Quantifying Uncertainty in Informational Systems: A Unified Framework with Applications to Language Models

Abstract

Uncertainty is an inherent aspect of informational systems, arising from various sources such as model assumptions, data limitations, computational constraints, and the intrinsic unpredictability of the systems themselves. This paper presents a unified framework for quantifying uncertainty by integrating entropy-based and Bayesian interpretations. We decompose total uncertainty into four distinct components: assumption uncertainty, data uncertainty, computational uncertainty, and inherent system uncertainty. Through mathematical formalization and a practical demonstration using language models, we illustrate how this framework provides actionable insights for improving model reliability and understanding system limitations. The framework's broader implications extend to enhancing decision-making in scientific, industrial, and societal contexts, offering a structured approach to navigating uncertainty in complex systems.

1. Introduction

Informational systems—ranging from machine learning models to physical simulations—are essential tools for understanding and predicting complex phenomena. However, these systems are invariably subject to uncertainty, which can stem from incomplete data, flawed assumptions, computational approximations, or the fundamental unpredictability of the system being modeled. Quantifying this uncertainty is critical for assessing the reliability of predictions and guiding improvements in modeling practices.

This paper introduces a comprehensive framework for quantifying uncertainty in informational systems by synthesizing entropy-based measures from information theory and Bayesian probabilistic methods. Our approach decomposes total uncertainty into four key components:

- **Assumption Uncertainty ((U_A))**: Discrepancies between the model's assumptions and the true system.
- **Data Uncertainty ((U_D))**: Imperfections in the input data, such as noise or incompleteness.
- **Computational Uncertainty ((U_C))**: Errors introduced by computational methods or approximations.
- **Inherent System Uncertainty ((U_I))**: Fundamental unpredictability intrinsic to the system.

By integrating these components into a unified model, we provide a structured way to measure and interpret uncertainty, enabling practitioners to identify its primary sources and take targeted actions. The framework is applied to language models as a case study, demonstrating its practical utility in a high-impact domain. We also explore the broader implications of this approach for understanding uncertainty in complex systems, including emergent phenomena where predictability is inherently limited.

2. Formalizing the Framework

2.1 Informational Systems

An informational system S is defined as a triplet:

$$S = (D, M, O)$$

where:

- D is the input data,
- M is the computational model,
- O is the output or prediction.

The system S transforms D into O via M, aiming to capture some aspect of a real-world phenomenon.

2.2 Components of Uncertainty

The total uncertainty in the system, $U(S)$, is decomposed into four components:

1. **Assumption Uncertainty (U_A)**: Arises from the model's structural assumptions (e.g., linearity, independence) that may not fully align with the true system.

2. **Data Uncertainty (U_D)**: Stems from imperfections in D, such as noise, sampling bias, or missing values.

3. **Computational Uncertainty (U_C)**: Results from approximations in implementing M, such as numerical errors or optimization limitations.

4. **Inherent System Uncertainty (U_I)**: Reflects the fundamental unpredictability of the system, which cannot be reduced by improving M or D.

2.3 Quantifying the Components

Each component is quantified as follows:

- **Assumption Uncertainty (U_A)**: Modeled using the Kullback-Leibler (KL) divergence between the true distribution P_{true} and the model's assumed distribution P_{model}:

$$U_A = D_{\text{KL}}(P_{\text{true}} | P_{\text{model}})$$

In practice, this can be approximated via generalization error or sensitivity analysis when P_{true} is unknown.

- **Data Uncertainty (U_D)**: For a dataset D, this can be quantified using statistical measures such as variance or Shannon entropy:

$$U_D = \text{Var}(D) \quad \text{or} \quad H(D) = -\sum p(d) \log p(d)$$

- **Computational Uncertainty (U_C)**: For numerical methods, this is often the discretization or approximation error:

$$U_C = | O_{\text{exact}} - O_{\text{approx}} |$$

- **Inherent System Uncertainty (U_I)**: Modeled as the irreducible variance or entropy when all other uncertainties are minimized:

$$U_I = \text{Var}(O | U_A, U_D, U_C \to 0)$$

In a Bayesian context, this aligns with aleatoric uncertainty, while U_A, U_D, and U_C contribute to epistemic uncertainty.

3. Mathematical Model for Total Uncertainty

The total uncertainty $U(S)$ is modeled as a weighted sum of the four components:

$$U(S) = w_A U_A + w_D U_D + w_C U_C + w_I U_I$$

where $w_A, w_D, w_C, w_I \geq 0$ are weights reflecting the relative importance of each component in a given context. This linear combination assumes independence between components for simplicity. In practice, interactions may exist (e.g., poor data amplifying computational errors), and future work could explore non-linear aggregations.

The entropy-based interpretation views $U(S)$ as the total entropy of the system’s output distribution, while the Bayesian perspective decomposes it into epistemic (reducible) and aleatoric (irreducible) uncertainty, with U_I representing the latter.

---

4. Application to Language Models

To illustrate the framework, we apply it to a language model predicting the next word in a sequence, a common task in natural language processing (NLP).

4.1 Setup

Consider a language model trained on a corpus to predict $p(y | x)$, where x is the input sequence and y is the next word. The total uncertainty is the entropy of the predictive distribution:

$$U(S) = H(p(y | x)) = -\sum_y p(y | x) \log p(y | x)$$

4.2 Decomposition

- **Inherent System Uncertainty (U_I)**: Represents linguistic ambiguity or randomness (e.g., multiple valid next words). Estimated as the average entropy across an ensemble of high-performing models:

$$U_I \approx \mathbb{E}_{\text{ensemble}}[H(p(y | x))]$$

- **Epistemic Uncertainty**: The reducible portion, computed as $U(S) - U_I$, and further decomposed:
  - **Data Uncertainty (U_D)**: Variability due to limited or noisy training data, measured by entropy differences across models trained on data subsets.
  - **Assumption Uncertainty (U_A)**: Variability across different model architectures (e.g., LSTM vs. Transformer), reflecting assumption mismatches.
  - **Computational Uncertainty (U_C)**: Variability due to random seeds, optimization paths, or precision limits.

4.3 Hypothetical Example

For the sequence “The cat sat on the,” the model predicts:

- “mat” (0.4),
- “chair” (0.3),
- “floor” (0.3).

Total uncertainty is (H = 1.08) bits. An ensemble of models yields an average entropy of 0.9 bits ((U_I)), suggesting 0.18 bits of epistemic uncertainty, which could be reduced by improving data, assumptions, or computation.

This decomposition reveals whether uncertainty arises from linguistic ambiguity ((U_I)) or modeling limitations, guiding practitioners accordingly.

5. Technical Implementation

Below is a Python script using SciPy to compute uncertainty components for a classification task, adaptable to language models with probabilistic outputs.

```
import numpy as np
from scipy.stats import entropy
from sklearn.utils import resample
from sklearn.linear_model import LogisticRegression
from sklearn.datasets import load_iris
from sklearn.model_selection import train_test_split

# Load dataset
iris = load_iris()
X, y = iris.data, iris.target
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Train ensemble on bootstrap samples
ensemble = []
for _ in range(10):
    X_boot, y_boot = resample(X_train, y_train)
    model = LogisticRegression(max_iter=200)
    model.fit(X_boot, y_boot)
    ensemble.append(model)

# Predict probabilities
probs = np.array([model.predict_proba(X_test) for model in ensemble])
avg_probs = np.mean(probs, axis=0)

# Total uncertainty: entropy of average predictions
total_uncertainty = entropy(avg_probs, axis=1)

# Aleatoric uncertainty: average entropy of individual predictions
entropies = entropy(probs, axis=2)
aleatoric_uncertainty = np.mean(entropies, axis=0)

# Epistemic uncertainty: total - aleatoric
epistemic_uncertainty = np.maximum(total_uncertainty - aleatoric_uncertainty, 0)

# Output results
print("Average Total Uncertainty:", np.mean(total_uncertainty))
print("Average Aleatoric Uncertainty:", np.mean(aleatoric_uncertainty))
```

```
print("Average Epistemic Uncertainty:", np.mean(epistemic_uncertainty))
```

Explanation

- **Ensemble**: Simulates variability in data and computation by training on bootstrap samples.
- **Total Uncertainty**: Entropy of the average predictive distribution.
- **Aleatoric Uncertainty ((U_I))**: Average entropy across ensemble predictions.
- **Epistemic Uncertainty**: Difference between total and aleatoric, capturing (U_A), (U_D), and (U_C).

For language models, replace the classifier with a model like a Transformer and use word probabilities.

---

6. Broader Themes and Implications

This framework extends beyond technical quantification to offer a conceptual lens for understanding predictability limits in complex systems. In domains like climate science, economics, or NLP, emergent phenomena often defy perfect modeling due to inherent uncertainty (U_I). By isolating this component, the framework clarifies where predictability ends and randomness begins.

Practically, it guides resource allocation—whether to collect more data (U_D), refine models (U_A), or optimize computation (U_C). Philosophically, it reflects a balance between knowable and unknowable aspects of reality, fostering humility in scientific inquiry and decision-making. For instance, in societal applications like policy modeling, acknowledging U_I prevents overconfidence in predictions.

The integration of entropy-based and Bayesian methods bridges information theory and probabilistic reasoning, aligning with existing uncertainty quantification efforts (e.g., in physics or AI safety) while offering a novel decomposition.

7. Conclusion

This paper presents a unified framework for quantifying uncertainty in informational systems, blending entropy-based and Bayesian perspectives. By decomposing uncertainty into assumption, data, computational, and inherent components, we provide a structured approach to understanding and mitigating uncertainty. Its application to language models demonstrates practical utility, while its broader implications illuminate the nature of predictability in complex systems. Future work could refine the mathematical aggregation of components, explore non-linear interactions, and validate the framework across diverse domains.

Quantifying uncertainty is both a practical tool and a philosophical stance, recognizing that our models are approximations of a reality that remains, in part, elusive.