6
0.8

$q = 0.6$

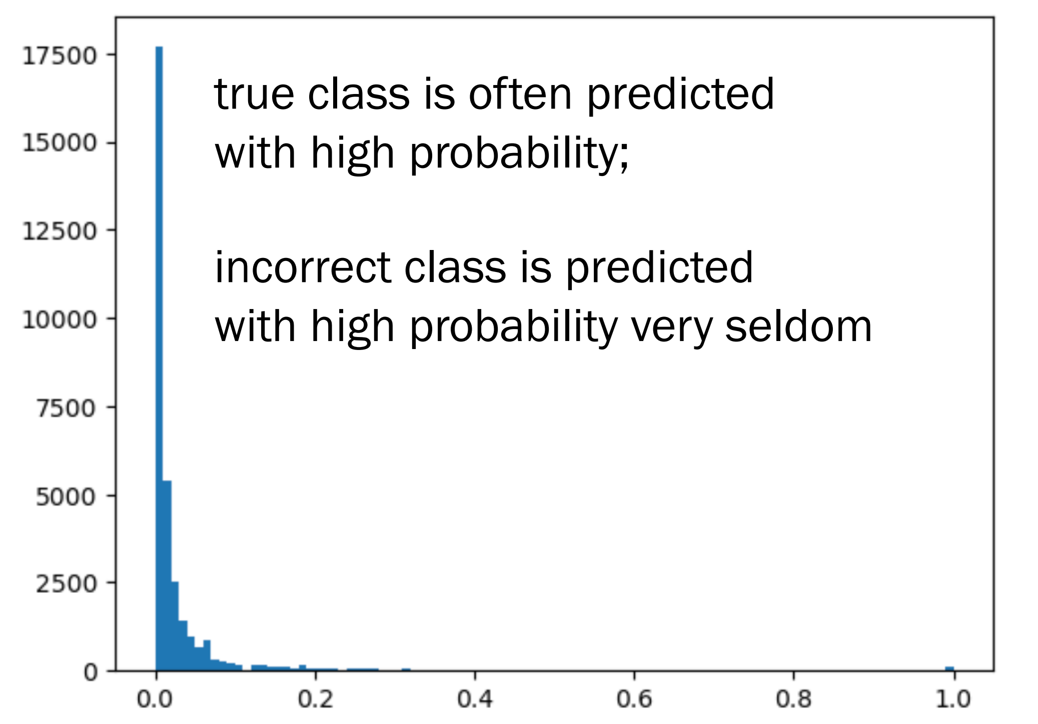
$\hat{q} = 0.72$

pred sets
2
1,2
1
3
2

How can we get a coverage guarantee no matter the quality of the estimator?

Good Classifier

conformal scores

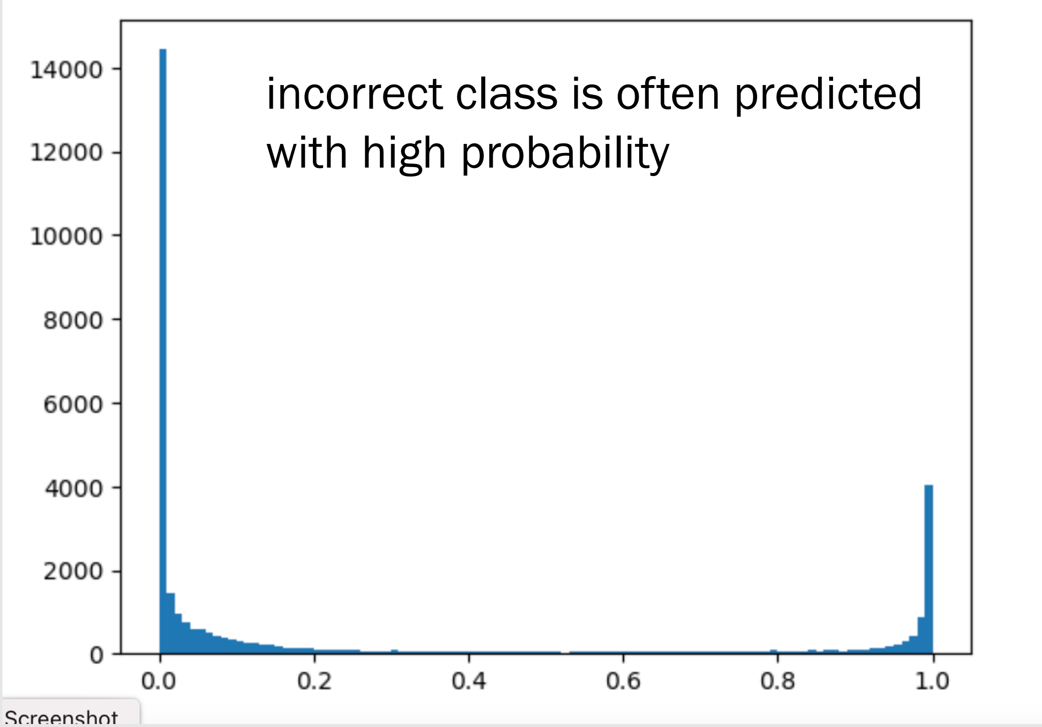


$\hat{q}$ (value corresponding to say 80% quantile of conformal scores) is low

only classes predicted with high probability (1- $\hat{q}$) are included in prediction sets

Bad Classifier

conformal scores



$\hat{q}$ (value corresponding to say 80% quantile of conformal scores) is high

even classes predicted with low probability (1- $\hat{q}$) are included in prediction sets

Devil is in the detail

- Marginal coverage guarantee doesn’t make single predictions particularly useful

Marginal means “on average”: given a new set of examples that is statistically equivalent to the calibration set, the predictions sets contain the true value “x”% of the time

- What we really want is conditional coverage: conditional on any split of the data, i.e. for every example

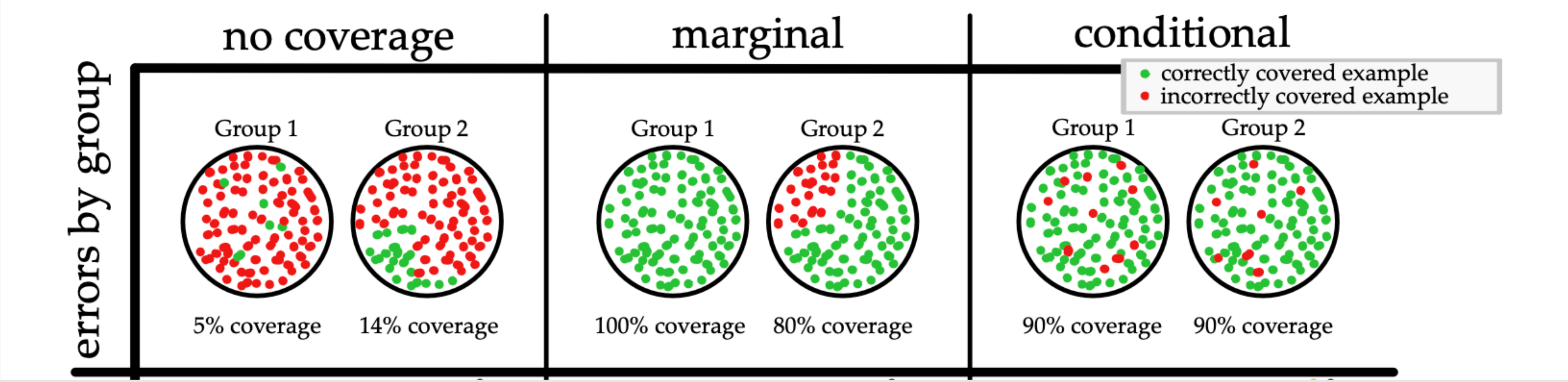


Figure from Angelopolous and Bates 2021

Devil is in the detail

- Marginal coverage guarantee doesn't make single predictions particularly useful

Marginal means "on average": given a new set of examples that is statistically equivalent to the calibration set, the predictions sets contain the true value "x"% of the time

- What we really want is conditional coverage: conditional on any split of the data, i.e. for every example

Unfortunately, this is relatively easy to obtain on classes, but impossible to achieve in general

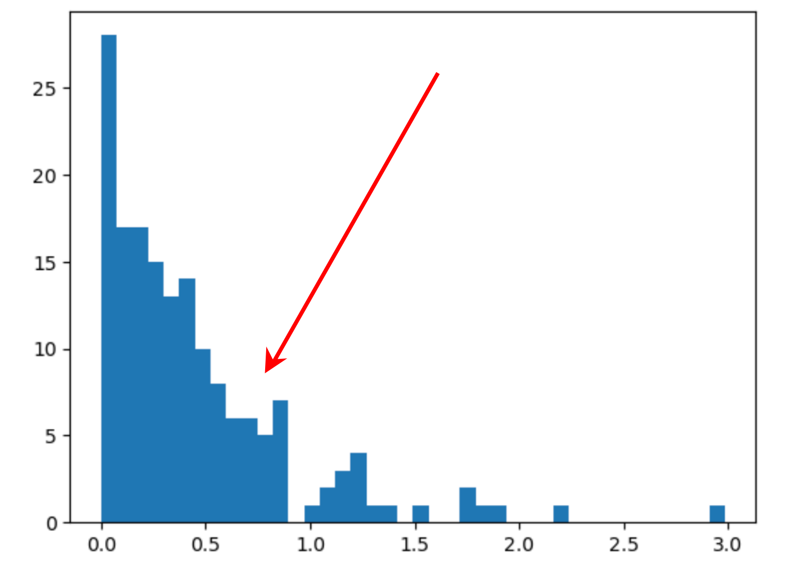
- A trivial solution to enforcing "including x% of the true classes" is to include all. Obviously useless

- Empirically, we look for small (not including too many) and adaptive (can recognize easy vs difficult examples; proxy for conditional coverage) prediction sets ⇒ look at size and spread in size of sets/intervals as diagnostics

Algorithms: APS, RAPS (Adaptive Pred Sets, Regularized Adaptive Pred Sets)

Regression

main difference is in the scores



Simplest version:

- Train regression model that outputs point predictions $\hat{y}$
- Pick desired coverage level α (e.g., 80%)
- Build conformal scores on calibration set (low = good) – what could work?
- e.g., absolute residuals $|y - \hat{y}|$
- Find quantile of residuals q corresponding to desired coverage, times finite sample correction $(n+1)/n$ ⇨ $\hat{q}$
- Build intervals by adding this number to/subtracting from point prediction: $[\hat{y} - \hat{q}\ ,\ \hat{y} + \hat{q}]$
- Marginal coverage still holds!

Let's say that we are interested in 80% coverage; alpha = 0.2. We find the corresponding 80% quantile of the absolute residuals, with the usual finite sample size correction:

```
alpha = 0.2
```

```
qhat = np.quantile(np.abs(y_calib_pred - y_calib), np.ceil((n+1)*(1-alpha))/n)
```

```
qhat
```

```
0.7055971501774867
```

We can now generate intervals by adding this quantity on either side of the point predictions:

```
intervals = np.hstack([(y_calib_pred - qhat).reshape(-1,1), (y_calib_pred + qhat).reshape(-1,1)])
```

To check calibration, we ask how often the true value is found in the intervals:

```
#whether true value is in interval

inint = np.array([intervals[i][0] < y_calib[i] < intervals[i][1] for i in range(len(y_calib))], dtype = int)
```

```
inint.mean() #Success! (Coverage is as expected)
```

```
0.806060606060606
```

We should also check that the coverage holds on the validation set:

```
y_val_pred = model.predict(X_val)
intervals_val = np.hstack([(y_val_pred - qhat).reshape(-1,1), (y_val_pred + qhat).reshape(-1,1)])
inint_val = np.array([intervals_val[i][0] < y_val[i] < intervals_val[i][1] for i in range(len(y_val))], dtype = int)
inint_val = np.array([intervals_val[i][0] < y_val[i] < intervals_val[i][1] for i in range(len(y_val))], dtype = int)
inint_val.mean() #Not bad! Should also marginalize over effect of split of data etc.
```

```
0.8363636363636363
```

Conformalized quantile regression

- Start from model that outputs quantiles

  (e.g., GradientBoostingRegressor w quantile loss)
- Pick desired coverage (e.g., 80%)
- Define up/low quantiles, e.g. $\widehat{GBRq}_{\text{low}} = 0.1$; $\widehat{GBRq}_{\text{up}} = 0.9$
- On calibration set, compute conformal scores

  $s(x, y) = \max(\widehat{GBRq}_{\text{low}} - y, y - \widehat{GBRq}_{\text{up}})$
- Median of s indicates typical correction needed by the intervals

  (negative = intervals were too wide; positive = too narrow)
- Find quantile q of scores s, with usual correction ⇨ $\hat{q}$
- $\hat{q}$ is added to both sides of the intervals for each example
- Adaptivity is achieved only through the original quantiles

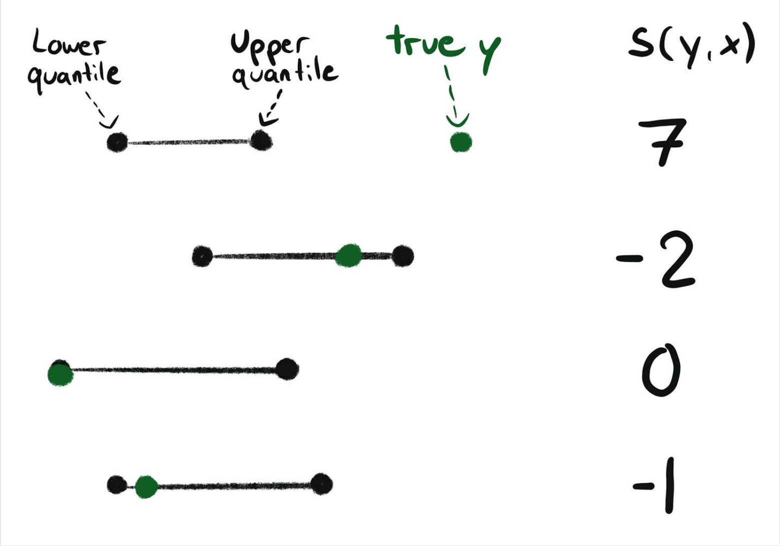


Figure by Christoph Molnar

References

- For the code relative to examples here:
https://github.com/vacquaviva/IntroConformalPredictions

- A paper comparing different UQ methods (including CP) for SED fitting:
https://www.mdpi.com/2674-0346/3/1/2

- Angelopoulos, Anastasios N., and Stephen Bates. “Conformal prediction: A gentle introduction.” Foundations and Trends® in Machine Learning 16.4 (2023): 494-591.
https://github.com/aangelopoulos/conformal-prediction

- Christoph Molnar has a series of (free) great blog posts and a book:
https://mindfulmodeler.substack.com/p/week-1-getting-started-with-conformal
https://christophmolnar.com/books/conformal-prediction/

- Conversation starter:

https://statmodeling.stat.columbia.edu/2024/02/20/when-do-we-expect-conformal-prediction-sets-to-be-helpful/

- Python Implementation:
Model Agnostic Prediction Interval Estimator: https://mapie.readthedocs.io/en/latest/

Summary: Uncertainty quantification methods

- Calibrating the uncertainties is always tricky. You can only do it if you have all sources of uncertainty under control. No free lunch.
- It is also necessary. Just like lunch.
- Conformal Predictions is a very active area of development (related: conformal risk control)
  - Methods that are proxies for conditional coverage are the most useful in practice
  - Question to ask: What is useful? What do you know about the data? Is it important to be conservative? Is adaptivity very important?
- Bayesian methods vs Conformal Predictions:
  - Do a different thing (produce calibrated sets/intervals vs produce posteriors)
  - – *Bayesian methods require assumptions about data distribution + priors*
  - – *Conformal Prediction Sets are always not always useful, even with guarantees*
- ¿Por qué no los dos?