软件学报 ISSN 1000-9825, CODEN RUXUEW
[doi: 10.13328/j.cnki.jos.007473] [CSTR: 32375.14.jos.007473]

E-mail: jos@iscas.ac.cn
http://www.jos.org.cn
Tel: +86-10-62562563

扩散模型引导的根因分析*

王浩天[1], 周学广[2], 王尚文[1], 靳若春[1], 黄万荣[1], 杨文婧[1], 王 戟[1]

[1](国防科技大学 计算机学院, 湖南 长沙 410073)

[2](海军工程大学 信息安全系, 湖北 武汉 430033)

通信作者: 王尚文, E-mail: wangshangwen13@nudt.edu.cn

摘 要: 根因分析是指找出引起复杂系统异常故障的根源因素. 基于因果关系的溯因方法基于结构因果模型, 是实现根因分析的最优选择之一. 目前大多数因果驱动的根因分析方法大都需要数据因果结构的发现作为前置条件, 这使得根因分析本身严重依赖于因果发现这一先验任务的效果. 最近, 基于得分函数的干预识别受到了广泛关注, 其通过对比干预前后的得分函数导数的方差来检测被干预的变量集合, 具备突破因果发现对根因分析约束的潜力. 然而, 主流的基于得分函数的干预识别大都受限于得分函数估计这一步骤, 其采用的解析求解方法并不能很好地对真实的高维复杂数据分布进行建模. 因此, 鉴于最近在数据生成中取得的进展, 提出一种扩散模型引导的根因分析策略. 具体来说, 所提方法首先利用扩散模型针对异常发生前后的数据分布对应的得分函数进行估计, 进而通过观察对加权融合后的总体得分函数的一阶导方差, 识别导致异常发生的根因变量集合. 此外, 为了进一步减小在识别过程中剪枝操作带来的扩散模型重复训练的开销, 提出一种可靠的估计策略, 其只需要训练一次扩散模型即可估计所有剪枝过程中对应节点的得分函数. 在仿真数据和真实数据上的实验结果表明, 所提出的方法实现了对于根因变量集合的精准识别. 此外, 相关的消融实验也表明, 扩散模型的引导作用对于表现提升至关重要.

关键词: 根因分析; 扩散模型; 结构因果模型

中图法分类号: TP311

中文引用格式: 王浩天, 周学广, 王尚文, 靳若春, 黄万荣, 杨文婧, 王戟. 扩散模型引导的根因分析. 软件学报. http://www.jos.org.cn/1000-9825/7473.htm

英文引用格式: Wang HT, Zhou XG, Wang SW, Jin RC, Huang WR, Yang WJ, Wang J. Diffusion-model-guided Root Cause Analysis. Ruan Jian Xue Bao/Journal of Software (in Chinese). http://www.jos.org.cn/1000-9825/7473.htm

Diffusion-model-guided Root Cause Analysis

WANG Hao-Tian[1], ZHOU Xue-Guang[2], WANG Shang-Wen[1], JIN Ruo-Chun[1], HUANG Wan-Rong[1], YANG Wen-Jing[1], WANG Ji[1]

[1](College of Computer Science, National University of Defense Technology, Changsha 410073, China)

[2](Department of Information Security, Naval University of Engineering, Wuhan 430033, China)

Abstract: Root cause analysis refers to identifying the underlying factors that lead to abnormal failures in complex systems. Causal-based backward reasoning methods, founded on structural causal models, are among the optimal approaches for implementing root cause analysis. Most current causality-driven root cause analysis methods require the prior discovery of the causal structure from data as a prerequisite, making the effectiveness of the analysis heavily dependent on the success of this causal discovery task. Recently, score function-based intervention identification has gained significant attention. By comparing the variance of score function derivatives before and after interventions, this approach detects the set of intervened variables, showing potential to overcome the constraints of causal discovery in root cause analysis. However, mainstream score function-based intervention identification is often limited by the score function estimation

* 基金项目: 国家自然科学基金 (62032024, 62372459, 62276273, 62302503); 国防科技大学青年自主创新科学基金 (ZK23-15); 高性能计算国家重点实验室自主开放课题 (202401-09)
收稿时间: 2024-11-09; 修改时间: 2025-04-21; 采用时间: 2025-05-26; jos 在线出版时间: 2025-09-28

step. The analytical solutions used in existing methods struggle to effectively model the real distribution of high-dimensional complex data. In light of recent advances in data generation, this study proposes a diffusion model-guided root cause analysis strategy. Specifically, the proposed method first estimates the score functions corresponding to data distributions before and after the anomaly using diffusion models. It then identifies the set of root cause variables by observing the variance of the first-order derivatives of the overall score function after weighted fusion. Furthermore, to solve the issue of computational overhead raised by the pruning operation, an acceleration strategy is proposed to estimate the score function from the initially trained diffusion model, avoiding the re-training cost of the diffusion model after each pruning operation. Experimental results on simulated and real-world datasets demonstrate that the proposed method accurately identifies the set of root cause variables. Furthermore, ablation studies show that the guidance provided by the diffusion model is critical to the improved performance.

Key words: root cause analysis; diffusion model; structural causal model (SCM)

根因分析 (root cause analysis, RCA) 指的是结合异常发生前后的数据, 对系统中引起异常的根本原因进行定位[1–5], 在识别引起系统故障的起因中起着至关重要的作用. 事实上, 现实世界中的复杂系统出现故障, 往往会造成严重的影响[4]. 如图 1(a) 所示, 为了确保云服务系统的可靠性和稳健性, 系统维护者通常会收集和分析关键性能指标如延迟、CPU/内存使用等度量数据和日志数据, 以定位可能出错的原因[3]. 然而, 大型复杂系统的复杂性和大量的监控数据使得手动分析根因变量这一过程既昂贵又容易出错. 因此, 如何实现快速有效的根因分析对于快速恢复系统服务、减少异常损失、确保大型复杂系统的持续运行至关重要[6–8].

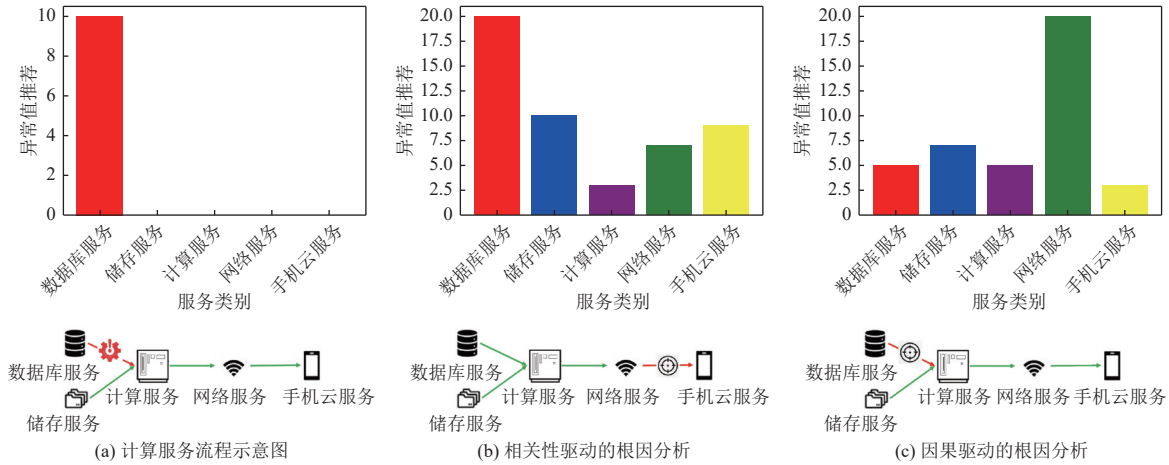

图 1 云计算服务的根因分析示意图

相关性驱动的异常检测和异常归因理论面向根因分析做出了一定的贡献, 例如基于异常值或者夏普利 (Shapley) 归因值来进行根因变量的排序和推荐[9]. 然而，基于相关关系的异常溯因方法往往会倾向于定位直接引起异常变量的因素变量, 而无法识别可能位于上游的根节点变量, 如图 1(b) 所示[10]. 与之相对地, 随着因果推断在社会科学和计算机科学的广泛发展, 基于因果关系的根因分析在近些年逐渐受到了研究社区的重视和关注[1–4,10,11]. 具体来说, 因果驱动的根因分析 (因果根因分析) 方法基于结构因果模型[12], 将异常的发生建模为某种形式的干预[3,10], 例如外源变量干预[10]和软干预[3]等, 进而异常发生前后的数据可以被定义为观测数据[12]和干预后数据[12]. 鉴于对因果结构的考虑以及变量间因果关系的建模, 因果驱动的根因分析方法相较于相关驱动的根因分析, 具备了识别上游根源节点的潜力, 如图 1(c) 所示.

进一步来说, 因果根因分析方法 (见图 1(c)) 大都遵循了因果骨架学习-根因变量定位这一两阶段的技术框架和技术路线[5,7,8]. 在第 1 阶段中, 因果根因分析旨在结合观测数据 (异常发生前) 和干预后数据 (异常发生后) 来识别数据对应的因果图结构. 换言之, 这一阶段的任务是结合多环境数据进行因果发现[13–15]. 基于第 1 阶段发现的因果图 (或者结构方程[10]), 因果根因分析进而利用随机游走[1]、条件独立性检测[3]、Shapley 值[9,10]等技术来实现根

因变量的定位. 尽管两阶段的因果根因方法在实现根源异常定位上取得了显著的优势和进展, 其自身固有的技术缺陷同样十分明显. 第一, 两阶段的根因分析方法设计本身包含了许多冗余且没必要的步骤. 具体来说, 根因分析和因果发现两个任务本质上的目标不一致, 前者旨在找到被干预的变量集合, 而后者旨在识别整个因果图, 即变量之间的边集合. 第二, 根因定位严重依赖于第 1 阶段因果发现的表现, 而到目前为止因果发现任务本身面向复杂数据的表现稳定性和可验证性都有待提升[13,16].

鉴于上述的动机, 本文旨在探索不依赖因果发现的高效端到端根因分析范式. 为了实现这一目标, 本文注意到近几年研究社区在结合多环境的干预变量识别任务中取得了一些突破[17−20]. 其主体思想是通过构造数据的特定统计量, 建立起统计量变化和跨环境干预之间的联系, 进而无需发现因果图直接识别干预变量. 然而, 这些已有方法大都围绕数据变量之间全线性关系的情况来进行设计[17,18], 很难满足现实世界复杂系统中蕴含非线性的数据分析需求. 为了克服这一难题, 研究者在文献 [20] 中, 围绕非线性的情况, 建立了变量得分函数和变量是否被干预之间的关联性. 美中不足的是, 这一研究严重依赖于通过斯坦等式[21]来实现数据分布得分函数的估计. 鉴于核方法驱动的斯坦估计[22]这一技术本身的复杂度随着样本量呈现出平方增长的趋势, 已有的非线性干预识别策略难以泛化到具备海量样本量的复杂现实系统根因分析问题中. 为了克服上述挑战, 本文面向非线性复杂系统, 提出了一种基于去噪扩散生成模型的根因分析策略 (diffusion-empowered root-cause analysis, DERCA). 具体来说, 本文基于训练好的扩散去噪模型, 实现了对数据分布得分函数的高效、精准估计. 进而, 如图 2 所示, 基于估计得到的得分函数和其二阶雅克比矩阵, 本文所提出的 DERCA 方法通过两条迭代的判定原则: (1) 判定变量在异常前/异常后数据中得分函数一阶导方差是否为 0, 进而判定其是否为叶节点变量; (2) 判定对应叶子节点变量的跨环境得分函数一阶导方差是否为 0, 进而判定其是否为根因变量; 来实现全部根因变量的识别和定位.

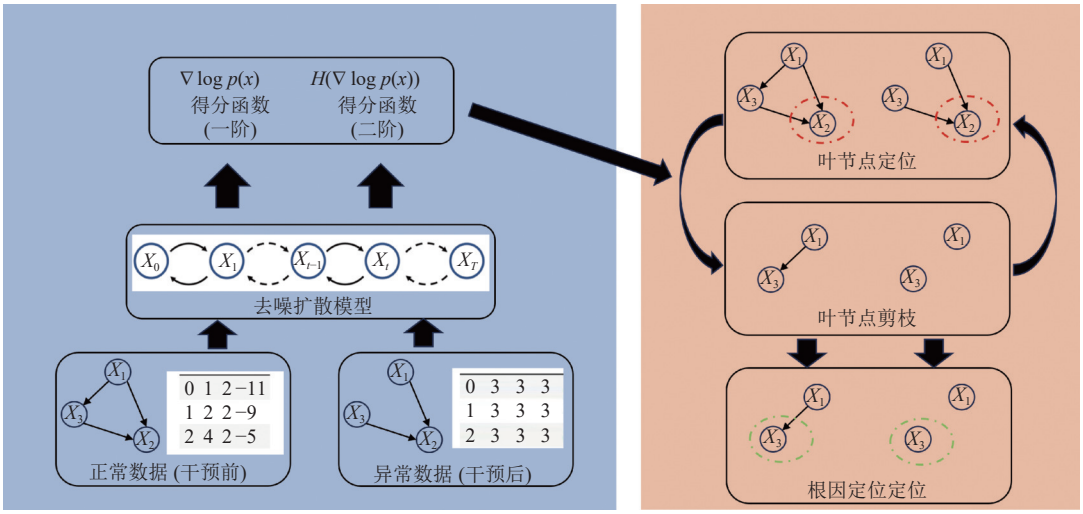

图 2 DERCA 方法框架示意图, 其中 $H(\cdot)$ 表示对应向量的 Hessian 矩阵

本文的主要贡献可以分为如下 3 个方面: (1) 本文通过得分函数直接建立起了变量干预和数据分布统计量之间的关联, 提出了首个不依赖因果发现且和样本数成线性复杂度的高效根因分析框架; (2) 本文通过去噪扩散模型, 实现了对于得分函数的高效估计. 和前面工作面向样本数的平方复杂度不同 (例如 iSCAN[20]方法), 本文所提出方法随样本数的线性复杂度可以支持具备海量样本的复杂系统根因分析; (3) 本文在仿真数据和真实数据上的实验验证了所提出策略的有效性和精确性.

1 根因分析以及相关工作

本文所涉及内容主要与根因分析和干预识别有关, 下面就相关研究现状给予介绍.

1.1 根因分析

根因分析指的是通过结合系统异常前后的数据, 揭示系统异常行为的过程[1]. 迄今为止, 根因分析主要的流派可以归纳为两类, 即先验知识完备的领域根因分析[23−27]和基于因果发现的根因定位[1,3,4,10,11]. 其中先验完备的领域根因分析主要针对在先验的因果结构已知的情况下, 如何定位引起异常故障的根因变量. 这一系列的研究内容大都面向具体的任务领域, 例如云服务系统[26]或者智能办公楼系统[24]. 然而, 并不是所有的领域, 或者一个领域内的所有子任务都可以获取到先验的因果图知识, 而上述领域根因方法在这样的应用场景中往往都会立即失效. 因此, 基于因果发现的根因定位通过两阶段的学习, 旨在先验因果知识不完备的时候学习相应的因果结构, 进而基于学习到的因果结构进行根因变量的识别和定位. 例如, 一种基于递归划分的 PC 拓展算法被提出[3], 其结合异常发生的前后数据, 通过进行节点增广和经典 PC 算法[13]的推广, 实现了高效的根因变量识别. 又如, 研究者在文献 [10] 中提出通过识别先验的结构因果模型 (structural causal model, SCM) 和结构不变的外源噪声干预操作, 来实现根因变量的识别. 最近, 面向实时增量数据流的根因分析[11]和面向跨模态数据的根因分析[1]相继被提出, 其通过深度学习的策略进行因果图的搜索, 并结合具体的下游任务场景上下文信息来实现具体的根因变量识别. 综上所述, 截止到目前的根因分析的基础思路是一致的, 即先进行因果图的获取 (先验知识或者因果发现), 再实现对应根因变量的识别定位. 然而, 本文注意到, 这一思路在实际应用会存在非常大的问题. 首先, 因果发现这一任务本身和根因分析的目标是不一致的, 前者的目标是在给定节点的情况下识别边的方向 (或者强度[28]), 而后者旨在识别因果图节点集合的一个子集 (因果图本身不是必须的). 这一技术路线的设计既不符合最小化的原则, 同样也会引入因果发现任务本身带来的误差传播. 其次, 因果发现这一前置任务本身仍处于探索的阶段, 无论从先验的可信赖假设到识别图的完备性 (马尔可夫等价类[13]) 再到验证评估难以实施这一问题[13], 其任何的误差都会传播到第 2 阶段, 即基于因果图的根因定位 (随机游走[4]或者马尔可夫链[11]) 本身严重依赖于上游因果图学习的精度. 综合上述两点缺陷, 本文旨在提出首个不依赖于因果发现的根因分析方法.

1.2 干预识别

干预识别指的是因果推断中的一类子任务, 其旨在结合干预前后的数据, 跨环境实现被干预节点集合的识别[14,17,18,20,29,30]. 其中, 贪婪搜索算法 (greedy equivalence search, GES)[13]被改进到干预-观测数据结合的多环境下[29], 而基于约束的因果发现框架在文献 [14] 中被推广到干预-观测联合数据的场景中, 并给出了干预变量识别的完备性理论. 然而, 基于约束的方法在面向高维数据以及样本量增加的场景下, 其性能表现会受制于条件独立性测试这一技术瓶颈[13]. 近两年, 部分工作尝试通过构造特定的统计量来识别变量是否被干预[14,18,20]. 具体来说, 在线性 SCM 模型的假设下, 一种基于精度矩阵 (变量协方差矩阵的逆) 变化的策略被提出[17]. 进而, 这一方法被推广到潜在混杂存在的情况下[30]. 为了克服线性数据这一假设的限制, iSCAN 方法在文献 [20] 中被提出, 其通过构建得分函数一阶导方差这一统计量, 用于识别变量是够跨环境被干预. 然而, iSCAN 方法的主要缺陷在于通过经典的斯坦等式 (Stein identify)[21]和基于核方法的斯坦估计[22]对于数据得分函数的估计. 这一方法缺乏对于复杂非线性数据的估计能力, 且由于具备样本量平方复杂度, 难以泛化到真实的大型数据集上. 本文面向得分函数一阶导方差这一统计量, 基于扩散模型的得分函数估计, 实现面向大规模数据的根因分析.

2 背景知识

2.1 结构因果模型

本节引入符号并正式定义问题设定. 我们使用 $[d]$ 来表示整数集合 $1,\ldots,d$. 此外, 本文用 $G=<[d],E>$ 来表示有向无环图 (directed acylic graph, DAG), 其中 $[d]$ 表示节点集合, $E\subset[d]\times[d]$ 表示了有向边的集合, 且 $(i,j)\in E$ 表示从节点 i 到节点 j 的一条边. 此外, 本文令 $X=(X_1,\ldots,X_d)$ 表示一个 d 维度的随机向量. 基于上述符号, 一个作用于 $X=(X_1,\ldots,X_d)$ 上的结构因果模型 (SCM) $\mathcal{M}=(X,f,P_U)$ 通常可以被定义为由以下形式的 d 个结构方程组成的集合 (对 $\forall j\in[d]$):

$$X_j=f_j(Pa_j,U_j),$$

其中, $Pa_j \subseteq \{X_1,\ldots,X_d \backslash X_j\}$ 表示 X_j 的直接原因集合 (即对应因果图中的父节点变量), U_j 代表了每个变量 X_j 对应的外源噪声变量. $f=\{f_j\}_{j=1}^{d}$ 代表了因果结构方程的集合 ($f_j: R^{|Pa_j|+1} \to R$); P_N 是噪声变量 U_j 的联合分布. 按照以往的惯例, 本文假设系统运维者对于系统本身的观察是充分的, 即不存在未观测的混杂变量 (马尔可夫假设), 因此 U_j 之间彼此独立. 此外, 结构因果模型 $\mathcal{M}$ 对应的因果图 G 可以通过为对每个变量 X_j, 从其对应的父节点变量集合 Pa_j 向 X_j 画一条有向边来实现. 鉴于马尔可夫假设, 可以立刻得到构建出来的图 G 一定是有向无环图.

2.2 数据分布和结构因果模型的映射

最后, 每个结构因果模型 $\mathcal{M}$ 定义了变量 X 上的唯一分布 P_X [12], 并且由于噪声变量的彼此独立性 (即马尔可夫假设), P_X 可以被进一步分解为如下形式:

$$P(X)=\prod_{j=1}^{d} P(X_j|Pa_j) \tag{1}$$

其中, $P(X_j|Pa_j)$ 又被称为 X_j 对应的因果机制. 然而, 鉴于广义的 SCM 存在两个不同的结构方程对应同一个数据分布这一反例现象的存在[12], 需要对 SCM 本身添加一定的、合理的约束才能够进行相关因果统计量的识别和分析[20,28,29]. 本工作将考虑一种被广泛采纳并且使用的 SCM 模型, 即加性噪声模型 (additive non-linear model, ANM).

定义 1. 加性噪声模型. 加性噪声模型是满足如下表达的 SCM 模型:

$$X_j = f_j(Pa_j)+U_j,\ \forall j \in [d].$$

基于对 f_j 和 U_j 进一步的一些参数化假设, ANM 对应的有向无环图可以通过观测数据进行识别. 例如, 当满足可信赖假设, 且 f_j 是线性的, U_j 是非高斯噪声时, 通常可以识别到对应得 DAG[28]. 又例如, 当 f_j 是线性且每个分量是三次可微分的, 那么潜在的 DAG 同样可以被识别[31,32].

针对加性噪声模型的进一步解释: 加性噪声模型在因果推断领域是一种较为通用的非线性 SCM 假设[20,28,29,33]. 这一假设放松了经典统计学中, 关于线性因果模型假设所带来的约束. 而外源噪声项以加性的形式引入这一假设, 同样带来了一定的限制. 但是值得注意的是, 这并非意味着加性模型不允许噪声项的非线性组合. 换言之, 诸多例如对等式两边取对数等的操作可以将加性噪声模型变换为 $f_j(Pa_j)$ 和 U_j 的非线性组合. 本文的后续实验将从加性噪声分别满足和不满足的角度进行进一步经验性的论证.

2.3 异常建模: 软干预和因果机制的变化

最后, 本文通过干预的形式为异常发生这一现象进行建模. 遵循之前的文献和研究经验, 不同场景下异常的发生可以被定义为不同的干预类型, 例如污水排放系统中的外源噪声干预[10]或者云服务器系统中的软干预[3]. 本文采用软干预这一概念来建模异常行为, 即被干预节点变量 X_j 的因果机制被改变, 而剩余变量节点的因果机制保持不变. 具体来说, 本文用 $X^n=\{X^{n,i}\}_{i=1}^{m_n}$ 和 $X^c=\{X^{c,i}\}_{i=1}^{m_c}$ 表示异常发生前和异常发生后的数据, 其中 m_n 和 m_c 分别表示正常数据和异常数据的样本量. 基于上述概念, 本文将异常发生后的数据, 即 $X^c=\{X^{c,i}\}_{i=1}^{m_c}$, 对应的分布刻画如下:

$$P^c(X)=\prod_{j\in[d]} P^c(X_j|Pa_j)=\prod_{j\in Inv} P^c(X_j|Pa_j)\prod_{j\notin Inv} P^n(X_j|Pa_j),$$

其中, Inv 表示被干预的节点变量集合. 此外, 令 G^c 和 G^n 分别表示干预后和干预前的因果图. 从上述分布可以看出, 异常发生前和异常发生后的数据分布发生变化的部分仅落在被干预节点的因果机制, 即 $P^c(X_j|Pa_j)\neq P^n(X_j|Pa_j)$ 对所有 Inv 中的节点成立. 在本文中, 有的时候会交替使用异常发生前 (观测数据) 和异常发生后 (干预数据) 这两组概念. 基于上述背景知识, 下面给出软干预和根因分析的形式化定义.

定义 2. 软干预. 某个节点变量 X_j 被软干预, 指的是其给定 Pa_j 的条件分布在干预前后发生了变化: $P^c(X_j|Pa_j)\neq P^n(X_j|Pa_j)$.

定义 3. 根因分析. 给定异常发生前和异常发生后的数据, 即 $X^n=\{X^{n,i}\}_{i=1}^{m_n}$ 和 $X^c=\{X^{c,i}\}_{i=1}^{m_c}$, 根因分析旨在识别集合 $Inv=\{X_j|P^c(X_j|Pa_j)\neq P^n(X_j|Pa_j),\ j\in[d]\}$.

为了方便阅读, 我们将本文涉及的相关重要符号说明以表 1 的形式进行梳理.

进而, 结合本文对根因分析问题的定义, 这里对 SCM 做出参数化的一些假设和限制, 以确保可识别性的成立.

假设 1. 非线性可微 SCM. 本文假设, 对于 SCM 中的每个结构函数 f_j, 其满足如下两个性质:

(1) f_j 在每个分量上是非线性的.

(2) 关于每个外源噪声变量 U_j, 其分布密度函数 $p^n_{N_j}$ 关于 U_j 值域的每个取值 u_j 有 $\frac{\partial^2}{(\partial u_j)^2}\log p^n_{N_j}(u_j)$ 等于某个和 n, j 相关的常数. 类似的, $\frac{\partial^2}{(\partial u_j)^2}\log p^c_{N_j}(u_j)$ 同样等于某个和 n, j 相关的常数.

本文注意到, 上述的假设在过去的许多文献中也同样被提出, 例如文献 [17,20,33].

表 1 本文重要符号说明表

符号	含义
X	样本矩阵
P	概率密度函数
G	因果图
$s(X)=\nabla_X \log p(X)$	得分函数
f	结构函数
n	样本数量
d	特征数量
H	海森矩阵
J	雅可比矩阵
$\mathcal{M}$	扩散模型
δ	得分函数的变化

2.4 去噪扩散模型

最初的扩散模型概念[34], 旨在往一个原始的数据分布 $x_0 \sim p_{\text{data}}(x)$ 中按照时间步添加呈现正态分布的噪声变量 $x_t = \mathcal{N}(0,(1-\alpha_t)I)$, 即 $q(x_t x_{t-1}) = \mathcal{N}\left(\sqrt{\alpha_t}x_{t-1},(1-\alpha_t)I\right)$, 这里 q 代表了前向过程的分布, α_t 代表了时间相关的某个标量, I 是单位矩阵, $\mathcal{N}$ 是正态分布. 前向过程的目标是将原始的数据分布 p_{data} 通过不断添加高斯噪声, 最终变成一个高斯分布. 和变分自编码器的逻辑类似[35], 扩散模型将 T 时间步后形成的噪声分布类比成隐变量分布, 并通过逆向过程来从 x_T 还原出 x_0. 基于扩散模型 (diffusion probabilistic model, DPM) 的概念, 去噪扩散模型 (denoising DPM, DDPM) 优化了噪声变量的学习过程, 提出了更加稳定、更加平滑的优化目标. 具体来说, DDPM 模型[36]通过在每一时刻从标准高斯分布 $\epsilon = \mathcal{N}(0,1)$ 中随机采样一个噪声, 代入 DPM 的前向过程中计算 $x_t = \sqrt{\alpha_t}x_\theta + \sqrt{1-\alpha_t}\epsilon$. 进而, DDPM 优化生成噪声的一致性:

$$\theta^* = \text{argmin}E_{x_0,t,\epsilon}\left[\lambda(t)\epsilon_\theta(x_t,t)-\epsilon_2^2\right],$$

其中, $\epsilon_\theta(x_t,t)$ 代表了神经网络的输出, θ 代表了神经网络的参数, $t \sim \mathcal{U}(o,T)$, $\lambda(t)$ 是某个依赖于时间步的加权函数. 进一步的, 基于分数匹配的扩散模型 (DDIM)[37]通过估计和匹配前向-逆向过程中得分函数的一致性来取代采样的一致性:

$$\theta^* = \text{argmin}E_{x_0,t,\epsilon}\left[s_\theta(x)-\nabla_x \log p_{\text{data}}(x)_2^2\right] \tag{2}$$

如果将对数概率密度方向类比成场的概念, 那么沿着采样过程中的分数, 即增长最快的场方向走, 就可以收敛到数据分布的高概率密度区域, 因此最终生成的样本一定符合数据的原始分布.

3 扩散模型引导的根因分析

3.1 得分函数支撑的因果干预识别

在文献 [33] 中, 得分函数首次被提出用于因果结构识别. 在本文中, 用 $s(X)$ 来表示随机向量 X 的得分函数向量, 且通过上标区分其在不同数据环境下 (n/c) 得到的得分函数, 用下标区分 $s(X)$ 的不同分量, 例如 $s(X)^n_j$ 表示在正常数据分布下得到的得分函数的第 j 个分量. 具体来说, 在 ANM 的 SCM 模型下, 得分函数具备如下的推导性质:

$$s_j(X)=\nabla_{X_j}\log p(X)=\nabla_{X_j}\log\prod_{i=1}^{d}p(X_i\mid Pa(X_i))=\nabla_{X_j}\sum_{i=1}^{d}\log p(X_i\mid Pa(X_i))=\nabla_{X_j}\sum_{i=1}^{d}\log p^{N_i}(X_i-f_i(Pa_i))$$
$$=\frac{\partial\log p^{N_i}(X_j-f_j(Pa_j))}{\partial X_j}-\sum_{i\in Ch(X_j)}\frac{\partial f_i(Pa_i)}{\partial X_j}\frac{\partial\log p^{N_i}(X_i-f_i(Pa_i))}{\partial X},$$

其中, $\nabla_{X_j}\log p(X)$ 代表了得分函数这一统计量的第 j 分量, $Ch(X_j)$ 表示节点 X_j 的子节点变量集合. 值得注意的是得分函数 $\nabla_X\log p(X)\in R^d$ 本身是一个向量函数. 进而, 基于上述的推导, 引理 1 可以桥接起得分函数和数据内生的因果结构.

引理 1. 现将变量在因果图中的结构和其得分函数的方差关系表达如下:

$$\begin{cases}X_j\text{在 }G^n\text{中是叶子节点, 如果}Var_X\left[\dfrac{\partial s_j^n(X)}{\partial X_j}\right]=0\\ X_j\text{在 }G^c\text{中是叶子节点, 如果}Var_X\left[\dfrac{\partial s_j^c(X)}{\partial X_j}\right]=0\end{cases}.$$

基于引理 1, 文献 [20] 进一步提出跨多环境数据下, 干预变量集合的识别引理.

引理 2. 令 $q(x)=w_c p^c(x)+w_n p^n(x)$ 为干预-观测数据分布的混合分布密度函数, 其中 w_c 和 w_n 分别为两个分布的权重. 进而, 令 $s_j(X)=\nabla_{X_j}\log q(X)$ 为混合密度函数的得分函数. 基于假设 1, 如下结论成立[20]:

$$\begin{cases}Var_X\left[\dfrac{\partial s_j(X)}{\partial X_j}\right]>0\Rightarrow X_j\text{在}G^n\text{和}G^c\text{中都是叶子节点且}X_j\in Inv\\ Var_X\left[\dfrac{\partial s_j^n(X)}{\partial X_j}\right]>0\cup Var_X\left[\dfrac{\partial s_j^c(X)}{\partial X_j}\right]>0\Rightarrow X_j\text{至少在一个因果图中是叶子节点}\end{cases}\tag{3}$$

上述框架的最大挑战之处在于对于分布得分函数, 即 $s_j(X)=\nabla_{X_j}\log q(X)$ 的估计. 这一估计的难度来源于对分布密度函数的建模. 尽管文献 [20] 本身对于得分函数的求解提出了基于核函数的斯坦估计方法, 其可以视为下面斯坦等式的近似:

$$\begin{cases}E_q\left[h(x)\nabla\log q(x)^{\top}+\nabla h(x)\right]=0\\ E_q\left[h(x)Diag(\nabla^2\log q(x))^{\top}\right]=E_q\left[\nabla^2_{\text{diag}}h(x)-h(x)Diag(\nabla\log q(x)\nabla\log q(x)^{\top})\right]\end{cases}.$$

其中, 函数 $h(x)$ 具备性质 $\lim\limits_{x\to\infty}h(x)q(x)=0$. 然而, 对上述等式的核估计[22]的复杂度是 $\mathrm{O}(m^2)$, 其中 m 为对应数据源的样本数量. 与此同时, 在面向非线性的复杂数据的时候, 基于核方法的估计并不能够很好地刻画数据分布本身.

3.2 基于扩散模型的海森矩阵估计

回顾引理 1 中项 $Var_X\left[\dfrac{\partial s_j(X)}{\partial X_j}\right]=0$, 计算得分函数一阶导数的方差实际上可以等价于求得分函数对应的海森矩阵的对角元素: $Var_X\left[\dfrac{\partial s_j(X)}{\partial X_j}\right]=Var_X\left[H(\log p(x))_{j,j}\right]$, 这里 $H(\log p(x))_{j,j}$ 表示海森矩阵的对角元素. 因此, 本文首先从扩散模型本身的性质出发, 证明扩散模型所得到的估计量本身足以支撑对得分函数海森矩阵的估计. 值得注意的是, 鉴于混合分布的得分函数 $s_j(X)$ 和 $s_j^n(X)$ 以及 $s_j^c(X)$ 的海森矩阵的估计逻辑是一样的, 本文只重点阐述如何利用扩散模型估计 $s_j(X)$ 的得分函数. 基于分部积分的方式, DDIM 的优化目标公式 (2) 可以被进一步化简为: $\arg\min\limits_{\theta}\dfrac{1}{2}E_{p_{\text{data}}(x)}\left[\text{tr}(J_x s_\theta(x))+\dfrac{1}{2}s_\theta(x)_2^2\right]$, 这里 $J_x s_\theta(x)$ 表示估计得到的得分函数 $s_\theta(x)$ 的雅克比矩阵, 而 tr 表示计算矩阵的迹. 鉴于 $J_x s_\theta(x)$ 的计算开销在训练的时候过于高昂, 本文通过引入去噪得分匹配 (denoising score matching, DSM)[38]来进一步优化雅克比矩阵. 具体来说, 本文通过加入足够小的噪声变量 $q_\sigma(\tilde{x}|x)\sim N(\tilde{x};x,\sigma^2 I)$ 对原始数据分布做扰动, 则加噪后的数据分布可以表达为 $q_\sigma(\tilde{x})=\int q_\sigma(\tilde{x}|x)\cdot p_{\text{data}}(x)\mathrm{d}x$. 进一步地, 如果添加的噪声足够小, 那么 $q_\sigma(\tilde{x})\approx p_{\text{data}}(x)$, 进而可以得到如下的替代优化表达:

$$\arg\min_{\theta}\frac{1}{2}E_{q_\sigma(\tilde{x}|x)p_{\text{data}}(x)}\left[s_\theta(\tilde{x})-\nabla_{\tilde{x}}\log q_\sigma(\tilde{x}|x)_2^2\right].$$

鉴于 DDIM 本身通过匹配得分函数来进行原始分布的拟合, 本文通过构建经 DDIM 训练好的模型, 在其推理阶段估计得到的得分函数的二阶导矩阵来进行对应海森矩阵元素的估计: $H(\log p(x))_{j,j} \approx J_x s_\theta(x)_{j,j}$. 值得注意的是, 由于训练阶段采用了 DSM 的方法, 原始雅克比矩阵的估计只需要在推理阶段做一次即可. 然而, 鉴于 iSCAN 中提出的判定原则公式 (3) 需要重复判定共同存在于 G^c 和 G^n 中的叶子节点, 每次剪枝之后重复雅克比矩阵的估计在特征变量数量较大的时候引起的开销仍然十分高昂. 鉴于上述考虑, 本文接下来设计了一种面向叶子节点剪枝的策略, 使得一次性面向全部节点估计出的海森矩阵值可以泛化到每一次剪枝操作之后的估计中.

3.3 剪枝前后的海森矩阵高效估计

首先, 基于 DAG 对应的数据分布可以被如 $P(X)=\prod_{j=1}^{d} P(X_j \mid Pa_j)$ 一样进行分解, 假设节点 X_k 被检测为跨干预前后的共同叶子节点变量, 那么对 X_k 剪枝后的数据分布可以写作如下.

引理 3. 剪枝后分布. 对 X_k 剪枝后的数据分布有如下表达:

$$p(x_{-k}) = \prod\nolimits_{j\in[d]\setminus k}^{d} P(X_j Pa_j).$$

进而, 本文推导出, 可以基于原始联合分布的得分函数 $s(x)=\log p(x)\in\mathcal{R}$ 来推导出剪枝后联合分布的得分函数 $s(x_{-k})=\log p(x_{-k})\in\mathcal{R}^{d-1}$.

引理 4. 差量表达. 剪枝前后的数据分布得分函数有如下表达关系:

$$\delta_j = s_j(x) - s_j(x_{-k}) = -\frac{\partial f_l(Pa_i)}{\partial X_j}\frac{\partial \log p^{N_l}(X_l - f_l(Pa_l))}{\partial X},$$

其中, δ_j 表示的是 $s(x)$ 和 $s(x_{-k})$ 在分量 j 上的差 ($j\in[d]\setminus k$).

证明: 首先, 回顾 ANM 中关于 $s_j(x)$ 和 $s_j(x_{-k})$ 的表达:

$$\begin{cases} s_j(x) = \dfrac{\partial \log p^{N_i}(X_j - f_j(Pa_j))}{\partial X_j} - \Sigma_{i\in Ch(x_j)}\dfrac{\partial f_i(Pa_i)}{\partial X_j}\dfrac{\partial \log p^{N_i}(X_i - f_i(Pa_i))}{\partial X} \\ s_j(x_{-k}) = \dfrac{\partial \log p^{N_i}(X_j - f_j(Pa_j))}{\partial X_j} - \Sigma_{i\neq l, i\in Ch(x_j)}\dfrac{\partial f_i(Pa_i)}{\partial X_j}\dfrac{\partial \log p^{N_i}(X_i - f_i(Pa_i))}{\partial X} \end{cases}.$$

通过对比, 可以得到:

$$\delta_j = -\frac{\partial f_k(Pa_i)}{\partial X_j}\frac{\partial \log p^{N_k}(X_k - f_k(Pa_k))}{\partial X}.$$

证毕.

定理 1. 海森矩阵 $H(s(X))$ 和对节点 X_k 剪枝前后的得分函数差值向量, 即 $\delta_k = \{\delta_j\}_{j=1,j\neq k}^{d}$ 之间的关系可以被表达如下:

$$\delta_k = -\nabla_{x_k}\log p(x)\cdot\frac{H_k(\log p(x))}{H_{k,k}(\log p(x))}.$$

证明: 首先将海森矩阵表达扩展如下:

$$\begin{aligned} H_{k,j}(s(x)) &= \frac{\partial}{\partial x_j}\left[\nabla_{x_k}\log p(x)\right] \\ &= \frac{\partial^2 \log p^{N_k}(X_k - f_k(Pa_k))}{\partial x^2}\cdot\frac{\mathrm{d}(x_k - f_k(Pa_k))}{\mathrm{d}x_j} \\ &= \frac{\partial^2 \log p^{N_k}(X_k - f_k(Pa_k))}{\partial x^2}\cdot\frac{\mathrm{d}(f_k(Pa_k))}{\mathrm{d}x_j}. \end{aligned}$$

这里由于 k 是叶子节点, 所以其得分函数一阶导的表达为:

$$\nabla_{x_k}\log p(x) = \frac{\partial \log p^{N_k}(X_k - f_k(Pa_k))}{\partial X_k}.$$

进而, 令 $j=k$, 可以得到:

$$H_{k,k}(s(x)) = \frac{\partial}{\partial x_k}[\nabla_{x_k}\log p(x)] = \frac{\partial^2 \log p^{N_k}(X_k - f_k(Pa_k))}{\partial x^2} \cdot \frac{\mathrm{d}(f_k(Pa_k))}{\mathrm{d}x_k}$$
$$= \frac{\partial^2 \log p^{N_k}(X_k - f_k(Pa_k))}{\partial x^2}.$$

进而, 定理等式右边可以写作如下表达:

$$\frac{H_{k,j}(s(x)) \cdot \nabla_{x_k} s(x)}{H_{k,k}(s(x))} = \frac{\dfrac{\partial^2 \log p^{N_k}(X_k - f_k(Pa_k))}{\partial x^2} \cdot \dfrac{\mathrm{d}(f_k(Pa_k))}{\mathrm{d}x_j} \dfrac{\partial \log p^{N_k}(X_k - f_k(Pa_k))}{\partial X}}{\dfrac{\partial^2 \log p^{N_k}(X_k - f_k(Pa_k))}{\partial x^2}}$$
$$= \frac{\partial \log p^{N_k}(X_k - f_k(Pa_k))}{\partial X} \cdot \frac{\mathrm{d}(f_k(Pa_k))}{\mathrm{d}x_j} = -\delta_j.$$

证毕.

上述一系列定理表明, 在剪枝的过程中无需重新训练扩散模型和对应的雅克比矩阵就可以实现从剪枝前分布到剪枝后分布的过渡.

3.4 基于剪枝差分的根因识别算法

基于第 3.2 和 3.3 节中的理论基础, 本节阐述如何通过定理 1 结合扩散模型本身在推理阶段的估计, 实现高效的大规模数据根因分析. 首先, 基于第 3.1 节中的扩散模型算法, 一种基础版本的根因分析策略是直接交替进行扩散模型的训练估计和共同叶节点变量的剪枝, 如算法 1 所示.

算法 1. 训练-剪枝交替的根因分析 (direct alternative root-cause analysis, DARCA).

输入: 异常发生前后数据 $X^n = \{X^{n,i}\}_{i=1}^{m_n}$ 和 $X^c = \{X^{c,i}\}_{i=1}^{m_c}$, 初始化的分数扩散模型 $\mathcal{M}_\theta$, $\mathcal{M}_\theta^n$ 和 $\mathcal{M}_\theta^c$, 阈值 t.

1. 初始化 $\mathcal{T} = [d]$, $S = \varnothing$.

当 $\mathcal{T} \neq \varnothing$:

2. 在数据 $X_{[d]\setminus\mathcal{T}}$ 上训练 $\mathcal{M}_\theta$, 实现对混合数据分布海森矩阵的估计:

$$H(\log p(x))_{j,j} \approx J_x s_\theta(x)_{j,j}.$$

3. 在 $X^n_{[d]\setminus\mathcal{T}}$ 和 $X^c_{[d]\setminus\mathcal{T}}$ 上分别训练 $\mathcal{M}_\theta^n$ 和 $\mathcal{M}_\theta^c$, 实现对于两个环境数据海森矩阵分别的估计:

$$\begin{cases} H^n(\log p(x))_{j,j} \approx J_x s^n_\theta(x)_{j,j} \\ H^c(\log p(x))_{j,j} \approx J_x s^c_\theta(x)_{j,j} \end{cases}.$$

4. 根据引理 2 中公式 (3) 的第 1 条原则, 得到共同的叶子节点估计:

$$L = \underset{j \in [d]\setminus\mathcal{T}}{\operatorname{argmin}} Score(j),$$

其中, $Score(j) = J_x s^n_\theta(x)_{j,j} + J_x s^c_\theta(x)_{j,j}$.

5. $\mathcal{T} = \mathcal{T} \setminus \{L\}$.

6. 根据引理 2 中公式 (3) 的第 2 条原则, 如果 $J_x s_\theta(x)_{j,j} < t$, 那么:

$$S = S \cup \{L\}.$$

输出: S.

---

1) 在算法 1 主循环中的第 4 步中, 本文将原始的共同叶子节点定义为估计方差在两个数据分布上的和最小的节点.

2) 算法 1 引入了阈值变量 t, 用来判定检测出来的共同叶子节点在混合分布中是否是被干预的.

然而, 上述的算法存在的主要问题就是每次剪枝的时候, 算法会在去掉叶子节点 L 后的数据 $X_{[d]\setminus\mathcal{T}}$, $X^n_{[d]\setminus\mathcal{T}}$ 和 $X^c_{[d]\setminus\mathcal{T}}$ 上重新训练扩散模型 $\mathcal{M}_\theta$, $\mathcal{M}_\theta^n$ 和 $\mathcal{M}_\theta^c$, 这样会导致扩散模型的再训练次数随着叶子节点数量的增加而增加, 进而导致算法本身的复杂度变得不切实际. 基于第 3.3 节中提出的定理 1, 本文进而通过建立起剪枝前-剪枝后的

得分函数差异, 进而只需要一次训练的扩散模型即可推导出剪枝后的得分函数及海森矩阵:

$$\begin{cases} \widehat{\delta_L} = -s_\theta(x)_L \cdot \dfrac{J_x s_\theta(x)_{r_{LJ}}}{J_x s_\theta(x)_{L,L}} \\ \widehat{\delta_L^n} = -s_\theta^n(x)_L \cdot \dfrac{J_x s_\theta^n(x)_{r_{LJ}}}{J_x s_\theta^n(x)_{L,L}} \\ \widehat{\delta_L^c} = -s_\theta^c(x)_L \cdot \dfrac{J_x s_\theta^c(x)_{r_{LJ}}}{J_x s_\theta^c(x)_{L,L}} \end{cases} \tag{4}$$

其中, r_{LJ} 表示雅可比矩阵的第 L 行, 而 $s_\theta(x)_L$ 和 $J_x s_\theta(x)_{L,L}$ 都是标量. 基于上述思路, 本文进而提出优化后的根因分析算法 (见算法 2). 这里算法 2 第 6 行通过已经存在的库来进行神经网络雅克比矩阵的计算 (本文采用 functorch 实现雅可比矩阵的高效计算[39]).

算法 2. 优化版本的根因分析 (optimized diffusion-guided root-cause analysis, ODRCA).

输入: 异常发生前后数据 $X^n = \{X^{n,i}\}_{i=1}^{m_n}$ 和 $X^c = \{X^{c,i}\}_{i=1}^{m_c}$, 初始化的分数扩散模型 $\mathcal{M}_\theta$, $\mathcal{M}_\theta^n$ 和 $\mathcal{M}_\theta^c$, 阈值 t.

1. 初始化 $\mathcal{T} = [d]$, $S = \varnothing$.

2. 在数据 $X_{[d]\setminus\mathcal{T}}$ 上训练 $\mathcal{M}_\theta$, 实现对混合数据分布海森矩阵的估计:

$$H(\log p(x))_{j,j} \approx J_x s_\theta(x)_{j,j}.$$

3. 在 $X^n_{[d]\setminus\mathcal{T}}$ 和 $X^c_{[d]\setminus\mathcal{T}}$ 上分别训练 $\mathcal{M}_\theta^n$ 和 $\mathcal{M}_\theta^c$, 实现对于两个环境数据海森矩阵分别的估计:

$$\begin{cases} H^n(\log p(x))_{j,j} \approx J_x s_\theta^n(x)_{j,j} \\ H^c(\log p(x))_{j,j} \approx J_x s_\theta^c(x)_{j,j} \end{cases}.$$

4. 当 $\mathcal{T} \neq \varnothing$:

5. 根据引理 3、引理 4 中的公式计算剪枝后新的得分函数;

6. 基于自动微分工具计算剪枝后的雅克比矩阵 $J_x^{[d]\setminus\mathcal{T}} s_\theta(x)$, $J_x^{[d]\setminus\mathcal{T}} s_\theta^n(x)$ 和 $J_x^{[d]\setminus\mathcal{T}} s_\theta^c(x)$;

7. 根据引理 2 中公式 (3) 的第 1 条原则, 得到共同的叶子节点估计:

$$L = \underset{j\in[d]\setminus\mathcal{T}}{\operatorname{argmin}} Score(j),$$

其中, $Score(j) = J_x^{[d]\setminus\mathcal{T}} s_\theta^n(x)_{j,j} + J_x^{[d]\setminus\mathcal{T}} s_\theta^c(x)_{j,j}$.

8. $\mathcal{T} = \mathcal{T}\setminus\{L\}$.

9. 根据引理 2 中公式 (3) 的第 2 条原则, 如果 $J_x^{[d]\setminus\mathcal{T}} s_\theta(x)_{j,j} < t$, 那么:

$$S = S \cup \{L\}.$$

输出: S.

3.5 复杂度分析

训练复杂度: 首先, 本文所提 ODRCA 方法将扩散模型的训练和根因定位中剪枝所需的迭代隔离开来, 因此训练的复杂度和迭代次数无关, 即只需要预训练扩散模型一次. 鉴于训练的轮次数是固定的; 扩散模型本身的优化函数没有涉及任何复杂的约束条件 (例如增广拉格朗日约束[40]); 且扩散模型层间最大神经元个数是常数, 因此, 本文训练的复杂度仅和样本量 n 成正比, 即 $\mathrm{O}(n)$.

根因定位复杂度: 一旦扩散模型被预训练好, 剩下的操作就是迭代 d 轮次, 进行共同叶子节点的筛选和被干预节点的判断. 首先, 最外侧的循环需要 d 次迭代; 其次, 内侧计算剪枝后更新的得分函数需要计算 $3|\mathcal{T}|$ 次; 最后, 考虑到自动微分工具计算更新得分函数的雅克比矩阵需要 $3(d-|\mathcal{T}|)$ 次, 以及排序所需要的 $\mathrm{O}(d)$ 次计算, 整体的计算复杂度为 $\mathrm{O}\left(n+\sum_{i=0}^{d} d\times(3i+3(d-i))\right) = \mathrm{O}(n+3d^3)$. 可以看到, 该复杂度和样本量呈线性关系, 将原本的 $\mathrm{O}(n^2)$ 的平方复杂度直接降到了线性复杂度, 因而具备了扩散到大型数据集进行根因分析的相关潜力.

4 实验分析

本节首先介绍了本文着重研究的 3 个问题、实验所用的数据集和评价标准, 然后设计实验对本文的方法进行验证并对实验结果进行分析讨论.

4.1 研究问题

为了评估所提出的扩散模型引导的根因分析策略, 即 DARCA 和其优化版本 ODRCA 的有效性和效率, 本文的实验部分主要集中于回答下面 4 个问题.

- RQ1: 本文所提出的 DARCA 和 ODRCA 方法能否准确地识别出根因变量集合?
- RQ2: 本文所提出的 DARCA 和 ODRCA 方法是否在面向大量样本数据的真实根因分析问题上具备效率上的明显优势?
- RQ3: 本文所提出的无需剪枝的估计策略在开放环境下的表现 (即真实根因变量集合大小未知的情况) 下如何?
- RQ4: 本文所依赖的加性噪声模型是否会影响到方法在面向非加性噪声模型的表现?

4.2 实验数据集

本文主要在仿真数据和真实的故障定位分析数据上进行相关的实验验证. 仿真数据主要包含了两种常见的因果图结构, 即 Erd″os-Rényi (ER) 和 scale free (SF) 图结构.

仿真协议说明: 本文采用的仿真协议主要面向两种常用的 ER 和 SF 图结构, 这也是之前因果发现、根因分析和干预文献最常用的两种图结构[10,17,26,29]. 本文通过改变节点之间的稀疏度, 在多组仿真协议上进行了测试. 具体的仿真协议如下.

- 干预之前 (正常数据 X^n): 本节从 ER (SF) 图结构中采样 d 个节点, 具体的 SCM 如下所示:

$$X_j = \sum_{i\in PA_j^n} \sin\left(X_i^2\right) + N_j.$$

- 干预之后 (正常数据 X^c): 本节随机选择大约 $0.2|d|$ 的节点数目做软干预, 将干预之前的 SCM 方程做了扰动:

$$\begin{cases} X_j = \sum_{i\in PA_j^n} \sin(X_i^2) + N_j, & j\text{未被干预} \\ X_j = \sum_{i\in PA_j^c} 4\cos(2X_i^2 - 3X_i) + N_j, & j\text{被干预} \end{cases}.$$

本节严格遵循软干预的定义, 即节点的父节点集合不变, 仅有节点的条件分布发生变化. 为了验证本节方法的有效性, 本节对外源噪声变量 N 设置了 3 种情况.

- N 服从标准正态分布.
- N 服从均匀分布.
- N 服从拉普拉斯分布.

在仿真协议中, 样本数目被设置为 5 000; 特征维度被设置成在 $[10,20,30,40,50]$ 之间变化. 本文汇报的实验结果为将随机种子设置为 $[5,50,500,5\,000,50\,000]$ 下测试结果的平均指标.

真实数据说明: WADI[1]数据集是在一个覆盖了 16 天的运行过程的线上水处理云实验平台上收集得到的. 整个 WADI 的服务系统由 123 个传感器和执行器组成. WADI 对应的服务系统在前 14 天正常运行, 然而在最后 2 天, WADI 系统遭受了攻击. 进而, 15 个系统故障在事后被相应的监测设备收集得到. SWaT 数据集是从一个水处理测试平台收集的, 该平台由 6 个阶段 (高层实体) 组成, 拥有 51 个传感器 (低层实体). SWaT 数据集[41]中包含了持续了 11 天收集到的 16 个系统故障. 并且, 6 个阶段的存在使得 SWaT 数据的收集更加多样化、数据分布更加复杂. 为了模拟开放环境下的根因分析, 即潜在的故障个数不一定知道, 本文在真实数据的实验中主要记录了各个基线方法的 Top-K 推荐精度, 其中 $K \in [1,3,5,7,10]$.

4.3 基线方法

本文用来比较验证的基线方法主要可以分为 3 类.

- 第 1 类是基于相关性的根因分析方法, 其中包括: ϵ-Diagnosis 方法[42], 其使用异常发生前后变量的 coefficient

of variation (COV) 属性变化来检测根因变量集合.

● 第 2 类是基于因果发现的根因分析方法, 其包括: CIRCA 方法[4], 其通过回归的方式, 结合干预前后的数据, 对预先给定好的因果图做根因分析; Ψ-PC 算法[3], 其改进了文献 [14] 中提出的面向观测-干预数据结合的马尔可夫等价类理论, 并针对根因分析问题做了递归式的加速 (原文中的方法名是 RCD, 但是本文采用 Ψ-PC 以表明其采用了 Ψ -马尔可夫等价理论来实现根因分析); MULAN[1]方法, 其结合了系统运行时的文本日志输出, 通过语言大模型来辅助构建相应的因果图和根因分析; UT-IGSP[29]方法, 其通过贪婪策略来搜索最稀疏的排列, 进而识别对应的根因集合.

● 第 3 类是无需因果发现的干预检测方法, 其包括: iSCAN 方法[20], 其利用得分函数方差的变化来识别干预变量集合; LinearEST 方法[17], 其在全线性 SCM 的假设下, 通过观测精度矩阵的变化来识别被干预元素的集合.

特别地, 对于 CIRCA 方法, 鉴于本文所测试的数据集没有先验的因果图知识, 本文通过预部署一个基于 PC 算法的因果发现模块来支持其输入条件. 此外, 对于 MULAN 方法, 鉴于本文提供的数据不能够支持其通过大模型进行的文本分析, 本文将其多模态输入改为单模态输入.

4.4 评估指标

对于方法在根因分析上的表现, 本文通过 $F1$ 分数 ($F1$ score, %) 来评价估计的根因集合 $\widehat{Inv}$ 和真实的根因集合 Inv 之间的差异:

$$\begin{cases} Precision = \dfrac{\sum_{j\in Inv} 1\left(j\in\widehat{Inv}\right)}{\sum_{j\in Inv} 1\left(j\in\widehat{Inv}\right)+\sum_{j\notin Inv} 1\left(j\in\widehat{Inv}\right)} \\ Recall = \dfrac{\sum_{j\in Inv} 1\left(j\in\widehat{Inv}\right)}{\sum_{j\in Inv} 1\left(j\in\widehat{Inv}\right)+\sum_{j\in Inv} 1\left(j\notin\widehat{Inv}\right)} \\ F1 = 2\times\dfrac{Precision\times Recall}{Precision+Recall} \end{cases}.$$

对于各个基线方法在开放环境下的根因分析表现 (真实的根因变量个数未知, 因此需要通过推荐前 k 个最可能的变量作为根因变量), 本节通过前 k 个推荐根因变量的精度 ($Precision@k$) 来衡量其表现 (这里默认变量已经被算法排序过):

$$Precision@k = \frac{\sum_{j\leqslant k} 1\left(j\in\widehat{Inv}\right)}{k}.$$

注意对于本文所提方法, 其 Top-K 推荐通过对于算法中 $J_x^{[d]\setminus\mathcal{T}} s_\theta(x)_{j,j}$ 的大小进行排序从而得到相应的 Top-K 个根因变量. 对于方法在根因分析数据上的效率, 本文通过计算平均每次训练+测试的运行时间 (min) 来测量.

4.5 实验设置

对于所有的基线方法, 本文遵循其在相应开源代码中的实现以及超参数选择方式. 对于所有条件独立性, 测试, 本文采用基于核方法的条件独立性测试 (KCI[43]). 对于本文方法用到的扩散模型, 其时间步骤 $T = 100$, 扩散模型方差的控制超参数在区间 [0.000 1,0.002] 之间线性增长. 在采样的时候, t 从一个均匀分布中被采样. 对于扩散模型内部的架构, 整个神经网络是具备 5 层线性 MLP 的全连接网络, 激活函数为 LeakyReLU, 同时在第一层配备了层归一化和 Dropout 机制[37]. 所有的实验都基于一块 NVIDIA A100 显卡进行.

4.6 RQ1: 各个方法的根因分析性能表现分析

如图 3 和图 4 所示, 本节在改变邻接密度, 即图本身结构的稀疏度的情况下, 分别测试了邻接密度为 5 和 10 下各个基线方法的表现情况.

● 基于相关性的根因分析方法 ϵ-Diagnosis 表现较差. 其根本原因和本文在第 1 节中引入因果根因分析的动机相吻合, 即相关性的方法非常容易定位到直接原因而非引起故障的上游根本原因.

● 基于条件独立性测试的 CIIRCA 和 Ψ-PC 表现同样不理想. 背后的原因主要是条件独立性测试这一工具目

前仍处于不稳定、尚待研究的状态, 尽管 KCI 已经是非线性估计中最好的测试工具之一, 但是面对样本量大、结构复杂的数据分布时候, 对应的因果发现效果会下降. 这也和本文的主旨相吻合: 基于因果发现的根因分析同样会受到因果发现这一流程本身精度的制约.

● 基于线性结构的 LinearEST 算法表现不太稳定, 并且效果也不是很理想 (平均指标在 50%–60% 摇摆). 这一现象的原因是线性模型的前件假设不满足本节生成的仿真数据. 在模型误识别的情况下, LinearEST 算法本身的理论基础会不成立.

● 基于非线性统计量构造的 iSCAN 算法表现优于大多数基线方法, 这也验证了本文的动机, 即非线性、不依赖先验因果发现任务的根因分析方法是未来根因分析框架设计的目标之一. 然而, iSCAN 本身对于数据分布的估计依赖于斯坦等式的核估计, 这一估计在面向复杂非线性结构的数据的时候, 其效果有待改进.

● 最后, 本文所提出的两个方法, DARCA 和 ODRCA 方法, 在各个参数设置下均取得了较好的根因识别效果. 这说明了扩散模型对于数据分布得分函数估计这一技术路线的有效性.

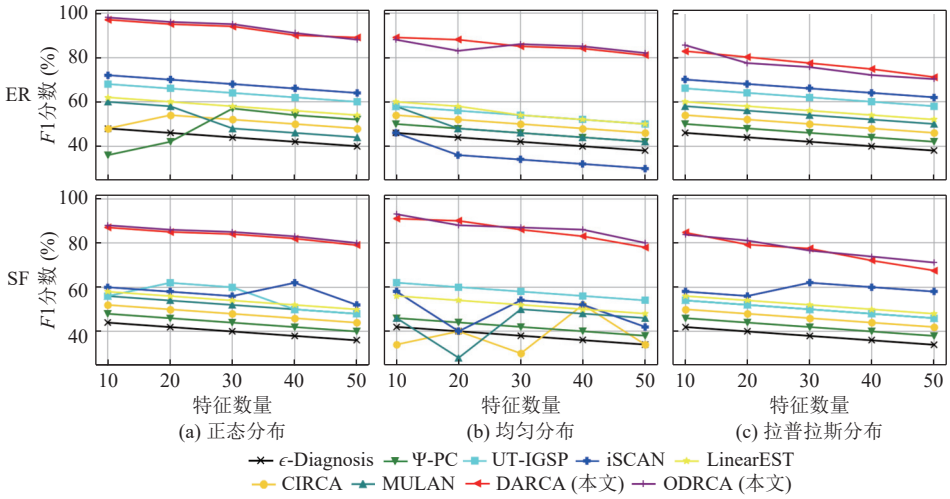

图 3 在平均邻接数量等于 5 的时候, 各个基线方法在仿真数据上的根因识别表现

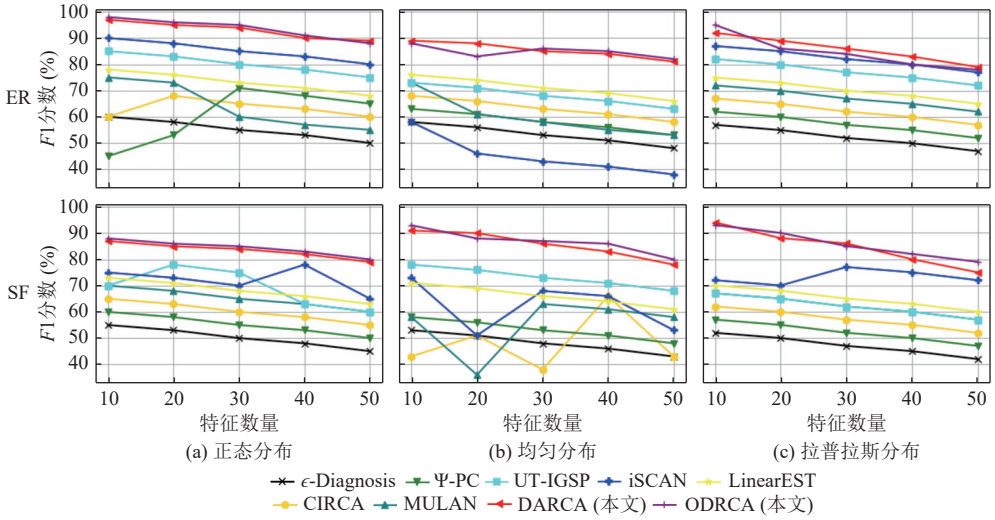

图 4 在平均邻接数量等于 10 的时候, 各个基线方法在仿真数据上的根因识别表现

4.7 RQ2: 各个方法的根因分析效率比较

通过在仿真数据上分别改变影响根因分析的两个关键因素, 即样本量和特征维度, 本节进而测试了除了浅层方法 (ϵ-Diagnosis、LinearEST 和 UT-IGSP) 方法之外, 其余所有基于深度学习的根因识别方法在仿真数据上的运行效率.

● 如图 5 所示, Ψ-PC 方法对于样本数量的增长较为敏感, 这是因为对应的 KCI 测试本身会随测试样本量的增长呈现出平方复杂度; 而 iSCAN 本身在面向超过 1 000 个样本时候的复杂度就因为平方的斯坦估计和剪枝交替进行, 而导致运行时间溢出. 具体来说, 图 5 呈现出来的趋势是因为其余方法的复杂度和运行时间过大导致: 例如, iSCAN 算法的运行时间超过了 230 min.

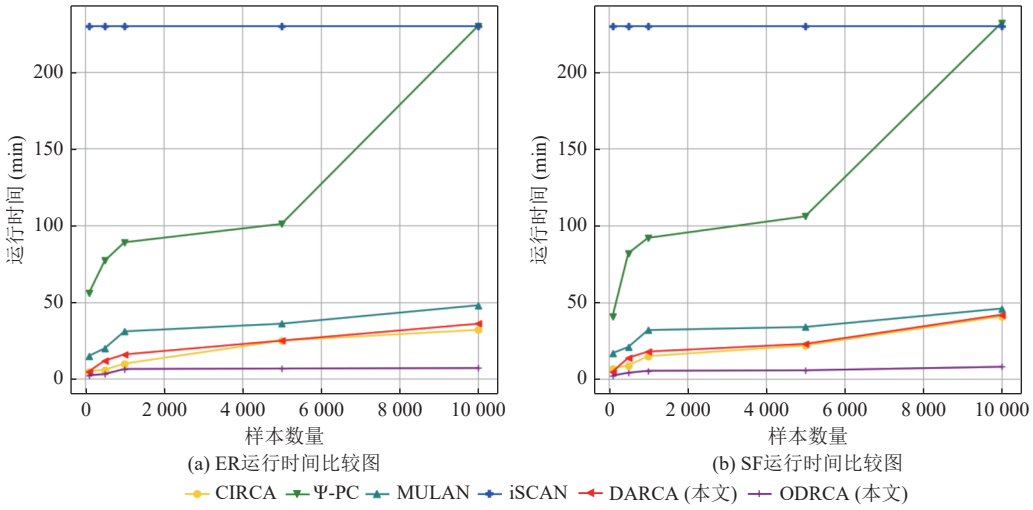

图 5 节点数量等于 50 的时候, 各个基线方法在仿真数据上的效率表现

● 如图 6 所示, 本文所提出的 DARCA 基线对于特征数量较为敏感, 因为剪枝-扩散预训练交替进行的模型会使每进行一剪枝就要进行一次扩散模型的预训练, 这样使得 DARCA 的训练成本随着特征增长而增加. 与和图 5 类似, 图 6 呈现的结果是由于其他对比基线算法的复杂度过高, 运行时间过长引起的; 而实际上, ODRCA 的实际实验结果和理论复杂度分析吻合.

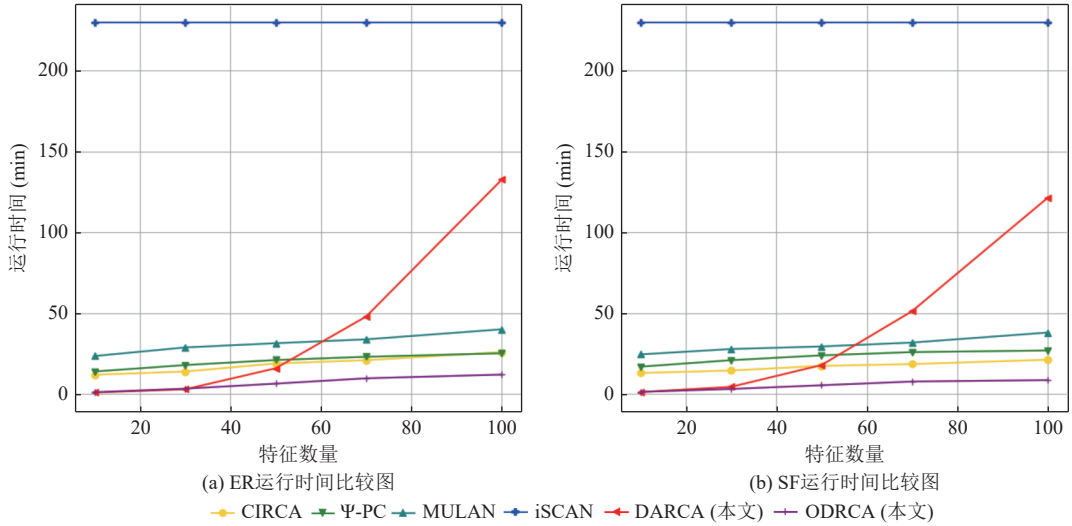

图 6 样本量等于 5 000 的时候, 各个基线方法在仿真数据上的效率表现

● 相对地, 本文面向 DARCA 内在的优化缺点, 进而提出的优化版本算法 ODRCA 无论是随着特征数量增长还是样本量增长, 都呈现出较高的运行效率, 其时间成本始终处于最低的状态, 这一分析结果也验证了本文所提出 ODRCA 方法, 即直接从一次训练的扩散模型推导剪枝后数据得分函数这一优化策略的有效性.

● 此外, 如表 2、表 3 所示, 我们将 ODRCA 方法在扩散模型预训练阶段的计算开销汇报如下: 由于 ODRCA 的扩散模型只训练一次, 其训练成本较小; 且由于扩散模型训练的时候采用固定大小的 batch size, 因此样本量不会影响扩散模型的训练时间 (时间步 $T = 100$); 由于扩散模型训练的时候网络架构 (嵌入层维度) 固定, 因此特征维度的增加不会特别影响到扩散模型的训练时间.

表 2　扩散模型训练时间随样本数量变化 ($d = 60$) (min)

样本数量	$n = 100$	$n = 500$	$n = 1000$	$n = 5000$	$n = 10000$
训练时间	2.08	2.11	2.12	2.13	2.13

表 3　扩散模型训练时间随特征维度变化 ($n = 5000$) (min)

样本数量	$d = 20$	$d = 40$	$d = 60$	$d = 80$	$d = 100$
训练时间	1.6	1.86	2.13	2.14	2.18

● 雅可比矩阵的计算开销分析. 最后, 我们将汇报仿真实验中每次迭代中需要计算雅可比矩阵的时间平均开销. 如表 4 所示, 可以看到随着特征维度增加, 雅可比矩阵的计算时间也在增长, 然而这样的计算开销是可以接受的 (尤其在通用的根因分析场景下 $d \leqslant 100$).

表 4　仿真实验 (ER 图) 在不同特征维度下雅可比矩阵计算时间的变化趋势 (s)

特征维度	$d = 50$	$d = 100$	$d = 200$	$d = 500$
雅可比计算时间	0.719	2.028	3.273	7.272

4.8 RQ3: 开放环境下的根因分析评测: 真实数据上的结果分析

表 5 和表 6 (加粗数据为最优值) 中的实验结果表明, 在面向真实场景的根因分析任务, 即推荐可能的 Top-K 个根因变量任务上, 所提出的 DARCA 和 ODRCA 方法表现同样十分优越. 这也进一步验证了本文的动机, 即面向真实复杂数据的时候, 扩散模型可以较好地对得分函数进行估计. 与此同时, 和基于核方法的估计 (iSCAN) 相比, 本文所提出的方法, 即 ODRCA, 同样在运行效率上取得了绝对的优势. 这样验证了本文所提出的无需剪枝, 直接估计得分函数这一策略的有效性.

表 5　WADI 真实服务系统故障定位的效果比较

方法	*Precision*@1 (%)	*Precision*@3 (%)	*Precision*@5 (%)	*Precision*@7 (%)	*Precision*@9 (%)	Time (min)
ϵ-Diagnosis	15.2	19.4	25.8	36.6	40.1	0.006 1
CIRCA	13.2	14.5	26.1	35.4	55.8	10.1
Ψ-PC	14.3	21.4	35.7	45.2	57.1	>60
MULAN	17.8	25.2	38.5	49.1	53.3	22.3
UT-IGSP	11.2	22.4	31.9	50.4	51.1	1.2
iSCAN	19.2	25.3	28.4	55.6	61.2	>60
LinearEST	10.1	16.3	21.4	32.7	38.2	0.05
DARCA (本文)	21.2	**32.7**	33.9	58.0	73.1	12.2
ODRCA (本文)	**25.8**	30.5	**34.6**	**61.2**	**78.5**	**5.7**

4.9 RQ4: 加性噪声假设 (ANM) 的影响

图 3 和图 4 展示了在开放环境根因推荐的任务下, 各个基线方法在 Top-K 推荐精度指标衡量下的表现对比. 具体来说, K 越大, 对推荐的容错率越高, Top-K 精度表现越好. 此外, 时间这一指标代表了各个基线方法在平均每

个样本的训练和推理的时间总和. 仿真数据的根因分析表现说明了, 当数据生成的机制满足加性噪声假设的时候, 所提出的方法效果表现较好, 这说明了本文理论和实际表现的一致性. 然而, 真实数据集 WADI 的内在数据生成机制明显不一定满足加性噪声假设. 与之相对地, 本文所提出的根因识别方法仍然在开放的根因推荐下取得了较好的表现效果, 而这一实验结果说明了本文所提出的根因分析方法本质上在面对非加性噪声数据的表现中仍然具有较好的鲁棒性. 这一实验现象也和基于得分函数的因果发现方法鲁棒性最好这一经验性结论[1], 是一致的. 此外, 与之相对的, LinearEST 方法在各个设置下的表现较差, 这也从侧面说明, 从线性到非线性的 SCM 假设带来的模型误识别 (misspecification) 影响比从加性噪声到非加性噪声带来的误识别影响大得多.

表 6　SWaT 真实服务系统故障定位的效果比较 (%)

方法	*Precision*@1	*Precision*@3	*Precision*@5	*Precision*@7	*Precision*@9
ϵ-Diagnosis	0.0	1.4	5.8	16.6	20.4
CIRCA	29.6	38.3	53.0	55.4	60.1
Ψ-PC	9.8	27.4	36.1	43.9	52.0
MULAN	17.8	25.2	34.1	44.7	49.3
UT-IGSP	13.4	20.3	25.6	33.8	41.2
iSCAN	20.1	27.3	38.9	45.7	53.1
LinearEST	16.4	24.4	29.8	40.1	48.7
DARCA (本文)	31.2	42.7	55.9	59.2	68.9
ODRCA (本文)	**35.2**	**46.8**	**56.9**	**62.7**	**71.4**

5　进一步实验补充和分析

5.1　RQ5: 面向未观测混杂变量的鲁棒性分析

为了检验本文所提出方法在未观测混杂变量下的表现, 我们补充了额外的仿真实验 (ER 图, 高斯噪声, 特征变量数 = 50): 首先, 我们在基于结构因果模型数据的时候, 筛选出子节点集合 $\geqslant 2$ 的节点; 进而, 对这些节点按照比例 ρ 随机筛选出未观测混杂变量; 最后, 生成整体的数据集, 但是丢掉筛选得到的混杂变量集合, 这样就得到了具有未观测混杂变量的节点集合; 此外, 本文确保根因变量不在未观测混杂变量中, 否则其无法被定位到. 具体实验结果如表 7 所示. 未观测混杂变量的比例在 40% 及以下的时候, 所提出 ODRCA 方法的效果具备一定的鲁棒性; 当未观测混杂变量的比例超过 40% 的时候, ODRCA 方法的效果会出现较大幅度的下降。

表 7　未观测混杂比例的影响分析

ρ	0.1	0.2	0.3	0.4	0.5
$F1$ (%)	93.8	92.1	88.7	85.2	63.4

5.2　RQ6: 扩散模型预训练的消融实验

当前, 扩散模型不能够在任意情况下支持对于数据分布的任意拟合. 例如, 信噪比很低的时候或者预训练数据量很小的时候, 扩散模型本身的拟合性能都会受到影响. 然而, 相比于之前的工作, 例如 iSCAN 中的核密度估计, 本文通过引入扩散模型已经极大地提升了对数据分布拟合的准确度和拟合过程的效率. 本小节将针对扩散模型本身的性能表现, 以及其是否精准识别了得分函数做进一步的消融实验.

● 得分函数的拟合情况. 在合成数据上, 通过调整数据的信噪比, 我们汇报扩散模型经过训练后对得分函数的拟合情况 (ER 图, 特征维度 $d = 30$): 首先, 我们设计了比较扩散模型估计的 $s_\theta(x)$ 和真实的得分函数 $s(x)$ 之间的差异的指标 (经验 L2-范数差异):

$$L_2(s) = \sum_{i=1}^{n} |s_\theta(x_i) - s(x_i)|^2.$$

进而, 我们分别通过对数据集添加加性的高斯噪声和乘性的高斯噪声 (均值为 0, 方差 σ 逐步增大, 原文仿真

实验方差为 1.0), 在结构因果模型的外源噪声为高斯变量 (此时可以通过闭式解得到潜在得分函数的表达和计算), 进而得到表 8. 如表 8 所示, 去噪扩散模型的训练表现对于训练数据的噪声添加展现了一定的鲁棒性, 但是当噪声程度越过一个阈值 ($\sigma = 2.0$) 的时候, 对于得分函数的拟合会受到影响.

表 8　噪声变化对扩散模型估计的影响

指标	$\sigma = 0.1$	$\sigma = 0.5$	$\sigma = 1.0$	$\sigma = 1.5$	$\sigma = 2.0$
$L_2(s)$	0.03±0.01	0.05±0.01	0.06±0.02	0.09±0.02	0.11±0.03
$F1$ (%)	93.7	93.7	92.1	91.6	90.3

● 扩散模型的样本效率. 在合成数据上, 通过调整扩散模型的训练数据量, 我们汇报扩散模型拟合结果和根因分析结果如下.

如表 9 所示, 去噪扩散模型的训练表现对于训练样本量的减小展现了一定的鲁棒性, 且训练样本量在 500 左右仍然可以维持 90% 以上的根因分析 $F1$ 指标; 同时, 我们也注意到训练样本量在过小的情况下, 任何机器学习方法都会失效且产生较大的偏差 (大数定律).

表 9　样本量变化对扩散模型估计的影响

指标	$n = 500$	$n = 1000$	$n = 1500$	$n = 3000$
$L_2(s)$	0.11±0.03	0.04±0.02	0.03±0.01	0.02±0.01
$F1$ (%)	90.5	92.1	93.8	94.1

● 扩散模型超参数的调整. 我们对于扩散模型训练过程中时间步 T 进行了进一步的经验性调整, 以检验本文超参数设置是否合理. 我们给出了进一步超参数调整的实验性结果 (仿真数据: ER 图, 高斯噪声), 通过调整时间步数参数 $T \in [50, 80, 100, 150]$, 我们将根因分析效果记录在表 10. 随着训练时间步 T 的增加, 得分函数的拟合误差, 即 $L_2(s)$ 逐步减小; 而在超过 $T = 100$ ($T = 150$) 下 $L_2(s)$ 已经收敛; 随着训练时间步 T 的增加, 根因分析模型的表现在 $T = 50$ 下有所下降, 这是因为扩散模型还未完全训练收敛 ($L_2(s)$ 较大); 而在 $T = 150$ 下根因分析的 F_1 指标几乎没有变化; 表 10 中的变化趋势反映了本文对时间步的选择是相对最优的.

表 10　不同时间步下的扩散模型性能以及下游根因分析指标

时间步	$T = 50$	$T = 80$	$T = 100$	$T = 150$
$L_2(s)$	0.12±0.05	0.03±0.01	0.02±0.01	0.02±0.01
$F1$ (%)	90.2	93.9	94.1	94.3

● 噪声方差在某个区间内线性增长这一结论是扩散模型进行分布拟合的通用操作, 本文中的区间 [0.000 1, 0.002] 是通过对于区间左侧和右侧同时进行搜索得到的: 区间左侧值在 [0.000 01, 0.000 1, 0.001, 0.01, 0.1] 中搜索; 区间右侧值在 [0.000 02, 0.000 2, 0.002, 0.02, 0.2] 中搜索; 搜索的时候保证区间左侧值小于右侧值.

6 总　结

本文提出了一种基于扩散模型的根因定位方法. 基于结构因果模型, 通过观察对于变量的方差变化, 本文提出通过扩散模型估计得到的得分函数来识别跨异常发生前后的根因变量. 为了避免识别过程中剪枝操作带来的模型重训练开销, 本文进而提出了无需剪枝的一次性估计得分函数估计策略. 在仿真数据和真实数据上的实验结果验证了本文所提出方法的有效性和高效性. 在未来的研究工作中, 以下几个方面的研究内容值得进一步关注: (1) 结合系统运行的实际数据进一步辅助根因变量的识别和定位. 在包括云服务器系统在内的大型软件系统中, 例如系统的运行日志等[6]数据是潜在可以支持根因定位的辅助变量. 具体来说, 云服务系统的依赖图通过 reverse 操作可以直接生成因果图. 如果有完整的、足以支撑依赖图还原的日志信息, 则完整的因果图可以被直接生成; 而生成的因果图可以辅助本文提出的扩散引导的根因分析方法 ODRCA 进行矫正和校验. 例如, 每次检验的叶子节点可以对照完整因果图进行校验; 此外, 如果日志信息不完全, 则可以基于现有的日志信息恢复出部分依赖图, 进而恢复

出因果图和因果边, 从而对根因分析方法进行信息量有限的矫正和检验; 如何过滤变量的冗余信息, 筛选有效的运行数据, 也是在把本文所提出的根因分析往具体的下游应用落地的关键所在. (2) 考虑面向潜在未观测混杂变量的根因分析. 目前主流的根因分析策略往往考虑混杂变量全部可观测, 即因果充分性条件成立[12]. 如何拓展该假设的边界, 利用因果敏感性分析等技术实现更加鲁棒的根因分析, 也是理论上值得突破的方向之一[44].

References

[1] Zheng LC, Chen ZZ, He JR, Chen HF. MULAN: Multi-modal causal structure learning and root cause analysis for microservice systems. In: Proc. of the 2024 ACM Web Conf. Singapore: ACM, 2024. 4107–4116. [doi: 10.1145/3589334.3645442]

[2] Dhaou A, Bertoncello A, Gourvénec S, Garnier J, Le Pennec E. Causal and interpretable rules for time series analysis. In: Proc. of the 27th ACM SIGKDD Conf. Knowledge Discovery & Data Mining. ACM, 2021. 2764–2772. [doi: 10.1145/3447548.3467161]

[3] Ikram A, Chakraborty S, Mitra S, Saini SK, Bagchi S, Kocaoglu M. Root cause analysis of failures in microservices through causal discovery. In: Proc. of the 36th Int'l Conf. Neural Information Processing Systems. New Orleans: Curran Associates Inc., 2022. 2259.

[4] Li MJ, Li ZY, Yin KL, Nie XH, Zhang WC, Sui K, Pei D. Causal inference-based root cause analysis for online service systems with intervention recognition. In: Proc. of the 28th ACM SIGKDD Conf. Knowledge Discovery and Data Mining. Washington: ACM, 2022. 3230–3240. [doi: 10.1145/3534678.3539041]

[5] Cheng Y, Wang L, Zhao XY. Review of root cause analysis research. Application Research of Computers, 2023, 40(4): 961–966 (in Chinese with English abstract). [doi: 10.19734/j.issn.1001-3695.2022.07.0450]

[6] Yu QY, Bai XY, Li MJ, Li QY, Liu T, Liu ZY, Pei D. Performance modeling and anomaly location of large microservice systems based on trace control flow analysis. Ruan Jian Xue Bao/Journal of Software, 2022, 33(5): 1849–1864 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/6209.htm [doi: 10.13328/j.cnki.jos.006209]

[7] Zhuang WJ, Zhang H. Research on micro-service fault detection based on deep learning. Computer Engineering and Applications, 2022, 58(16): 326–332 (in Chinese with English abstract). [doi: 10.3778/j.issn.1002-8331.2012-0553]

[8] Jing YH, He B, Zhang LX, Li TX, Wang JY, Liu C. PASER: Root cause location model for additive multidimensional KPIs. Ruan Jian Xue Bao/Journal of Software, 2022, 33(2): 738–750 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/6212.htm [doi: 10.13328/j.cnki.jos.006212]

[9] Xia HC, Li X, Pang Y, Liu JF, Ren K, Xiong L. P-Shapley: Shapley values on probabilistic classifiers. Proc. of the VLDB Endowment, 2024, 17(7): 1737–1750. [doi: 10.14778/3654621.3654638]

[10] Budhathoki K, Minorics L, Bloebaum P, Janzing D. Causal structure-based root cause analysis of outliers. In: Proc. of the 39th Int'l Conf. Machine Learning. Baltimore: PMLR, 2022. 2357–2369.

[11] Wang DJ, Chen ZZ, Fu YJ, Liu YC, Chen HF. Incremental causal graph learning for online root cause analysis. In: Proc. of the 29th ACM SIGKDD Conf. Knowledge Discovery and Data Mining. Long Beach: ACM, 2023. 2269–2278. [doi: 10.1145/3580305.3599392]

[12] Halpern JY. A modification of the Halpern-pearl definition of causality. In: Proc. of the 24th Int'l Conf. Artificial Intelligence. Buenos Aires: AAAI Press, 2015. 3022–3033.

[13] Glymour C, Zhang K, Spirtes P. Review of causal discovery methods based on graphical models. Frontiers in Genetics, 2019, 10: 524. [doi: 10.3389/fgene.2019.00524]

[14] Jaber A, Kocaoglu M, Shanmugam K, Bareinboim E. Causal discovery from soft interventions with unknown targets: Characterization and learning. In: Proc. of the 34th Int'l Conf. Neural Information Processing Systems. Vancouver: Curran Associates Inc., 2020. 801.

[15] Spirtes P, Zhang K. Causal discovery and inference: Concepts and recent methodological advances. Applied Informatics, 2016, 3(1): 3. [doi: 10.1186/s40535-016-0018-x]

[16] Montagna F, Mastakouri AA, Eulig E, Noceti N, Rosasco L, Janzing D, Aragam B, Locatello F. Assumption violations in causal discovery and the robustness of score matching. In: Proc. of the 37th Int'l Conf. Neural Information Processing Systems. New Orleans: Curran Associates Inc., 2024. 2050.

[17] Varici B, Shanmugam K, Sattigeri P, Tajer A. Scalable intervention target estimation in linear models. In: Proc. of the 35th Int'l Conf. Neural Information Processing Systems. Curran Associates Inc., 2021. 115.

[18] Varici B, Shanmugam K, Sattigeri P, Tajer A. Intervention target estimation in the presence of latent variables. In: Proc. of the 38th Conf. Uncertainty in Artificial Intelligence. Eindhoven: PMLR, 2022. 2013–2023.

[19] Yang K, Katcoff A, Uhler C. Characterizing and learning equivalence classes of causal DAGs under interventions. In: Proc. of the 35th Int'l Conf. Machine Learning. Stockholm: PMLR, 2018. 5541–5550.

[20] Chen TY, Bello K, Aragam B, Ravikumar P. iSCAN: Identifying causal mechanism shifts among nonlinear additive noise models. In:

Proc. of the 37th Int'l Conf. Neural Information Processing Systems. New Orleans: Curran Associates Inc., 2023. 1935.

[21] Stein C. A bound for the error in the normal approximation to the distribution of a sum of dependent random variables. In: Proc. of the 6th Berkeley Symp. Mathematical Statistics and Probability. Berkeley: University of California Press, 1972. 583–602.

[22] Li YZ, Turner RE. Gradient estimators for implicit models. In: Proc. of the 6th Int'l Conf. Learning Representations. Vancouver: OpenReview.net, 2018.

[23] Cheng W, Zhang K, Chen HF, Jiang GF, Chen ZZ, Wang W. Ranking causal anomalies via temporal and dynamical analysis on vanishing correlations. In: Proc. of the 22nd ACM SIGKDD Int'l Conf. Knowledge Discovery and Data Mining. San Francisco: ACM, 2016. 805–814. [doi: 10.1145/2939672.2939765]

[24] Capozzoli A, Lauro F, Khan I. Fault detection analysis using data mining techniques for a cluster of smart office buildings. Expert Systems with Applications, 2015, 42(9): 4324–4338. [doi: 10.1016/j.eswa.2015.01.010]

[25] Duan PT, He ZZ, He YH, Liu FD, Zhang AQ, Zhou D. Root cause analysis approach based on reverse cascading decomposition in QFD and fuzzy weight ARM for quality accidents. Computers & Industrial Engineering, 2020, 147: 106643. [doi: 10.1016/j.cie.2020.106643]

[26] Meng Y, Zhang SL, Sun YQ, Zhang RR, Hu ZL, Zhang YY, Jia CY, Wang ZG, Pei D. Localizing failure root causes in a microservice through causality inference. In: Proc. of the 28th IEEE/ACM Int'l Symp. Quality of Service (IWQoS). Hangzhou: IEEE, 2020. 1–10. [doi: 10.1109/IWQoS49365.2020.9213058]

[27] Soldani J, Brogi A. Anomaly detection and failure root cause analysis in (micro) service-based cloud applications: A survey. ACM Computing Surveys (CSUR), 2022, 55(3): 59. [doi: 10.1145/3501297]

[28] Shimizu S. Lingam: Non-Gaussian methods for estimating causal structures. Behaviormetrika, 2014, 41(1): 65–98. [doi: 10.2333/bhmk.41.65]

[29] Wang YH, Solus L, Yang KD, Uhler C. Permutation-based causal inference algorithms with interventions. In: Proc. of the 31st Int'l Conf. Neural Information Processing Systems. Long Beach: Curran Associates Inc., 2017. 5824–5833.

[30] Yang YQ, Salehkaleybar S, Kiyavash N. Learning unknown intervention targets in structural causal models from heterogeneous data. In: Proc. of the 27th Int'l Conf. Artificial Intelligence and Statistics. Valencia: PMLR, 2024. 3187–3195.

[31] Hoyer PO, Janzing D, Mooij J, Peters J, Schölkopf B. Nonlinear causal discovery with additive noise models. In: Proc. of the 22nd Int'l Conf. Neural Information Processing Systems. Vancouver: Curran Associates Inc., 2008. 689–696.

[32] Peters J, Mooij JM, Janzing D, Schölkopf B. Causal discovery with continuous additive noise models. The Journal of Machine Learning Research, 2014, 15(1): 2009–2053.

[33] Rolland P, Cevher V, Kleindessner M, Russel C, Janzing D, Schölkopf B, Locatello F. Score matching enables causal discovery of nonlinear additive noise models. In: Proc. of the 39th Int'l Conf. Machine Learning. Baltimore: PMLR, 2022. 18741–18753.

[34] Sohl-Dickstein J, Weiss E, Maheswaranathan N, Ganguli S. Deep unsupervised learning using nonequilibrium thermodynamics. In: Proc. of the 32nd Int'l Conf. Machine Learning. Lille: PMLR, 2015. 2256–2265.

[35] Kingma DP, Salimans T, Poole B, Ho J. Variational diffusion models. In: Proc. of the 35th Int'l Conf. Neural Information Processing Systems. Curran Associates Inc., 2021. 1660.

[36] Ho J, Jain A, Abbeel P. Denoising diffusion probabilistic models. In: Proc. of the 34th Int'l Conf. Neural Information Processing Systems. Vancouver: Curran Associates Inc., 2020. 574.

[37] Song Y, Sohl-Dickstein J, Kingma DP, Kumar A, Ermon S, Poole B. Score-based generative modeling through stochastic differential equations. In: Proc. of the 9th Int'l Conf. Learning Representation. OpenReview.net, 2021.

[38] Ramzi Z, Remy B, Lanusse F, Starck JL, Ciuciu P. Denoising score-matching for uncertainty quantification in inverse problems. arXiv:2011.08698, 2020.

[39] Ma JM, Li XH, Wang ZH, Zhang XC, Yan SG, Chen YT, Zhang YQ, Jin MX, Jiang LJ, Liang Y, Yang C, Lin DH. A holistic functionalization approach to optimizing imperative tensor programs in deep learning. In: Proc. of the 61st ACM/IEEE Design Automation Conf. San Francisco: ACM, 2024. 200. [doi: 10.1145/3649329.3658483]

[40] Lachapelle S, Brouillard P, Deleu T, Lacoste-Julien S. Gradient-based neural dag learning. In: Proc. of the 8th Int'l Conf. Learning Representations. Addis Ababa: OpenReview.net, 2020.

[41] Mathur AP, Tippenhauer NO. SWaT: A water treatment testbed for research and training on ICS security. In: Proc. of the 2016 Int'l Workshop on Cyber-physical Systems for Smart Water Networks (CySWater). Vienna: IEEE, 2016. 31–36. [doi: 10.1109/CySWater.2016.7469060]

[42] Shan HS, Chen Y, Liu HF, Zhang YP, Xiao X, He XF, Li M, Ding W. ϵ-Diagnosis: Unsupervised and real-time diagnosis of small-window long-tail latency in large-scale microservice platforms. In: Proc. of the 2019 World Wide Web Conf. San Francisco: ACM, 2019. 3215–3222. [doi: 10.1145/3308558.3313653]

[43] Doran G, Muandet K, Zhang K, Schölkopf B. A permutation-based kernel conditional independence test. In: Proc. of the 30th Conf. on Uncertainty in Artificial Intelligence. Quebec City: AUAI Press, 2014. 132–141.

[44] Cinelli C, Kumor D, Chen B, Pearl J, Bareinboim E. Sensitivity analysis of linear structural causal models. In: Proc. of the 36th Int'l Conf. Machine Learning. Long Beach: PMLR, 2019. 1252–1261.

附中文参考文献

[5] 程燕, 王磊, 赵晓永. 根因分析研究综述. 计算机应用研究, 2023, 40(4): 961–966. [doi: 10.19734/j.issn.1001-3695.2022.07.0450]

[6] 于庆洋, 白晓颖, 李明杰, 李奇原, 刘涛, 刘泽胤, 裴丹. 基于调用链控制流分析的大型微服务系统性能建模与异常定位. 软件学报, 2022, 33(5): 1849–1864. http://www.jos.org.cn/1000-9825/6209.htm [doi: 10.13328/j.cnki.jos.006209]

[7] 庄卫金, 张鸿. 基于深度学习的微服务故障检测研究. 计算机工程与应用, 2022, 58(16): 326–332. [doi: 10.3778/j.issn.1002-8331.2012-0553]

[8] 靖宇涵, 何波, 张凌昕, 李天星, 王敬宇, 刘聪. PASER: 加性多维 KPI 异常根因定位模型. 软件学报, 2022, 33(2): 738–750. http://www.jos.org.cn/1000-9825/6212.htm [doi: 10.13328/j.cnki.jos.006212]

作者简介

王浩天, 博士, 助理研究员, 主要研究领域为因果推断, 根因分析, 因果大模型.

周学广, 博士, 教授, 博士生导师, CCF 高级会员, 主要研究领域为网络内容安全, 基于密码的中文信息处理, 基于知识推理的网络对抗.

王尚文, 博士生, CCF 专业会员, 主要研究领域为智能化软件工程, 软件维护与演化.

靳若春, 博士, 助理研究员, CCF 专业会员, 主要研究领域为数据库, 数据质量, 智能数据管理.

黄万荣, 博士, 副研究员, 主要研究领域为智能软件, 机器学习.

杨文婧, 博士, 研究员, 博士生导师, 主要研究领域为以数据为中心的人工智能, 因果推断和可信机器学习, 多智能体分布式控制.

王戟, 博士, 研究员, 博士生导师, CCF 会士, 主要研究领域为可信软件, 智能软件, 新兴软件.