PUNCC:
Empowering trustworthy AI with reliable predictive uncertainty quantification

Joseba DALMAU
Mouhcine MENDIL
Uncertainty: A random world
Predictive Uncertainty
Epistemic vs Aleatoric Uncertainty

?
Epistemic vs Aleatoric Uncertainty

12/25 = 0.48 are H
Epistemic vs Aleatoric Uncertainty

3543 / 5000 = 0.7086 are H

11/01/2024
Epistemic vs Aleatoric Uncertainty

**Epistemic Uncertainty:**
- Unrelated to the phenomenon itself
- Can be reduced with:
  - Additional Observation
  - Noise Reduction
  - Model Complexity

3543 / 5000 = 0.7086
are H

**Aleatoric Uncertainty:**
- Inherent to the phenomenon
- Cannot be reduced
Uncertainty Quantification

Quantify the uncertainty of a model based on different parameters:

- Number of observations,
- prior knowledge,
- complexity of the model
- ...
Uncertainty Quantification

Classification

Object Detection
Why Uncertainty Quantification?

- **Evaluate** the accuracy of the model
- Design **trustworthy** ML components
- Help in the **certification** process of AI-based modules
How to Quantify Uncertainty?

- Model calibration (e.g. Platt scaling, temperature scaling, …)

- Bayesian Networks

- Model Complexity

- …
Conformal Prediction

• Model agnostic
• Post-processing
• Finite sample

\[ P(Y_{new} \in \hat{C}(X_{new})) \geq 1 - \alpha \]
Conformal Prediction

Black-box model

Input $X_{new}$  $\rightarrow$ Prediction $\hat{Y}_{new}$
Conformal Prediction

User-defined risk level: $\alpha$

Probabilistic Guarantee: $P(Y_{new} \in \hat{C}(X_{new})) \geq 1 - \alpha$
A statistical approach

We are given a black-box predictor
A statistical approach

Use a representative calibration dataset to measure errors
A statistical approach

Learn prediction intervals with probabilistic guarantees
A statistical approach

The intervals can be adaptive to heteroskedasticity
Beyond Regression and Split-CP

- Regression, Classification, Anomaly Detection, Object Detection,…
- Jacknife+, CV+
- …
Conformal Prediction in Practice with PUNCC
Conformal Prediction: Procedure

Model without UQ (or uncalibrated) → Conformalization (Postprocessing) → Predictive UQ

Conformal prediction
Conformal Prediction: Procedure

- Model without UQ (or uncalibrated)
- Conformalization (Postprocessing)
- Predictive UQ

Additional data
Light computation

Probabilistic guarantees on error rate
PUNCC Library (Predictive UNcertainty Calibration and Conformalization)
PUNCC for different ML Tasks

Most likely class predicted by model

Prediction set with probabilistic guarantees

Conformalization

Conformal prediction

Isolation Forest

Conformalized Isolation Forest
PUNCC: Regression

Conformal prediction

Diabetes progression in one year

Body Mass Index (standardized)
PUNCC: Classification

- Most likely class predicted by model: $\{5\}$ with 99.7% confidence
- Prediction set with probabilistic guarantees: $\{5, 6\}$ with 99.7% and 0.03% confidence

Conformalization
PUNCC: Object Detection
PUNCC: Anomaly Detection

Isolation Forest

Conformalized Isolation Forest

Normal
Anomaly
PUNCC for different ML Tasks
Conformal Prediction in few lines of code

**Conformal Regression**

```python
from deel.puncc.regression import SplitCP

# Instantiate conformal predictor
cp_alg = SplitCP(Predictor)

# Compute calibration scores
cp_alg.fit(X_calib, y_calib)

# Generate prediction sets
y_pred, y_low, y_high = cp_alg.predict(X_new, alpha=0.1)
```
Conformal Prediction in few lines of code

Conformal Classification

```python
from deel.puncc.classification import APS

# Instantiate conformal predictor
cp_alg = APS(Predictor)

# Compute calibration scores
cp_alg.fit(X_calib, y_calib)

# Generate prediction sets
y_pred, set_pred = cp_alg.predict(X_new, alpha=0.1)
```
Conformal Prediction in few lines of code

**Conformal Object Detection**

```python
from deel.puncc.object_detection import SplitBoxWise

# Instantiate conformal predictor
cp_alg = SplitBoxWise(Predictor)

# Compute calibration scores
cp_alg.fit(X_calib, y_calib)

# Generate prediction sets
y_pred, y_inner, y_outer = cp_alg.predict(X_new, alpha=0.1)
```
Interoperability

- PUNCC supports popular data types and ML libraries and more …

- Can work on top of UQ models and libraries
Demo I: Classification

- Pretrained classifier within existing ML pipeline

Source: D. Decoste
Demo II: Object Detection

- Predictions accessible via API

Model API → Conformalization → Conformal Predictor

Calibration Data

https://cocodataset.org/
Demo III: Regression

- Model to be selected and trained from scratch

![Diagram showing model and data flow](source: https://scikit-learn.org)

**Table: Dataset Attributes**

<table>
<thead>
<tr>
<th>Number of Instances:</th>
<th>442</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of Attributes:</td>
<td>First 10 columns are numeric predictive values</td>
</tr>
<tr>
<td>Target:</td>
<td>Column 11 is a quantitative measure of disease progression one year</td>
</tr>
</tbody>
</table>

**Attribute Information**

- age: age in years
- sex
- bmi: body mass index
- bp: average blood pressure
- s1 t.c: total serum cholesterol
- s2 LDL: low-density lipoproteins
- s3 HDL: high-density lipoproteins
- s4 t.ch: total cholesterol / HDL
- s5 log: possibly log of serum triglycerides level
- s6 glu: blood sugar level

Source: https://scikit-learn.org
PUNCC Project

- Documentation
- Tutorials
- Tests
- Updates
- Open to contributions

https://github.com/deel-ai/puncc
Thanks for your attention!
Conditional Guarantees

90% marginal coverage
\[ P(Y_{new} \in \hat{C}(X_{new})) \geq 0.9 \]

90% conditional coverage
\[ \forall x, P(Y_{new} \in \hat{C}(X_{new}) \mid X_{new} = x) \geq 0.9 \]

Note that: conditional coverage ⇒ marginal coverage