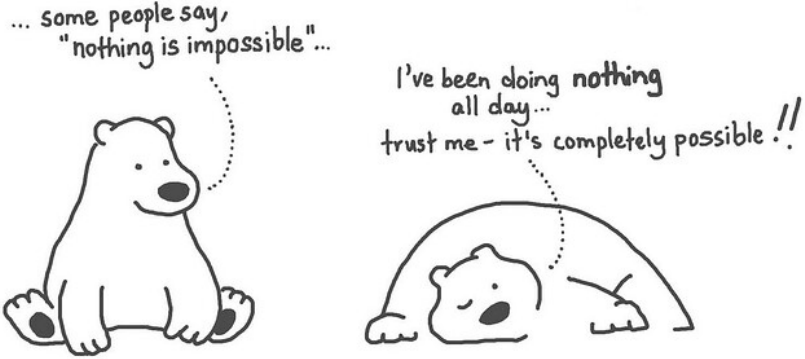
PROBABILITY: CERTAIN, POSSIBLE, IMPOSSIBLE



Uncertainty reasoning and machine learning

Introduction to notions of calibrated and valid predictions

Vu-Linh Nguyen

Chaire de Professeur Junior, Laboratoire Heudiasyc
Université de technologie de Compiègne

AOS4 master courses

A predictive system

- perceives **a training data set** (consisting of input-output pairs which specify individuals of a population) and **a hypothesis space** (consisting of the possible classifiers),

A predictive system

- perceives **a training data set** (consisting of input-output pairs which specify individuals of a population) and **a hypothesis space** (consisting of the possible classifiers),
- and seeks a classifier that **optimizes** its chance of making accurate predictions with respect to some given **evaluation criterion** (which is typically a loss function or an accuracy metric) which reflects how good/bad the predictive system is.

Optimization problem should be described after declaring

- a training (+ validation) data set,
- a hypothesis space,
- an evaluation criterion,
- and a notion of an optimal classifier.

Optimization Problem: “Spam in Emails" Example

What optimization problem do you want to solve?

- Using a decision tree to predict “Spam in Emails"

Optimization Problem: “Cat Dog classification" Example

What optimization problem do you want to solve?

- Using a convolutional neural network (CNN) to predict images as either a cat or a dog

Objectives

After this lecture students should be able to

- describe commonly used notions of classifier calibration [10]
- describe a few calibration errors and calibration methods [10]
- describe commonly used notions of coverage [1]
- describe a few coverage metrics and conformal procedures [1]

Outline

- Classifier Calibration
  - Introduction
  - Notions
  - Calibration Errors
  - Post-hoc Calibration
  - Other methods
- Conformal Prediction

Outline

- Classifier Calibration
  - Introduction
  - Notions
  - Calibration Errors
  - Post-hoc Calibration
  - Other methods
- Conformal Prediction

A Weather Forecasting Example

A Weather Forecasting Example

- Forecaster: “the probability of rain tomorrow in Compiègne is 80%"
- How could we interpret this forecast?

A Weather Forecasting Example (cont.)

- On about 80% of the days when the whether conditions are like tomorrow's, you would experience rain in Compiègne?
- It will rain in 80% of the land area of Compiègne?
- It will rain in 80% of the time?

A Weather Forecasting Example (cont.)

- On about 80% of the days when the whether conditions are like tomorrow's, you would experience rain in Compiègne?
- It will rain in 80% of the land area of Compiègne?
- It will rain in 80% of the time?

Determining the degree to which a forecaster is well-calibrated

- cannot be done on a per-forecast basis,
- but requires looking at a sufficiently large and diverse set of forecasts.

Why Calibration Matters?

A well-calibrated classifier is expected to

- generate estimated class probabilities, which are consistent with what would naturally occur.

Why Calibration Matters?

A well-calibrated classifier is expected to

- generate estimated class probabilities, which are consistent with what would naturally occur.

If (heterogeneous) classifiers can be well-calibrated,

- their estimated class probabilities may be of the same “scale" and may be combined
- they can be further compared given the same/similar levels of predictive performance.

Outline

- Classifier Calibration
  - Introduction
  - Notions
  - Calibration Errors
  - Post-hoc Calibration
  - Other methods
- Conformal Prediction

Notions of Calibration (Mentioned in Lecture 3)

Confidence calibration [3]:

$$P(y = \arg\max_{y \in \mathcal{Y}} \theta_y | \boldsymbol{x} \text{ such that } \max_{y \in \mathcal{Y}} \theta_y | \boldsymbol{x} = \beta) = \beta, \forall \beta \in [0, 1]. \quad (1)$$

Notions of Calibration (Mentioned in Lecture 3)

Confidence calibration [3]:

$$P(y = \arg\max_{y \in \mathcal{Y}} \theta_y | \boldsymbol{x} \text{ such that } \max_{y \in \mathcal{Y}} \theta_y | \boldsymbol{x} = \beta) = \beta, \forall \beta \in [0, 1]. \tag{1}$$

Classwise calibration [12]:

$$P(y \text{ such that } \theta_y | \boldsymbol{x} = \beta_y) = \beta_y, y \in \mathcal{Y}, \beta_y \in [0, 1]. \tag{2}$$

- May be harder to ensure, compared to **confidence calibration**

Notions of Calibration (Mentioned in Lecture 3)

Confidence calibration [3]:

$$P(y = \arg\max_{y \in \mathcal{Y}} \theta_y | \boldsymbol{x} \text{ such that } \max_{y \in \mathcal{Y}} \theta_y | \boldsymbol{x} = \beta) = \beta, \forall \beta \in [0, 1]. \tag{1}$$

Classwise calibration [12]:

$$P(y \text{ such that } \theta_y | \boldsymbol{x} = \beta_y) = \beta_y, y \in \mathcal{Y}, \beta_y \in [0, 1]. \tag{2}$$

- May be harder to ensure, compared to **confidence calibration**

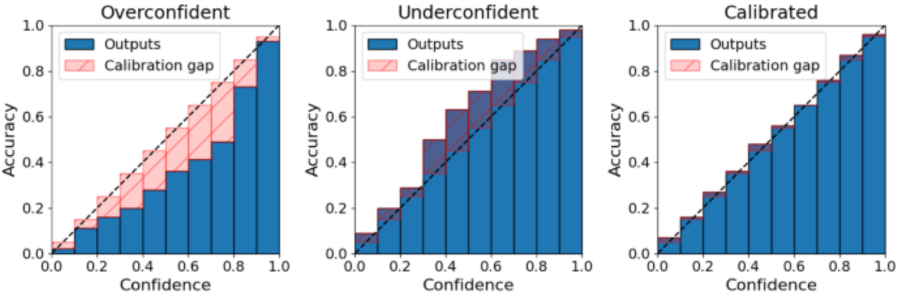
Distribution calibration [4]:

$$P(y \text{ such that } \boldsymbol{\theta} | \boldsymbol{x} = \boldsymbol{q}) = \boldsymbol{q}, \forall \boldsymbol{q} \in \triangle^{|\mathcal{Y}|}, \tag{3}$$

where $\triangle^{|\mathcal{Y}|}$ is the $|\mathcal{Y}|$-dimensional simplex

- May be harder to ensure, compared to the **above notions**.

Notions of Calibration with Examples



(a) Underconfidence (b) Overconfidence (c) Calibrated classifier

Confidence calibration: Examples [2]

Notions of Calibration with Examples (Exercise 1)

Basic setup (rephrased from an example in [10]):

- A dataset contains 40 instances
- A model ***h*** which partitions the input space into 4 regions:

# **instances**	**Predicted probabilities**	**Class distributions**
10	(0.3,0.3,0.4)	(4,2,4)
10	(0.4,0.3,0.3)	(3,4,3)
10	(0.4,0.6,0.0)	(5,5,0)
10	(0.3,0.6,0.1)	(2,7,1)

Notions of Calibration with Examples (Exercise 1)

Basic setup (rephrased from an example in [10]):

- A dataset contains 40 instances
- A model ***h*** which partitions the input space into 4 regions:

# instances	Predicted probabilities	Class distributions
10	(0.3,0.3,0.4)	(4,2,4)
10	(0.4,0.3,0.3)	(3,4,3)
10	(0.4,0.6,0.0)	(5,5,0)
10	(0.3,0.6,0.1)	(2,7,1)

Question: Check if the following statements are correct

- ***h*** is not confidence-calibrated
- ***h*** is classwise-calibrated
- ***h*** is not distribution-calibrated

Notions of Calibration with Examples (Solution 1.1)

Basic setup (rephrased from an example in [10]):

# instances	Predicted probabilities	Class distributions
10	(0.3,0.3,**0.4**)	(4,2,4)
10	(**0.4**,0.3,0.3)	(3,4,3)
10	(0.4,**0.6**,0.0)	(5,5,0)
10	(0.3,**0.6**,0.1)	(2,7,1)

Statement: ***h*** is not confidence-calibrated

$$P(y = \arg\max_{y \in \mathcal{Y}} \theta_y | \boldsymbol{x} \text{ such that } \max_{y \in \mathcal{Y}} \theta_y | \boldsymbol{x} = \beta) = \beta, \forall \beta \in [0,1]. \qquad (4)$$

- $\beta = 0.4$: $P = {}^{(4+3)}/_{20} = {}^{7}/_{20} \neq 0.4$
- $\beta = 0.6$: $P = {}^{(5+7)}/_{20} = {}^{12}/_{20} = 0.6$

Notions of Calibration with Examples (Solution 1.2)

Basic setup (rephrased from an example in [10]):

# Instances	Predicted probabilities	Class distributions
10	(0.3,0.3,0.4)	(4,2,4)
10	(0.4,0.3,0.3)	(3,4,3)
10	(0.4,0.6,0.0)	(5,5,0)
10	(0.3,0.6,0.1)	(2,7,1)

Statement: ***h*** is classwise-calibrated

$$P(y \text{ such that } \theta_y | \boldsymbol{x} = \beta_y) = \beta_y, y \in \mathcal{Y}, \beta_y \in [0,1]. \tag{5}$$

- $y_1 \wedge \beta_1 = 0.3$: $P = {}^{(2+4)}/_{20} = 0.3$, $y_1 \wedge \beta_1 = 0.4$: $P = {}^{(3+5)}/_{20} = 0.4$
- $y_2 \wedge \beta_2 = 0.3$: $P = {}^{(2+4)}/_{20} = 0.3$, $y_2 \wedge \beta_2 = 0.6$: $P = {}^{(5+7)}/_{20} = 0.6$
- $y_3 \wedge \beta_3 = 0.4$: $P = {}^{4}/_{10} = 0.4$, $y_3 \wedge \beta_3 = 0.3$: $P = {}^{3}/_{10} = 0.3$
- $y_3 \wedge \beta_3 = 0.0$: $P = {}^{0}/_{10} = 0.0$, $y_3 \wedge \beta_3 = 0.1$: $P = {}^{1}/_{10} = 0.1$

Notions of Calibration with Examples (Solution 1.3)

Basic setup (rephrased from an example in [10]):

# Instances	Predicted probabilities	Class distributions
10	(0.3,0.3,0.4)	(4,2,4)
10	(0.4,0.3,0.3)	(3,4,3)
10	(0.4,0.6,0.0)	(5,5,0)
10	(0.3,0.6,0.1)	(2,7,1)

Statement: ***h*** is not distribution-calibrated

$$P(y \text{ such that } \boldsymbol{\theta} | \boldsymbol{x} = \boldsymbol{q}) = \boldsymbol{q}, \forall \boldsymbol{q} \in \triangle^{|\mathscr{Y}|}, \tag{6}$$

- $\boldsymbol{q} = (0.3, 0.3, 0.4)$: $P = (4/10, 2/10, 4/10) = (0.4, 0.2, 0.4) \neq (0.3, 0.3, 0.4)$
- $\boldsymbol{q} = (0.4, 0.3, 0.3)$: $P = (3/10, 4/10, 3/10) = (0.3, 0.4, 0.3) \neq (0.4, 0.3, 0.3)$
- $\boldsymbol{q} = (0.4, 0.6, 0.0)$: $P = (5/10, 5/10, 0/10) = (0.5, 0.5, 0.0) \neq (0.4, 0.6, 0.0)$
- $\boldsymbol{q} = (0.3, 0.6, 0.1)$: $P = (2/10, 7/10, 1/10) = (0.2, 0.7, 0.1) \neq (0.3, 0.6, 0.1)$

A Note on Classifier Calibration (Exercise 2)

Consider three notions of classifier calibration:

- Confidence calibration [3]:

$$P(y = \arg\max_{y \in \mathcal{Y}} \theta_y | \boldsymbol{x} \text{ such that } \max_{y \in \mathcal{Y}} \theta_y | \boldsymbol{x} = \beta) = \beta, \forall \beta \in [0,1]. \tag{7}$$

- Classwise calibration [12]:

$$P(y \text{ such that } \theta_y | \boldsymbol{x} = \beta_y) = \beta_y, y \in \mathcal{Y}, \beta_y \in [0,1]. \tag{8}$$

- Distribution calibration [4]:

$$P(y \text{ such that } \boldsymbol{\theta} | \boldsymbol{x} = \boldsymbol{q}) = \boldsymbol{q}, \forall \boldsymbol{q} \in \triangle^{|\mathcal{Y}|}, \tag{9}$$

  where $\triangle^{|\mathcal{Y}|}$ is the $|\mathcal{Y}|$-dimensional simplex.

Prove that these notions are equivalent for binary classification?

A Note on Classifier Calibration (Exercise 3)

Consider three notions of classifier calibration:

- Confidence calibration [3]:

$$P(y = \arg\max_{y \in \mathcal{Y}} \theta_y | \boldsymbol{x} \text{ such that } \max_{y \in \mathcal{Y}} \theta_y | \boldsymbol{x} = \beta) = \beta, \forall \beta \in [0,1]. \quad (10)$$

- Classwise calibration [12]:

$$P(y \text{ such that } \theta_y | \boldsymbol{x} = \beta_y) = \beta_y, y \in \mathcal{Y}, \beta_y \in [0,1]. \quad (11)$$

- Distribution calibration [4]:

$$P(y \text{ such that } \boldsymbol{\theta} | \boldsymbol{x} = \boldsymbol{q}) = \boldsymbol{q}, \forall \boldsymbol{q} \in \triangle^{|\mathcal{Y}|}, \quad (12)$$

where $\triangle^{|\mathcal{Y}|}$ is the $|\mathcal{Y}|$-dimensional simplex.

A Note on Classifier Calibration (Exercise 3)

Consider three notions of classifier calibration:

- Confidence calibration [3]:

$$P(y = \arg\max_{y \in \mathscr{Y}} \theta_y | \boldsymbol{x} \text{ such that } \max_{y \in \mathscr{Y}} \theta_y | \boldsymbol{x} = \beta) = \beta, \forall \beta \in [0,1]. \tag{10}$$

- Classwise calibration [12]:

$$P(y \text{ such that } \theta_y | \boldsymbol{x} = \beta_y) = \beta_y, y \in \mathscr{Y}, \beta_y \in [0,1]. \tag{11}$$

- Distribution calibration [4]:

$$P(y \text{ such that } \boldsymbol{\theta} | \boldsymbol{x} = \boldsymbol{q}) = \boldsymbol{q}, \forall \boldsymbol{q} \in \triangle^{|\mathscr{Y}|}, \tag{12}$$

where $\triangle^{|\mathscr{Y}|}$ is the $|\mathscr{Y}|$-dimensional simplex.

Prove that $\boldsymbol{h}(\boldsymbol{x}) = P(\mathscr{Y})$, $\forall$ $\boldsymbol{x}$, is perfectly calibrated?

Notes on Classifier Calibration (Cont.)

Comments on confidence/classwise/distribution calibration:

- **Well-calibrated classifiers may perform poorly**.
- Using calibration error as the only criterion to assess classifiers might not be a good idea ...
- **Well-calibrated and accurate classifiers** would be useful in practice!
- They would be seen as **notions of marginal calibration** $\longleftarrow$ **population level**

Outline

- Classifier Calibration
  - Introduction
  - Notions
  - Calibration Errors
  - Post-hoc Calibration
  - Other methods
- Conformal Prediction

Calibration Error: The Binary Case

Binary estimated calibration error (Binary-ECE):

- Specify a number M of bins
- Apply equal-width binning to $\theta_1|\boldsymbol{x}$ on $\mathbf{D}$
- For each bin $\mathbf{B}_m$, compute average probability $\bar{s}(\mathbf{B}_m)$ and the proportion of positives $\bar{y}(\mathbf{B}_m)$

$$\bar{s}(\mathbf{B}_m) = \frac{1}{|\mathbf{B}_m|} \sum_{\boldsymbol{x} \in \mathbf{B}_m} \theta_1|\boldsymbol{x}$$

$$\bar{y}(\mathbf{B}_m) = \frac{1}{|\mathbf{B}_m|} \sum_{\boldsymbol{x} \in \mathbf{B}_m} y$$

- Compute Binary-ECE

$$\text{Binary-ECE}(\mathbf{D}) = \sum_{m=1}^{M} \frac{|\mathbf{B}_m|}{|\mathbf{D}|} \left|\bar{y}(\mathbf{B}_m) - \bar{s}(\mathbf{B}_m)\right|$$

Calibration Error: The Binary Case (Exercise 4)

Basic setup:

- A given data set $\mathbf{D} = \{(\boldsymbol{x}_n, y_n) | n = 1, \ldots, N\}$ with $y \in \{0, 1\}$
- The proportion of instances with $y = 1$ is $0.5 + \epsilon$
- The decision rule is $0/1$ loss ℓ and the number of bins is 10

Questions:

- Show that there is at least one classifier with

$$\text{Binary-ECE}(\mathbf{D}) = 0.0 \text{ and } \frac{1}{N}\sum_{n=1}^{N} \ell(y_n^*, y_n) = 0.0$$

Calibration Error: The Binary Case (Exercise 4)

Basic setup:

- A given data set $\mathbf{D} = \{(\boldsymbol{x}_n, y_n) | n = 1, \ldots, N\}$ with $y \in \{0, 1\}$
- The proportion of instances with $y = 1$ is $0.5 + \epsilon$
- The decision rule is $0/1$ loss ℓ and the number of bins is 10

Questions:

- Show that there is at least one classifier with

$$\text{Binary-ECE}(\mathbf{D}) = 0.0 \text{ and } \frac{1}{N}\sum_{n=1}^{N} \ell(y_n^*, y_n) = 0.0$$

- Show that there is at least one classifier with

$$\text{Binary-ECE}(\mathbf{D}) = 0.0 \text{ and } \frac{1}{N}\sum_{n=1}^{N} \ell(y_n^*, y_n) = 0.5 - \epsilon$$

Calibration Error: The Binary Case (Exercise 4)

Basic setup:

- A given data set $\mathbf{D} = \{(\boldsymbol{x}_n, y_n) | n = 1, \ldots, N\}$ with $y \in \{0, 1\}$
- The proportion of instances with $y = 1$ is $0.5 + \epsilon$
- The decision rule is $0/1$ loss ℓ and the number of bins is 10

Questions:

- Show that there is at least one classifier with

$$\text{Binary-ECE}(\mathbf{D}) = 0.0 \text{ and } \frac{1}{N} \sum_{n=1}^{N} \ell(y_n^*, y_n) = 0.0$$

- Show that there is at least one classifier with

$$\text{Binary-ECE}(\mathbf{D}) = 0.0 \text{ and } \frac{1}{N} \sum_{n=1}^{N} \ell(y_n^*, y_n) = 0.5 - \epsilon$$

- Can we find worse perfectly calibrated classifiers?

Calibration Error: The Binary Case (Exercise 5)

Basic setup:

- A given data set $\mathbf{D} = \{(\boldsymbol{x}_n, y_n) | n = 1, \ldots, N\}$ with $y \in \{0, 1\}$
- The proportion of instances with $y = 1$ is $\alpha \neq 0.5$
- The decision rule is $0/1$ loss ℓ and the number of bins is M

Questions:

- Show that there is at least one classifier with

$$\text{Binary-ECE}(\mathbf{D}) = 0.0 \text{ and } \frac{1}{N}\sum_{n=1}^{N} \ell(y_n^*, y_n) = 0.0$$

Calibration Error: The Binary Case (Exercise 5)

Basic setup:

- A given data set $\mathbf{D} = \{(\boldsymbol{x}_n, y_n) | n = 1, \ldots, N\}$ with $y \in \{0, 1\}$
- The proportion of instances with $y = 1$ is $\alpha \neq 0.5$
- The decision rule is $0/1$ loss ℓ and the number of bins is M

Questions:

- Show that there is at least one classifier with

$$\text{Binary-ECE}(\mathbf{D}) = 0.0 \text{ and } \frac{1}{N}\sum_{n=1}^{N} \ell(y_n^*, y_n) = 0.0$$

- Show that there is at least one classifier with

$$\text{Binary-ECE}(\mathbf{D}) = 0.0 \text{ and } \frac{1}{N}\sum_{n=1}^{N} \ell(y_n^*, y_n) = \min(\alpha, 1 - \alpha)$$

Calibration Error: The Binary Case (Exercise 5)

Basic setup:

- A given data set $\mathbf{D} = \{(\boldsymbol{x}_n, y_n) | n = 1, \ldots, N\}$ with $y \in \{0, 1\}$
- The proportion of instances with $y = 1$ is $\alpha \neq 0.5$
- The decision rule is $0/1$ loss ℓ and the number of bins is M

Questions:

- Show that there is at least one classifier with

$$\text{Binary-ECE}(\mathbf{D}) = 0.0 \text{ and } \frac{1}{N}\sum_{n=1}^{N} \ell(y_n^*, y_n) = 0.0$$

- Show that there is at least one classifier with

$$\text{Binary-ECE}(\mathbf{D}) = 0.0 \text{ and } \frac{1}{N}\sum_{n=1}^{N} \ell(y_n^*, y_n) = \min(\alpha, 1 - \alpha)$$

- Can we find worse perfectly calibrated classifiers?

Classwise Calibration Error

Estimated classwise calibration error (classwise-ECE):

- For each class $y \in \mathcal{Y}$, consider y as class 1 and the others as 0
- Compute Binary-ECE for class $y \in \mathcal{Y} \longrightarrow \text{Binary-ECE}_y(\mathbf{D})$
- Compute classwise-ECE

$$\text{classwise-ECE}(\mathbf{D}) = \frac{1}{|\mathcal{Y}|} \sum_{y \in \mathcal{Y}} \text{Binary-ECE}_y(\mathbf{D})$$

Classwise Calibration Error (Exercise 6)

Basic setup:

- A given data set $\mathbf{D} = \{(\boldsymbol{x}_n, y_n) | n = 1, \ldots, N\}$ with $y \in \{0, 1, 2\}$
- The proportions of instances with $(y = 0, y = 1, y = 2)$ are $(\alpha_0, \alpha_1, \alpha_2)$
- The decision rule is $0/1$ loss ℓ and the number of bins is M

Questions:

- Can we find at least one classifier with

$$\text{classwise-ECE}(\mathbf{D}) = 0.0 \text{ and } \frac{1}{N}\sum_{n=1}^{N} \ell(y_n^*, y_n) = 0.0$$

Classwise Calibration Error (Exercise 6)

Basic setup:

- A given data set $\mathbf{D} = \{(\boldsymbol{x}_n, y_n) | n = 1, \ldots, N\}$ with $y \in \{0, 1, 2\}$
- The proportions of instances with $(y = 0, y = 1, y = 2)$ are $(\alpha_0, \alpha_1, \alpha_2)$
- The decision rule is $0/1$ loss ℓ and the number of bins is M

Questions:

- Can we find at least one classifier with

$$\text{classwise-ECE}(\mathbf{D}) = 0.0 \text{ and } \frac{1}{N}\sum_{n=1}^{N} \ell(y_n^*, y_n) = 0.0$$

- Show that there is at least one classifier with

$$\text{classwise-ECE}(\mathbf{D}) = 0.0 \text{ and } \frac{1}{N}\sum_{n=1}^{N} \ell(y_n^*, y_n) = 1 - \max(\alpha_0, \alpha_1, \alpha_2)$$

Classwise Calibration Error (Exercise 6)

Basic setup:

- A given data set $\mathbf{D} = \{(\boldsymbol{x}_n, y_n) | n = 1, \ldots, N\}$ with $y \in \{0, 1, 2\}$
- The proportions of instances with $(y = 0, y = 1, y = 2)$ are $(\alpha_0, \alpha_1, \alpha_2)$
- The decision rule is $0/1$ loss ℓ and the number of bins is M

Questions:

- Can we find at least one classifier with

$$\text{classwise-ECE}(\mathbf{D}) = 0.0 \text{ and } \frac{1}{N} \sum_{n=1}^{N} \ell(y_n^*, y_n) = 0.0$$

- Show that there is at least one classifier with

$$\text{classwise-ECE}(\mathbf{D}) = 0.0 \text{ and } \frac{1}{N} \sum_{n=1}^{N} \ell(y_n^*, y_n) = 1 - \max(\alpha_0, \alpha_1, \alpha_2)$$

- Can we find worse perfectly calibrated classifiers?

Confidence Calibration Error

Confidence-ECE is the weighted average difference between accuracy and average confidence across all bins:

$$\text{Confidence-ECE}(\mathbf{D}) = \sum_{m=1}^{M} \frac{|\mathbf{B}_m|}{|\mathbf{D}|} \left|\text{accuracy}(\mathbf{B}_m) - \text{confidence}(\mathbf{B}_m)\right| \quad (13)$$

- $\text{accuracy}(\mathbf{B}_m)$: Average accuracy in bin $\mathbf{B}_m$
- $\text{confidence}(\mathbf{B}_m)$: Average confidence in bin $\mathbf{B}_m$

Confidence Calibration Error (Exercise 7)

Basic setup:

- A given data set $\mathbf{D} = \{(\boldsymbol{x}_n, y_n) | n = 1, \ldots, N\}$ with $y \in \{0, 1, 2\}$
- The proportions of instances with $(y = 0, y = 1, y = 2)$ are $(\alpha_0, \alpha_1, \alpha_2)$
- The decision rule is $0/1$ loss ℓ and the number of bins is M

Questions:

- Show that there is at least one classifier with

$$\text{Confidence-ECE}(\mathbf{D}) = 0.0 \text{ and } \frac{1}{N}\sum_{n=1}^{N} \ell(y_n^*, y_n) = 0.0$$

Confidence Calibration Error (Exercise 7)

Basic setup:

- A given data set $\mathbf{D} = \{(\boldsymbol{x}_n, y_n) | n = 1, \ldots, N\}$ with $y \in \{0, 1, 2\}$
- The proportions of instances with $(y = 0, y = 1, y = 2)$ are $(\alpha_0, \alpha_1, \alpha_2)$
- The decision rule is $0/1$ loss ℓ and the number of bins is M

Questions:

- Show that there is at least one classifier with

$$\text{Confidence-ECE}(\mathbf{D}) = 0.0 \text{ and } \frac{1}{N}\sum_{n=1}^{N} \ell(y_n^*, y_n) = 0.0$$

- Show that there is at least one classifier with

$$\text{Confidence-ECE}(\mathbf{D}) = 0.0 \text{ and } \frac{1}{N}\sum_{n=1}^{N} \ell(y_n^*, y_n) = 1 - \max(\alpha_0, \alpha_1, \alpha_2)$$

Confidence Calibration Error (Exercise 7)

Basic setup:

- A given data set $\mathbf{D} = \{(\boldsymbol{x}_n, y_n) | n = 1, \ldots, N\}$ with $y \in \{0, 1, 2\}$
- The proportions of instances with $(y = 0, y = 1, y = 2)$ are $(\alpha_0, \alpha_1, \alpha_2)$
- The decision rule is $0/1$ loss ℓ and the number of bins is M

Questions:

- Show that there is at least one classifier with

$$\text{Confidence-ECE}(\mathbf{D}) = 0.0 \text{ and } \frac{1}{N}\sum_{n=1}^{N} \ell(y_n^*, y_n) = 0.0$$

- Show that there is at least one classifier with

$$\text{Confidence-ECE}(\mathbf{D}) = 0.0 \text{ and } \frac{1}{N}\sum_{n=1}^{N} \ell(y_n^*, y_n) = 1 - \max(\alpha_0, \alpha_1, \alpha_2)$$

- Can we find worse perfectly calibrated classifiers?

Notes on Classifier Errors (Homework)

Basic setup:

- Choose some calibration error
- Choose your favorite classifier
- Choose one data set you want to work with

Notes on Classifier Errors (Homework)

Basic setup:

- Choose some calibration error
- Choose your favorite classifier
- Choose one data set you want to work with

Compute & compare:

- Train your favorite classifier
- Do post-hoc calibration (see next slides)
- Compute the calibration error

Notes on Classifier Errors (Homework)

Basic setup:

- Choose some calibration error
- Choose your favorite classifier
- Choose one data set you want to work with

Compute & compare:

- Train your favorite classifier
- Do post-hoc calibration (see next slides)
- Compute the calibration error

- Estimate the prior distribution $P(\mathscr{Y})$ using MLE and/or DM
- Use $\boldsymbol{h}(\boldsymbol{x}) = P(\mathscr{Y}), \forall\ \boldsymbol{x}$
- Compute the calibration error

Outline

- Classifier Calibration
  - Introduction
  - Notions
  - Calibration Errors
  - Post-hoc Calibration
  - Other methods
- Conformal Prediction

How to learn well-calibrated and accurate classifiers[1]?

[1] I would be rich if I knew a very good answer :)

How to learn well-calibrated and accurate classifiers[1]?

Learn a well-calibrated classifier (a good strategy?)

- **Basic setup**: A hypothesis space (classifiers) and a calibration error
- **Problem**: Find a classifier which optimizes the calibration error

[1] I would be rich if I knew a very good answer :)

How to learn well-calibrated and accurate classifiers[1]?

Learn a well-calibrated classifier (a good strategy?)

- **Basic setup**: A hypothesis space (classifiers) and a calibration error
- **Problem**: Find a classifier which optimizes the calibration error

Learn a well-calibrated and accurate classifier (better?)

- **Basic setup**: A hypothesis space (classifiers) and an evaluation criterion
- **Basic setup (cont.)**: A hypothesis space (calibrators) and a calibration error
- **Problem**: Find an accurate classifier which optimizes the calibration error

[1] I would be rich if I knew a very good answer :)

Post-hoc calibration methods

- assume a reasonably accurate pre-trained model is given,
- calibrate the soft/probabilistic output of the pre-trained model.

Post-hoc calibration methods

- assume a reasonably accurate pre-trained model is given,
- calibrate the soft/probabilistic output of the pre-trained model.

Seek a (reasonably) **accurate pre-trained model**:

- a training (+ validation) data set,
- a hypothesis space (classifiers),
- an evaluation criterion,
- and a notion of an optimal classifier.

Post-hoc calibration methods

- assume a reasonably accurate pre-trained model is given,
- calibrate the soft/probabilistic output of the pre-trained model.

Seek a (reasonably) **accurate pre-trained model**:

- a training (+ validation) data set,
- a hypothesis space (classifiers),
- an evaluation criterion,
- and a notion of an optimal classifier.

Seek a(n reasonably) **good calibrator**:

- a training (+ validation) data set,
- a hypothesis space (calibrators),
- an evaluation criterion,
- and a notion of an optimal calibrator.

Empirical Binning

Basic Setup:

- Binary classification: $\mathcal{Y} := \{0, 1\}$
- Loss function: $\ell(y', y) = \mathbb{1}(y' \neq y)$
- Prediction: $y_\ell^{\boldsymbol{\theta}} = \mathbb{1}(\theta_y | \boldsymbol{x} > 0.5)$

Empirical Binning

Basic Setup:

- Binary classification: $\mathscr{Y} := \{0,1\}$
- Loss function: $\ell(y', y) = \mathbb{1}(y' \neq y)$
- Prediction: $y_\ell^{\boldsymbol{\theta}} = \mathbb{1}(\theta_y|\boldsymbol{x} > 0.5)$

Steps:

- Apply equal-width binning to $\theta_1|\boldsymbol{x}$ on **D**
- For each bin $\mathbf{B}_m \longrightarrow$ use $\overline{y}(\mathbf{B}_m)$

Empirical Binning (Exercise 7)

Basic Setup:

- Binary classification: $\mathscr{Y} := \{0, 1\}$
- Loss function: $\ell(y', y) = \mathbb{1}(y' \neq y)$
- Prediction: $y_\ell^{\boldsymbol{\theta}} = \mathbb{1}(\theta_y | \boldsymbol{x} > 0.5)$

Empirical Binning (Exercise 7)

Basic Setup:

- Binary classification: $\mathcal{Y} := \{0, 1\}$
- Loss function: $\ell(y', y) = \mathbb{1}(y' \neq y)$
- Prediction: $y_\ell^{\boldsymbol{\theta}} = \mathbb{1}(\theta_y | \boldsymbol{x} > 0.5)$

Steps:

- Apply equal-width binning to $\theta_1 | \boldsymbol{x}$ on $\mathbf{D}$
- For each bin $\mathbf{B}_m \longrightarrow$ use $\overline{y}(\mathbf{B}_m)$

Empirical Binning (Exercise 7)

Basic Setup:

- Binary classification: $\mathcal{Y} := \{0, 1\}$
- Loss function: $\ell(y', y) = \mathbb{1}(y' \neq y)$
- Prediction: $y_\ell^{\boldsymbol{\theta}} = \mathbb{1}(\theta_y | \boldsymbol{x} > 0.5)$

Steps:

- Apply equal-width binning to $\theta_1 | \boldsymbol{x}$ on $\mathbf{D}$
- For each bin $\mathbf{B}_m \longrightarrow$ use $\overline{y}(\mathbf{B}_m)$

Question: Empirical Binning optimizes binary-ECE($\mathbf{D}$)?

Platt Scaling

Basic Setup:

- Binary classification: $\mathscr{Y} := \{0, 1\}$
- Loss function: $\ell(y', y) = \mathbb{1}(y' \neq y)$
- Prediction: $y_\ell^{\boldsymbol{\theta}} = \mathbb{1}(\theta_y | \boldsymbol{x} > 0.5)$

Platt Scaling

Basic Setup:

- Binary classification: $\mathscr{Y} := \{0, 1\}$
- Loss function: $\ell(y', y) = \mathbb{1}(y' \neq y)$
- Prediction: $y_\ell^{\boldsymbol{\theta}} = \mathbb{1}(\theta_y|\boldsymbol{x} > 0.5)$

Learn a **logistic transformation** of the classifier

$$P(y = 1|\boldsymbol{x}) \approx \frac{1}{1 + \exp\left(A(\boldsymbol{\theta}|\boldsymbol{x}) + B\right)} \tag{14}$$

- Estimate A and B: fit the regressor **via maximum likelihood**
- **Multi-class classification**: Platt Scaling $\longleftarrow$ Platt Scaling + z
- z $\in$ $\{$One-vs-All, One-vs-One$\}$

Isotonic Regression (The Same Basic Setup)

Fits a **non-parametric isotonic regressor**,

- which outputs a step-wise non-decreasing function $f|\boldsymbol{x}$

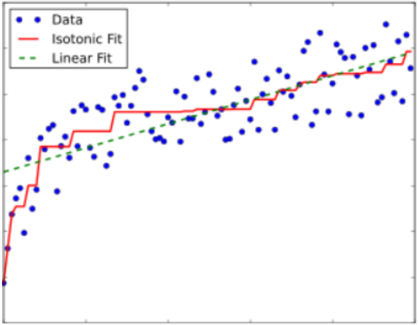
$$\text{minimize} \quad \sum_{(y,\boldsymbol{x})\in\boldsymbol{D}} (y - f|\boldsymbol{x})^2 \quad \text{s.t.} \quad f|\boldsymbol{x} \geq f|\boldsymbol{x} \text{ if } \boldsymbol{\theta}|\boldsymbol{x} \geq \boldsymbol{\theta}|\boldsymbol{x}' \qquad (15)$$

Isotonic Regression (The Same Basic Setup)

Fits a **non-parametric isotonic regressor**,

- which outputs a step-wise non-decreasing function $f|\boldsymbol{x}$

$$\text{minimize} \quad \sum_{(y,\boldsymbol{x})\in \boldsymbol{D}} (y - f|\boldsymbol{x})^2 \quad \text{s.t.} \quad f|\boldsymbol{x} \geq f|\boldsymbol{x} \text{ if } \boldsymbol{\theta}|\boldsymbol{x} \geq \boldsymbol{\theta}|\boldsymbol{x}' \tag{15}$$



An example of isotonic regression (solid red line)

Beta Calibration (The Same Basic Setup)

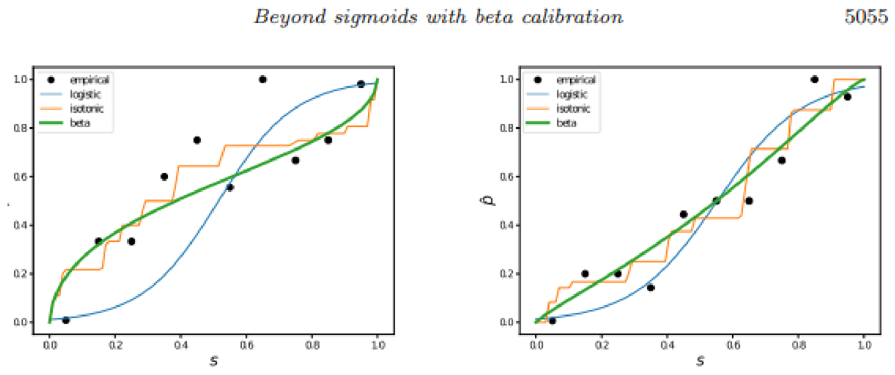
Learn a **beta calibration map**

$$P(y = 1|\boldsymbol{x}) \approx \frac{1}{1 + 1/\left(\exp(c)\frac{(\theta|\boldsymbol{x})^a}{(1-\theta|\boldsymbol{x})^b}\right)} \tag{16}$$

There are some requirements [5]:

- each calibration is monotonically non-decreasing $\longrightarrow a, b \geq 0$
- c is some real number

Practical Examples [6]



(a) Adaboost – landsat-satellite
(b) Naive Bayes – vowel

Notes on Post-hoc Calibration (Homework)

Basic setup:

- Choose some calibration error
- Choose your favorite classifier
- Choose one data set you want to work with

Notes on Post-hoc Calibration (Homework)

Basic setup:

- Choose some calibration error
- Choose your favorite classifier
- Choose one data set you want to work with

Compute & compare:

- Train your favorite classifier
- Do post-hoc calibration (see previous slides)
- Compute the average 0/1 loss + calibration error

Notes on Post-hoc Calibration (Homework)

Basic setup:

- Choose some calibration error
- Choose your favorite classifier
- Choose one data set you want to work with

Compute & compare:

- Train your favorite classifier
- Do post-hoc calibration (see previous slides)
- Compute the average 0/1 loss + calibration error

- Estimate the prior distribution $P(\mathscr{Y})$ using MLE and/or DM
- Use $\boldsymbol{h}(\boldsymbol{x}) = P(\mathscr{Y}), \forall\ \boldsymbol{x}$
- Compute the average 0/1 loss + calibration error

Potential Impact [8]

Basic Setup:

- run 10×10-fold stratified cross-validation ⟶ average the results
- UC = The uncalibrated model (trained using the entire training set)
- PS = UC + Platt scaling (training set = 2/3 train + 1/3 calibration)
- VA = UC + Venn-Abers (training set = 2/3 train + 1/3 calibration)
- Compare Accuracy (1 - 0/1 loss) and Binary-ECE
- 25 data sets for binary classification

Potential Impact [8]

Basic Setup:

- run 10×10-fold stratified cross-validation ⟶ average the results
- UC = The uncalibrated model (trained using the entire training set)
- PS = UC + Platt scaling (training set = 2/3 train + 1/3 calibration)
- VA = UC + Venn-Abers (training set = 2/3 train + 1/3 calibration)
- Compare Accuracy (1 - 0/1 loss) and Binary-ECE
- 25 data sets for binary classification

Classifiers:

- UC = RF: Random forest
- UC = xGBoost: Extreme Gradient Boosting

Data set characteristics [8]

Data set	#instances	#features	Class distr.	Data set	#instances	#features	Class distr.
colic	375	59	134/223	kc2	369	21	270/99
creditA	690	42	383/307	kc3	325	39	283/42
diabetes	768	8	500/268	liver	341	6	142/199
german	955	27	283/672	pc1req	104	8	55/49
haberman	283	3	204/79	pc4	1343	37	1166/177
heartC	302	22	164/138	sonar	208	60	97/111
heartH	293	20	187/106	spect	218	22	24/194
heartS	270	13	150/120	spectf	267	44	55/212
hepatitis	155	19	123/32	transfusion	502	4	371/131
iono	350	33	225/125	ttt	958	27	332/626
je4042	270	8	136/134	vote	517	16	429/144
je4243	363	8	161/202	wbc	463	9	225/263
kc1	1192	21	877/315				

Accuracy [8]

	RF			xGB		
Data sets	UC	PS	VA	UC	PS	VA
colic	.838	.819	.818	.840	.832	.824
creditA	.850	.849	.837	.845	.854	.832
diabetes	.763	.759	.753	.736	.736	.715
german	.665	.703	.703	.623	.704	.703
haberman	.661	.721	.712	.587	.721	.721
heartC	.833	.822	.814	.788	.778	.772
heartH	.793	.808	.784	.720	.771	.768
heartS	.824	.816	.808	.807	.804	.793
hepati	.837	.829	.814	.800	.813	.768
iono	.936	.929	.918	.909	.911	.914
je4042	.758	.729	.727	.704	.744	.756
je4243	.626	.630	.618	.606	.642	.628
kc1	.710	.717	.716	.691	.716	.721
kc2	.781	.771	.769	.762	.753	.767
kc3	.849	.858	.848	.868	.868	.862
liver	.718	.694	.683	.701	.686	.683
pc1req	.696	.622	.673	.615	.567	.683
pc4	.896	.889	.888	.897	.887	.888
sonar	.714	.677	.684	.736	.683	.668
spect	.883	.890	.873	.858	.885	.867
spectf	.803	.791	.793	.809	.779	.783
transfusion	.655	.698	.694	.657	.699	.677
ttt	.918	.893	.891	.874	.889	.883
vote	.819	.801	.814	.801	.776	.779
wbc	.949	.941	.946	.929	.931	.933
Mean	.791	.786	.783	.766	.777	.775

Binary-ECE [8]

	RF			xGB		
Data sets	UC	PS	VA	UC	PS	VA
colic	.062	.031	.024	.093	.057	.036
creditA	.031	.025	.045	.098	.064	.061
diabetes	.018	.049	.036	.162	.044	.046
german	.091	.019	.007	.198	.009	.009
haberman	.144	.041	.043	.307	.068	.077
heartC	.042	.025	.031	.133	.047	.038
heartH	.051	.036	.059	.183	.056	.074
heartS	.042	.073	.070	.118	.080	.076
hepati	.039	.073	.075	.121	.077	.119
iono	.049	.041	.061	.067	.041	.071
je4042	.056	.044	.037	.188	.074	.076
je4243	.091	.049	.047	.271	.052	.070
kc1	.090	.049	.059	.177	.072	.071
kc2	.073	.065	.020	.172	.042	.067
kc3	.054	.037	.052	.085	.038	.054
liver	.042	.036	.020	.174	.030	.046
pc1req	.079	.132	.116	.247	.096	.133
pc4	.030	.024	.010	.058	.037	.023
sonar	.066	.120	.124	.146	.164	.146
spect	.063	.054	.052	.097	.051	.061
spectf	.028	.052	.042	.148	.054	.056
transfusion	.204	.092	.118	.227	.074	.095
ttt	.157	.044	.037	.073	.074	.067
vote	.088	.111	.096	.156	.146	.110
wbc	.027	.029	.047	.048	.023	.048
Mean	.069	.054	.053	.150	**.063**	**.069**

PyCalib

Python library for classifier calibration

User installation

The PyCalib package can be installed from Pypi with the command

```
pip install pycalib
```

Documentation

The documentation can be found at https://classifier-calibration.github.io/PyCalib/

sklearn.calibration.CalibratedClassifierCV

class `sklearn.calibration.CalibratedClassifierCV`(*estimator=None, *, method='sigmoid', cv=None, n_jobs=None, ensemble=True, base_estimator='deprecated'*) [source]

Outline

- Classifier Calibration
  - Introduction
  - Notions
  - Calibration Errors
  - Post-hoc Calibration
  - Other methods
- Conformal Prediction

(Hopefully) Calibration During Training [10]

- Calibration error $\longrightarrow$ a regularization term
- Mixup: regularization $\approx$ augmentation + label smoothing effect

(Hopefully) Calibration During Training [10]

- Calibration error $\longrightarrow$ a regularization term
- Mixup: regularization $\approx$ augmentation + label smoothing effect
- Few others (see [10][section 5.6] and elsewhere)

A Regularization Approach [7]

Optimization problem should be described after declaring

- a training (+ validation) data set,
- a hypothesis space,
- an evaluation criterion,
- and a notion of an optimal classifier.

A Regularization Approach [7]

Optimization problem should be described after declaring

- a training (+ validation) data set,
- a hypothesis space,
- an evaluation criterion,
- and a notion of an optimal classifier.

- (criterion) = (negative log-likelihood) + λ * (calibration error)

A Regularization Approach [7]

Optimization problem should be described after declaring

- a training (+ validation) data set,
- a hypothesis space,
- an evaluation criterion,
- and a notion of an optimal classifier.

- (criterion) = (negative log-likelihood) + λ * (calibration error)
- (calibration error) should be trainable (differentiable, ...)

A Regularization Approach (cont.) [7]

Remark: ECE = Confidence-ECE

E#	Dataset	Model	ECE		Accuracy	
			Baseline	MMCE	Baseline	MMCE
1	MNIST	LeNet 5	0.5%	**0.2%**	99.24%	99.26%
2	CIFAR 10	Resnet 50	4.3%	**1.2%**	93.1%	93.4%
3	CIFAR 10	Resnet 110	4.6%	**1.1%**	93.7%	94.0%
4	CIFAR 10	Wide Resnet 28-10	4.5%	**1.6%**	94.1%	94.2%
5	CIFAR 100	Resnet 32	19.6%	**6.9%**	67.0%	67.7%
6	CIFAR 100	Wide Resnet 28-10	15.0%	**8.9%**	74.0%	76.6%
7	Birds CUB 200	Inception-v3	2.6%	**2.3%**	78.2%	77.9%
8	20 Newsgroups	Global Pooling CNN	16.5%	**6.5%**	74.2%	73.9%
9	IMDB Reviews	HAN	4.9%	**0.4%**	86.8%	86.3%
10	SST Binary	Tree LSTM	7.4%	**5.9%**	88.6%	88.7%
11	HAR time series	LSTM	7.6%	**5.9%**	89.4%	90.3%

Outline

- Classifier Calibration
- Conformal Prediction
  - Notions
  - Coverage Metrics
  - Conformal Procedures

Outline

Coverage as Another Notion of Calibration [1]

Figure 1: Prediction set examples on Imagenet. *We show three progressively more difficult examples of the class* **`fox squirrel`** *and the prediction sets (i.e., $\mathcal{C}(X_{\text{test}})$) generated by conformal prediction.*

Coverage as Another Notion of Calibration [1]

Figure 1:* *Prediction set examples on Imagenet. *We show three progressively more difficult examples of the class* **`fox squirrel`** *and the prediction sets (i.e., $\mathcal{C}(X_{\text{test}})$) generated by conformal prediction.*

General setting:

- We wish to produce a (possibly empty) **set-valued prediction** for each query instance.
- We wish to guarantee that **the probability of covering the true class** is bounded by the chosen significance level $\sigma \in [0, 1]$.

Marginal and Conditional Coverage

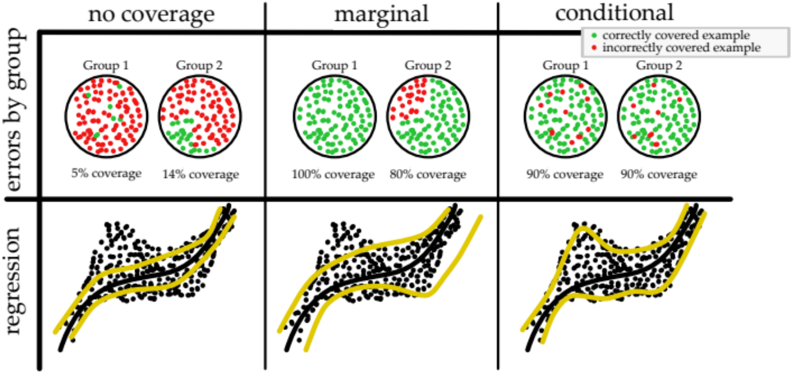


Figure 10: Prediction sets with various notions of coverage: *no coverage, marginal coverage, or conditional coverage (at a level of 90%). In the marginal case, all the errors happen in the same groups and regions in X-space. Conditional coverage disallows this behavior, and errors are evenly distributed.*

Population Level: Marginal Coverage

- Data set = $\mathbf{D}_{\text{train}}$ + $\mathbf{D}_{\text{calibration}}$ + $\mathbf{D}_{\text{test}}$
- They are expected to come from the same distribution
- Learn a predictor (classifier/regressor) $\mathbf{h}$ using $\mathbf{D}_{\text{train}}$

Population Level: Marginal Coverage

- Data set = $\mathbf{D}_{\text{train}}$ + $\mathbf{D}_{\text{calibration}}$ + $\mathbf{D}_{\text{test}}$
- They are expected to come from the same distribution
- Learn a predictor (classifier/regressor) **h** using $\mathbf{D}_{\text{train}}$
- Use $\mathbf{D}_{\text{calibration}}$ and **h** to construct for each $\boldsymbol{x}_{\text{test}} \in \mathbf{D}_{\text{test}}$ a $Y_{\text{test}} \subset \mathscr{Y}$ s.t.

$$1 - \alpha \leq P(y_{\text{test}} \in Y_{\text{test}})$$

where $\alpha \in [0,1]$ is a user-chosen error rate.

Population Level: Marginal Coverage

- Data set = $\mathbf{D}_{\text{train}}$ + $\mathbf{D}_{\text{calibration}}$ + $\mathbf{D}_{\text{test}}$
- They are expected to come from the same distribution
- Learn a predictor (classifier/regressor) **h** using $\mathbf{D}_{\text{train}}$
- Use $\mathbf{D}_{\text{calibration}}$ and **h** to construct for each $\boldsymbol{x}_{\text{test}} \in \mathbf{D}_{\text{test}}$ a $Y_{\text{test}} \subset \mathscr{Y}$ s.t.

$$1 - \alpha \leq P(y_{\text{test}} \in Y_{\text{test}})$$

where $\alpha \in [0, 1]$ is a user-chosen error rate.

Marginal Coverage (Exercise 8)

- Prove that if we always predict $Y_{\text{test}} := \mathcal{Y}$ we can always produce perfect conformal predictions w.r.t. the notion of marginal coverage with any chosen significance level $\sigma \in [0, 1]$.

Marginal Coverage (Exercise 8)

- Prove that if we always predict $Y_{\text{test}} := \mathscr{Y}$ we can always produce perfect conformal predictions w.r.t. the notion of marginal coverage with any chosen significance level $\sigma \in [0,1]$.
- Prove that if we know the prior distribution $P(\mathscr{Y})$, we can always produce perfect conformal predictions w.r.t. the notion of marginal coverage with any chosen significance level $\sigma \in [0,1]$.

Marginal Coverage (Homework)

Basic setup:

- Choose your favorite classifier + data set

Marginal Coverage (Homework)

Basic setup:

- Choose your favorite classifier + data set

Compute & compare:

- Train your favorite classifier
- Apply the chosen conformal procedure (see next slides)
- Compute the coverage metrics with different α

Marginal Coverage (Homework)

Basic setup:

- Choose your favorite classifier + data set

Compute & compare:

- Train your favorite classifier
- Apply the chosen conformal procedure (see next slides)
- Compute the coverage metrics with different α
- Estimate the prior distribution $P(\mathscr{Y})$ using MLE and/or DM
- For each given α, always returns the set of classes whose prior probabilities are at least α
- Compute the coverage metrics with different α

Marginal Coverage (Homework)

Basic setup:

- Choose your favorite classifier + data set

Compute & compare:

- Train your favorite classifier
- Apply the chosen conformal procedure (see next slides)
- Compute the coverage metrics with different α
- Estimate the prior distribution $P(\mathscr{Y})$ using MLE and/or DM
- For each given α, always returns the set of classes whose prior probabilities are at least α
- Compute the coverage metrics with different α
- Always return $Y_{\text{test}} := \mathscr{Y}$
- Compute the coverage metrics with different α

Group Level: Group-Balanced Conformal Prediction

- Prior information $\longrightarrow$ partition $\mathbf{D}$ into G groups $\mathbf{D}^g$
- We then ask for group-balanced coverage

$$1 - \alpha \leq P\left(y_{\text{test}} \in Y_{\text{test}} | \boldsymbol{x}_{\text{test}} \in \mathbf{D}^g\right), g = 1, \ldots, G. \tag{17}$$

Group Level: Group-Balanced Conformal Prediction

- Prior information $\longrightarrow$ partition $\mathbf{D}$ into G groups $\mathbf{D}^g$
- We then ask for group-balanced coverage

$$1-\alpha \leq P\left(y_{\text{test}} \in Y_{\text{test}} | \boldsymbol{x}_{\text{test}} \in \mathbf{D}^g\right), g = 1, \ldots, G. \tag{17}$$

Class-Conditional Conformal Prediction:

- Partition $\mathbf{D}$ into $|\mathcal{Y}|$ groups, one per class $y \in \mathcal{Y}$

$$1-\alpha \leq P\left(y_{\text{test}} \in Y_{\text{test}} | y_{\text{test}} = y\right), y \in \mathcal{Y}. \tag{18}$$

Group Level: Group-Balanced Conformal Prediction

- Prior information $\longrightarrow$ partition $\mathbf{D}$ into G groups $\mathbf{D}^g$
- We then ask for group-balanced coverage

$$1-\alpha \leq P\left(y_{\text{test}} \in Y_{\text{test}} | \boldsymbol{x}_{\text{test}} \in \mathbf{D}^g\right), g = 1, \ldots, G. \tag{17}$$

Class-Conditional Conformal Prediction:

- Partition $\mathbf{D}$ into $|\mathcal{Y}|$ groups, one per class $y \in \mathcal{Y}$

$$1-\alpha \leq P\left(y_{\text{test}} \in Y_{\text{test}} | y_{\text{test}} = y\right), y \in \mathcal{Y}. \tag{18}$$

Other examples:

- Group patients into demographic groups
- Group set-valued predictions into groups of equal cardinality

Group Level: Group-Balanced Conformal Prediction

- Prior information $\longrightarrow$ partition **D** into G groups $\mathbf{D}^g$
- We then ask for group-balanced coverage

$$1-\alpha \leq P\left(y_{\text{test}} \in Y_{\text{test}} | \boldsymbol{x}_{\text{test}} \in \mathbf{D}^g\right), g = 1, \ldots, G. \tag{17}$$

Class-Conditional Conformal Prediction:

- Partition **D** into $|\mathcal{Y}|$ groups, one per class $y \in \mathcal{Y}$

$$1-\alpha \leq P\left(y_{\text{test}} \in Y_{\text{test}} | y_{\text{test}} = y\right), y \in \mathcal{Y}. \tag{18}$$

Other examples:

- Group patients into demographic groups
- Group set-valued predictions into groups of equal cardinality

Comment (AOS4): Shouldn't we always predict $Y_{\text{test}} := \mathcal{Y}$?

Individual Level: Conditional Coverage

Problem: construct for each $\boldsymbol{x}_{\text{test}} \in \mathbf{D}_{\text{test}}$ a $Y_{\text{test}} \subset \mathscr{Y}$ s.t.

$$1 - \alpha \leq P(y_{\text{test}} \in Y_{\text{test}} | \boldsymbol{x}_{\text{test}})$$

where $\alpha \in [0, 1]$ is a user-chosen error rate.

Individual Level: Conditional Coverage

Problem: construct for each $\boldsymbol{x}_{\text{test}} \in \mathbf{D}_{\text{test}}$ a $Y_{\text{test}} \subset \mathscr{Y}$ s.t.

$$1 - \alpha \leq P(y_{\text{test}} \in Y_{\text{test}} | \boldsymbol{x}_{\text{test}})$$

where $\alpha \in [0, 1]$ is a user-chosen error rate.

Comments [1]:

- A stronger property than the marginal/group coverage
- In the most general case, conditional coverage is impossible to achieve [11]
- $\longrightarrow$ check how close our procedure comes to approximating it

Individual Level: Conditional Coverage

Problem: construct for each $\boldsymbol{x}_{\text{test}} \in \mathbf{D}_{\text{test}}$ a $Y_{\text{test}} \subset \mathscr{Y}$ s.t.

$$1 - \alpha \leq P(y_{\text{test}} \in Y_{\text{test}} | \boldsymbol{x}_{\text{test}})$$

where $\alpha \in [0, 1]$ is a user-chosen error rate.

Comments [1]:

- A stronger property than the marginal/group coverage
- In the most general case, conditional coverage is impossible to achieve [11]
- $\longrightarrow$ check how close our procedure comes to approximating it

Comment (AOS4): Shouldn't we always predict $Y_{\text{test}} := \mathscr{Y}$?

Conformal Risk Control

- We have constructed prediction sets that bound the miscoverage

$$P(y_{\text{test}} \in Y_{\text{test}}) \geq 1 - \alpha \equiv 1 - P(y_{\text{test}} \in Y_{\text{test}}) \leq \alpha \tag{19}$$

$$\equiv P(y_{\text{test}} \notin Y_{\text{test}}) \leq \alpha \tag{20}$$

- We haven't taken into account the cardinality[2] $|Y_{\text{test}}|$

[2] Still remember $Y_{\text{test}} := \mathcal{Y}$?

Conformal Risk Control

- We have constructed prediction sets that bound the miscoverage

$$P(y_{\text{test}} \in Y_{\text{test}}) \geq 1-\alpha \equiv 1 - P(y_{\text{test}} \in Y_{\text{test}}) \leq \alpha \tag{19}$$

$$\equiv P(y_{\text{test}} \not\in Y_{\text{test}}) \leq \alpha \tag{20}$$

- We haven't taken into account the cardinality[2] $|Y_{\text{test}}|$
- We can consider both the miscoverage and cardinality using

$$\ell(y_{\text{test}}, Y_{\text{test}}) \tag{21}$$

$\longrightarrow$ any bounded loss function that shrinks as $|Y_{\text{test}}|$ grows.

- We may construct prediction sets that bound the expected loss

$$\mathrm{E}\left[\ell(y_{\text{test}}, Y_{\text{test}})|\boldsymbol{x}\right] = \sum_{y_{\text{test}} \in \mathcal{Y}} \ell(y_{\text{test}}, Y_{\text{test}}) * P(y_{\text{test}}|\boldsymbol{x}) \leq \alpha \tag{22}$$

[2] Still remember $Y_{\text{test}} := \mathcal{Y}$?

Outline

- Classifier Calibration
- Conformal Prediction
  - Notions
  - Coverage Metrics
  - Conformal Procedures

Population Level: Empirical Coverage[3]

- Empirical coverage (EC) metric is defined as

$$\text{EC-metric}(\mathbf{D}_{\text{test}}) = \frac{1}{|\mathbf{D}_{\text{test}}|} \sum_{\boldsymbol{x}_{\text{test}} \in \mathbf{D}_{\text{test}}} \mathbb{1}(y_{\text{test}} \in Y_{;\text{test}}) \tag{23}$$

[3] Should we always predict $Y_{\text{test}} := \mathscr{Y}$?

Population Level: Empirical Coverage[3]

- Empirical coverage (EC) metric is defined as

$$\text{EC-metric}(\mathbf{D}_{\text{test}}) = \frac{1}{|\mathbf{D}_{\text{test}}|} \sum_{\boldsymbol{x}_{\text{test}} \in \mathbf{D}_{\text{test}}} \mathbb{1}(y_{\text{test}} \in Y_{;\text{test}}) \tag{23}$$

- If we consider

$$P(y_{\text{test}} \in Y_{\text{test}}) \longleftarrow \frac{1}{|\mathbf{D}_{\text{test}}|} \sum_{\boldsymbol{x}_{\text{test}} \in \mathbf{D}_{\text{test}}} \mathbb{1}(y_{\text{test}} \in Y_{\text{test}}) \tag{24}$$

- then we might claim the relation

$$\text{EC-metric}(\mathbf{D}_{\text{test}}) \leq P(y_{\text{test}} \in Y_{\text{test}}) \tag{25}$$

[3] Should we always predict $Y_{\text{test}} := \mathscr{Y}$?

Group Level: Feature-Stratified Coverage Metric[4]

- Feature information $\longrightarrow$ partition $\mathbf{D}$ into G groups $\mathbf{D}^g$
- Feature-stratified coverage (FSC) metric is defined as

$$\text{FSC-metric}(\mathbf{D}_{\text{test}}) = \min_{g \in \{1,\dots,G\}} \frac{1}{\left|\mathbf{D}^g_{\text{test}}\right|} \sum_{\boldsymbol{x}_{\text{test}} \in \mathbf{D}^g_{\text{test}}} \mathbb{1}(y_{\text{test}} \in Y_{\text{test}}) \tag{26}$$

[4] Should we always predict $Y_{\text{test}} := \mathscr{Y}$?

Group Level: Feature-Stratified Coverage Metric[4]

- Feature information $\longrightarrow$ partition $\mathbf{D}$ into G groups $\mathbf{D}^g$
- Feature-stratified coverage (FSC) metric is defined as

$$\text{FSC-metric}(\mathbf{D}_{\text{test}}) = \min_{g\in\{1,\dots,G\}} \frac{1}{\left|\mathbf{D}^g_{\text{test}}\right|} \sum_{\boldsymbol{x}_{\text{test}}\in\mathbf{D}^g_{\text{test}}} \mathbb{1}(y_{\text{test}} \in Y_{\text{test}}) \tag{26}$$

- If we consider (the instances within each $\mathbf{D}^g_{\text{test}}$ equally and)

$$P(y_{\text{test}} \in Y_{\text{test}}|\boldsymbol{x}_{\text{test}}) \longleftarrow \frac{1}{\left|\mathbf{D}^g_{\text{test}}\right|} \sum_{\boldsymbol{x}_{\text{test}}\in\mathbf{D}^g_{\text{test}}} \mathbb{1}(y_{\text{test}} \in Y_{\text{test}}) \tag{27}$$

- then we might claim the relation

$$\text{FSC-metric}(\mathbf{D}_{\text{test}}) \leq P(y_{\text{test}} \in Y_{\text{test}}|\boldsymbol{x}_{\text{test}}), \forall \boldsymbol{x}_{\text{test}} \in \mathbf{D}_{\text{test}} \tag{28}$$

[4] Should we always predict $Y_{\text{test}} := \mathscr{Y}$?

Group Level: Size-Stratified Coverage Metric[5]

- Cardinality $|Y| \longrightarrow$ partition $\mathbf{D}$ into G groups $\mathbf{D}^g$
- Size-Stratified Coverage (SSC) metric is defined as

$$\text{SSC-metric}(\mathbf{D}_{\text{test}}) = \min_{g \in \{1,\ldots,G\}} \frac{1}{\left|\mathbf{D}^g_{\text{test}}\right|} \sum_{\boldsymbol{x}_{\text{test}} \in \mathbf{D}^g_{\text{test}}} \mathbb{1}(y_{\text{test}} \in Y_{\text{test}}) \qquad (29)$$

[5] Should we always predict $Y_{\text{test}} := \mathcal{Y}$?

Group Level: Size-Stratified Coverage Metric[5]

- Cardinality $|Y| \longrightarrow$ partition $\mathbf{D}$ into G groups $\mathbf{D}^g$
- Size-Stratified Coverage (SSC) metric is defined as

$$\text{SSC-metric}(\mathbf{D}_{\text{test}}) = \min_{g \in \{1,\dots,G\}} \frac{1}{\left|\mathbf{D}^g_{\text{test}}\right|} \sum_{\boldsymbol{x}_{\text{test}} \in \mathbf{D}^g_{\text{test}}} \mathbb{1}(y_{\text{test}} \in Y_{\text{test}}) \tag{29}$$

- If we consider the instances within each $\mathbf{D}^g_{\text{test}}$ equally and

$$P(y_{\text{test}} \in Y_{\text{test}} | \boldsymbol{x}_{\text{test}}) \approx \frac{1}{\left|\mathbf{D}^g_{\text{test}}\right|} \sum_{\boldsymbol{x}_{\text{test}} \in \mathbf{D}^g_{\text{test}}} \mathbb{1}(y_{\text{test}} \in Y_{\text{test}}) \tag{30}$$

- then we might claim the relation

$$\text{SSC-metric}(\mathbf{D}_{\text{test}}) \leq P(y_{\text{test}} \in Y_{\text{test}} | \boldsymbol{x}_{\text{test}}), \forall \boldsymbol{x}_{\text{test}} \in \mathbf{D}_{\text{test}} \tag{31}$$

[5] Should we always predict $Y_{\text{test}} := \mathscr{Y}$?

Cover. Metrics Have often Been Coupled with Prediction Size

This can (hopefully) be done by using, for example,

- a loss considering both the miscoverage and cardinality,
- a suitable conformal procedure (see next slides),
- and so on.

Outline

- Classifier Calibration
- Conformal Prediction
  - Notions
  - Coverage Metrics
  - Conformal Procedures

Split Conformal Prediction: Steps

- Learn a classifier **h** using $\mathbf{D}_{\text{train}}$
- Define the score function $s(\boldsymbol{x}, y) \in \mathbb{R}$, which should depend on **h**.
- Larger $s \longrightarrow$ worse agreement between $\boldsymbol{x}$ and y.

Split Conformal Prediction: Steps

- Learn a classifier **h** using $\mathbf{D}_{\text{train}}$
- Define the score function $s(\boldsymbol{x}, y) \in \mathbb{R}$, which should depend on **h**.
- Larger $s \longrightarrow$ worse agreement between $\boldsymbol{x}$ and y.
- Let $M = |\mathbf{D}_{\text{validation}}|$, compute

$$s_1 = s(\boldsymbol{x}_1, y_1), \ldots, s_M = s(\boldsymbol{x}_M, y_M), (\boldsymbol{x}_m, y_m) \in \mathbf{D}_{\text{validation}}$$

- Sort the calibration scores $s_1, \ldots, s_M$ in the decreasing order
- Find $\frac{(n+1)(1-\alpha)}{n}$ quantile q_α of the calibration scores

Split Conformal Prediction: Steps

- Learn a classifier **h** using $\mathbf{D}_{\text{train}}$
- Define the score function $s(\boldsymbol{x}, y) \in \mathbb{R}$, which should depend on **h**.
- Larger $s \longrightarrow$ worse agreement between $\boldsymbol{x}$ and y.
- Let $M = |\mathbf{D}_{\text{validation}}|$, compute

$$s_1 = s(\boldsymbol{x}_1, y_1), \ldots, s_M = s(\boldsymbol{x}_M, y_M), (\boldsymbol{x}_m, y_m) \in \mathbf{D}_{\text{validation}}$$

- Sort the calibration scores $s_1, \ldots, s_M$ in the decreasing order
- Find $\frac{(n+1)(1-\alpha)}{n}$ quantile q_α of the calibration scores
- For any $\boldsymbol{x}_{\text{test}}$, predict

$$Y_{\text{test}} = \{y \in \mathcal{Y} \text{ s.t. } s(\boldsymbol{x}_{\text{test}}, y) \leq q_\alpha\} \tag{32}$$

Split Conformal Prediction: A Marginal Coverage Seeker

Conformal coverage guarantee [1, 9]:

- Suppose $(\boldsymbol{x}_m, y_m) \in \mathbf{D}_{\text{validation}}$ and $(\boldsymbol{x}_{\text{test}}, y_{\text{test}})$ are independent and identically distributed (i.i.d.). Then the following holds:

$$1 - \alpha \leq P(y_{\text{test}} \in Y_{\text{test}}) \tag{33}$$

Split Conformal Prediction: A Marginal Coverage Seeker

Conformal coverage guarantee [1, 9]:

- Suppose $(\boldsymbol{x}_m, y_m) \in \mathbf{D}_{\text{validation}}$ and $(\boldsymbol{x}_{\text{test}}, y_{\text{test}})$ are independent and identically distributed (i.i.d.). Then the following holds:

$$1 - \alpha \leq P(y_{\text{test}} \in Y_{\text{test}}) \tag{33}$$

Assumptions:

- Larger $s \longrightarrow$ worse agreement between $\boldsymbol{x}$ and y.
- $(\boldsymbol{x}_m, y_m) \in \mathbf{D}_{\text{validation}}$ and $(\boldsymbol{x}_{\text{test}}, y_{\text{test}})$ are independent i.i.d.

Assumptions of I.I.D.

Independence:

- The occurrence or value of one data point does not provide any information about the occurrence or value of another data point.
- The data points are not influenced by each other and that there is no hidden structure or correlation among them.

Assumptions of I.I.D.

Independence:

- The occurrence or value of one data point does not provide any information about the occurrence or value of another data point.
- The data points are not influenced by each other and that there is no hidden structure or correlation among them.

Identical distribution:

- The data points are drawn from the same underlying distribution.

Split Conformal Prediction: A Smallest Average Size Seeker

Average size [9][Remark 4] is defined as

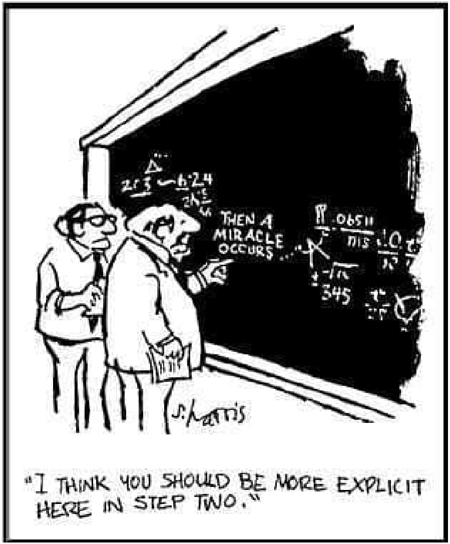
$$E(Y) = \sum_{y \in \mathcal{Y}} P(y \in Y) \tag{34}$$

Other procedures [1]

Conformal prediction can also be adapted to handle

- unsupervised outlier detection
- covariate/distribution shift
- multilabel classification

Remember to Check the Underlying Assumptions



github.com/aangelopoulos/conformal-prediction

README.md

Conformal Prediction

rigorous uncertainty quantification for any machine learning task

paper arXiv | website Berkeley | conda env | license MIT | Views 33k | hits 8576

This repository is the easiest way to start using conformal prediction (a.k.a. conformal inference) on real data.

References I

[1] A. N. Angelopoulos and S. Bates.
A gentle introduction to conformal prediction and distribution-free uncertainty quantification.
arXiv preprint arXiv:2107.07511, 2021.

[2] J. Gawlikowski, C. R. N. Tassi, M. Ali, J. Lee, M. Humt, J. Feng, A. Kruspe, R. Triebel, P. Jung, R. Roscher, et al.
A survey of uncertainty in deep neural networks.
Artificial Intelligence Review, pages 1–77, 2023.

[3] C. Guo, G. Pleiss, Y. Sun, and K. Q. Weinberger.
On calibration of modern neural networks.
In *Proceedings of the 34th International Conference on Machine Learning (ICML)*, pages 1321–1330, 2017.

[4] M. Kull and P. Flach.
Novel decompositions of proper scoring rules for classification: score adjustment as precursor to calibration.
In *Proceedings of the 2015th European Conference on Machine Learning and Knowledge Discovery in Databases (ECML-PKDD)*, pages 68–85, 2015.

[5] M. Kull, T. Silva Filho, and P. Flach.
Beta calibration: a well-founded and easily implemented improvement on logistic calibration for binary classifiers.
In *Artificial Intelligence and Statistics*, pages 623–631. PMLR, 2017.

[6] M. Kull, T. M. Silva Filho, and P. Flach.
Beyond sigmoids: How to obtain well-calibrated probabilities from binary classifiers with beta calibration.
Electronic Journal of Statistics, 11:5052–5080, 2017.

[7] A. Kumar, S. Sarawagi, and U. Jain.
Trainable calibration measures for neural networks from kernel mean embeddings.
In *Proceedings of the 37th International Conference on Machine Learning (ICML)*, pages 2805–2814. PMLR, 2018.

References II

[8] H. Löfström, T. Löfström, U. Johansson, and C. Sönströd.
Investigating the impact of calibration on the quality of explanations.
Annals of Mathematics and Artificial Intelligence, pages 1–18, 2023.

[9] M. Sadinle, J. Lei, and L. Wasserman.
Least ambiguous set-valued classifiers with bounded error levels.
Journal of the American Statistical Association, 114(525):223–234, 2019.

[10] T. Silva Filho, H. Song, M. Perello-Nieto, R. Santos-Rodriguez, M. Kull, and P. Flach.
Classifier calibration: a survey on how to assess and improve predicted class probabilities.
Machine Learning, pages 1–50, 2023.

[11] V. Vovk.
Conditional validity of inductive conformal predictors.
In *Proceedings of the 4th Asian conference on machine learning (ACML)*, pages 475–490. PMLR, 2012.

[12] B. Zadrozny and C. Elkan.
Transforming classifier scores into accurate multiclass probability estimates.
In *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining (SIGKDD)*, pages 694–699, 2002.