因果引擎 TM：让机器学习更智能

从相关性到因果性的革命

龚鹤阳

August 4, 2025

今天我们要讲什么?

1. 传统机器学习有什么问题?
2. 因果引擎 ™ 是怎么工作的?
3. 五种工作模式
4. 因果引擎的性能有多好?
5. 怎么使用因果引擎?
6. 理论基础（简单版）
7. 适用场景
8. 总结与展望

传统机器学习的局限性

传统机器学习做什么?

- 学习数据中的相关性
- 就像：看到乌云就说要下雨
- 但不知道为什么下雨

不知道里面发生了什么

问题出现了:

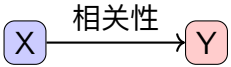
- 数据有噪音就不准了
- 每个人的差异被当作" 干扰"
- 只能告诉你" 是什么"，不知道" 为什么"

核心问题

现实世界的数据总是有噪音的，传统方法应付不了!

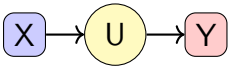
我们需要什么样的解决方案?

传统机器学习



- 简单直接
- 容易过拟合
- 噪音敏感

因果机器学习



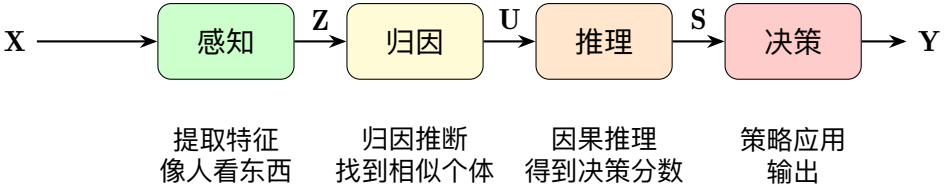
个体特征

- 理解机制
- 抗噪能力强
- 个性化预测

关键洞察

每个个体都有独特的特征 U，这不是噪音，而是有用的信息!

因果引擎 ™ 的四个步骤



核心思想

就像医生看病：先观察症状，推断病因，再对症下药！

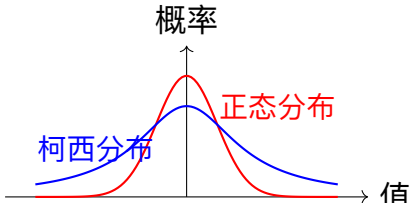
为什么选择柯西分布?

柯西分布的特点:

- 有" 重尾巴"——更好地处理极端情况
- 数学性质好——计算简单高效
- 更符合现实——人与人差异很大

打个比方:

- 正态分布: 大部分人都差不多
- 柯西分布: 承认有些人就是很特别

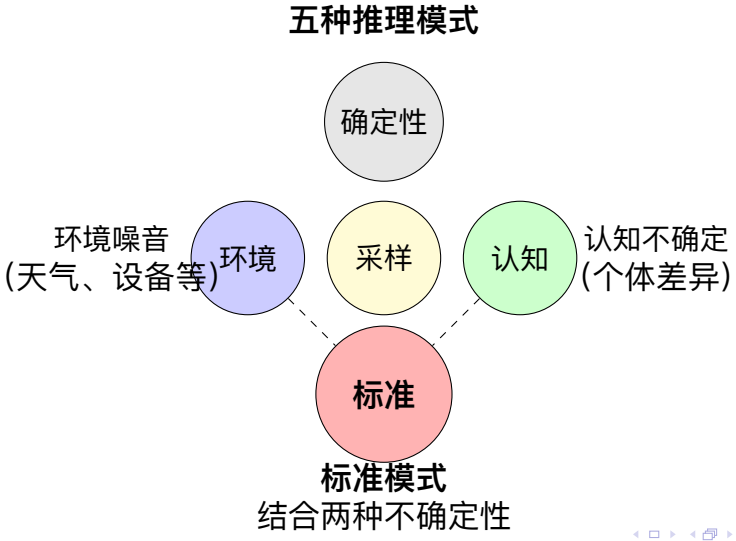


柯西分布的" 尾巴" 更长

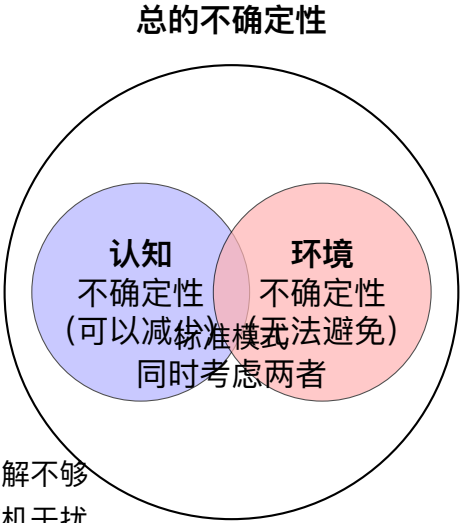
简单理解

柯西分布让我们的模型能更好地理解" 特殊" 的个体

五种推理模式



不确定性的分解

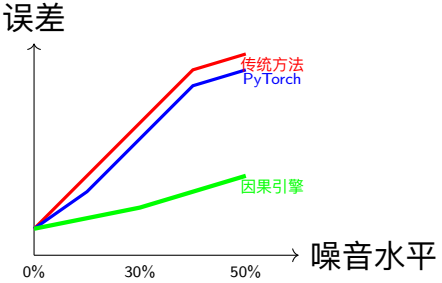


认知：我们对个体了解不够

环境：测量误差、随机干扰

抗噪音能力测试：回归任务

随着噪音增加，性能如何变化?



30% 标签噪音下的表现：

- 传统 MLP：误差 47.60
- PyTorch：误差 45.32
- **因果引擎：误差 11.41**

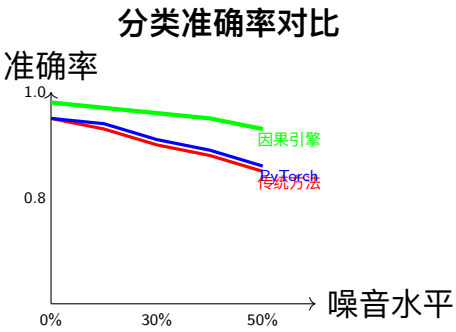
惊人提升

性能提升 76%!

关键发现

噪音越多，因果引擎的优势越明显!

抗噪音能力测试：分类任务



30% 标签噪音下的准确率：

- 传统 MLP：88.50%
- PyTorch：88.75%
- **因果引擎：92.25%**

显著改善

错误率降低 31%！

实际意义

在现实的噪音数据中，因果引擎提供更可靠的预测

安装和基本使用

安装很简单

```
pip install causal-sklearn
```

基本使用示例

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

# 生成数据
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)

# 创建和训练模型
```

高级配置

自定义网络结构

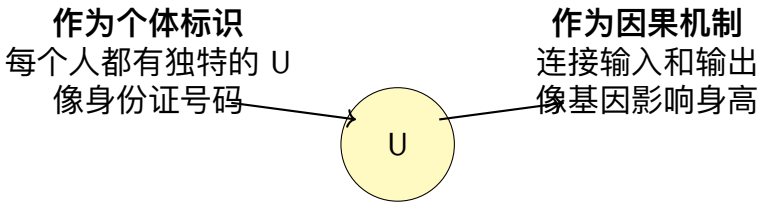
```
model = MLPCausalRegressor(
    # 感知层配置
    perception_hidden_sizes=(128, 64),
    perception_activation='relu',
    perception_dropout=0.2,

    # 推理层配置
    abduction_hidden_sizes=(32,),

    # 推理模式
    inference_mode='standard',

    # 训练配置
    learning_rate_init=0.001,
```

个体选择变量 U 的双重身份



统一数学框架

$$P(\text{输出}|\text{输入}) = \int P(\text{输出}|U) \cdot P(U|\text{输入})\, dU$$

$P(U|\text{输入})$：从观察推断原因　　$P(\text{输出}|U)$：从原因预测结果

简单理解

U 既是每个人的" 身份证"，又是连接原因和结果的" 桥梁"

为什么这种方法有效?

1. **正确的思维方式**
   - 个体差异是信息，不是噪音
   - 学习通用规律，而不是死记硬背
2. **量化不确定性**
   - 明确区分" 不知道" 和" 随机性"
   - 让模型知道自己" 不知道什么"
3. **数学优雅**
   - 柯西分布的线性稳定性
   - 可以直接计算，不需要复杂的采样
4. **实际有效**
   - 在多个真实数据集上验证
   - 在噪音环境中显著优于传统方法

什么时候用因果引擎?

特别适合的场景:

- ✓ 数据有很多噪音的时候
- ✓ 需要理解个体差异
- ✓ 医疗诊断（个性化）
- ✓ 金融风险评估
- ✓ 推荐系统
- ✓ 异常检测

优势不明显的场景:

- × 数据非常干净
- × 纯图像分类
- × 不需要解释性的场景
- × 计算资源极其有限

经验法则

当数据质量不确定或需要鲁棒性时，因果引擎是理想选择

真实数据集上的表现

数据集	传统 MLP	因果引擎	提升
加州房价	0.65	0.78	+20%
葡萄酒质量	0.55	0.71	+29%
波士顿房价	0.62	0.74	+19%
糖尿病	0.41	0.52	+27%

* 在 20% 标签噪音下的 R^2 分数

关键洞察

即使在中等噪音水平下，因果引擎也能提供显著的性能改善

核心贡献

1. **新的机器学习范式**
   - 从学习相关性到学习因果关系
   - 把个体差异当作特征，而不是噪音
2. **实用的实现**
   - 完全兼容 scikit-learn API
   - 高效的分析计算
   - 容易集成到现有工作流程
3. **出色的鲁棒性**
   - 在噪音环境中性能优异
   - 适合现实世界的混乱数据

一句话总结

因果引擎通过理解" 为什么" 而不仅仅是" 是什么"，为机器学习带来新的可能性

未来发展方向

- **理论扩展**
  - 扩展到其他概率分布
  - 多任务因果学习
  - 时间序列因果推断
- **应用拓展**
  - 大规模数据集优化
  - 与深度学习架构集成
  - 领域特定定制
- **工具生态**
  - 可视化工具
  - 自动超参数调优
  - 云部署支持

谢谢大家！

有问题可以讨论

让机器学习变得更智能