Coding exercise: Conformal inference
Foundations of Machine Learning - Oxford Summer School 2025

Maximilian Kasy

In this problem, you are asked to implement and evaluate some conformal inference procedures in Python. This problem builds on the first coding exercise, using Scikit-learn.

Your code should run from start to end in one execution, producing all the output. Output and discussion of findings should be integrated in a report generated in a Jupyter Notebook. Figures and tables should be clearly labeled and interpretable. The findings should be discussed in the context of the theoretical results that we derived in class.

1. **Preparation - Discrete Classification**

   Load a discrete outcome dataset from the scikit-learn package. For this exercise, use the **Wine dataset** (`sklearn.datasets.load_wine()`) which contains 3 classes of wine based on chemical analysis,
   or alternatively the **Breast Cancer dataset** (`sklearn.datasets.load_breast_cancer()`) for binary classification.

   Split the dataset into three parts: a **training set** (60%) used for initial model fitting, a **calibration set** (20%) used for conformal inference calibration, and a **test set** (20%) used for final evaluation.

   Fit a penalized logistic regression model using scikit-learn's `LogisticRegression` with L1 or L2 regularization. Use cross-validation on the training set to find the optimal penalty parameter λ that minimizes classification error.

2. **Conformal Inference for Classification**

   Implement conformal inference for the classification problem using the following procedure. Let $j(y, x)$ be the rank of $\hat{f}(y|x)$ across all possible labels y, where $\hat{f}(y|x)$ represents the predicted probability of class y given features x. Define the conformity score as:

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

   where $\mathbf{1}(\cdot)$ is the indicator function.

   Follow these steps: (a) Train your penalized logistic regression model on the training set using the optimal penalty found in the previous problem. (b) For each point (x_i, y_i) in the calibration set, compute the conformity score $s(x_i, y_i)$. (c) For a desired confidence level $1 - \alpha$ (e.g., $\alpha = 0.1$ for 90% confidence), find the $(1 - \alpha)(n + 1)/n$-th quantile of the calibration scores, where n is the size of the calibration set. (d) For each test point x, construct the prediction set $C(x) = \{y : s(x, y) \geq \text{quantile}\}$.

Evaluate the method on the hold-out test sample by computing: the **coverage** (fraction of test points where the true label is in the prediction set), the **average set size** (mean number of labels in the prediction sets), and the **conditional coverage** (coverage rates for each true class).

3. **k-Nearest Neighbors Classification**

   Repeat the conformal inference procedure from the previous problem, but replace the penalized logistic regression with k-nearest neighbors classification. Use cross-validation on the training set to select the optimal number of neighbors k. Use the same conformity score definition as in the previous problem. Compare the coverage and efficiency (average prediction set size) with the logistic regression results, and discuss any differences in performance between the two methods.

4. **Continuous Outcome Regression**

   Switch to a continuous outcome dataset.
   Use the **Boston Housing dataset** (`sklearn.datasets.load_boston()`) or the **California Housing dataset** (`sklearn.datasets.fetch_california_housing()`) for predicting housing prices.

   Implement conformal inference for regression using ridge regression with penalty parameter selected via cross-validation as the model, and the conformity score:

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

   where $\hat{f}(x)$ is the predicted value from the ridge regression model.

   For evaluation, construct prediction intervals $[\hat{f}(x) - q, \hat{f}(x) + q]$ where q is the appropriate quantile of calibration residuals. Compute empirical coverage on the test set, calculate average interval width, and plot prediction intervals for a subset of test points compared with true values.

5. **Quantile Regression Conformal Inference**

   Using the same continuous outcome dataset from the previous problem, implement conformal inference with quantile regression. Fit quantile regression models for quantiles $\tau_{\alpha/2}$ and $\tau_{1-\alpha/2}$ (e.g., 0.05 and 0.95 for $\alpha = 0.1$). Let $t_{\alpha/2}(x)$ and $t_{1-\alpha/2}(x)$ be the predicted quantiles.

   Use the conformity score:

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

   Note that you can use scikit-learn's `QuantileRegressor`. The conformity score measures how far the true value y falls outside the predicted quantile interval, with negative scores indicating the point lies within the predicted interval.

   Compare coverage and interval widths with the ridge regression approach from the previous problem. Analyze whether quantile regression provides more adaptive intervals and discuss the trade-offs between the two approaches.

For each problem, provide: (1) **Code implementation** with clear comments, (2) **Results summary** including coverage rates, prediction set sizes (classification) or interval widths (regression), (3) **Visualization** of prediction sets/intervals for selected examples, (4) **Discussion** of the strengths and limitations of each approach, and (5) **Comparison** across different methods and datasets.

Suggested Dataset Alternatives

Classification:

- `sklearn.datasets.load_digits()` for handwritten digit recognition (10 classes),
- `sklearn.datasets.load_iris()` for iris flower classification (3 classes), or
- `sklearn.datasets.make_classification()` for synthetic classification data.

Regression:

- `sklearn.datasets.load_diabetes()` for diabetes progression prediction,
- `sklearn.datasets.make_regression()` for synthetic regression data, or
- any regression dataset from `sklearn.datasets`.