

Problem set 2: Conformal prediction

Foundations of Machine Learning, Hilary term 2027

Maximilian Kasy

What this problem set is about

Conformal prediction takes *any* prediction method, however complicated, and turns its predictions into prediction intervals (for continuous outcomes) or prediction sets (for discrete outcomes) that are guaranteed to contain the true outcome with probability at least $1 - \alpha$. In lectures, we proved this guarantee using nothing more than the exchangeability of the data.

In this problem set you will implement conformal prediction for a regression problem (Problem 1) and a classification problem (Problem 2), and check the coverage theorem numerically. Just as importantly, you will explore what the theorem does *not* promise: it guarantees coverage on average, but not for every subgroup of the data, and not when the data are not exchangeable.

How this problem set works

1. **Software.** This problem set has to be solved in Python, using scikit-learn. It builds on Problem set 1.
2. **Your report.** Your code should run from start to end in one execution, producing all the output. Code, output, and discussion should be integrated in a single report (a Jupyter notebook), which you submit as a pdf. Please label all figures and tables clearly, so that someone who has not read your code can understand them. For reference: my solution runs in less than a minute on a laptop.
3. **Predictions.** Throughout the problem set, you will find blue boxes labelled *Predict* asking you to **predict** what you will find, *before* you run the corresponding code. Please take these seriously: they are the most useful part of the exercise. Write down your prediction in your report, in a few sentences, together with a brief reason. Keep your prediction in the report even if it turns out to be wrong; your predictions are not assessed on whether they are correct. At the end of each problem, a short section asks you to look back at your predictions: Were they right? If not, what did you overlook? This is the discussion part of your report; there is no separate discussion to write.
4. **Checkpoints.** The grey boxes list numbered *checkpoints*, [C2.1], [C2.2], and so on: specific numbers that your code should produce. For many of them, the theory from class tells you what the value should be; in that case, the box says so. If your number differs from the theoretical prediction by more than Monte Carlo error, say so in your report and try to explain why. At the end of your report, print all checkpoints in one block, one per line, in exactly this format:

CHECKPOINT C2.1 = 0.9002

for instance using `print(f"CHECKPOINT C2.1 = {value:.4f}")`.

5. **Preparing for the exam.** Each problem ends with a green box of *questions for the exam*. You do not need to answer these in your report. The exam will contain questions similar to these, about figures like the ones you produce here, and about short pieces of code from my solution. If you have done the problem set yourself and thought about your predictions, you should find them straightforward.

FROM THE LECTURES split conformal prediction

We use the following procedure throughout, with $\alpha = 0.1$ (that is, we are aiming for 90% coverage).

1. Train a prediction model on a *training set*.
2. Choose a *conformity score* $s(x, y)$, which is large when the outcome y is unusual given x , according to the trained model.
3. For the n observations in a separate *calibration set*, compute the scores $s_i = s(X_i, Y_i)$.
4. Let $\hat{q}$ be the $\lceil (n+1)(1-\alpha) \rceil$ -th smallest of the calibration scores $s_1, \dots, s_n$. (If $\lceil (n+1)(1-\alpha) \rceil > n$, set $\hat{q} = \infty$.) Please implement this step yourself, rather than using `np.quantile`, so that you see exactly what is going on.
5. For a new observation with covariates x , the prediction set contains all outcomes that would not look too unusual: $\mathcal{C}(x) = \{y : s(x, y) \leq \hat{q}\}$.

Coverage theorem. If the calibration observations and the new observation are exchangeable (for instance i.i.d.), then

$$P(Y_{test} \in \mathcal{C}(X_{test})) \geq 1 - \alpha.$$

In fact, as long as the scores have no ties, the coverage probability equals exactly $\lceil (n+1)(1-\alpha) \rceil / (n+1)$. The probability here is over the draw of the calibration data and of the new observation, holding the trained model fixed.

How we will check the theorem. The theorem is about an average over many possible calibration sets. To see this average, we keep the trained model fixed, and then repeat the following $R = 500$ times: randomly split the held-out data (all the data not used for training) into two halves, a calibration set and a test set; compute $\hat{q}$ from the calibration set; and record how well the prediction sets do on the test set.

Problem 1: Prediction intervals for house prices

Data

We use the California Housing data again (`fetch_california_housing()` in `sklearn.datasets`): median house values y for 20,640 districts, and 8 covariates. Split the data into a training set (60%) and a held-out set (40%), using

```
train_test_split(X, y, train_size=0.6)
```

Below, “income” always refers to the first covariate, `MedInc`, the median income in a district.

(a) Intervals from a point prediction

The simplest approach takes a point prediction $\hat{f}(x)$, and asks how far the outcome is from it. On the training set, fit ridge regression with standardized covariates and a cross-validated penalty:

```
make_pipeline(StandardScaler(), RidgeCV(alphas=np.logspace(-3, 3, 13)))
```

Use the absolute residual as the conformity score, $s(x, y) = |y - \hat{f}(x)|$. The prediction set is then simply the interval $\mathcal{C}(x) = [\hat{f}(x) - \hat{q}, \hat{f}(x) + \hat{q}]$.

(b) Intervals from quantile regression

A more flexible approach first estimates, for each x , a range in which y is likely to fall, and then uses conformal prediction to correct this range if needed. On the training set, fit models $t_{\alpha/2}(x)$ and $t_{1-\alpha/2}(x)$ for the 5% and 95% conditional quantiles of y given x , using

```
HistGradientBoostingRegressor(loss="quantile", quantile=tau)
```

with `tau` equal to 0.05 and 0.95. Use the score from the slides, $s(x, y) = \max(t_{\alpha/2}(x) - y, y - t_{1-\alpha/2}(x))$, which measures how far y falls outside the estimated range (and is negative if y falls inside). The prediction set is the interval $\mathcal{C}(x) = [t_{\alpha/2}(x) - \hat{q}, t_{1-\alpha/2}(x) + \hat{q}]$.

(c) Evaluation

PREDICT P2.1

before you run the code

- Will the average coverage across the 500 splits be above, below, or close to 90%? Will it differ between the two methods?
- How much do you expect the coverage to vary from one calibration/test split to the next: by about 0.1%, 1%, or 10%?
- Which of the two methods will have narrower intervals on average?
- House prices are much more variable in high-income districts. Which method will have better coverage in the districts with the highest income?

For both methods, use the $R = 500$ repeated calibration/test splits described above to compute, for each split: the coverage on the test set, the average width of the intervals, and the coverage within each quintile of income (with the quintile cutoffs computed from the training data).

WHAT TO REPORT

Figure 2.1: Two panels.

- (i) Histograms of the test-set coverage across the 500 splits, for both methods, with a vertical line at the theoretical coverage $\lceil (n+1)(1-\alpha) \rceil / (n+1)$.
- (ii) A bar chart of the average coverage within each income quintile, for both methods, with a horizontal line at $1 - \alpha$.

[C2.1]: the theoretical coverage $\lceil (n+1)(1-\alpha) \rceil / (n+1)$, for your calibration sample size n .

[C2.2]: average coverage, ridge. Theory: [C2.1].

[C2.3]: average coverage, quantile regression. Theory: [C2.1].

[C2.4]: average coverage in the highest income quintile, ridge.

[C2.5]: average coverage in the highest income quintile, quantile regression.

Please also report the average interval width of both methods (no checkpoint).

(d) What if the data are not exchangeable?

The coverage theorem relies on the calibration data and the test data being exchangeable: in particular, they need to come from the same distribution. Let us see what happens when they do not. Using the ridge model and score from part (a), compute $\hat{q}$ using only the held-out districts with income *at or below* the median income in the held-out data. Then compute the coverage for the held-out districts with income *above* the median.

PREDICT P2.2

before you run the code

Will the coverage for the high-income districts be above 90%, below 90%, or about 90%? Why?

WHAT TO REPORT

[C2.6]: the coverage for the high-income districts.

(e) Looking back at your predictions

Return to your predictions P2.1 and P2.2. For each one, say in a sentence or two whether it was right, and if not, what you overlooked. Where you can, relate your answer to the coverage theorem, and to the difference between unconditional and conditional coverage.

QUESTIONS FOR THE EXAM

1. State the coverage theorem for split conformal prediction. Which assumption does it require about the data?

2. Compute $\lceil (n+1)(1-\alpha) \rceil / (n+1)$ for $\alpha = 0.1$ and $n = 19$. What happens to $\hat{q}$ and to the prediction sets when $n = 4$?
3. Panel (i) of Figure 2.1 shows that coverage varies across calibration/test splits. Does this contradict the coverage theorem? Why or why not?
4. Why do the ridge intervals all have the same width?
5. Use panel (ii) of Figure 2.1 to explain the difference between unconditional and conditional coverage. Which method comes closer to conditional coverage, and why?
6. In part (d), which assumption of the coverage theorem is violated? Explain why coverage fails in the direction that it does.

Problem 2: Prediction sets for handwritten digits

Data and model

We now turn to a classification problem: recognizing handwritten digits, as in our first lecture notebook. Load the digits data (`sklearn.datasets.load_digits()`), with 1,797 images of digits $0, \dots, 9$. Split the data into a training set (60%) and a held-out set (40%), using

```
train_test_split(X, y, train_size=0.6, stratify=y)
```

On the training set, fit penalized multinomial logistic regression with a cross-validated penalty,

```
make_pipeline(StandardScaler(), LogisticRegressionCV(Cs=10, cv=5, max_iter=5000))
```

and let $\hat{f}(y|x)$ denote the predicted probability of digit y given image x (from `predict_proba`).

(a) Two scores

We consider the two scores for classification discussed in the slides.

- The first score is one minus the predicted probability of the label,

$$s_1(x, y) = 1 - \hat{f}(y|x).$$

- The second score adds up the predicted probabilities of all labels that are at least as likely as y ,

$$s_2(x, y) = \sum_{y'} \hat{f}(y'|x) \cdot \mathbf{1}(\hat{f}(y'|x) \geq \hat{f}(y|x)).$$

For example, if the model is 70% sure that an image is a 3, 20% that it is an 8, and 10% that it is a 5, then $s_2(x, 3) = 0.7$, $s_2(x, 8) = 0.9$, and $s_2(x, 5) = 1$.

For both scores, the prediction set is $\mathcal{C}(x) = \{y \in \{0, \dots, 9\} : s(x, y) \leq \hat{q}\}$.

(b) Class-conditional calibration

The slides discuss how to obtain coverage conditional on groups, by computing a separate quantile for each group. Here the groups are the true labels. For the score s_1 , implement this: for each digit y , compute a separate $\hat{q}_y$, using only the calibration observations whose true label is y , and let $\mathcal{C}(x) = \{y : s_1(x, y) \leq \hat{q}_y\}$.

(c) Evaluation

PREDICT P2.3

before you run the code

- Which of the three procedures (score s_1 , score s_2 , and score s_1 with class-conditional calibration) will produce the largest prediction sets on average?

- Can a prediction set be empty? For which of the procedures?
- Will all ten digits be covered with probability close to 90%? Which procedure would you trust most for the digit that is hardest to recognize?

For each of the three procedures, use $R = 500$ repeated calibration/test splits of the held-out data to compute, for each split: the coverage, the average size of the prediction sets, the share of empty prediction sets, and the coverage for each true digit.

WHAT TO REPORT

Figure 2.2: A bar chart with the average coverage for each true digit (horizontal axis), for each of the three procedures, with a horizontal line at $1 - \alpha$. Report the average set size of each procedure in the legend.

[C2.7]: the theoretical coverage, for your calibration sample size.

[C2.8]: average coverage, score s_1 . Theory: [C2.7].

[C2.9]: average set size, score s_1 .

[C2.10]: average share of empty prediction sets, score s_1 .

[C2.11]: average set size, score s_2 .

[C2.12]: the lowest average coverage across the 10 digits, score s_1 .

[C2.13]: the lowest average coverage across the 10 digits, score s_1 with class-conditional calibration.

(d) Looking back at your predictions

Return to your prediction P2.3. For each part, say in a sentence or two whether it was right, and if not, what you overlooked.

QUESTIONS FOR THE EXAM

1. Compute s_1 and s_2 for all labels, given predicted probabilities (0.6, 0.3, 0.1) for three classes. Which labels are in the prediction set if $\hat{q} = 0.85$, for each score?
2. Why can the prediction sets based on s_1 be empty? Why does this not contradict the coverage theorem?
3. The slides say that a problem with the score s_1 is that prediction sets are small when uncertainty is high. Explain why, and how the score s_2 addresses this.
4. Why does class-conditional calibration guarantee coverage of at least $1 - \alpha$ for each digit? What is the cost?