因果学习挑战赛代码与解决方案介绍

——中传智能音频与大数据团队

目录

环境和包的配置

环境：

- python3.8

包：

- ylearn
- causallearn
- pandas
- numpy
- sklearn
- matplotlib
- csv
- io
- random

复现方法

简单版：

使用压缩包中已附上的预处理后的 train_input.csv 和 test_input.csv，运行因果效应估计程序（causal_estimator.py）即可复现本工作。

完整版：

首先运行数据预处理文件（data_processing.py），运行时长约 3~4 分钟，得到 train_input.csv 和 test_input.csv，作为因果效应估计程序（causal_estimator.py）的输入，得到预测后的结果。

注：由于预处理中的随机森林算法存在一定的随机性，此方案生成的结果可能与提交的结果存在 0.01 以内的偏差。

程序介绍

数据预处理（data_processing.py）

功能：非数值型特征转为数值型，随机森林预测缺失数据

输入：原始数据 train.csv、test.csv

输出：train_input.csv、test_input.csv

因果发现（causal_graph.py）

功能：调用 causal-learn 中的工具包，进行数据分析，画出多种因果图

输入：train_input.csv、test_input.csv

输出：PC、FCI、CD-NOD、GRaSP、GES 五种方法得到的因果图

注意：需提前安装 causallearn 包

参数调整（parameters_identifier.py）

功能：寻找合适的混淆因子和因果效应估计参数

输入：train_input.csv、test_input.csv，同时需要自己设置希望研究的因子或参数选择范围

输出：输出为 treatment=1 或 2 时的平均治疗效应，保存至 test_V.csv 或 test_dml_parameters.csv 文件中

因果效应估计（causal_estimator.py）：

功能：根据数据预处理、因果发现、因子选择等结果，预测 treatment=0 或 1 时的因果效应

输入：train_input.csv、test_input.csv

输出：result.csv

解决方案介绍

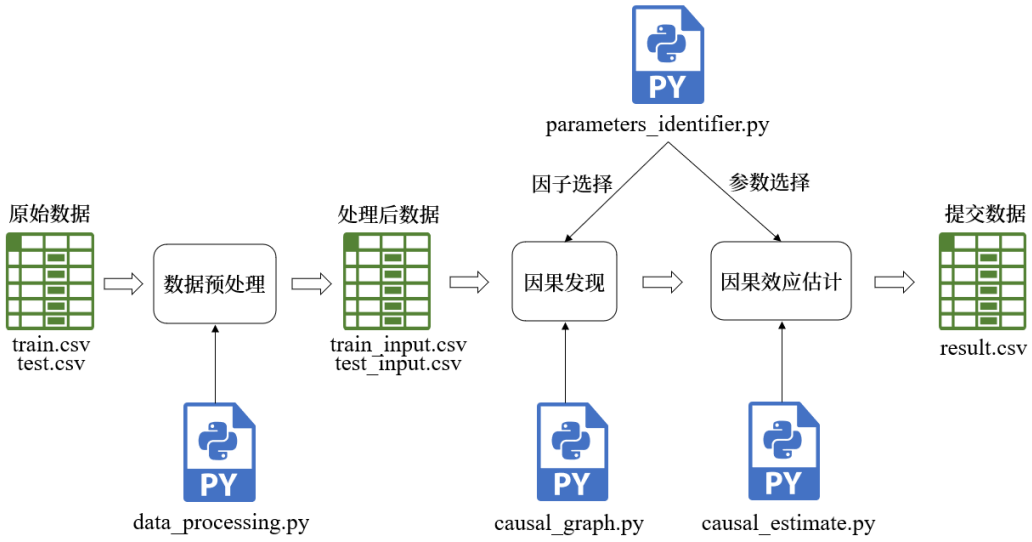

图 1 方案的技术路线

本团队的解决方案主要包括数据预处理、因果发现和因果效应估计三个部分，如图 1 所示。具体工作介绍如下。

1. 数据预处理

（1）数值转换

- 字符串类型定性转换：将文本类型的数据按类别转换为数值类型，具体的，V_8、V_10、V14、V_26 中的 yes 赋值为 1，no 赋值为 0
- 缺失项赋值：缺失值赋值，方便后续的类型转换和预测，赋值原则：选取原始特征列中不存在的数值，以便缺失值/非缺失值的区分，赋值大小并不影响预测效果，只用作区分缺失值。
- 类型转换：将非数值型转为数值型

（2）随机森林回归预测缺失值

- 构建新的特征矩阵：计算原始数据中所有特征与目标特征（含缺失项的特征）的相关系数，选择相关系数较高的特征构成新的特征矩阵。
- 数据划分：将目标特征划分为已知部分和需要预测的部分。
- 随机森林模型预测：根据已知数据拟合随机森林模型，用得到的模型进行未知部分的预测，用得到的预测结果填补原缺失数据。

2. 因果发现与因果效应估计

（1） 因果发现

本方案的因果发现过程使用了 causal-learn 工具，这是由 CMU 张坤老师主导，多个团队（CMU 因果研究团队、DMIR 蔡瑞初老师团队、宫明明老师团队和 Shohei Shimizu 老师团队）联合开发出品的因果发现算法平台。本方案使用 PC（Peter-Clark）、FCI（ Fast Causal Inference）、CD-NOD、GES（Greedy Equivalence Search）、GRaSP（Greedy relaxation of the sparsest permutation ）五种算法，分别绘制因果图，并固定了 treatment 到 outcome 的因果方向，根据绘制的因果图，我们选取了可以作为 confounders 的候选因子。

（2） 因果效应估计

根据赛题的评判标准，我们将 treatment=1 和 2 时的平均治疗效应（ATE）作为判断结果是否准确的指标。通过积累每日提交的结果与 NRMSE 得分之间的关系，判断出合理的 ATE（treatment=1）约为 1.3 左右，ATE（treatment=2）约为 13 左右，我们以此作为后续选取因子和参数的依据。

因果效应估计部分我们使用了 ylearn 的双机器学习（double mechine learning，

dml）模型。随机选取 confounders 列表中的元素，计算以它们为 confounders 条件下得到的 ATE（treatment=1）和 ATE（treatment=2），选取规定范围内出现次数最多的元素作为最终的 confounders 列表。比赛后期，考虑到样本的不平衡以及可能存在的样本选择性偏差，我们对 treatment=1 或 2 的情况分别进行估计。

确定 confounders 后，我们循环计算了不同 dml 参数下模型得到的 ATE 结果，选取最合理的参数作为最终模型的参数，对预处理后的模型进行因果估计，并得到最终的结果。

其他

由于程序中包含随机森林回归和随机森林分类等具有随机性的算法，程序结果可能出现一定的波动，经测试，在不同设备上误差不超过 0.01，我们在代码中的相应位置进行了标注与说明。