Human-Machine Calibration

Using Conformal Prediction for Trustworthy Evaluation

July 5, 2025

Outline

What is Human-Machine Calibration?

Core Definition

Human-Machine Calibration bridges the gap between **machine metrics** and **human judgment** using statistical frameworks to provide trustworthy, interpretable evaluation.

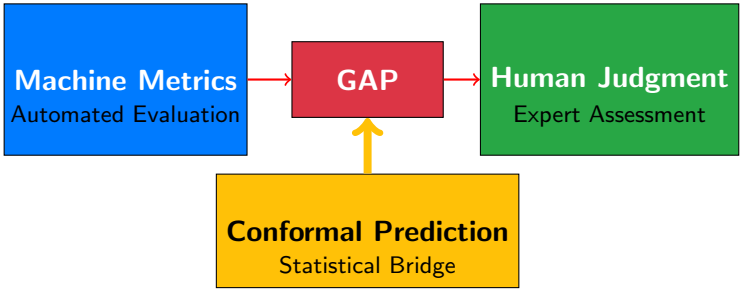
Key Components:

- **Uncertainty Quantification**: Measure prediction reliability
- **Statistical Guarantees**: Provide mathematical confidence bounds
- **Conformal Prediction**: Framework for prediction sets with coverage guarantees
- **Active Learning**: Minimize human labeling effort through smart sampling

Why It Matters

In regulated domains and high-stakes applications, we need reliable ways to know when to trust machine predictions and when to seek human oversight.

The Calibration Challenge



The Problem: Machine metrics may not align with human judgment, especially in:

- Subjective evaluation tasks
- Complex reasoning scenarios
- Domain-specific nuances
- High-stakes decisions

Why Calibration Matters?

Key Challenges:

Machine metrics $\neq$ Human judgment

Automated systems may score differently from domain experts, leading to misaligned outcomes.

Need for trustworthy evaluation

Stakeholders require confidence in AI system decisions, especially in regulated or high-risk scenarios.

Importance in regulated domains

Financial services, healthcare, and legal applications demand rigorous uncertainty quantification.

Critical Issues:

- Subjectivity in evaluation
- Cost of human labeling
- Need for uncertainty quantification
- Regulatory compliance requirements
- Risk management needs

Solution Requirements:

- Statistical rigor
- Computational efficiency
- Practical interpretability
- Scalable implementation

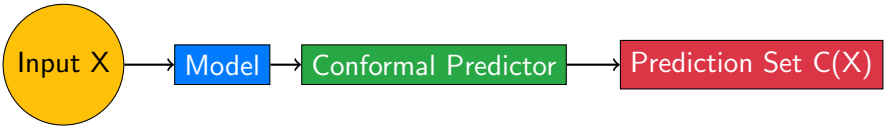
Conformal Prediction Overview

Core Concept

Conformal Prediction is a **statistical framework** for creating prediction sets with **guaranteed coverage probabilities** and rigorous **uncertainty quantification**.

Key Properties:

- **Statistical Guarantees**: Coverage probability $P(Y \in C(X)) \geq 1 - \alpha$
- **Distribution Free**: No assumptions about data distribution
- **Finite Sample Validity**: Works with limited training data
- **Exchangeability**: Assumes data points are exchangeable



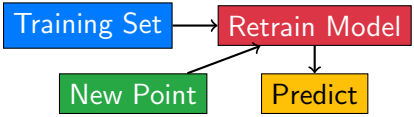
Two Main Approaches

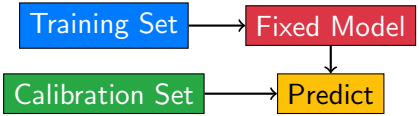
Transductive (Full) Conformal Prediction

- Retrains model with each new prediction
- Higher accuracy and tighter bounds
- More computationally expensive
- Suitable for small datasets

Split Conformal Prediction

- Uses separate calibration dataset
- More computationally efficient
- Slightly less accurate bounds
- Suitable for large-scale applications





Choice Criteria

Use **Transductive** for highest accuracy with small datasets. Use **Split** for computational efficiency with large-scale production systems.

Non-Conformity Scores

Purpose

Non-conformity scores measure how **unusual** or **unexpected** a prediction is compared to the training data patterns.

Two Main Types:

1. Negative Logit Score

$$\alpha = -\log\left(\frac{p(y)}{1-p(y)}\right) \tag{1}$$

Properties:

- Based on prediction confidence
- Distance from decision boundary
- Best when confidence matters
- Captures model uncertainty

2. Residual Score

$$\alpha = |y_{\text{true}} - p_{\text{predicted}}| \tag{2}$$

Properties:

- Direct error measurement
- Absolute prediction difference
- Best for error quantification
- Simple and interpretable

Implementation Steps

Training

1. Pick metrics to calibrate
2. Train model on human-labeled dataset
3. Calculate non-conformity scores for each data point
4. Determine quantile threshold: Set confidence level 1- (e.g., 95%)

Prediction

5. Assign hypothetical labels to new unlabeled observation
6. Compare non-conformity scores to quantile threshold
7. Classify new observation: {0}, {1}, {0,1}, {}

Mathematical Foundation:

$$\hat{q} = \text{Quantile}\left(\{S_1, \ldots, S_n\}, \frac{\lceil (n+1)(1-\alpha) \rceil}{n}\right) \tag{3}$$

Prediction Set Construction

Conformal Prediction Results

The output is a **prediction set** that contains the true label with probability 1-.

Possible Outcomes:

- $\{\mathbf{0}\}$: Confident prediction of class 0
- $\{\mathbf{1}\}$: Confident prediction of class 1
- $\{\mathbf{0,1}\}$: Uncertain prediction - both classes possible
- $\{\}$: Empty set - very unusual observation (rare)

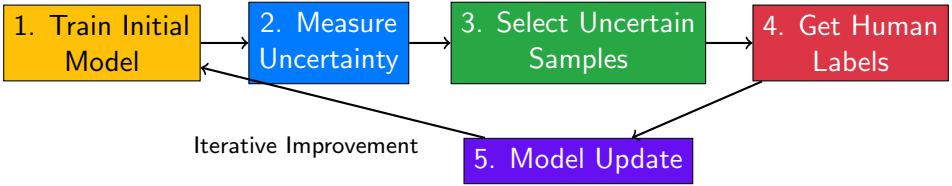
Coverage Guarantee:

$$P(Y \in C(X)) \geq 1 - \alpha$$

Active Learning with Conformal Prediction

Objective

Minimize human labeling effort by strategically selecting the most **uncertain** examples for manual annotation.



Uncertainty Measurement:

$$U(x) = \text{size}(C(x)) \quad \text{and} \quad x_{\text{select}} = \arg\max_x U(x) \tag{4}$$

Benefits:

Reduces human labeling costs by 50-80% and focuses expert attention on difficult cases, improving model performance efficiently

Active Learning Process Details

Model Update Workflow:

1. **Update Training Set**: Add newly labeled examples
2. **Retrain Model**: Improve predictions with new data (if necessary)
3. **Recalibrate Conformal Predictor**: Update uncertainty estimates

Selection Strategies:

- **Largest Prediction Sets**: Select samples with $|C(x)| = 2$
- **Highest Uncertainty**: Use prediction set size as uncertainty measure
- **Diversity Sampling**: Ensure coverage across different data regions
- **Balanced Selection**: Mix uncertain and representative samples

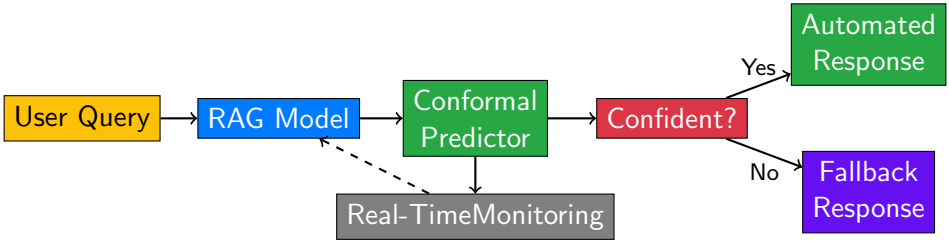
Stopping Criteria: Continue active learning until

- Desired performance level achieved
- Human labeling budget exhausted
- Uncertainty levels stabilize
- Model convergence detected

Safety Mechanisms in Model Operation

1. **Fallback Mechanisms**
   - Set confidence bounds using prediction intervals
   - Trigger safe responses when predictions fall outside thresholds
   - Default to human review or simplified responses
2. **Decision Gating**
   - Route uncertain predictions to human reviewers
   - Apply stricter thresholds for high-stakes scenarios
   - Escalate complex queries automatically
3. **Real-Time Monitoring**
   - Continuously validate using calibrated metrics
   - Detect concept drift through prediction patterns
   - Monitor coverage probability maintenance
4. **Context-Aware Alerts**
   - Configure alerts based on conformal scores
   - Adapt thresholds to different query types
   - Implement escalation hierarchies

Production Deployment Framework



Decision Logic:

- **High Confidence** (small prediction sets): Automated response
- **Low Confidence** (large sets): Fallback, human review, escalation

Implementation Best Practices

Getting Started:

1. **Choose Appropriate Method**: Split conformal for scale, transductive for accuracy
2. **Select Right Scores**: Negative logit for confidence, residual for error measurement
3. **Set Conservative Thresholds**: Start with 95% confidence, adjust based on domain
4. **Validate on Historical Data**: Test coverage guarantees before production
5. **Implement Safety Mechanisms**: Always have fallback options

Common Pitfalls to Avoid:

- Ignoring exchangeability assumptions
- Using inadequate calibration data
- Setting thresholds without domain validation
- Neglecting concept drift monitoring
- Over-relying on automation without human oversight

Success Factors:

- Strong statistical foundation
- Domain expert involvement
- Continuous monitoring and updating
- Clear escalation procedures
- Regular recalibration schedules

Key Takeaways

1. **Calibration is Essential for Trustworthy AI**
   - Machine metrics often differ from human judgment
   - Statistical frameworks provide reliability guarantees
   - Uncertainty quantification enables safe automation
2. **Conformal Prediction Provides Rigorous Foundation**
   - Mathematical guarantees on coverage probability
   - Distribution-free and finite-sample valid
   - Practical implementation with clear steps
3. **Active Learning Maximizes Efficiency**
   - Strategic sampling reduces labeling costs
   - Uncertainty-guided selection improves models
   - Iterative improvement maintains quality
4. **Safety Mechanisms Enable Production Deployment**
   - Fallback strategies handle edge cases
   - Real-time monitoring detects issues
   - Context-aware alerts prevent failures