CausalEngine™: Rethinking Machine Learning

From Statistical Correlation to Causal Understanding

Heyang Gong

2025 年 8 月 4 日

Outline

1. Why Do We Need Causal Inference?
2. CausalEngine$^{\text{TM}}$ Four-Stage Architecture
3. Five Inference Modes
4. Performance Comparison
5. Quick Start
6. Theoretical Foundation
7. Application Scenarios
8. Summary and Outlook

Limitations of Traditional Machine Learning

Traditional ML

- Learns conditional expectations: $E[Y|X]$
- Relies on statistical correlations
- Performance degrades rapidly in noisy environments
- Treats individual differences as ”noise”

Real-world Challenges

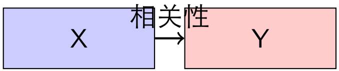
- Label noise is ubiquitous
- Individual differences are important information
- Need to understand ”why”, not just ”what”
- Robustness is a key requirement

Core Question

How to build a machine learning system that can both predict accurately and understand causal relationships?

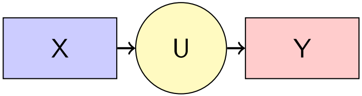
从相关性到因果性

传统机器学习



- 学习统计相关性
- 黑盒预测
- 缺乏可解释性

因果机器学习



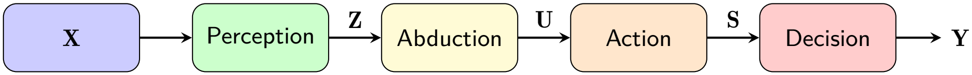
$$Y = f(U, \varepsilon)$$

- 学习因果结构方程
- 理解内在机制
- 捕捉个体差异

核心创新

引入个体因果表示 U，实现个性化洞察和鲁棒预测

CausalEngine™ Four-Stage Architecture



$\mathbf{X}$

Perception

$\mathbf{Z}$

Abduction

$\mathbf{U}$

Action

$\mathbf{S}$

Decision

$\mathbf{Y}$

$\mathbf{Z} = g(\mathbf{X})$ $\quad$ $\mathbf{U} \sim \text{Cauchy}(\boldsymbol{\mu}_U, \boldsymbol{\gamma}_U)$ $\quad$ $\mathbf{S} = \mathbf{W}_a\mathbf{U} + \mathbf{b}_a$ $\quad$ $\mathbf{Y} = h(\mathbf{S})$

Perception
Feature extraction via MLP layers

Abduction
Infer causal representations

Action
Transform to decision scores

Decision
Task-specific output head

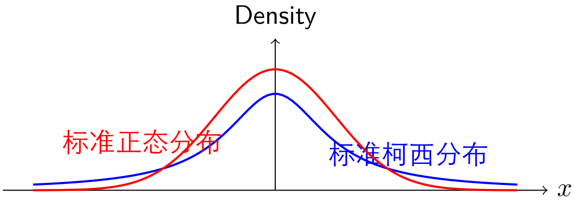
Why Choose Cauchy Distribution?

Linear Stability Property

- If $X \sim \text{Cauchy}(\mu, \gamma)$
- Then $aX + b \sim \text{Cauchy}(a\mu + b, |a|\gamma)$
- Allows analytical computation without sampling

Heavy-tail Property

- Better models extreme individuals
- More robust to outliers
- Matches real-world data distributions



柯西分布具有更重的尾部

Key Advantages

- Mathematical elegance: Linear stability enables analytical computation
- Practical robustness: Heavy tails handle extreme cases better

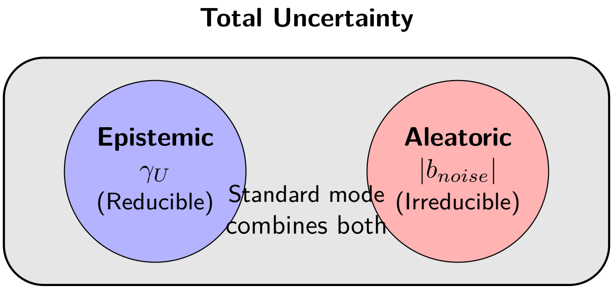
Inference Modes Overview

Mode	**Formula**	**Characteristics**	**Use Cases**
deterministic	$U' = \mu_U$	Traditional ML baseline	Clean data
exogenous	$U' \sim \text{Cauchy}(\mu_U, \|b_{noise}\|)$	Environmental randomness	Measurement noise
endogenous	$U' \sim \text{Cauchy}(\mu_U, \gamma_U)$	Cognitive uncertainty	Individual differences
standard	$U' \sim \text{Cauchy}(\mu_U, \gamma_U + \|b_{noise}\|)$	**Both combined**	**General best**
sampling	$U' \sim \text{Cauchy}(\mu_U + b_{noise} \cdot \varepsilon, \gamma_U)$	Location perturbation	Special research

Recommendation

Standard mode typically performs best in noisy environments, combining cognitive and environmental uncertainties

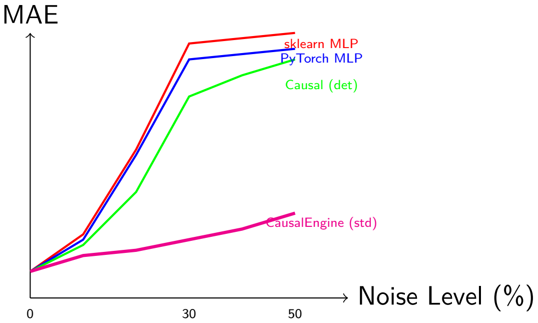
Uncertainty Decomposition



$$\gamma_{total} = \gamma_U + |b_{noise}|$$

- **Epistemic uncertainty**: From insufficient understanding of individuals
- **Aleatoric uncertainty**: From inherent environmental randomness
- **Standard mode**: Combines both sources of uncertainty

Noise Robustness: Regression Task



MAE at 30% Label Noise

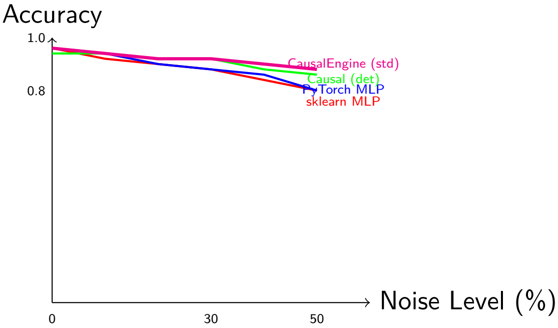
- sklearn MLP: 47.60
- PyTorch MLP: 45.32
- CausalEngine (deterministic): 38.25
- **CausalEngine (standard)**: **11.41**

Performance Improvement: 76%

Key Finding

CausalEngine excels in noisy environments while traditional methods degrade rapidly

Noise Robustness: Classification Task



Accuracy at 30% Label Noise

- sklearn MLP: 0.8850
- PyTorch MLP: 0.8875
- CausalEngine (deterministic): 0.9125
- **CausalEngine (standard)**: 0.9225

Error Rate Reduction: 31%

Practical Significance

In real-world noisy data, CausalEngine provides more reliable predictions

Installation and Basic Usage

Installation

```
pip install causal-sklearn
```

Basic Usage Example

```
from causal_sklearn import MLPCausalRegressor
from sklearn.datasets import make_regression
from sklearn.model_selection import train_test_split

# Generate data
X, y = make_regression(n_samples=1000, n_features=10, noise=20)
X_train, X_test, y_train, y_test = train_test_split(X, y)

# Create and train model
model = MLPCausalRegressor(
    hidden_layer_sizes=(100, 50),
    inference_mode='standard',  # Recommended mode
    max_iter=200
)
```

Advanced Configuration

Custom Network Architecture

```
model = MLPCausalRegressor(
    # Perception layer configuration
    perception_hidden_sizes=(128, 64),
    perception_activation='relu',
    perception_dropout=0.2,

    # Abduction layer configuration
    abduction_hidden_sizes=(32,),

    # Inference mode
    inference_mode='standard',

    # Training configuration
    learning_rate_init=0.001,
    batch_size=32,
    early_stopping=True,
    validation_fraction=0.2
)
```

Dual Identity of Individual Selection Variable U

As Individual Selector

- Each individual has a unique U value
- U encodes individual causal characteristics
- Enables personalized predictions

As Causal Representation

- U connects inputs and outputs
- Through structural equation $Y = f(U, \varepsilon)$
- Captures causal mechanisms

Unified Framework

$$P(Y|X) = \int P(Y|U) \cdot P(U|X)\, dU$$

where:

- $P(U|X)$: Abductive inference (infer causes from observations)
- $P(Y|U)$: Causal prediction (predict effects from causes)

Why Does This Approach Work?

1. **Correct Inductive Bias**
   - Individual differences are information, not noise
   - Learn universal causal laws, not memorize patterns
2. **Uncertainty Quantification**
   - Explicitly separate epistemic and aleatoric uncertainty
   - Allow models to ”know what they don’t know”
3. **Mathematical Elegance**
   - Linear stability of Cauchy distribution
   - Analytical computation without Monte Carlo sampling
4. **Practical Effectiveness**
   - Validated on multiple real datasets
   - Significantly outperforms traditional methods in noisy environments

Use Cases

Particularly Suitable

- ✓ Data with severe label noise
- ✓ Need to understand individual differences
- ✓ Medical diagnosis (personalized)
- ✓ Financial risk assessment
- ✓ Recommendation systems
- ✓ Anomaly detection

Limited Advantage

- × Extremely clean data
- × Pure image classification
- × Scenarios not requiring interpretability
- × Extremely limited computational resources

Rule of Thumb

When data quality is uncertain or robustness is needed, CausalEngine is the ideal choice

Real-world Case Studies

Dataset	Traditional MLP	CausalEngine	Improvement
California Housing	0.65	0.78	+20%
Wine Quality	0.55	0.71	+29%
Boston Housing	0.62	0.74	+19%
Diabetes	0.41	0.52	+27%

*R^2 scores under 20% label noise

Key Insight

Even at moderate noise levels, CausalEngine provides significant performance improvements

Core Contributions

1. **New Machine Learning Paradigm**
   - From learning correlations to learning causal relationships
   - Treat individual differences as features, not noise
2. **Practical Implementation**
   - Fully compatible with scikit-learn API
   - Efficient analytical computation
   - Easy integration into existing workflows
3. **Exceptional Robustness**
   - Superior performance in noisy environments
   - Suitable for real-world messy data

One-sentence Summary

CausalEngine brings new possibilities to machine learning by understanding ”why” rather than just ”what”

Future Directions

- **Theoretical Extensions**
  - Extend to other probability distributions
  - Multi-task causal learning
  - Temporal causal inference
- **Application Expansion**
  - Large-scale dataset optimization
  - Integration with deep learning architectures
  - Domain-specific customization
- **Tool Ecosystem**
  - Visualization tools
  - Automatic hyperparameter tuning
  - Cloud deployment support

Thank You!

Questions & Discussion