Human-Machine Calibration

Using Conformal Prediction

Why Calibration Matters?

- Machine metrics ≠ Human judgment
- Need for trustworthy evaluation
- Importance in regulated domains

Key Challenges

- Subjectivity in evaluation
- Cost of human labeling
- Need for uncertainty quantification

H2O.ai

Conformal Prediction

- Statistical framework for prediction sets
- Guaranteed coverage probabilities
- Uncertainty quantification

Two Main Approaches

Transductive (Full) Conformal Prediction

- Retrains with each prediction
- Higher accuracy, more computational cost

Split Conformal Prediction

- Uses separate calibration set
- More efficient, slightly less accurate

Statistical Guarantees

- Coverage probability: $P(Y \in C(X)) \geq 1 - \alpha$
- Exchangeability assumption
- Finite sample validity
- Distribution Free

Steps

1. Pick a machine metrics to calibrate
2. Train the Model on Entire Labeled Dataset

   Logistics, Isotonic Regression, Monotonic xgboost, etc.
3. Calculate Nonconformity Scores for Each Data Point
4. Determine the Quantile Threshold
   – Sort the nonconformity scores
   – Set a confidence level 1 - α (e.g., 95%)
5. Assign Hypothetical Labels to the New Unlabeled Observation
6. Compare Nonconformity Scores to the Quantile Threshold
7. Classify the New Observation: {0}, {1}, {0,1},{}

Non Conformity Scores

1. Negative Logit Score
   - Based on prediction confidence
   - Distance from decision boundary

   α = -log(p(y)/(1-p(y)))
   where p(y) = predicted probability for true class

2. Residual Score
   - Direct error measurement
   - |true_label - predicted_probability|

   α = |ytrue - p(ypositive)|
   where p(ypositive) = predicted probability for positive class

Choosing the Right Score

- Negative Logit: When confidence matters
- Residual: When error measurement is key

Implementation Detail

1. Model Training

   Train model M on labeled data D = {(xi, yi)}

2. Nonconformity Calculation

   For each point (x, y):
   α = nonconformity_score(M, x, y)

3. Threshold Determination

   Q = (n+1)(1-α)/n # n = number of calibration points
   τ = quantile(non_conformity_scores, Q)

4. Prediction Set Construction

   C(x) = {y : nonconformity_score(M, x, y) ≤ τ}

5. Decision Making

   For each test point:
   if nonconformity_score < τ:
       include in prediction set
   else:
       exclude from prediction set

Active Learning

1. Train initial model
2. Measure Uncertainty
3. Select Sample
4. Get Human Labels
5. Model update
   - Update training set
   - Retrain model if necessary
   - Recalibrate conformal predictor

```
2. Calculate uncertainty for each unlabeled point:
   U(x) = size(C(x))  # size of prediction set

3. Select points for labeling:
   X_select = argmax_x U(x)
```

H2O.ai

Key Takeaways

- Conformal prediction provides rigorous uncertainty quantification
- Choice of method depends on computational resources
- Active learning optimizes human labeling effort
- Framework enables trustworthy automated evaluation