NETRCA: AN EFFECTIVE NETWORK FAULT CAUSE LOCALIZATION ALGORITHM

（NETRCA：一种有效的网络故障原因定位算法）

1 文献信息

作者：C Zhang，Z Zhou，Y Zhang，L Yang，K He，Q Wen，L Sun

年份：2022 年

作者单位：达摩研究院（阿里巴巴），阿里云（阿里巴巴）

2 研究问题和意义

2.1 研究问题概述

本文研究的主要问题是网络故障原因的定位算法。

由于网络庞大复杂，我们不可能即时知道网络中具体哪个部分发生了故障，只能从我们探测到的数据集中去推断。我从网上了解到一些故障探测指标，包括设备状态异常警报、链路流量异常、协议状态报告、接口会回环等。这些警告属于网络中不同的层次，只是表面上可观测的信息，而接下来要用定位算法，从不同的数据集中具体分析故障的根本原因，来迅速解决网络故障。可以把算法看做树形结构，数据集是源节点，要找的根本原因就是源节点可以走到的根节点。

2.2 研究现状和挑战

目前已经有了许多自动和智能的根本原因分析算法。这些算法在以前的网络中运行得很好，但目前的 5G 无线网络日益复杂，以前的算法效果变差了。主要有三个挑战。

1.**网络深度增加**，数据集和故障原因之间的关系错综复杂，源节点（探测到的现象）到根节点（根本故障原因）的因果路径会传播错误，影响关系复杂，单一节点故障引起连锁反应，往往很难找到根本原因。

2.**缺乏足够多的标签来识别不同的数据集。**无法充分利用原始数据的信息。因此必须对通信具体细节研究才能缩小归因范围。

3.**提取到的时间序列和各个网络节点的关系是多对多的。** 一个网络节点的变化会影响多个序列，一个序列的值又受到多个网络节点的共同影响。不同变量互相影响，使得因果分析难以实现。

2.3 本论文的创新

本文提出了网络故障原因定位算法 netRCA。它的基础是 XGBoost，在此基础上集成了特征工程、数据扩充和集合模型（规则集学习、预测归因模型、图算法）。本算法提高了网络故障原因分析的正确率。

表 2-1　netRCA 算法的创新点和解决的问题

特征工程	从数据集中获取更多有效信息，找到关系
数据扩充	应对标记数据不足的情况，获得更多标记数据来训练
集合模型	更准确的分析数据特征和根本原因的因果关系

3　netRCA 算法概述

3.1　算法总结构

图 3-1　netRCA 算法结构

如图，netRCA 算法包含三个步骤。第一步是特征工程，原始数据通过特征工程算法生成各种特征集。第二步是数据扩充，根据数据的相似性，通过数据扩充算法把未标记数据打上标签。第三步是集合模型算法。它的基础是 XGBoost。再通过规则集学习、预测归因模型、图算法进一步分析故障根本原因的因果关系。最后得出最可能的根本原因。

3.2　特征工程

原数据可识别的特征很少，这一步是在生成更多的特征，来充分提取数据中的信息。特征分类如下：

表 3-1　特征类型和生成方法

时间特征	数据统计参数，均值、最值、标准差等。
方向相关特征	对于 5G 多天线，测量波束方向和节点的欧几里得距离
属性特征	估计每个特征相对于特征 0（目标特征）的重要性（因果性）具体推导在预测归因模型里面。
交互特征	计算两个互相影响的特征的关系，得出它们的统计信息

生成了这些特征以后，数据集的信息就得到了进一步利用。我的理解是特征工程是找出因果关系的前置准备。特征就是数据提供的信息，不同的特征往往表明了不同的根本原因的影响。特征工程做好了，对数据本身的提取就算完成了。接下来都是对这些特征进行分析。

3.3　数据扩充

我的理解是算法做训练依赖打上根本原因标签的数据集。实际上一半以上的训练集是没有标签的。数据扩充就是把未标记的训练数据加上标签，来扩充训练数据集。这可以极大的扩充训练数据，特别是防止数据太少造成的过拟合现象。

目标：

1.发现训练样本之间的相似性，给相似的数据打上一样的根本原因标签。

2.给相似时间戳的样本打上一样的根本原因标签。这是为了加强对拥有多个根本原因（故障）的数据的分析。经过研究发现，有大量一分钟的时间片段内根本原因 123 共同产生作用。所以时间戳可以作为打上原因标记的依据。

方法：

Eros 算法。Eros 的基础是 Frobenius 范数。设 A 和 B 是两个 MTS 矩阵。把 A 和 B 的协方差矩阵进行奇异值分解，其中的右特征向量矩阵分别为$V_A = [a_1,...,a_n]$和$V_B = [b_1,...,b_n]$。A 和 B 代表的两个数据集的相似度为

$$\mathrm{Eros}(\mathbf{A},\mathbf{B},w) = \sum_{i=1}^{n} w_i |< a_i, b_i >|$$

其中$\omega = [\omega_1,...,\omega_n]$是每一列的权重，取决于特征向量的归一化值。满足$\sum_{i=1}^{n} \omega_i = 1$。

按照公式就可以计算出两个数据集的相似度，从而为未标记数据集打上标签。**我的理解是，这样的算法的好处就是可以衡量不同长度的多元序列的相似性，适应样本长度不一的情况。**

3.4 集合模型

经过前面两步的处理，本步骤就是找出故障根本原因。本算法是几个部分集合在一起。它的基础是已有的 XGBoost 算法，然后增加了规则集学习、预测归因模型、图算法。

（1） 规则集学习

针对问题：

不同特征互相影响，特征和根本原因的关系是非线性的。不能简单地叠加不同特征值来寻找原因。

解决算法：

在目标和特征之间建立逻辑关系，制定一个决策规则集。可以用“if……then”的逻辑来表示这些关系。**我理解可以把寻找根本原因的过程用一个状态转移图来表示。**当遇到特征 A，转移到状态 1，又遇到特征 B，转移到状态 2。最终状态走到根本原因所在的状态，搜索完成。这个规则集的学习可以用 Skope 规则。它是一个决策树结构。流程如下。

① 把初始状态（特征 0）当根节点，中间状态和根本原因当内部节点和叶节点。

② 生成可能的特征和根本原因的因果路径树。

③ 根据精度和召回率，小于阈值的路径是坏路径，剔除。

④ 留下的树就是从特征到根本原因的规则集。

（2） 预测归因模型

针对问题：

对于特征 0 来说，不同的特征和它的关系不同，从而导致特征之间有重要性的差别。**我的理解是，一个故障导致的异常数据可能使得特征 0 发生剧变，那么这个异常代表的特征很可能就指向故障原因。**而对特征 0 影响小的其他特征，它更可能是故障的次生影响。所以在把特征加入路径前，需要估计它们的重要程度。

解决算法：

这是基于 SHAPLEY 值的算法。给定一组特征 S，f 为内部特征（特征的集合）和特征 0 之间的关系，是 XGBoost 算法计算出来的。x_T 是仅包含 T 集合中的特征的 x 的子集，则特征 i 的重要性$\phi(i)$为

$$\phi(i) = \sum_{T \subseteq S \setminus \{i\}} \frac{|T|!(p - |T| - 1)!}{p!} (f(\boldsymbol{x}_{T \cup \{i\}}) - f(\boldsymbol{x}_T)),$$

这个算法仍然存在问题。函数 f 只在所有特征值都准备好才能计算，而且它需要计算所有可能的顺序添加路径得到的收益，时延很高。所以一般用简化的算法，计算特征集 S 的 f 函数和除去了特征 i 的平均值的 f 函数的差。公式如下

$$\phi(i) \approx |f(\boldsymbol{x}_S) - f([\boldsymbol{x}_{S \setminus \{i\}}, \bar{x}_i])|$$

这个公式计算快，误差小。只需要给定一个$\phi(i)$阈值，删除那些不重要的特征，就能更精准地找到根本原因。

我理解这个过程就像去噪，去除由于故障的次生影响和其他干扰影响，找准最主要的特征进行原因分析。

（3） 图算法

算法思想：

图算法提供了从已有的特征到根本原因的一个完整算法过程。它是以上两个算法的结合体，同时考虑了因果关系和特征的相关性。考虑因果关系，用来理清特征的相互影响关系。考虑特征的相关性，能剔除一些不重要的特征，避免因为牵强的特征联系造成归因错误，形成假阳性。

另外，图算法的求解基于如此事实：在图遍历过程中访问某一根本原因旁边的特征越多，该根本原因越可能是要求解的原因。

算法过程：

对于给定的特征 i,可以用皮尔逊相关性计算它和特征 0 之间的相关性。其中 fi 是特征 i 的单变量时间数据。

$$S_i = \left| \frac{\sum_{t=1}^{T}([\mathbf{f}_i]_t - \bar{\mathbf{f}}_i)([\mathbf{f}_0]_t - \bar{\mathbf{f}}_0)}{\sqrt{\sum_{t=1}^{T}([\mathbf{f}_i]_t - \bar{\mathbf{f}}_i)^2}\sqrt{\sum_{t=1}^{T}([\mathbf{f}_0]_t - \bar{\mathbf{f}}_0)^2}} \right|$$

这个式子只是表示数据的相似性，不等于因果性。特征之间的因果性参照规则集学习算法来画出，表现为各个特征相连的一棵树。相邻节点表示具有因果关系。这里结合因果性和相关性得出两个特征的相关指标ω_{ij}。$\omega_{ij} = \frac{A_{ij}S_j}{\sum_j A_{ij}S_j}$。其中$A_{ij}$为因果系数，如果两个节点相连为 1，否则为 0。$\omega_{ij}$计算的是一个节点走到另一个的权重。下面给出流程。

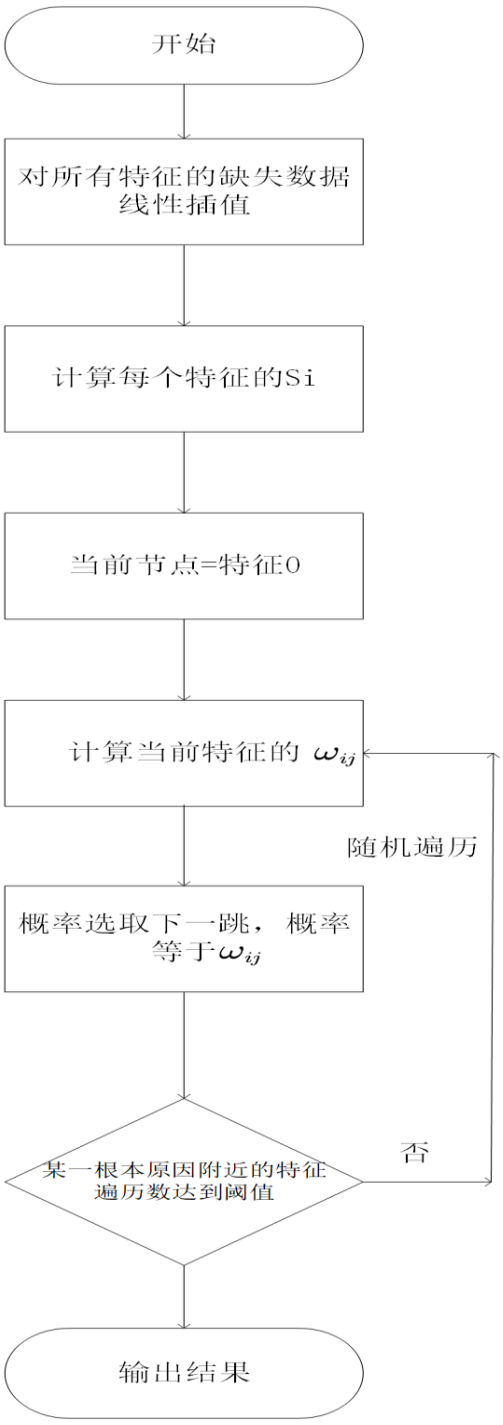

图 3-2　图算法流程图

图算法给出了一个清晰的算法流程。它兼顾了因果性和相似性。

4　实验结论

4.1　实验方法

数据集包括一个固定因果关系图和特征数据集，只有约 45%的样本被标记为根本原因故障，而其他样本则没有标记。为了检验算法的作用，采用对比法，基本的 XGB 模型、XGB 加上特征工程、XGB 加上特征工程加图算法、完整的 NetRCA 算法分别对数据进行分析。每个

真阳性+1 分，假阳性-1 分，最终分数归一化。

4.2 实验结论

对比实验结果如下

Table 1: Ablation studies of the proposed NetRCA model.

Models	Root1 acc	Root2 acc	Root3 acc	Final Score
XGB	0.9828	0.97849	0.9957	0.78139
XGB+FE	0.9957	0.97849	0.9914	0.86611
XGB+FE+Graph	0.9957	0.97849	0.9914	0.87917
Proposed NetRCA	0.9957	0.98495	0.9914	**0.91778**

图 4-1　不同模型的性能对比

下面是我得出的结论

表 4-1　不同算法的作用

特征工程	分数提高 8%，提取特征增加了原始数据的信息，使得过拟合大大降低，能够发现更多关联性
图算法	提高 1%。图算法有利于捕捉特征的因果关系，去除不必要的联系，消除假阳性
数据扩充	对根本原因 2 的分析更准确。因为根本原因 2 的标签数据不足，用数据扩充能得到更多有用信息，加强学习

4.3 我的总结

本文提出一种新的网络故障根源定位算法。本算法有效加强了故障定位的准确度。它充分的利用了现有的模型，在这个基础上去集合其他模型，从而推断出因果性。并且运用了扩充数据、特征工程这些算法来获取更多信息。

本文的创新并不是用通信网本身的理论，而是把网络抽象成数学模型，用经典的算法去解决。特征工程基于数理统计，特征向量基于 Frobenius 范数，图算法和规则集模型基于图论的搜索算法和相关性的理论。**这说明应用层的人工智能算法和通信网是分不开的**，我们要学会把实际问题转化为数学模型，用成熟的经典理论去解决。这样会让我们的学习更扎实，更有办法去解决实际问题。