# Conformal prediction via regression-as-classification (ICLR 2024)

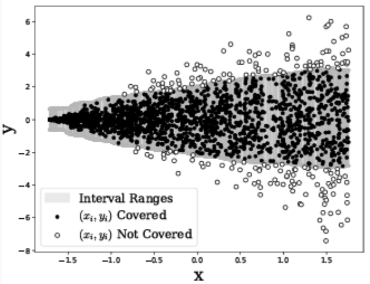
February 25, 2025

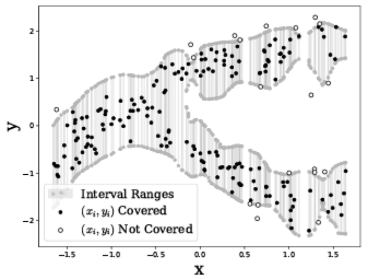
Seoul National University

# Introduction

- CP techniques aim to construct a prediction set that, for given test inputs, is guaranteed to contain the true (unknown) output with high probability.
- The set is built using a *conformity score*, which, roughly speaking, indicates the similarity between a new test example and the training examples.
- For regression, CP can be challenging when the output distribution is heteroscedastic, multimodal, or skewed (Lei & Wasserman, 2014).
- The main challenge lies in the design of the *conformity score*.
- We convert regression to a classification problem and then use CP for classification to obtain CP sets for regression.
- To preserve the ordering of the continuous-output space, we design a new loss function and make necessary modifications to the CP classification techniques.

# Examples





- **Left**: Example where the output distribution is heteroscedastic.
- **Right**: Example where the output distribution is bimodal.

# BACKGROUND ON CONFORMAL PREDICTION

- Given a new input $x_{new}$, CP techniques aim to construct a set that contains the true but unknown output $y_{new}$ with high probability.
- Assuming that a pair of input-output variables $(x, y)$ has a joint density $p(x, y)$ and a conditional density $p(y|x)$, oracle prediction sets for the output $y$ can be constructed as

$$\{z \in \mathbb{R} : p(x, z) \geq \tau_\alpha\} \quad \text{or} \quad \{z \in \mathbb{R} : p(z \mid x) \geq \tau_{\alpha,x}\},$$

  where the thresholds $\tau_\alpha$ and $\tau_{\alpha,x}$ are selected to ensure that the corresponding sets have a probability mass that meets or exceeds prescribed confidence level $1 - \alpha \in (0, 1)$.
- As the ground-truth distribution is unknown, we rely on estimating these uncertainty sets using the density estimators $\hat{p}(x, y)$ and $\hat{p}(x|y)$
- Without a stronger distribution assumption, the finite-sample guarantee is typically not upheld.

# BACKGROUND ON CONFORMAL PREDICTION

- Conformal Prediction has arisen as a method for yielding sets that do hold finite-sample guarantees.
- Given a observed instance $(x_{new}, y_{new})$ where $y_{new}$ is unknown, Conformal Prediction (Vovk et al., 2005) constructs a set of values that contains $y_{new}$ with high probability without knowing the underlying data distribution.
- This property is guaranteed under mild assumption that the data satisfies exchangeability.
- The set is called the conformal set and is built using a conformity score, denoted by $\sigma(x, y)$, which measures how appropriate an output value is for a given input example.

# BACKGROUND ON CONFORMAL PREDICTION

- There are many ways to build the conformity score, but they all involve splitting the data into a training set $\mathcal{D}_{tr}$ and a calibration set $\mathcal{D}_{cal}$
- Often, a prediction model $\mu_{tr}(x)$ is built using the training set, and then a conformity score is obtained using this model along with the calibration set.
- The conformal set merely gathers the points with larger conformity scores:

  $$\{z \in \mathbb{R} : \sigma(x_{new}, z) \geq Q_{1-\alpha}(\mathcal{D}_{cal})\},$$

  where $Q_{1-\alpha}(\mathcal{D}_{cal})$ is the $(1-\alpha)$ quantile od the conformity scored on the calibration data.
- This set provably contains $y_{new}$ with probability larger than $1-\alpha$ for any finite sample size and without assumption on the ground-truth distribution.

# BACKGROUND ON CONFORMAL PREDICTION

- There are many design choices for this conformity score.
- For example, one can choose a prediction model $\mu_{tr}(x)$ as an estimate of the conditional expectation and $\sigma(x, y) = -|y - \mu_{tr}(x)|$.
- The corresponding conformal set is a single interval centered around the prediction $\mu_{tr}(x)$ and of constant length $Q_{1-\alpha}(\mathcal{D}_{cal})$ for any example $x_{new}$, without taking into account its variability.
- However, in situations where the underlying data distribution demonstrates skewness or heteroscedasticity, we may desire a more flexible conformity score.
- We explore established density estimation in Classification Conformal Prediction that are already performing effectively.

# CP via Regression-as-Classification

# Classification Conformal Prediction

- We aim to compute a conformity function that accurately predicts the appropriateness of a label for a specific data point.
- Typically, practitioners perform conformal prediction for classification with probability estimates from Softmax neural network that covers $K$ output logits using cross-entropy loss.
- Let us denote the parametrized density

$$q_\theta(\cdot|x) = \text{softmax}(f_\theta(x)), \text{where softmax}(v)_j = \frac{exp(v_j)}{\sum_{k=1}^{K} exp(v_k)},$$

where $f_\theta : \mathbb{R}^d \rightarrow \mathbb{R}^K$
as the outputted discrete probability distribution over the labels of the input $x$.

# Classification Conformal Prediction

- Traditionally, we fit our neural network by minimizing the cross-entropy loss on the training set:

  $$\hat{\theta} \in argmin_{\theta} \sum_{i=1}^{n} KL(\delta_{y_i} || q_{\theta}(\cdot|x_i)).$$

  Here, $\delta_{y_i}$ is the Dirac Distribution with all of the probabilitistic mass on $y_i$

- A natural conformity score is simply the probability of a label according to the learned conditional distribution, i.e.,

  $$\sigma(x, y) = q_{\hat{\theta}}(y|x)$$

  assuming that we acquired a $\hat{\theta}$ that has minimized the traditional cross-entropy loss function on the training dataset.

# Regression to Classification approach

- The distribution of labels in the regression scenario is continuous, and learning a continuous distribution directly using a neural network is challenging(Rothfuss et al., 2019).
- It would be desirale to use similar methods for both classification and regression conformal prediction.
- We simply turn a regression problem into a classification problem by binning the range space.
- Specifically, generate $K$ bins with $K$ equally spaced numbers covering the interval $\mathcal{Y} = [y_{min}, y_{mx}]$, where $y_{min}$ (or $y_{max}$) is the minimum (or maximum) value of the labels observed in the training set.

# Regression to Classification approach

- Explicitly, we define our discretization of the label space as

$$\hat{\mathcal{Y}} = \{\hat{y}_1, ..., \hat{y}_K\} \text{ where } \hat{y}_{k+1} = \hat{y}_k + \frac{\hat{y}_K - \hat{y}_1}{K-1} \text{ with } \hat{y}_1 = y_{min} \text{ and } \hat{y}_K = y_{max}$$

- These values $\hat{y} \in \mathcal{Y}$ from the midpoints for each bin of our discretization.
- To unify classication and regression conformal prediction, a simple solution is to employ the Classification Conformal Prediction model with discrete labels $\tilde{y}_i = argmin_{\hat{y} \in \hat{\mathcal{Y}}} |y_i - \hat{y}|$
- This will aid in training the neural network with modified labels through cross-entropy loss, resulting in a discrete distribution of $q_\theta(\cdot|x)$
- To compute conformity scores for all labels, employ linear interploation from the discrete probability function $q_\theta(\cdot|x)$ to generate the continuous distribution $\bar{q}_\theta(\cdot|x)$. for any $y$ between $\hat{y}_k$ and $\hat{y}_{k+1}$

$$\bar{q}_\theta(y|x) = \gamma_k q_\theta(\hat{y}_k|x) + (1-\gamma_k) q_\theta(\hat{y}_{k+1}|x),\ \gamma_k = \frac{\hat{y}_{k+1} - y}{\hat{y}_{k+1} - \hat{y}_k}$$

# Data Fitting

- Problem with employing CrossEntropy loss in the classification conformal prediction is that *any structural relationships between classes are disregarded*
- For classification context, no structure exists between classes, and each class is independent.
- However, within the regression setting, the labels adhere to an ordinal structure.
- Need to a new loss function that incentivizes the allocation of probabilistic mass not only to the corret bin but also to the neighboring bins.

# Data Fitting

- Given an input and output pair $(x, y)$, our goal is to determine a density estimate $q_\theta(\cdot|x)$ that assigns low probability to points that are far to the true label $y$, i.e.,

$$q_\theta(\hat{y}|x_i) \text{ high when the loss } l(\hat{y}, y_i) \text{ is small}$$

$$\text{with } l(\hat{y}, y_i) = |y_i - \hat{y}|^p,\ p > 0$$

- Hence, a natural desideratum for learning the probability density function $q_\theta$ is that their product $l(y, \hat{y})q(\hat{y}|x)$ is small in expectation.
- we propose to find a ditribution $q_\theta$ minimizing the loss

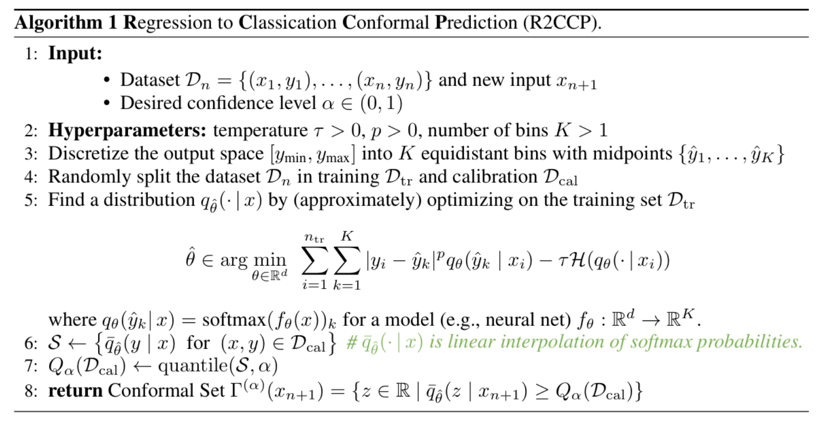
$$\mathbb{E}_{\hat{y}\sim q(\cdot|x)}[l(y, \hat{y})] = \sum_{k=1}^{K} l(y, \hat{y}_k)q(\hat{y}_k|x)$$

# Entropy Regularization

- Although the proposed loss function better encodes the connection between bins, it tends toward outputting Dirac distribution.
- For smoothness we rely on a classical entropy regularization technique for learning density estimators(Wainwright & Jordan, 2008)
- Formally, we can calculate the entropy of our probability distribution by using the Shannon entropy $\mathcal{H}$ of the produced probability distribution $q(\cdot|x)$ as a penalty term as follows:

$$\mathcal{H}(q_\theta(\cdot|x)) = \sum_{k=1}^{K} q_\theta(\hat{y}_k|x) log\, q_\theta(\hat{y}_k|x)$$

# Summary



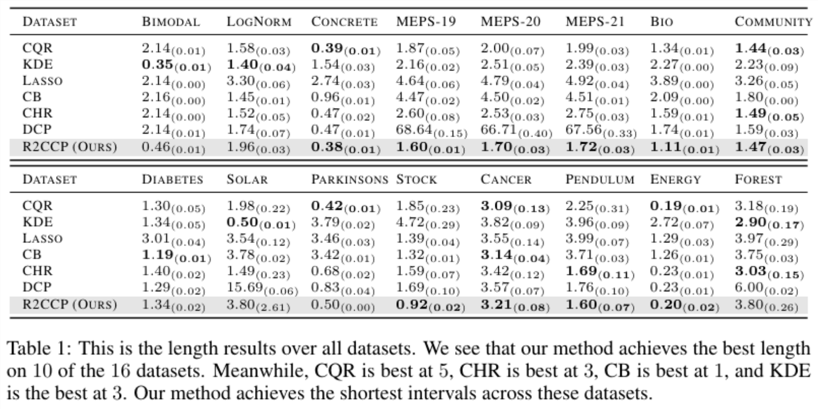
**Algorithm 1** **R**egression to **C**lassication **C**onformal **P**rediction (R2CCP).

```
Input:
    • Dataset $\mathcal{D}_n = \{(x_1, y_1), \ldots, (x_n, y_n)\}$ and new input $x_{n+1}$
    • Desired confidence level $\alpha \in (0, 1)$
Hyperparameters: temperature $\tau > 0$, $p > 0$, number of bins $K > 1$
Discretize the output space $[y_{\text{min}}, y_{\text{max}}]$ into $K$ equidistant bins with midpoints $\{\hat{y}_1, \ldots, \hat{y}_K\}$
Randomly split the dataset $\mathcal{D}_n$ in training $\mathcal{D}_{\text{tr}}$ and calibration $\mathcal{D}_{\text{cal}}$
Find a distribution $q_{\hat{\theta}}(\cdot \mid x)$ by (approximately) optimizing on the training set $\mathcal{D}_{\text{tr}}$

    $$\hat{\theta} \in \arg\min_{\theta \in \mathbb{R}^d} \sum_{i=1}^{n_{\text{tr}}} \sum_{k=1}^{K} |y_i - \hat{y}_k|^p q_\theta(\hat{y}_k \mid x_i) - \tau \mathcal{H}(q_\theta(\cdot \mid x_i))$$

    where $q_\theta(\hat{y}_k | x) = \text{softmax}(f_\theta(x))_k$ for a model (e.g., neural net) $f_\theta : \mathbb{R}^d \to \mathbb{R}^K$.
$\mathcal{S} \leftarrow \{\bar{q}_{\hat{\theta}}(y \mid x) \text{ for } (x, y) \in \mathcal{D}_{\text{cal}}\}$ # $\bar{q}_{\hat{\theta}}(\cdot \mid x)$ is linear interpolation of softmax probabilities.
$Q_\alpha(\mathcal{D}_{\text{cal}}) \leftarrow \text{quantile}(\mathcal{S}, \alpha)$
return Conformal Set $\Gamma^{(\alpha)}(x_{n+1}) = \{z \in \mathbb{R} \mid \bar{q}_{\hat{\theta}}(z \mid x_{n+1}) \geq Q_\alpha(\mathcal{D}_{\text{cal}})\}$
```

# Experiments

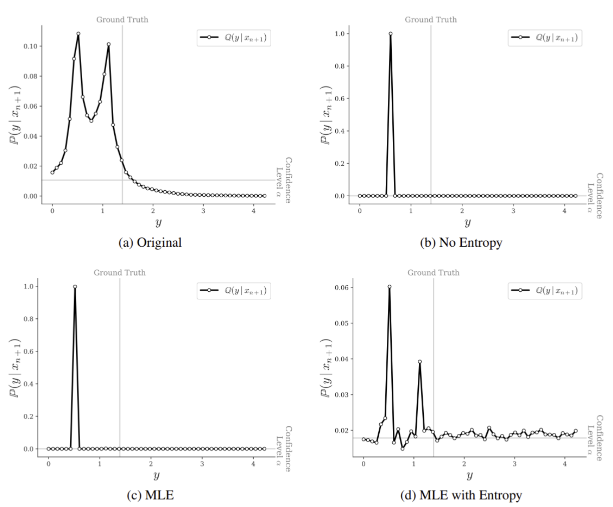


| Dataset | Bimodal | LogNorm | Concrete | MEPS-19 | MEPS-20 | MEPS-21 | Bio | Community |
|---|---|---|---|---|---|---|---|---|
| CQR | $2.14_{(0.01)}$ | $1.58_{(0.03)}$ | **$0.39_{(0.01)}$** | $1.87_{(0.05)}$ | $2.00_{(0.07)}$ | $1.99_{(0.03)}$ | $1.34_{(0.01)}$ | **$1.44_{(0.03)}$** |
| KDE | **$0.35_{(0.01)}$** | **$1.40_{(0.04)}$** | $1.54_{(0.03)}$ | $2.16_{(0.02)}$ | $2.51_{(0.05)}$ | $2.39_{(0.03)}$ | $2.27_{(0.00)}$ | $2.23_{(0.09)}$ |
| Lasso | $2.14_{(0.00)}$ | $3.30_{(0.06)}$ | $2.74_{(0.03)}$ | $4.64_{(0.06)}$ | $4.79_{(0.04)}$ | $4.92_{(0.04)}$ | $3.89_{(0.00)}$ | $3.26_{(0.05)}$ |
| CB | $2.16_{(0.00)}$ | $1.45_{(0.01)}$ | $0.96_{(0.01)}$ | $4.47_{(0.02)}$ | $4.50_{(0.02)}$ | $4.51_{(0.01)}$ | $2.09_{(0.00)}$ | $1.80_{(0.00)}$ |
| CHR | $2.14_{(0.00)}$ | $1.52_{(0.05)}$ | $0.47_{(0.02)}$ | $2.60_{(0.08)}$ | $2.53_{(0.03)}$ | $2.75_{(0.03)}$ | $1.59_{(0.01)}$ | **$1.49_{(0.05)}$** |
| DCP | $2.14_{(0.01)}$ | $1.74_{(0.07)}$ | $0.47_{(0.01)}$ | $68.64_{(0.15)}$ | $66.71_{(0.40)}$ | $67.56_{(0.33)}$ | $1.74_{(0.01)}$ | $1.59_{(0.03)}$ |
| R2CCP (Ours) | $0.46_{(0.01)}$ | $1.96_{(0.03)}$ | **$0.38_{(0.01)}$** | **$1.60_{(0.01)}$** | **$1.70_{(0.03)}$** | **$1.72_{(0.03)}$** | **$1.11_{(0.01)}$** | **$1.47_{(0.03)}$** |

| Dataset | Diabetes | Solar | Parkinsons | Stock | Cancer | Pendulum | Energy | Forest |
|---|---|---|---|---|---|---|---|---|
| CQR | $1.30_{(0.05)}$ | $1.98_{(0.22)}$ | **$0.42_{(0.01)}$** | $1.85_{(0.23)}$ | **$3.09_{(0.13)}$** | $2.25_{(0.31)}$ | **$0.19_{(0.01)}$** | $3.18_{(0.19)}$ |
| KDE | $1.34_{(0.05)}$ | **$0.50_{(0.01)}$** | $3.79_{(0.02)}$ | $4.72_{(0.29)}$ | $3.82_{(0.09)}$ | $3.96_{(0.09)}$ | $2.72_{(0.07)}$ | **$2.90_{(0.17)}$** |
| Lasso | $3.01_{(0.04)}$ | $3.54_{(0.12)}$ | $3.46_{(0.03)}$ | $1.39_{(0.04)}$ | $3.55_{(0.14)}$ | $3.99_{(0.07)}$ | $1.29_{(0.03)}$ | $3.97_{(0.29)}$ |
| CB | **$1.19_{(0.01)}$** | $3.78_{(0.02)}$ | $3.42_{(0.01)}$ | $1.32_{(0.01)}$ | **$3.14_{(0.04)}$** | $3.71_{(0.03)}$ | $1.26_{(0.01)}$ | $3.75_{(0.03)}$ |
| CHR | $1.40_{(0.02)}$ | $1.49_{(0.23)}$ | $0.68_{(0.02)}$ | $1.59_{(0.07)}$ | $3.42_{(0.12)}$ | **$1.69_{(0.11)}$** | $0.23_{(0.01)}$ | **$3.03_{(0.15)}$** |
| DCP | $1.29_{(0.02)}$ | $15.69_{(0.06)}$ | $0.83_{(0.04)}$ | $1.69_{(0.10)}$ | $3.57_{(0.07)}$ | $1.76_{(0.10)}$ | $0.23_{(0.01)}$ | $6.00_{(0.02)}$ |
| R2CCP (Ours) | $1.34_{(0.02)}$ | $3.80_{(2.61)}$ | $0.50_{(0.00)}$ | **$0.92_{(0.02)}$** | **$3.21_{(0.08)}$** | **$1.60_{(0.07)}$** | **$0.20_{(0.02)}$** | $3.80_{(0.26)}$ |

Table 1: This is the length results over all datasets. We see that our method achieves the best length on 10 of the 16 datasets. Meanwhile, CQR is best at 5, CHR is best at 3, CB is best at 1, and KDE is the best at 3. Our method achieves the shortest intervals across these datasets.

# Experiments



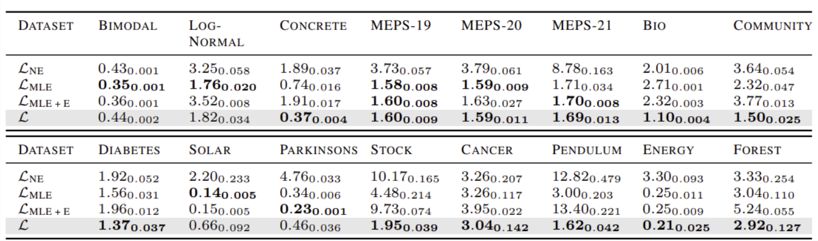
(a) Original

(b) No Entropy

(c) MLE

(d) MLE with Entropy

# Experiments



| Dataset | Bimodal | Log-Normal | Concrete | MEPS-19 | MEPS-20 | MEPS-21 | Bio | Community |
|---|---|---|---|---|---|---|---|---|
| $\mathcal{L}_{\text{NE}}$ | $0.43_{0.001}$ | $3.25_{0.058}$ | $1.89_{0.037}$ | $3.73_{0.057}$ | $3.79_{0.061}$ | $8.78_{0.163}$ | $2.01_{0.006}$ | $3.64_{0.054}$ |
| $\mathcal{L}_{\text{MLE}}$ | $\mathbf{0.35_{0.001}}$ | $\mathbf{1.76_{0.020}}$ | $0.74_{0.016}$ | $\mathbf{1.58_{0.008}}$ | $\mathbf{1.59_{0.009}}$ | $1.71_{0.034}$ | $2.71_{0.001}$ | $2.32_{0.047}$ |
| $\mathcal{L}_{\text{MLE + E}}$ | $0.36_{0.001}$ | $3.52_{0.008}$ | $1.91_{0.017}$ | $\mathbf{1.60_{0.008}}$ | $1.63_{0.027}$ | $\mathbf{1.70_{0.008}}$ | $2.32_{0.003}$ | $3.77_{0.013}$ |
| $\mathcal{L}$ | $0.44_{0.002}$ | $1.82_{0.034}$ | $\mathbf{0.37_{0.004}}$ | $\mathbf{1.60_{0.009}}$ | $\mathbf{1.59_{0.011}}$ | $\mathbf{1.69_{0.013}}$ | $\mathbf{1.10_{0.004}}$ | $\mathbf{1.50_{0.025}}$ |

| Dataset | Diabetes | Solar | Parkinsons | Stock | Cancer | Pendulum | Energy | Forest |
|---|---|---|---|---|---|---|---|---|
| $\mathcal{L}_{\text{NE}}$ | $1.92_{0.052}$ | $2.20_{0.233}$ | $4.76_{0.033}$ | $10.17_{0.165}$ | $3.26_{0.207}$ | $12.82_{0.479}$ | $3.30_{0.093}$ | $3.33_{0.254}$ |
| $\mathcal{L}_{\text{MLE}}$ | $1.56_{0.031}$ | $\mathbf{0.14_{0.005}}$ | $0.34_{0.006}$ | $4.48_{0.214}$ | $3.26_{0.117}$ | $3.00_{0.203}$ | $0.25_{0.011}$ | $3.04_{0.110}$ |
| $\mathcal{L}_{\text{MLE + E}}$ | $1.96_{0.012}$ | $0.15_{0.005}$ | $\mathbf{0.23_{0.001}}$ | $9.73_{0.074}$ | $3.95_{0.022}$ | $13.40_{0.221}$ | $0.25_{0.009}$ | $5.24_{0.055}$ |
| $\mathcal{L}$ | $\mathbf{1.37_{0.037}}$ | $0.66_{0.092}$ | $0.46_{0.036}$ | $\mathbf{1.95_{0.039}}$ | $\mathbf{3.04_{0.142}}$ | $\mathbf{1.62_{0.042}}$ | $\mathbf{0.21_{0.025}}$ | $\mathbf{2.92_{0.127}}$ |