软件学报 ISSN 1000-9825, CODEN RUXUEW
Journal of Software, 2023,34(3):1126−1147 [doi: 10.13328/j.cnki.jos.006781]

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

基于联邦学习的跨源数据错误检测方法*

陈 璐[1], 郭宇翔[1], 葛丛丛[2], 郑白桦[3], 高云君[1]

[1](浙江大学 计算机科学与技术学院, 浙江 杭州 310027)

[2](华为云计算公司 数据智能创新 Lab, 浙江 杭州 310052)

[3](School of Computing and Information Systems, Singapore Management University, Singapore)

通信作者: 高云君, E-mail: gaoyj@zju.edu.cn

摘 要: 随着海量数据的涌现和不断积累, 数据治理成为提高数据质量、最大化数据价值的重要手段. 其中, 数据错误检测是提高数据质量的关键步骤, 近年来引起了学术界及工业界的广泛关注. 目前, 绝大多数错误检测方法只适用于单数据源场景. 然而在现实场景中, 数据往往不集中存储与管理. 不同来源且高度相关的数据能够提升错误检测的精度. 但由于数据隐私安全问题, 跨源数据往往不允许集中共享. 鉴于此, 提出了一种基于联邦学习的跨源数据错误检测方法 FeLeDetect, 以在数据隐私保证的前提下, 利用跨源数据信息提高错误检测精度. 为了充分捕获每一个数据源的数据特征, 首先提出一种基于图的错误检测模型 GEDM, 并在此基础上设计了一种联邦协同训练算法 FCTA, 以支持在各方数据不出本地的前提下, 利用跨源数据协同训练 GEDM. 此外, 为了降低联邦训练的通信开销和人工标注成本, 还提出了一系列优化方法. 最后, 在 3 个真实数据集上进行了大量的实验. 实验结果表明: (1) 相较于 5 种现有最先进的错误检测方法, GEDM 在本地场景和集中场景下, 错误检测结果的 *F*1 分数平均提高了 10.3%和 25.2%; (2) FeLeDetect 错误检测结果的 *F*1 分数较本地场景下 GEDM 的结果平均提升了 23.2%.

关键词: 数据治理; 数据质量; 错误检测; 联邦学习

中图法分类号: TP311

中文引用格式: 陈璐, 郭宇翔, 葛丛丛, 郑白桦, 高云君. 基于联邦学习的跨源数据错误检测方法. 软件学报, 2023, 34(3): 1126−1147. http://www.jos.org.cn/1000-9825/6781.htm

英文引用格式: Chen L, Guo YX, Ge CC, Zheng BH, Gao YJ. Cross-source Data Error Detection Approach Based on Federated Learning. Ruan Jian Xue Bao/Journal of Software, 2023, 34(3): 1126−1147 (in Chinese). http://www.jos.org.cn/1000-9825/6781.htm

Cross-source Data Error Detection Approach Based on Federated Learning

CHEN Lu[1], GUO Yu-Xiang[1], GE Cong-Cong[2], ZHENG Bai-Hua[3], GAO Yun-Jun[1]

[1](College of Computer Science and Technology, Zhejiang University, Hangzhou 310027, China)

[2](Data Intelligence Innovation Lab, Huawei Cloud Computing Technologies Co. Ltd., Hangzhou 310052, China)

[3](School of Computing and Information Systems, Singapore Management University, Singapore)

Abstract: With the emergence and accumulation of massive data, data governance has become an important manner to improve data quality and maximize data value. Error detection is crucial for improving data quality, which has attracted a surge of interests from both industry and academia. Various detection methods tailored for a single data source have been proposed. Nevertheless, in many real-world scenarios, data is not centrally stored and managed. Different sources of correlated data can be employed to improve the accuracy of error detection. Unfortunately, due to privacy/security issues, cross-source data is often not allowed to be integrated centrally. To this end, this study proposes FeLeDetect, a cross-source data error detection method based on federated learning. First, a graph-based error detection

* 基金项目: 国家重点研发计划(2021YFC3300303); 国家自然科学基金(62025206, 61972338, 62102351)
本文由“大数据治理的理论与技术”专题特约编辑杜小勇教授、杨晓春教授和童咏昕教授推荐.
收稿时间: 2022-05-13; 修改时间: 2022-07-29, 2022-09-07; 采用时间: 2022-09-23; jos 在线出版时间: 2022-10-27

model (GEDM) is presented to capture sufficient data features from each data source. Then, the study investigates a federated co-training algorithm (FCTA) to collaboratively train GEDM over different data sources without privacy leakage. Furthermore, the study designs a series of optimization methods to reduce the communication cost during the federated learning and the manual labeling efforts. Extensive experiments on three real-life datasets demonstrate that GEDM achieves an average improvement of 10.3% F1-score in the local scenario and 25.2% F1-score in the centralized scenario, outperforming all the five existing state-of-the-art competitors for a single data source; and FeLeDetect further enhances local GEDM in terms of F1-score by 23.2% on average.

Key words: data governance; data quality; error detection; federated learning

随着移动设备、物联网设备的普及以及互联网技术的飞速发展, 海量数据不断涌现和积累. 通过对海量数据的分析与挖掘, 可以获取潜在的数据价值. 错误检测往往是数据分析流程中的第一步[1,2], 这是因为数据错误将严重影响数据质量[3], 误导下游的决策分析, 甚至对企业造成巨额经济打击[4]. 因此, 近年来, 数据错误检测受到了学术界与工业界的广泛关注和深入研究.

数据错误的原因多种多样, 例如人为输入的错别字、整合不同来源数据时造成的不一致等. 常见的错误类型包括错别字、数据缺失、格式错误、违反数据一致性规则等. 在现实场景中, 数据错误往往是异质且稀疏的[4], 这加剧了错误检测的难度. 目前, 错误检测方法主要可以分为以下几类: (1) 离群值检测[5−7], 基于统计方法检测出明显偏离数据整体分布的错误值; (2) 重复值检测[8,9], 通过识别指向同一实体的元组以删除重复值; (3) 格式错误检测[10], 通过预先定义的模式(如日期格式 2000/01/01)以检测出格式错误; (4) 违反规则检测[11,12], 通过给定的数据规则检测违反规则的数据错误; (5) 违反外部知识检测, 利用外部知识(如主数据[13]、知识库[14])检测数据错误. 以上这些方法都是针对单一数据源的, 然而在真实场景中, 即使描述真实世界中同一组实体的数据往往也不是集中存储与管理的. 结合跨源数据信息, 一些仅依赖单源信息难以被识别出的错误能够更容易地被检测出来.

为了便于理解, 图 1 给出了一个示例. 来自不同数据源(DBLP 与 ACM)的两张数据表记录了一些计算机科学领域的会议和期刊论文信息. 每张表中, 相同 ID 的元组描述的是同一篇论文信息. 其中, 有两个数据单元存在替换错误(以高亮表示, 记作 E_D 和 E_A), 分别位于 DBLP 数据表的第 1 条元组(该论文正确的发表信息为“SIGMOD Conference”而非“SIGMOD Record”)以及 ACM 数据表的第 1 条元组(论文正确的标题为“A Compact B-Tree”而非“Bit-Sliced Index Arithmetic”). 错误 E_D 仅凭 DBLP 数据表很难被检测出来, E_A 仅凭 ACM 数据表也难以识别. 换言之, 这两处错误难以被上述面向单源数据的错误检测方法检测出来: (1) “SIGMOD Record”和“Bit-Sliced Index Arithmetic”均为其所在数据表属性域中的合法取值, 且不能被视作离群值; (2) 两数据表均不存在重复记录, 因而无法利用重复记录检测识别错误; (3) 两处错误取值在语法和语义层面均为正确, 且不存在格式问题; (4) 两处错误在单表中无法被数据规则捕获; (5) 不存在关于这两个数据表的主数据(master data)或外部知识, 且建立主数据库或外部知识库需耗费大量的人力成本和计算资源. 因此, 利用外部知识进行错误检测有一定难度.

DBLP

D.Title	D.Pages	D.Authors	D.Venue	D.Year
A Compact B-Tree	533-541	Ivan T. Bowman, Peter Bumbulis	SIGMOD Record	2002
Bit-Sliced Index Arithmatic	47-57	Elizabeth J. O’Neil, Denis Rinfreet, ect.	SIGMOD Conference	2001
Updating XML	413-424	Daniel S. Weld, etc.	SIGMOD Conference	2001

ACM

A.Title	A.Keywords	A.Authors	A.Venue	A.Year
Bit-Sliced Index Arithmatic	B-Tree	Ivan T. Bowman, Peter Bumbulis	International Conference on Management of Data	2002
Bit-Sliced Index Arithmatic	Bit-Sliced	Elizabeth J. O’Neil, Denis Rinfreet, ect.	International Conference on Management of Data	2001
Updating XML	XML	Daniel S. Weld, etc.	International Conference on Management of Data	2001

图 1 跨源数据集示例

然而, 若将 DBLP 与 ACM 传输至同一数据中心, 合并为一联合数据集, 如图 2 所示, 错误检测的难度将大大降低. 尽管这两个数据表来自不同数据源(DBLP 和 ACM), 但它们描述的是同一组实体(论文)的相关信息. 通过观察可以发现, 对于同一篇论文, D.Venue 的取值“SIGMOD Conference”和 A.Venue 的取值“International Conference on Management of Data”存在高频共现的规律, 因而可以推测它们指代同一个学术会议的名称. 然而, 表 DBLP 与表 ACM 中第 1 条元组的 Venue 值打破了“SIGMOD Conference”与“International Conference on Management of Data”的高频共现规律, 因此容易推断出至少有一个数据单元的值是错误的. 由

于数据错误并不是随机分布的[15], 因此可以通过机器学习的方式学习数据集中错误分布的规律, 以预测哪一个数据单元取值有误的可能性更大, 从而进行更精确的推断. 例如, 在该数据集中, D.Venue 属性域中存在与“SIGMOD Record”十分相似的合法取值“SIGMOD Conference”, 而在 A.Venue 属性域中不存在与“International Conference on Management of Data”相似的取值, 因此相对而言, “SIGMOD Record”与“SIGMOD Conference”发生混淆的概率更高. 若训练集中的数据错误符合上述分布规律, 模型将学习到该知识并应用于测试集. 基于此, 本文研究跨源数据错误检测方法, 其主要有以下三大挑战.

(1) 如何表征不同粒度的数据特征? 如上文所述, 真实场景下, 由于数据错误的异质性, 导致其难以统一表征. 一种有效应对错误异质性的方法是利用机器学习技术, 将错误检测视作二分类问题: 给定一个数据集和一些训练标签, 通过学习错误数据单元与正确数据单元的特征, 以预测各个数据单元取值是否错误. 所以, 如何有效表征不同粒度的数据单元特征, 对高精度的错误检测至关重要.

(2) 如何保证跨源错误检测中的数据隐私安全? 如图 2 所示, 跨源数据能够有效地提升错误检测的质量. 然而, 由于数据隐私安全的原因(如欧洲颁布的《通用数据保护条例》[16]), 在真实应用场景中, 不同来源的数据往往不允许传输至公共数据中心进行集成. 因而亟需研究一种跨源错误检测方法, 使其不仅能够有效地利用跨源数据信息以提升错误检测质量, 而且还能保证各数据源的数据隐私安全.

(3) 如何减少跨源数据错误检测的通信代价? 在跨源错误检测过程中, 不同数据源之间需要进行频繁的信息交换以获取必要的跨源信息, 由此造成的通信开销不容忽视. 因此, 如何在保证错误检测精度的同时, 尽可能地降低跨源错误检测过程所需的通信开销也是一大挑战.

DBLP-ACM

D.Title	D.Pages	D.Authors	D.Venue	D.Year	A.Title	A.Keywords	A.Authors	A.Venue	A.Year
A Compact B-Tree	533-541	Ivan T. Bowman, Peter Bumbulis	SIGMOD Record	2002	Bit-Sliced Index Arithmatic	B-Tree	Ivan T. Bowman, Peter Bumbulis	International Conference on Management of Data	2002
Bit-Sliced Index Arithmatic	47-57	Elizabeth J. O’Neil, Denis Rinfreet, ect.	SIGMOD Conference	2001	Bit-Sliced Index Arithmatic	Bit-Sliced	Elizabeth J. O’Neil, Denis Rinfreet, ect.	International Conference on Management of Data	2001
Updating XML	413-424	Daniel S. Weld, etc.	SIGMOD Conference	2001	Updating XML	XML	Daniel S. Weld, etc.	International Conference on Management of Data	2001

图 2 集中式联合数据集

为了应对上述挑战, 本文提出了一种基于联邦学习的跨源错误检测方法 FeLeDetect. 首先, 考虑到图结构能够有效地表示关系型数据特征[17], 本文设计了一种基于图的错误检测模型 GEDM (graph-based error detection model), 以捕获每个数据源不同粒度的特征(包括属性值级特征、属性级特征和元组级特征), 从而为分类器提供精确的分类信号. 尽管 GEDM 能够有效捕获输入数据集的数据特征, 但当输入数据集为纵向划分的子集时(如图 1 中的 DBLP), 由于无法访问另一相关数据集(如图 1 中的 ACM), 造成 GEDM 性能受限. 鉴于联邦学习技术[18]能够在保证数据隐私的前提下协同训练机器学习模型, 在 GEDM 的基础上, 本文提出了一种信息无损的联邦协同训练算法 FCTA (federated co-training algorithm), 以协同训练部署在不同数据源的 GEDM, 在确保跨源数据的隐私安全前提下, 使算法能够捕获跨源数据特征, 以达到接近于在联合数据集(如图 2 所示)上运行 GEDM 的效果. 最后, 为了降低联邦训练过程中的通信开销,本文进一步提出了 3 种优化策略: (1) 数据去重, 以确保跨源交换的数据中没有重复信息; (2) 量化压缩, 将连续数据进行量化压缩, 以降低数据编码所需的字节数; (3) 降频传输, 根据相似性阈值执行过滤优化, 以降低数据交换的频率.

本文工作的主要贡献可以总结为以下 4 点.

(1) 提出了一种基于联邦学习的跨源数据错误检测方法 FeLeDetect. 该方法利用跨源数据, 在隐私保护的前提下, 大大提高了数据错误检测的精度.

(2) 设计了一种基于图的错误检测模型 GEDM. 该模型能够捕获每个数据源不同粒度的丰富数据特征, 以支持高质量的错误检测结果.

(3) 提出了一种信息无损的联邦协同训练算法 FCTA, 以协同训练部署在不同数据源的模型 GEDM, 确保跨源数据的隐私安全. 此外, 本文还提出了若干优化策略以降低联邦训练的通信开销.

(4) 在 3 个真实数据集上进行了充分的实验评估, 实验结果表明, 相较于 5 种现有先进的错误检测方法,

GEDM 与 FeLeDetect 有效地提高了错误检测的精度.

1 相关工作

本节介绍相关工作. 第 1.1 节回顾错误检测的相关工作. 第 1.2 节介绍联邦学习的相关工作.

1.1 错误检测

数据错误检测是提高数据质量的关键流程. 传统的错误检测方法可以分为定量方法和定性方法两类: 定量方法[5−7]利用统计知识, 根据数据预期的统计分布识别数据错误; 定性的方法利用规则[11,12]、模式[10,19]或外部知识[13,14]检测数据错误. 然而, 由于错误的异质性, 这些方法难以检测出关系表中同时存在的多类型错误, 造成错误检测的召回率偏低. 近年来, 基于机器学习/深度学习技术的错误检测方法[2,4,20−23]受到广泛关注, 其旨在通过学习的方式捕获数据错误的特征, 利用训练标签以指导分类器做出较为精准的判断, 从而有效地解决了错误异质性挑战. 在此类错误检测方法中, HoloDetect[2]和 Raha[20]取得了最佳的错误检测效果: HoloDetect[2]通过表征数据内在的语义和语法特征, 并根据人为给定的数据一致性规则检测错误数据单元; Raha[20]系统地生成一组错误检测算法的配置参数, 并将算法的输出编码为每个数据单元的特征向量, 以学习数据错误的特征. Raha 取得了与 HoloDetect 相当的错误检测效果, 然而, Raha 配置与运行多种错误检测算法的过程十分耗时. 与Raha相比, 本文提出的基于图的错误检测模型GEDM不仅能够充分捕获不同粒度的数据特征, 而且在效率上优于 Raha, 这将在第 5.4 节的实验部分得以验证.

1.2 联邦学习

联邦学习[24]是一种分布式机器学习技术, 以支持不同的数据参与方在不披露自身原始数据的前提下共同建立机器学习模型. 目前, 联邦学习技术已成功应用于多个领域, 比如智能推荐[25]、智慧医疗[26]、金融犯罪检测[27]等. 因此, 一些联邦学习框架也被相应提出, 如 FedATT[28]、FedProx[29]、FedKD[30]等. 这些计算框架主要面向横向联邦学习(即数据横向划分). 本文将纵向联邦学习技术应用于数据治理领域, 利用跨源数据信息提高错误检测精度, 从而提升数据质量. 由于联邦学习在模型训练过程中需要各数据参与方进行频繁的数据交换, 因此, 如何降低联邦训练过程中由于数据频繁交换带来的高昂通信代价, 是联邦学习领域的一个研究重点[31−33]. 本文在提出基于联邦学习的错误检测方法 FeLeDetect 后, 进一步设计了若干有效的通信优化方案, 从不同层面减少跨源部署的模型 GEDM 在协同训练过程中的高通信量. 这些优化策略在保证错误检测有效性的前提下, 大大降低了 FeLeDetect 的通信代价.

2 基础知识

本节介绍本文工作的基础知识. 第 2.1 节介绍数据错误与错误检测的基本概念. 第 2.2 节介绍联邦学习的相关知识. 第 2.3 节给出本文所研究问题的具体定义.

2.1 数据错误与错误检测

给定一个含错误数据的关系表 $D=\{t_1,t_2,\ldots,t_n\}$, 其属性集 $A=\{a_1,a_2,\ldots,a_n\}$. 其中, 每个元组(记录)$t_i\in D$ 是一组数据单元的集合$\{t_i[a_1],t_i[a_2],\ldots,t_i[a_m]\}$. 每个元组描述一个真实世界中的实体. 将某数据单元 $t_i[a_j]$在表 D 中的取值记 v_{ij}, 用于描述实体 t_i在某特征维度 a_j下的特征值. 将数据单元 $t_i[a_j]$的真实值记作 v_{ij}^*. 若 $v_{ij}\neq v_{ij}^*$, 则称该数据单元为错误数据单元, v_{ij}为一个数据错误. 例如, 图 1 中展示了两个关系表 DBLP 和 ACM(分别简记作 D 和 D'). 每个关系表包含了 3 条元组. 由于数据单元 $t_1[a_4]$和 $t_1'[a_1]$的真实值分别为“SIGMOD Conference”和“A Compact B-Tree”, 均不等于其在 D 和 D'中的取值, 因此, 数据单元 $t_1[a_4]$和 $t_1'[a_1]$均为错误数据单元.

错误检测的目的是识别出关系表中所有的错误数据, 即图 1 中的 $t_1[a_4]$和 $t_1'[a_1]$. 本文将错误检测视作一个二分类问题: 给定一个含有错误数据的关系表 D 以及一些训练标签 L, 错误检测旨在为每个数据单元 $t_i[a_j]$赋予一个分类标签 $\hat{y}\in\{0,1\}$, 其中, $\hat{y}=1$表示该单元为错误数据单元, $\hat{y}=0$ 表示该单元为正确数据单元.

2.2 联邦学习

在现实世界中, 不同组织/机构所持有的数据往往各自定义各自管理. 不同来源的数据形成孤岛, 难以流通共享. 由于行业竞争、隐私安全以及复杂的管理机制等, 即便是同一公司不同部门间的数据集成都面临巨大阻力[34]. 实现跨机构、跨组织的数据共享更加困难. 为了应对这一挑战, 谷歌公司提出了联邦学习[18]的概念. 联邦学习是一种分布式机器学习技术, 旨在利用分布于不同位置的数据共同建立机器学习模型, 同时保证跨源数据的隐私安全. 因此, 利用联邦学习技术可以在不牺牲隐私的前提下, 有效地利用跨源数据信息. 根据数据参与方的特征空间以及所描述实体空间的不同, 联邦学习技术可以分为 3 类: 横向联邦学习、纵向联邦学习和迁移学习[34]. 横向联邦学习适用于数据参与方共享相同的特征空间但实体空间不同的场景(即各参与方的数据由横向划分而来); 纵向联邦学习适用于数据参与方共享相同的实体空间但特征空间不同的情况(即各参与方的数据由纵向划分而来); 迁移学习则关注数据参与方的特征空间与实体空间均不相同的场景.

本文研究的跨源错误检测与纵向联邦学习研究的场景高度相关. 正如图 1 所示, (1) DBLP 和 ACM 描述的是同一组实体(会议和期刊论文)的信息; (2) 图 1 中, 双方各自持有的本地数据集 DBLP 与 ACM 可以视作由图 2 的联合数据集 DBLP-ACM 纵向划分而来. 为了实现跨源数据错误检测, 一个最直接的方法是: 先将数据持有方 F 和 F' 各自所有的数据 D 和 D' 传输至一个公共数据中心后, 合并为一个联合数据集 $D_S=D\cup D'$; 而后, 在此数据集 D_S 上建立一个机器学习模型 M_S. 但正如上文所述, 出于隐私安全的考量, 这种集中式的建模方式在现实场景中往往是不可行的. 与之不同, 基于纵向联邦学习的跨源错误检测旨在利用数据集 D 和 D' 共同建立错误模型 M_F, 且保证跨源数据不出本地. 所以, 其无需不同来源的原始数据执行传输和集中共享, 从而保证了跨源数据的隐私安全. 此外, 基于联邦学习的方法要求联邦模型 M_F 的效果接近于数据集中场景下所建立的模型 M_S.

2.3 问题定义

本文关注关系型数据的错误检测问题.

- 一方面, 跨源数据可以提升错误检测的质量. 如图 1 所示, 现有的单源错误检测方法难以有效地检测出图 1 中的替换错误. 然而, 利用跨源数据信息可以提高错误检测的精度. 因此, 遵循相关研究工作所做的一般性假设[35,36], 本文假定不同数据持有方都愿意参与该跨源错误检测过程, 以提高自身的数据质量.
- 另一方面, 数据隐私是一个亟待解决的全球性问题. 为此, 本文研究了隐私保护下跨源数据错误检测问题.

定义 1(隐私保护下跨源数据错误检测). 给定两个含有错误数据的关系表 D 和 D', 其分别为不同数据持有方 F 和 F' 所有(尽管在定义中只涉及两个数据持有方, 但本文提出的方法可以拓展至多数据持有方. 然而, 由于复杂的管理和利益分配机制, 多数据持有方在真实的纵向联邦场景中并不常见[35]). 考虑到数据隐私安全, 双方均不允许原始数据离开本地. 隐私保护下跨源数据错误检测旨在结合跨源数据 D 和 D' 的信息, 在保证双方数据隐私安全的前提下, 检测出数据 D 和 D' 中的所有错误.

由于不允许跨源数据的集中共享, 本文利用联邦学习技术进行跨源数据错误检测. 正如上文所述, 双方各自持有的数据可以视作由联合数据集 D_s 纵向划分而来, 因此, 该问题属于纵向联邦学习的研究范畴. 一个典型的纵向联邦学习系统主要包括两个部分, 分别为加密实体对齐和联邦学习[34]. 加密实体对齐旨在利用现有技术[35]识别出两个数据源中指向同一实体的记录. 本文遵循纵向联邦学习研究的一般性假设: 两个数据集的记录已被完全对齐, 即已完成加密实体对齐, 仅关注联邦学习阶段[36,37]. 例如, 图 1 中的关系数据表 DBLP 与 ACM 已完全对齐, 相同 ID 的元组均描述同一篇论文的信息.

3 基于图的错误检测模型

本节介绍基于图的错误检测模型 GEDM. 第 3.1 节介绍多关系图模型构建. 第 3.2 节提出了一种基于图的

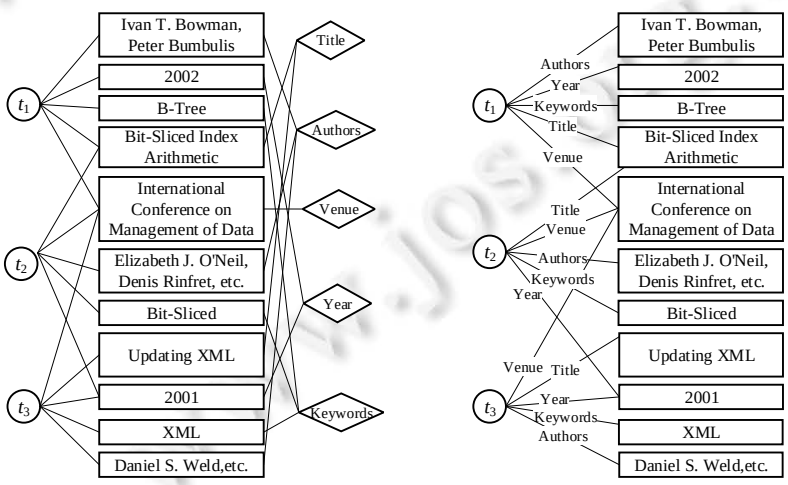
错误检测模型 GEDM, 该模型能够捕获每个数据源中不同粒度的丰富特征.

3.1 多关系图构建

近年来研究表明[17], 将关系型数据表示为图结构有利于捕获数据的内在特征. 鉴于此, 本文先将关系型数据转化为图, 而后进行建模. Cappuzzo 等人[17]提出的 EMBDI 模型将关系表中的每一个元组(行)、每一个属性(列)以及每一个数据单元的取值建模为图中的 3 类节点. 这些节点根据关系表所描述的逻辑关系进行连接. 图 3(a)给出了一种基于图 1 中 ACM 数据集建立的 EMBDI 模型, 其中, 以矩形表示的节点(如“Ivan T. Bowman, Peter Bumbulis”)对应于表 ACM 中数据单元的取值, 圆形节点(如 t_1)对应表 ACM 的第 1 条元组, 菱形节点(例如“Title”)对应于表 ACM 中的 Title 属性. 然而, 该图模型存在两个问题.

- 其一, 利用该模型对关系表进行转化建模会产生结构复杂的大规模图, 其包含了大量的节点和边. 存储大图需要耗费大量的存储资源, 同时, 对于后续过程中基于图的训练也是极大的挑战.
- 其二, 该图模型并未考虑边的语义信息. 例如, 连接元组节点和属性节点的边与连接属性节点和单元值节点的边在语义上并不相同, 而该图模型并未区分这些不同语义的边.

为了解决这两点不足, 本文利用多关系图模型 MRG[38]以实现关系表到图的转化. 该多关系图模型为一个二部图, 即存在两类节点, 分别为元组级节点(记为 t)和值节点(记为 v). 这两类节点与三部图模型中定义的元组节点和单元值节点相同. 与之不同的是, 多关系图模型引入了属性级边(记为 a)的概念, 以连接 t 节点和 v 节点. 具体来说, 在原关系表中, 若元组 t 在属性 a 上取值为 v, 则在对应的多关系图中将 t 节点与 v 节点通过属性级边 a 相连. 图 3(b)描述了图 1 中的 ACM 数据集按照多关系图模型 MRG 进行转化的示例. 例如, 连接元组级节点 t_1 和值节点“2002”的是属性“Year”代表的边, 对应的语义为: ACM 数据集中第 1 个元组所描述的论文发表年份是 2002 年. 对比图 3(a)与图 3(b)可以看出, 对于同一个关系表而言, 相较于 EMBDI 模型, MRG 构建的图的边数和节点数都有所减少. 特别是边数从 $2mn$ 降低至 mn, 其中, m 表示原关系表中的属性个数(列数), n 表示原关系表中的元组数(行数). 不仅如此, 多关系图模型中的每条边都被赋予了语义信息(属性名), 这有利于表征不同节点之间的语义关系.



(a) EMBDI (b) MRG

图 3 EMBDI 与 MRG 图构建对比示例

3.2 模型设计

图 4 展示了基于图的错误检测模型 GEDM 框架. GEDM 由 3 个部分组成: (1) 图构建; (2) 基于图的特征提取; (3) 二分类器. 具体而言:

- 在模型训练(或错误检测)阶段, GEDM 先将训练数据集 T(或完整数据集 D)转化为一个多关系图.

- 接着，利用图神经网络提取 3 个不同维度的数据特征：元组级特征、属性级特征以及属性值级特征(简称为值级特征).
- 最后，使用一个二分类器判断每一个数据单元取值是否错误.

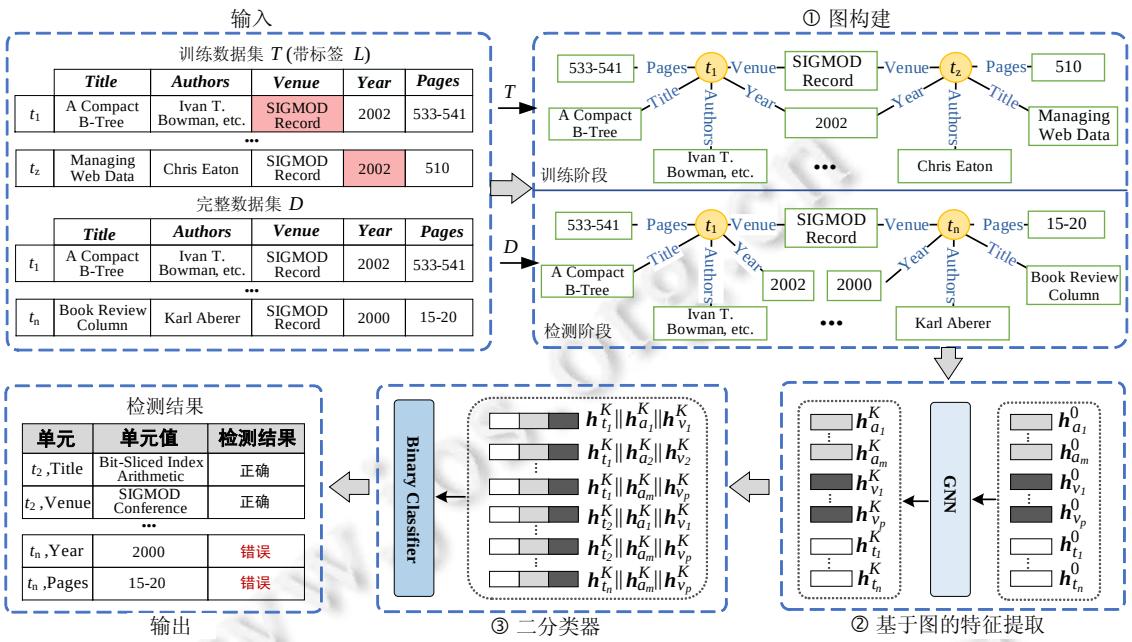


图 4 错误检测模型 GEDM 框架

鉴于在第 3.1 节已经介绍了 MRG 多关系图构建的过程，下面仅介绍基于图的特征提取和二分类器模块.

- 基于图的特征学习

将数据表转化为图之后，需要对图中每个节点和边进行特征表示，以捕获原数据表中丰富的数据特征. 近年来，图神经网络(GNN)[39]受到了普遍关注. 得益于其强大的特征捕获能力，图神经网络在许多领域都受到了广泛关注[40−42]. 为此，GEDM 利用图神经网络来提取特征，以支持高精度的错误检测任务. 给定一个多关系图 G，GEDM 首先随机初始化图中的元组级节点、值级节点和属性级边，初始特征向量并分别记作 $\boldsymbol{h}_t^0$,$\boldsymbol{h}_v^0$ 和 $\boldsymbol{h}_a^0$；接着，利用本文提出的多粒度图卷积神经网络 MGGCN，不断更新特征向量以学习蕴含不同粒度数据特征的向量最终表示. 图卷积神经网络 MGGCN 的每个卷积层都设计了针对元组级节点、值级节点和属性级边的卷积操作，以聚合图上邻居节点特征，充分捕获图结构信息. 与文献[43]类似，MGGCN 采用了归纳式学习的思想，在训练过程中只使用训练集 T 而非完整数据集 D，以达到更高的效率. 具体来说，模型训练的目的是通过学习以获得最优的模型参数，而非获得图上每个节点和边的特征向量表示[43]. 一旦模型训练完毕，在错误检测阶段，即可利用训练好的模型快速地表示出整个数据集 D 中每个数据单元的特征向量. 接下来详细介绍本文提出的多粒度图卷积神经网络 MGGCN 中各层的卷积操作.

➢ 元组级节点卷积操作：在 MGGCN 的第 k ($\geqslant$1)层，元组级节点 t 的特征向量通过以下公式计算得到:

$$\boldsymbol{h}_{\mathcal{N}(t)}^k = AGG_{(a,v)\in\mathcal{N}(t)}(\boldsymbol{W}_{ta}^k\boldsymbol{h}_a^{k-1}\odot\boldsymbol{W}_{tv}^k\boldsymbol{h}_v^{k-1}) \tag{1}$$

$$\boldsymbol{h}_t^k = \sigma(\boldsymbol{W}_t^k(\boldsymbol{h}_t^{k-1}\,\|\,\boldsymbol{h}_{\mathcal{N}(t)}^k)) \tag{2}$$

其中，$\boldsymbol{W}_{ta}^k$, $\boldsymbol{W}_{tv}^k$, $\boldsymbol{W}_t^k$ 为待学习的权重矩阵；$\boldsymbol{h}_t^{k-1}$, $\boldsymbol{h}_v^{k-1}$, $\boldsymbol{h}_a^{k-1}$ 为由第(k−1)层网络计算得到的特征向量；$(a,v)\in\mathcal{N}(t)$表示与元组级节点 t 直接相连的属性级边和单元级节点的集合；$AGG(\cdot)$是一个聚合函数，在本文中使用均值函数；$\odot$为哈达玛乘积；$\sigma(\cdot)$为激活函数，本文使用 tanh 函数实现；$\|$为拼接操作符.

➢ 值级节点卷积操作：在 MGGCN 的第 k ($\geqslant$1)层，值级节点 v 的特征向量通过以下公式计算得到:

$$\boldsymbol{h}_{\mathcal{N}(v)}^{k}=AGG_{(a,t)\in\mathcal{N}(v)}(\boldsymbol{W}_{va}^{k}\boldsymbol{h}_{a}^{k-1}\odot\boldsymbol{W}_{vt}^{k}\boldsymbol{h}_{t}^{k-1}) \tag{3}$$

$$\boldsymbol{h}_{v}^{k}=\sigma(\boldsymbol{W}_{v}^{k}(\boldsymbol{h}_{v}^{k-1}\|\boldsymbol{h}_{\mathcal{N}(v)}^{k})) \tag{4}$$

其中, $\boldsymbol{W}_{va}^{k}$, $\boldsymbol{W}_{vt}^{k}$, $\boldsymbol{W}_{v}^{k}$ 为待学习的权重矩阵, $\boldsymbol{h}_{t}^{k-1}$, $\boldsymbol{h}_{v}^{k-1}$, $\boldsymbol{h}_{a}^{k-1}$ 为由第(k–1)层网络计算得到的特征向量, $(a,t)\in\mathcal{N}(v)$表示与值级节点 v 直接相连的属性级边和元组级节点的集合.

- 属性级边卷积操作: 在 MGGCN 的第 k ($\geqslant$1)层, 属性级边 a 的特征向量通过以下公式计算得到:

$$\boldsymbol{h}_{a}^{k}=\boldsymbol{W}_{a}^{k}\boldsymbol{h}_{a}^{k-1} \tag{5}$$

其中, $\boldsymbol{W}_{a}^{k}$ 为待学习的权重矩阵, $\boldsymbol{h}_{a}^{k-1}$ 为由第(k–1)层网络计算得到的特征向量.

- 二分类器

经过若干层 MGGCN 的堆叠, 得到每个元组/值级节点以及属性级边的最终特征向量表示. 对于原关系数据表中的某数据单元 v=t[a], 将与之对应的节点和边的最终特征向量 $\boldsymbol{h}_{t}$, $\boldsymbol{h}_{a}$, $\boldsymbol{h}_{v}$ 进行拼接, 得到一个新的向量 $\boldsymbol{c}$=$\boldsymbol{h}_{t}\|\boldsymbol{h}_{a}\|\boldsymbol{h}_{v}$, 并作为二类器 C 的输入. 该二分类器输出一个类别标签 $\hat{y}$ 以指示该数据单元取值正确或错误. 二分类器 C 由一个两层全连接神经网络和 Softmax 层构成, 使用 ReLu 激活函数, 并选择交叉熵损失(记作$\mathcal{L}$)作为目标函数, 定义如下:

$$\mathcal{L}=\frac{1}{M}\sum_{i=1}^{M}\mathcal{L}_{i}=-\frac{1}{M}\sum_{i=1}^{M}y_{i}\log p_{i}+(1-y_{i})\log(1-p_{i}) \tag{6}$$

其中, M 表示训练集的样本数目, y_i 为第 i 个训练样本的真实标签值(y_i=1 表示错误数据单元, y_i=0 表示正确数据单元), p_i 表示该训练样本被预测为错误数据的概率. 注意, 上述 MGGCN 网络是与二分类器 C 一起训练的.

4 联邦错误检测

本节介绍联邦错误检测方法 FeLeDetect. 首先介绍 FeLeDetect 的框架, 其次介绍 FeLeDetect 的技术细节, 最后提出了若干优化方法以减少联邦训练过程中的通信开销和人工标注成本.

4.1 FeLeDetect框架

基于第 3 节提出的错误检测模型 GEDM, 本节提出了基于联邦学习的跨源错误检测方法 FeLeDetect. 给定两个数据参与方 F 和 F', 分别持有数据 D 和 D'. 训练集 $T\subset D$ 和 $T'\subset D'$包含的元组已经完成对齐. 正如上文所述, F 和 F'不允许原始数据离开本地. FeLeDetect 利用协同训练算法 FCTA 在数据集 T 和 T'上协同训练 GEDM, 在此期间不涉及任何原始数据的交换, 从而保证了双方的数据隐私安全. 为方便起见, 下文只使用两个数据参与方进行阐述, 但 FeLeDetect 方法可拓展至多数据参与方的场景.

FeLeDetect 在数据持有方 F 和 F'本地分别部署错误检测模型 GEDM, 并根据算法 FCTA 进行协同训练. 正如第 3.2 节所述, GEDM 的训练包含 3 个步骤: 多关系图构建、基于图的特征提取以及二分类. FeLeDetect 错误检测方法的框架如图 5 所示, 首先, 数据参与双方各自利用 MRG 模型将关系数据转化为图 G 和 G'; 接着, 在特征学习阶段, 双方遵循 FCTA 算法进行数据交换以捕获跨源数据的数据特征, 从而对图 G 和 G'上的节点与边进行协同特征表示. 值得注意的是, 在二分类阶段, 双方无须进行任何的数据/标签的交换. 这是因为二分类器的输入是经过协同表示的特征向量, 该向量已充分捕获双方数据集的特征, 因而无须再进行数据交换.

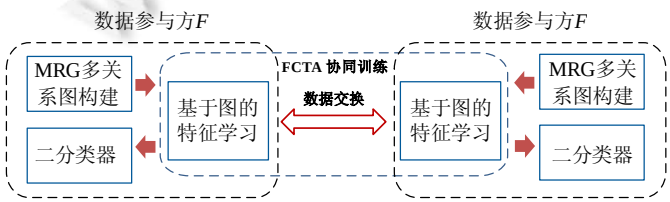


图 5 跨源错误检测方法 FeLeDetect

以数据参与方 F 为例, 算法 1 描述了 FeLeDetect 训练阶段的基本流程.

算法 1. FeLeDetect 跨源数据错误检测算法(以数据参与方 F 为例).

输入: 训练集 T, 训练集标签 L, MGGCN 的深度 K, 训练代数(epoch) N_{ep}, 每代的迭代(iteration)次数 N_{it}, 负责产生种子的数据参与方标号 I.

输出: 训练后的模型参数 Θ_F.

1 利用 MRG 模型将数据集 T 转化为图 G_F
2 **if** $F.index=i$ **then**
3 产生种子 *seed*, 并将其发送给 F'
4 **else**
5 接收由 F' 发送的种子 *seed*
6 初始化本地模型 GEDM 的参数 Θ_F
7 $\boldsymbol{h}_t^0,\boldsymbol{h}_a^0,\boldsymbol{h}_v^0 \leftarrow$random $(\forall a,v,t\in G_F)$ /*随机初始化 G_F 中节点与边的特征向量*/
8 **for** *epoch*=1 to N_{ep}
9 **for** *iter*=1 to N_{it}
10 **for** k=1 to K
11 执行协同训练算法 FCTA 正向传播, 得到 G_F 上各节点和边的特征向量 $\boldsymbol{h}_t^k$, $\boldsymbol{h}_a^k$, $\boldsymbol{h}_v^k$
12 分类 C_F 接受 $\boldsymbol{h}_t^k \| \boldsymbol{h}_a^k \| \boldsymbol{h}_v^k$ $(\forall(t,a,v)\in G_F)$为输入, 输出预测标签 $\hat{y}$
13 利用标签集 L 计算损失函数, 并反向传播更新参数 Θ_F /*根据公式(6)的目标函数更新参数*/
14 **return** Θ_F

算法 1 的输入为训练数据集 T, 训练标签 L, MGGCN 网络的深度 K, 最大训练代数 N_{ep}, 每代的迭代次数 N_{it}(采用 mini-batch 训练方式)以及负责种子产生方的标记 i. 算法先将数据集 T 转化为对应的图 G_F(第 1 行), 而后生成种子或等待接收对方生成的种子(第 2–15 行). 这是为了保证双方用相同的种子 *seed* 来初始化各自本地模型 GEDM 的参数 Θ_F, 从而使得双方模型的初始化参数相同(第 6 行). 这是联邦学习中的常见设置[42]. Θ_F 包括 MGGCN 和二分类器 C_F 的待学参数. 接下来, 算法分别随机初始化 $\boldsymbol{h}_a^0$, $\boldsymbol{h}_v^0$, 并使用种子 *seed* 随机初始化 $\boldsymbol{h}_t^0$ (第 7 行). 注意, 不同数据参与方使用相同 *seed* 来初始化 $\boldsymbol{h}_t^0$, 以保证各数据参与方中对应元组节点的初始特征向量一致. 这是由于不同数据源中对应的元组级节点表示的是相同的实体(数据集已对齐). 然后, 算法执行 N_{ep} 个 epoch 的训练. 这里采用 mini-batch 训练方式, 即每个 epoch 由 N_{it} 次迭代(iteration)组成. 每次迭代利用 FCTA 协同训练算法对 K 层 MGGCN 网络进行前向传播计算, 以得到网络的输出(第 10、11 行). 接着, 将 $\boldsymbol{h}_t^k \| \boldsymbol{h}_a^k \| \boldsymbol{h}_v^k$ 输入二分类器 C_F, 输出预测标签 $\hat{y}$ (第 12 行), 计算损失函数并反向传播, 以更新 MGGCN 和 C_F 的模型参数(第 13 行). 最后, 算法输出训练后的模型参数(第 14 行).

值得注意的是, 协同训练结束后, 需要进行协同错误检测. 具体而言, 双方将各自的数据集 D 和 D' 转化为对应的图, 利用训练好的模型, 遵循算法 FCTA 执行一次 MGGCN 网络以及二分类器的前向传播计算. 双方的分类器 C_F 和 $C_{F'}$ 的输出即为错误检测的结果.

4.2 协同训练算法FCTA

在 FeLeDetect 中, 数据交换发生在特征学习阶段. 以数据持有方 F 为例, 为了得到第 $k(\geq 1)$层 MGGCN 网络输出的特征向量, 首先需要与数据参与方 F' 交换彼此在第$(k-1)$层得到的部分节点和边的特征向量. 在完成与 F' 的数据交换后, 参与方 F 按如下公式完成第 $k(\geq 1)$层 MGGCN 中元组级节点特征向量的更新:

$$\boldsymbol{h}_{\mathcal{N}(t)\cup\mathcal{N}'(t)}^k = AGG_{(a,v)\in\mathcal{N}(t)\cup\mathcal{N}'(t)}(\boldsymbol{W}_{ta}^k\boldsymbol{h}_a^{k-1} \odot \boldsymbol{W}_{tv}^k\boldsymbol{h}_v^{k-1}) \tag{7}$$

$$\boldsymbol{h}_t^k = \sigma(\boldsymbol{W}_t^k(\boldsymbol{h}_t^{k-1} \| \boldsymbol{h}_{\mathcal{N}(t)\cup\mathcal{N}'(t)}^k)) \tag{8}$$

其中, $\boldsymbol{W}_{ta}^k$, $\boldsymbol{W}_{tv}^k$, $\boldsymbol{W}_t^k$ 为待学习的权重矩阵, $\boldsymbol{h}_t^{k-1}$, $\boldsymbol{h}_v^{k-1}$, $\boldsymbol{h}_a^{k-1}$ 为由第$(k-1)$层网络计算得到的特征向量, $\mathcal{N}(t)$表示图 G 中与 t 节点直接相连的边和节点的集合, $\mathcal{N}'(t)$表示图 G'中与 t 节点直接相连的边和节点的集合.

第 k 层 MGGCN 值级节点和属性级边的特征向量更新方式与第 3 节中 GEDM 相同, 见公式(3)–(5).由于双方的训练过程是高度对称的, 因此这里以数据参与方 F 为例描述训练流程. 算法 2 给出了以 F 为例的协同训练算法 FCTA 的流程.

算法 2. 协同训练算法 FCTA(以数据参与方 F 为例).

输入: MGGCN 第(k–1) (k≥1)层的属性级边、元组级节点和值级节点的特征向量 $\boldsymbol{h}_a^{k-1}$, $\boldsymbol{h}_t^{k-1}$, $\boldsymbol{h}_v^{k-1}$.

输出: MGGCN 第 k (k≥1)层的属性级边、元组级节点和值级节点的特征向量 $\boldsymbol{h}_a^{k}$, $\boldsymbol{h}_t^{k}$, $\boldsymbol{h}_v^{k}$.

1 将 $\boldsymbol{h}_a^{k-1}$, $\boldsymbol{h}_v^{k-1}$ 发送给参与方 F'($\forall(a,v)\in G_F$);

2 接收由参与方 F'传来的 $\boldsymbol{h}_a'^{k-1}$, $\boldsymbol{h}_v'^{k-1}$ ($\forall(a,v)\in G_F$)

3 计算 $\boldsymbol{h}_t^{k}$ ($\forall t\in G_F$) /*根据公式(7)、(8)更新节点 t 对应的特征向量*/

4 计算 $\boldsymbol{h}_v^{k}$ ($\forall t\in G_F$) /*根据公式(3)、(4)更新节点 v 对应的特征向量*/

5 计算 $\boldsymbol{h}_a^{k}$ ($\forall t\in G_F$) /*根据公式(5)更新边 a 对应的特征向量*/

6 **return** $\boldsymbol{h}_t^{k}$, $\boldsymbol{h}_v^{k}$, $\boldsymbol{h}_a^{k}$

算法 2 以 MGGCN 网络第(k–1) (k≥1)层的特征向量 $\boldsymbol{h}_a^{k-1}$, $\boldsymbol{h}_t^{k-1}$, $\boldsymbol{h}_v^{k-1}$ 为输入, 输出更新后的第 k (k≥1)层对应的特征向量. FCTA 先将上一层 MGGCN 网络输出的属性级边和值级节点对应的特征向量 $\boldsymbol{h}_a^{k-1}$, $\boldsymbol{h}_v^{k-1}$ 与 F'进行交换(第 1、2 行), 而后依次根据公式(7)、(8)更新元组级节点的特征向量, 根据公式(3)、(4)更新值级节点的特征向量以及根据公式(5)更新属性级边的特征向量(第 3–5 行). 最后, FCTA 返回第 k (k≥1)层 MGGCN 各节点和边的特征向量.

值得注意的是, 由于协同训练可以视作在数据参与双方分别构建了虚拟图 G_v=G∪G', 并基于该虚拟图进行协同特征表示, 因此, 协同训练机制保证双方得到的特征向量在同一嵌入空间内, 因而无须进行类似于文献[17]所述的数据参与双方的嵌入空间重对齐操作. 此外, 在协同训练以及错误检测的过程中, 数据参与双方只需交换属性级边的特征向量与值级节点的特征向量. 期间, 双方并不需要对元组级节点的特征向量进行交换.下面从数据信息分析的角度阐述其具体原因.

- 数据信息分析

首先给出数据信息的定义.

定义 2(数据信息). 基于图的错误检测方法旨在学习各节点与边的特征向量, 以捕获复杂的结构关系. 对于元组级节点 t, 其在特征学习的过程中所需的数据信息为与其直接相连的属性级边与值级节点$(a,v)\in\mathcal{N}(t)$的特征向量; 对于值级节点 v, 其在特征学习的过程中所需的数据信息为与其直接相连的属性级边与元组级节点$(a,t)\in\mathcal{N}(v)$的特征向量; 对于属性级边 a, 其在特征学习的过程中仅需要自身的特征向量.

为了便于分析协同训练算法 FCTA 捕获数据信息的情况, 这里引入 3 个错误检测场景.

(1) 本地场景(L): 数据参与双方仅使用本地数据进行错误检测.

(2) 联邦场景(F): 数据参与双方利用 FeLeDetect 进行联邦错误检测, 此过程不涉及原始数据交换.

(3) 集中场景(C): 数据参与双方先将各自的数据传输至一个数据中心, 而后在合并后的数据上进行错误检测.

定义 3(信息损失). 在给定集中场景下的多关系图 G, 其中, $(t_i,a_j,v_k)\in G\wedge(t_i,a_m,v_n)\in G$, $(a_j\neq a_m,v_k\neq v_n)$. 在本地场景下, 数据参与双方 F 和 F'分别构建本地多关系图 G_F 和 $G_{F'}$. 由于数据纵向分割, 在 G_F 中, $(t_i,a_j,v_k)\in G_F\wedge(t_i,a_m,v_n)\notin G_F$, 在 $G_{F'}$中, $(t_i,a_m,v_n)\in G_{F'}\wedge(t_i,a_j,v_k)\notin G_{F'}$. 因此, 在本地场景下, 数据参与方 F 在更新 t_i 的特征向量时的信息损失为(a_m,v_n)的特征向量; 数据参与方 F'的信息损失同理.

定理 1. 相较于集中场景下错误检测模型 GEDM 直接利用跨源数据信息, 联邦场景下, FeLeDetect 方法利用协同训练算法 FCTA 间接使用跨源数据, 可以保证跨源信息的无损性.

证明: 首先证明 FCTA 在更新元组级节点的特征向量时无信息损失. 在集中场景下, GEDM 基于合并数据集 T_S=T∪T', 利用公式(1)得到元组级节点的特征向量; 在联邦场景下, 数据参与方 F/F'在数据集 T/T'上利用

公式(7)得到元组级节点的特征向量. 由于$\mathcal{N}(t)=\mathcal{N}(t)\cup\mathcal{N}'(t)$, 故公式(7)中的$(a,v)\in\mathcal{N}(t)\cup\mathcal{N}'(t)$与公式(1)中的$(a,v)\in\mathcal{N}(t)$相等. 在 FCTA 更新元组级节点的特征向量前, 双方交换各自的属性级边和值级节点的特征向量. 因此, 更新是基于双方数据信息的, 这就解决了在本地场景下仅用单源数据更新元组级节点而造成的信息损失问题. 与集中场景相比, 根据 FCTA 进行的元组级节点特征向量的更新满足跨源数据信息的无损性.

接着证明 FCTA 在更新属性级边的特征向量时, 保证跨源数据信息无损. 如上所述, 属性级边利用公式(5)进行更新. 每次更新仅依赖自身上一次的更新结果. 所以在联邦场景下, 无需任何数据交换即可保证信息无损.

最后证明 FCTA 在更新值级节点的特征向量时无信息损失. 在集中场景下, GEDM 基于合并数据集$T_S=T\cup T'$, 利用公式(3)得到元组级节点的特征向量; 在联邦场景下, 仅基于本地数据集 T 或 T', FCTA 利用公式(3)得到元组级节点的特征向量. 容易看出, $(a,t)\in\mathcal{N}(v)=(a,t)\in\mathcal{N}(v)$且$(a,t)\in\mathcal{N}'(v)=(a,t)\in\mathcal{N}(v)$. 对于属性级边, 其更新只依赖自身上一次更新结果; 对于元组级节点, 各数据参与方的初始特征向量均由相同的种子生成. 经过更新的特征向量已被证明其包含无损的跨源数据信息, 故无须进行交换. 因此, 数据参与方 F 的本地数据中的$(a,t)\in\mathcal{N}(v)$与集中场景下合并数据中的$(a,t)\in\mathcal{N}(v)$相同, 且信息完整. 所以, FCTA 中数据参与双方无须交换元组级节点的特征向量, 且值级节点的更新可以按照公式(3)、(4), 并仅利用本地数据完成.

- 数据隐私分析

由于隐私保护是跨源数据错误检测问题的一大挑战, 确保 FCTA 在数据交换过程中没有隐私泄露风险至关重要. 具体来说, FCTA 涉及两种类型的数据交换.

(1) 双方初始特征向量的交换. 由于双方各节点和边的初始特征向量是随机初始化的, 其不包含任何与原始数据相关的信息, 因而无法通过初始特征向量推测出原始数据. 这使得交换初始特征向量没有隐私泄露的风险.

(2) 经过第 $k(k\geqslant 1)$层网络更新的特征向量的交换. 该神经网络中间层结果交换亦无隐私泄露风险, 原因是: 尽管有研究表明, 可以由神经网络的中间层结果推断原始输入[44], 但数据参与双方的原始输入均为经随机初始化的特征向量, 即原始输入中不包含任何隐私信息.

综上, FCTA 的数据交换机制保证了跨源数据的隐私安全. □

4.3 优化方法

在 FeLeDetect 协同训练 GEDM 期间, 需要频繁交换不同数据参与方的中间结果, 这给网络通信带来了很大的压力. 本文从3个方面提出了不同的优化技术以减少通信代价. 同时, 针对有监督学习中存在的人工标注代价问题, 本文也提出了一种自动化标注策略以减少训练集中所需的人工标注成本.

- 数据去重

在多关系图里, 某些不同的元组级节点与相同的值级节点相连, 这将造成某些相同数据的多次交换. 如图 3(b)中, 元组级节点 t_1, t_2 和 t_3 都与同一个值级节点 v: “International Conference on Management of Data”相连. 因此, 在向另一数据参与方传输节点信息时, 需要传输的内容为 $\{1:\boldsymbol{h}_v^k\}$, $\{2:\boldsymbol{h}_v^k\}$, $\{3:\boldsymbol{h}_v^k\}$, 这导致该值级节点“International Conference on Management of Data”的特征向量 $\boldsymbol{h}_v^k$ 被传输 3 次. 不仅如此, 许多元组级节点与值级节点被相同类型的属性级边相连, 这也将造成大量相同类型的边对应的特征向量 $\boldsymbol{h}_a^k$ 被多次传输. 实际上, 图上不同的属性级边的数目取决于数据集的属性数目. 为了避免重复不必要的重复数据交换, 对于相同的值级节点或属性级节点的特征向量, 在数据传输过程中仅传输一次, 即将重复数据去除后再进行数据传输. 延续上文所述的例子, 经过去重操作后, 向另一方传输的数据为$\{\{1,2,3\}:\boldsymbol{h}_v^k\}$. 本文用矩阵 $\boldsymbol{M}_V=[\boldsymbol{h}_{v1}^k,\boldsymbol{h}_{v2}^k,...,\boldsymbol{h}_{vp}^k]$ 表示去重后的值级节点特征向量, 矩阵 $\boldsymbol{M}_A=[\boldsymbol{h}_{a1}^k,\boldsymbol{h}_{a2}^k,...,\boldsymbol{h}_{am}^k]$ 表示去重后的属性级边的特征向量, 以避免相同特征向量的多次交换. 这里, p 表示数据集中不同属性值的数量, m 表示数据集中不同属性的数量. 以图 1 的 DBLP 数据集为例, p=13, m=5.

- 量化压缩

由于矩阵 $\boldsymbol{M}_A$ 一般远小于矩阵 $\boldsymbol{M}_V$, 故 FeLeDetect 的通信代价主要由传输矩阵 $\boldsymbol{M}_V$ 引起. 因此, 可设法进一步降低传输矩阵 $\boldsymbol{M}_V$ 的通信代价. 由于使用了激活函数, 矩阵 $\boldsymbol{M}_V$ 中每个值都是介于[−1,1]之间的实数. 为了进一步降低通信代价, 本文对这些实值进行离散化处理以压缩其大小. 具体来说, 将区间[−1,1]划分为 2^{η}个子区间. 例如, 当η=2时, 区间[−1,1]被划分为 4 个子区间, 分别为[−1,−0.5), [−0.5,0), [0,0.5)以及[0.5,1]. 这样, 仅需 2 比特的数据就能表示这 4 个子区间. 原矩阵中, 每个实值都落入被重新划分的某个子区间中, 因而可以被近似为子区间的区间端点. 例如, 落入子区间[−1,−0.5)的实值都被近似为−1. 近似后的量化值与原始值的误差上限为 $2^{1-\eta}$=0.5. 这样, 原先需要占用 32 比特(4 B)带宽的一个实数值, 经过量化压缩后, 仅需η比特编码.若η=16, 则矩阵 $\boldsymbol{M}_V$ 大小将压缩为原先的一半, 且量化误差仅为 2^{-15}.

- 降频传输

在 FeLeDetect 协同训练 GEDM 的过程中, 需要进行多代(epoch)的训练以使得神经网络模型收敛. 一代意味着使用全部训练数据对模型(包括 MGGCN 和二分类器)进行一次训练, 每代训练过程中又包含多个迭代轮次(iteration). 在每个轮次中, MGGCN 都需要交换每层网络的中间结果. 然而, 某个轮次的中间结果(如 $\boldsymbol{M}_V$)可能与前一个轮次的结果非常相似, 尤其是在网络参数趋于稳定之后. 因此, 可以设法降低数据交换的频次以进一步减小训练过程的通信开销. 具体而言, 若 MGGCN 在两个相邻轮次中产生的中间结果非常相似, 则双方可以取消本次中间结果的交换. 给定一个当前轮次的中间结果矩阵 $\boldsymbol{M}_V$, 上一轮次的中间结果矩阵 $\boldsymbol{M}'_V$ 以及一个传输阈值τ, 若 $Sim(\boldsymbol{M}_V,\boldsymbol{M}'_V)=\|\boldsymbol{M}_V-\boldsymbol{M}'_V\|_F<\tau$, 则将上一轮次交换的中间结果 $\boldsymbol{M}'_V$ 近似为本轮产生的中间结果, 从而避免 $\boldsymbol{M}_V$ 的交换. 其中, $\|\cdot\|_F$ 表示相似性度量函数. 在本文实验中, 采用最简单的欧式距离以度量相似性.

- 标注策略

由于本文将错误检测问题视作二分类问题, 因此, 二分类器需要带标签的训练数据以学习如何对数据单元进行分类. 然而, 数据标注需要领域专家的参与, 该过程耗费大量的人力成本, 因而往往成为实际应用中的瓶颈. HoloDetect[2]利用数据增强技术减少对人工标注的需求, 但其仍要求整个数据集规模 5%–10%的人工标注量. 对于一个包含 100 000 个元组和 8 个属性的数据集来说, 10%意味着需要人工标注 80 000 个数据单元, 这样的人工作业量是不可小觑的. Raha[20]结合了聚类和标签传播技术将人工标注量限制在了常数级别, 即不随着数据集规模的增长而增长. 但它无法直接应用在跨源数据的场景下. 鉴于此, 本文设计了一种自动标注策略以降低人工标注成本, 并为跨源数据产生带标签的训练数据集.

假设两个数据持有方 F 和 F', 分别持有数据集 $D=\{t_1,t_2,\ldots,t_n\}$和 $D'=\{t'_1,t'_2,...,t'_n\}$. 数据集已经完成加密实体对齐, 即每个元组对$(t_j,t'_j)(1\leqslant j\leqslant n)$表示同一个实体. 标注策略按如下步骤进行: 首先, F 通过随机采样一个数据子集 $S=\{t_j|t_j\in D\}$, 其中, $|S|=n\cdot\alpha\%$; 其次, 利用替换加密策略对 S 中每个元组中的每个数据单元的值进行加密, 得到加密后的数据子集 $E(S)$; 最后, F 将 $E(S)$和采样索引集 $I=\{j|t_j\in S\}$传给另一数据持有方 F'. 其中, 索引集 I 指示原数据集 D 中被采样出的元组标号. 当数据持有方 F'收到 $E(S)$和 I 后, 根据 I 对数据集 D'进行采样得到 $S'=\{t'_j|j\in I\}$, 并将 S'与 $E(S)$中指向同一实体的元组进行拼接来得到一个合并数据集 $S_C=\{E(t_j)\|t'_j|E(t_j)\in E(S),t'_j\in S'\}$. 注意, S_C 共有(m_1+m_2)个属性, 其中, m_1 为数据集 D 的属性数量, m_2 为数据集 D'的属性数量. F'将 S_C 作为错误检测工具 Raha 的输入数据集, 以获得每个数据单元的标签. F'保留与 S'相关的数据标签, 并将属于 $E(S)$的数据标签传输给 F. 为了保证标签的质量, F'仅接受 Raha 返回的置信度大于一定阈值(实验中设置为 90%)的标签. 至此, 数据标注流程结束. 双方都获得了一部分的数据标签, 因而产生了带标签的训练数据集 T 和 T'. 在此期间, 双方没有暴露各自的原始数据. 由于自动标注策略借助 Raha 来产生标签, 故所需的人工标注的代价与 Raha 一致, 为常数级别.

由于上述过程采用了替换加密技术, 故 $E(S)$丢失了原数据的语义信息, 但原数据的隐私得到了保护. 同时, Raha 依旧可以检测出一些数据错误(比如违反属性值的依赖关系和格式错误等). 这是因为替换加密相当于对属性值进行了一个匿名空间映射, 原始值的依赖关系在加密空间得以保持; 同时, 与语义无关的错误(如

格式错误等)也能够被检测出来. 为了简便起见, 实验采用替换密码来加密采样数据子集 $S=\{t_j|t_j\in D\}$, 这是一个经典密码. 尽管已有许多的高级密码技术被提出并广泛应用, 但替换密码一直是密码学领域十分重要的一种加密策略. 目前常用的现代密码(比如 DES 和 AES)都使用经典密码作为其基本构建模块[45]. 在实践中, 参与双方可以协商并选择更为复杂的高级加密技术来加密 S, 以达到更高的安全性.

5 实验分析

本节在真实数据集上进行实验评估, 主要目的是: (1) 证明 GEDM 的有效性与先进性; (2) 验证 FeLeDetect 的检测精度优于仅使用单源数据的各种本地检测方法, 且与集中场景下 GEDM 的检测精度相当; (3) 验证通信优化策略的有效性. 此外, 实验还测试了 GEDM 和 FeLeDetect 的运行时间以评估其效率, 验证了 FeLeDetect 在不同错误率及错误类型分布下的有效性, 并测试了 FeLeDetect 的可扩展性. 第 5.1 节介绍本文的实验数据.第 5.2 节介绍实验设置与评价指标. 第 5.3 节介绍实验的实现细节. 第 5.4 节给出实验结果, 并对实验结果进行分析.

5.1 实验数据

本文使用 3 个公开的真实数据集进行实验测试. 表 1 给出了所使用数据集的统计信息, 其中, 错误类型包括替换错误(SE)、格式问题(FI)、缺失值(MV)以及违反属性间依赖规则(VAD). 如引言部分所述, 替换错误(SE)在单源的情况下往往难以检测出来, 需要结合跨源数据信息; 所使用的数据集中包含多类型错误, 体现了数据错误的异质性.

表 1 实验数据集

数据集		规模	数据错误数量	错误类型
D-A	DBLP	2224×4	444	SE, FI
	ACM	2224×4	444	SE, MV
Flights	Flights1	2445×4	1 879	SE, MV, FI
	Flights2	2445×4	2 972	SE, MV, VAD
Adult	Adult1	97864×4	19 481	SE
	Adult2	97864×4	19 535	SE

DBLP-ACM 数据集(简记为 D-A)[46]是实体对齐研究领域常用的一个公开数据集, 该数据集包含两个数据表 DBLP 和 ACM, 分别记录了一些计算机科学领域的会议和期刊论文信息. 两个数据表模式均为(title,author, venue,year). 实验中使用已经完成实体对齐的数据版本. 由于数据集 DBLP 和 ACM 本身隶属于不同组织, 故可以合理假设双方不允许原始数据出本地. 本实验对 DBLP 和 ACM 数据集的 venue 属性列的一些数据单元中注入替换错误, 这是因为 venue 属性域中的许多值十分相似, 在真实场景下极易发生替换错误. 具体来说, 这里随机选取了一些 venue 属性列下的数据单元, 将该单元的原始值替换为 venue 属性域中与其最相似的另一值, 即与原始值编辑距离最小的新值. 同时还向 DBLP 的 year 属性列和 ACM 的 title 属性列单元注入格式错误(非法字符)和缺失错误, 以模拟现实场景中普遍存在的数据错误异质性现象. 默认情况下, 注入错误的错误类型比为 80%. 这里, 错误类型比定义为替换错误数据单元的数量与所有错误数据单元数量的比率.

Flights 数据集[47]记录了 2011 年 12 月份每天从不同数据源收集的关于 1 200 多个航班的信息, 其中, 每天收集到的信息记录在同一张表中. 由于不同来源的数据质量参差不齐, 故表中存在许多的错误, 包括替换错误、缺失值、格式错误以及违反属性间依赖规则. 这些错误是数据集中真实存在的, 因而实验中不再人为注入错误. 实验中随机选取了 12 月 1 日当天收集的数据, 数据属性集为(source,flight,scheduled departure,actual departure,scheduled arrival,actual arrival). 此外, 为了模拟两个数据参与方不愿共享数据的场景, 人为地将该数据表 Flights 纵向分割为两个子表, 即 Flights1 (source,scheduled departure,scheduled arrival)和 Flights2 (flight, actual departure,actual arrival), 并假设它们各自被一个数据持有方拥有.

Adult 数据集来自 UCI 机器学习库(http://archive.ics.uci.edu/ml/), 该数据集是从美国 1994 年人口普查数据库中抽取而来, 包含 97 684 条记录. 实验中选取其中的 8 个属性, 并将其纵向分割为两个不相交的子表, 即

Adult1 (age,education,race,sex)和 Adult2 (maritalstatus,relationship,country,income), 并假设它们各自被一个数据持有方所有. 与 D-A 相似, 实验分别向 age, race 和 realtionship 属性上的某些数据单元注入替换错误.

由于存在真实错误的数据集通常没有真值(ground truth), 因此无法评估错误检测算法的有效性[48]. 大量错误检测工作都使用人工注入错误的方式进行实验[49,50]. 相关研究[15,51]表明, 在真实世界, 一个数据集中大约有 5%的数据单元因各种不同原因而存在数据错误, 因此, 本文在向 D-A 和 Adult 数据集人为注入错误时, 默认将错误率设为 5%. 这里, 错误率被定义为错误的数据单元数与数据集中全部数据单元数的比率. 由于替换错误的检测难度较大, 默认将 D-A 的错误类型比设为 80%. 这里, 错误类型比定义为替换错误数据单元的数量与所有错误数据单元数量的比率. 除非明确指出, 在以下实验中, 都使用上述默认的 5%错误率和 80%的错误类型比. 值得强调的是, 尽管在此设置了默认的错误率和错误类型比, 本文还进行了错误率及错误类型比的变化实验, 证明了 FeLeDetect 适用于不同错误率和错误类型比的场景.

5.2 实验设置与评价指标

实验选择了 5 种不同类型的现有错误检测方法作为基准: (1) DBoost[52], 一个异常点检测框架, 集成了多个基于机器学习方法的检测模型; (2) NADEEF[11], 一种基于规则的错误检测方法, 其允许用户指定数据一致性规则并根据规则进行错误检测; (3) KATARA[14], 一种基于外部知识库(KB)的错误检测方法; (4) Metadata-driven[22]方法(简记为 Meta), 一种融合了多种元数据的集成式错误检测方法; (5) Raha[20], 一个最先进的(SOTA)基于机器学习的错误检测系统(HoloDetect[2]和 Raha 都是基于机器学习方法的先进错误检测系统. 由于 Raha 公开了源代码, 本文使用 Raha 作为基准方法进行对比. 同时, 文献[53]证实, Raha 可在更低的人力成本条件下达到与 HoloDetect 相当的错误检测精度). 由于 Raha 的错误检测结果具有随机性, 这里遵循 Raha 研究者的实验评估方式, 即对其相关的评价指标报告 10 次独立运行结果的均值.

实验设置了 3 种不同场景.

(1) 本地场景(local, L), 每个数据持有方仅通过本地数据进行错误检测. 实验在本地场景下, 对本文提出的模型 GEDM 以及 5 个基准模型进行了测试.

(2) 联邦场景(federated, F), 数据持有方无须进行原始数据交换, 协同训练错误检测模型. 在联邦场景下, 对本文提出的 FeLeDetect 错误检测方法进行实验, 并证明其先进性.

(3) 集中场景(centralized, C), 所有数据持有方先将各自数据汇聚至一个数据中心, 而后基于合并数据集进行错误检测. 该设置是为了验证联邦场景下 FeLeDetect 方法与集中场景下 GEDM 方法的检测精度相当. 在集中场景下,对本文提出的 GEDM 和 5 种基准模型分别进行了测试.

实验采用不同的评价指标对所提出的方法进行全面评估, 报告了:

(1) 精确率 P, 即被检测为错误的数据单元中真正为错误单元的概率.

(2) 召回率 R, 即实际的错误数据单元中被检测为错误的概率.

(3) $F1$ 分数, 即精确率与召回率的调和平均数: $F1=2\times(P\times R)/(P+R)$.

(4) 模型训练时间 TT 以及模型测试时间(即在完整数据上错误检测的时间)DT.

(5) 模型训练的通信代价 V, 以字节为单位, 表示协同训练中双方的数据通信量.

5.3 实现细节

FeLeDetect 方法实现中需使用 PyTorch 库[54]. 实验使用 SGD 作为模型训练的优化算法, 并将学习率设置为 0.01, 动量设置为 0.9. 经过一些初步实验评估, 将数据集 DBLP-ACM、Adult、Flights 的批大小分别设置为 16, 128 和 512. 在训练过程中, 对所有数据集进行 300 个 epoch 的训练. 在 FeLeDetect 的数据自动标注过程中, 将 DBLP-ACM 和 Flights 数据集的标注采样率α设为 20%. 由于 Adult 数据集的规模较大, 将α设为 5%已经足够支持模型训练. 实验中, 随机将标注数据集划分为训练集(60%)和验证集(40%), 返回在验证集上 $F1$ 分数最高的检查点, 并在完整数据集上, 利用训练好的模型进行错误检测. 在默认情况下, 设置通信优化参数 η=4, τ=1.5. 所有实验均在 Dell 服务器上进行, 具体配置为: 英特尔志强 4110 CPU 处理器, 128 GB 内存, 4 块

英伟达 GeForce RTX2080ti GPU. 本实验所有测试程序使用 Python 语言编写(相关的实验代码公开发布于 https://github.com/ZJUDAILY/FeLeDetect).

5.4 实验结果与分析

本节在 3 个数据集上对 GEDM 和 FeLeDetect 进行全面评估, 并与前述 5 种基准方法对比.

- GEDM 的有效性与效率

首先, 在本地场景(L)与集中场景(C)下, 对比 GEDM 与其他 5 种基准方法的错误检测效果, 并使用精确率 *P*、召回率 *R* 以及 *F*1 分数评估错误检测精度; 同时给出了不同错误检测方法所需时间 *DT* (以秒为单位)以评估错误检测效率. 为公平起见, 这里只报告 GEDM 的错误检测时间(即模型的测试时间), 模型的训练时间 *TT* 将在补充实验部分给出. 表 2 给出了本地场景(L)与集中场景(C)下, GEDM 和其他 5 种基准方法的错误检测结果,并用粗体表示最优 *F*1 分数.

表 2 不同错误检测方法在不同检测场景下的结果精度 *P*, *R* 和 *F*1 对比

本地错误检测场景

检测方法	D-A						Flights						Adult					
	DBLP			ACM			Flights1			Flights2			Adult1			Adult2		
	P	*R*	*F*1	*P*	*R*	*F*1	*P*	*R*	*F*1	*P*	*R*	*F*1	*P*	*R*	*F*1	*P*	*R*	*F*1
dBoost	0.17	1.00	0.29	0.33	0.20	0.25	0.78	0.87	0.82	0.72	0.49	0.58	0.63	0.36	0.45	0.23	0.90	0.36
NADEEF	0.44	0.21	0.28	0.50	0.21	0.29	0.08	0.12	0.09	0.40	1.00	0.57	0	0	0	0	0	0
KATARA	0.12	1.00	0.21	0	0	0	0.02	0.13	0.56	0.02	0.14	0.02	0.02	0.29	0.08	0.02	0.10	0.02
Meta	0.45	0.22	0.29	1.00	0.01	0.01	1.00	0.87	**0.93**	0.70	0.68	0.68	1.00	0	0	1.00	0	0
Raha	0.57	0.32	0.42	0.65	0.63	0.64	0.98	0.87	0.92	0.71	0.70	0.70	0.50	0.83	0.62	0.74	0.93	0.82
GEDM	0.68	0.33	**0.45**	0.78	0.80	**0.79**	1.00	0.87	**0.93**	0.71	0.73	**0.72**	0.76	0.75	**0.75**	0.86	0.90	**0.88**

集中错误检测场景

检测方法	D-A						Flights						Adult					
	DBLP			ACM			Flights1			Flights2			Adult1			Adult2		
	P	*R*	*F*1	*P*	*R*	*F*1	*P*	*R*	*F*1	*P*	*R*	*F*1	*P*	*R*	*F*1	*P*	*R*	*F*1
dBoost	0.10	0.20	0.13	0.31	1.00	0.48	0.78	0.87	0.82	0.67	0.30	0.41	0.67	0.36	0.45	0	0	0
NADEEF	0.10	1.00	0.17	0.10	1.00	0.17	0.38	0.41	0.40	0.40	1.00	0.57	0	0	0	0	0	0
KATARA	0.12	1.00	0.21	0	0	0	0.02	0.13	0.06	0.02	0.01	0.01	0.05	0.29	0.08	0.02	0.10	0.02
Meta	1.00	0.21	0.35	1.00	0.01	0.01	1.00	0.87	**0.93**	0.70	0.68	0.69	1.00	0	0	1.00	0	0
Raha	0.67	0.37	0.47	0.68	0.61	0.65	0.97	0.88	0.92	0.66	0.75	0.70	0.67	1.00	0.80	0.87	0.90	0.88
GEDM	1.00	0.72	**0.84**	1.00	0.84	**0.91**	0.98	0.88	**0.93**	0.73	0.74	**0.73**	0.92	0.99	**0.96**	0.99	0.90	**0.95**

联邦错误检测场景

检测方法	D-A						Flights						Adult					
	DBLP			ACM			Flights1			Flights2			Adult1			Adult2		
	P	*R*	*F*1	*P*	*R*	*F*1	*P*	*R*	*F*1	*P*	*R*	*F*1	*P*	*R*	*F*1	*P*	*R*	*F*1
FeLeDetect	1.00	0.72	**0.84**	1.00	0.84	**0.91**	1.00	0.87	**0.93**	0.73	0.74	**0.73**	0.92	1.00	**0.96**	1.00	0.90	**0.95**

在精度方面, 首先可以看到, 在本地场景与集中场景下, GEDM 的 *F*1 分数都高于其他 5 种基准方法. 具体来说, 在本地场景与集中场景下, GEDM 的 *F*1 分数较最优的基准方法平均提高了 10.3%和 25.2%. 尤其是在集中场景下, 当跨源数据合并后, GEDM 能够有效地捕获合并数据的特征, 因而能够同时达到高精确率和高召回率. 但其他基准方法无法同时达到高精确率和高召回率. 具体而言, dBoost 由于使用启发式思想将统计概率上的离群点视作数据错误, 导致其精确率较低; NADEEF 的低精确率是由于基于规则的检测方法普遍存在粗粒度的问题; KATARA 由于不可避免地将某些 KBs 上的关系(realtion)与关系型数据上的属性进行错误匹配, 导致其精确率低下, 同时, 由于有限的 KBs 无法完全覆盖关系数据中涵盖的知识, 因而其召回率也相对较低; Meta 由于所集成的错误检测方法难以覆盖所有错误类型, 导致其表现出较低的召回率; Raha 虽然是目前最优的错误检测方法, 但其精度仍不及 GEDM, 这是因为其不能充分地捕获数据集中不同维度的丰富特征.相反, 本文提出的 GEDM 不仅能够捕获属性值的语义特征, 而且还能对不同属性以及不同元组之间的依赖关系进行表征. 注意: 由于数据集 Flights1 中大部分的错误(超过 85%)都是缺失值, 而缺失值的检测难度较低, 因而 GEDM 在 Flights1 上没有表现出明显的优越性. 其次, 注意到, dBoost、NADEEF 以及 KATARA 在集中场景下的错误检测效果并没有优于本地场景. 这是因为这些方法无法有效地捕获不同属性间的相关性. 因此, 即使在集中式场景下, 合并后的数据集拥有更多属性(更多维度的数据特征), 错误检测的精度也没有实质上的提

升. 具体来说, dBoost 是按每个属性列独立进行检测错误, 且基于数据集调整相关参数配置, 所以在集中式场景下, dBoost 基于合并后的完整数据集进行参数选择, 而该全局参数对于双方的本地数据集来说未必都是最优参数配置, 这可能导致其在集中场景下的检测效果甚至不如本地场景. NADEEF 基于一致性规则进行错误检测, 但由于其粗粒度的检测方式, 当属性数量增多时(在集中场景下), 反而可能造成检测结果的更多误判. KATARA 利用外部知识库进行错误检测, 因此在集中场景下, 知识库中的关系与数据表中属性错误匹配的几率更高, 也无法保证集中场景下错误检测的精度有所提升. 而其他几种方法, 由于其能够有效地表征不同属性间的关系, 在集中场景下的检测效果自然优于本地场景.

在运行效率方面, 从表 3 中可以看出, 在本地场景与集中场景下, GEDM 均优于其他基准方法. 例如, dBoost需耗费大量时间在不同的参数配置下运行, 以获得最终结果; KATARA需要仔细检查每个KB并与数据表中的属性进行匹配, 因而也需要较长运行时间; Raha 需要运行不同错误检测策略并进行数据单元聚类和标签传播, 这些过程都是极其耗时的, 因而其运行时间最长. 相比之下, GEDM 的错误检测十分高效. 这是因为模型一旦训练完毕, 在完整数据集上, 只需对模型进行一次前向传播计算, 即可获得错误检测结果.

表 3 不同错误检测方法在不同检测场景下的检测时间 *DT* 对比

检测方法	本地错误检测场景						集中错误检测场景			联邦错误检测场景		
	D-A		Flights		Adult		D-A	Flights	Adult	D-A	Flights	Adult
	DBLP	ACM	Flights1	Flights2	Adult1	Adult2						
dBoost	2.21	2.37	2.35	2.64	74.29	77.06	5.29	6.06	211.85	–	–	–
NADEEF	0.13	1.05	1.06	1.30	1.76	2.11	1.98	1.42	2.97	–	–	–
KATARA	9.28	9.35	9.62	9.64	40.02	40.02	14.70	15.45	100.72	–	–	–
Meta	2.20	2.16	2.18	2.66	14.69	17.90	4.41	3.10	29.04	–	–	–
Raha	14.97	15.91	14.44	13.73	1059.06	1266.30	27.04	31.58	2397.48	–	–	–
GEDM	**0.10**	**0.09**	**0.23**	**0.22**	**1.29**	**1.27**	**0.12**	**0.36**	**3.12**	–	–	–
FeLeDetect	–	–	–	–	–	–	–	–	–	9.43	9.46	386.9

- FeLeDetect 的有效性与效率

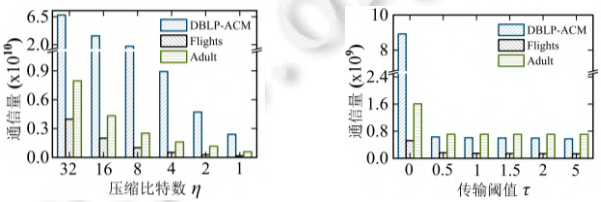
考虑到隐私保护, 跨源数据往往不允许被传输至公共数据中心进行集成. 因此, 集中场景下的数据错误检测受到很大的阻碍. 鉴于此, 针对单源数据的错误检测方法只能在某个数据持有方本地进行, 且无法获取到与该数据源相关的其他数据源的信息. 在联邦场景(F)下, 将本文提出的FeLeDetect方法与本地场景(L)和集中场景(C)下的各基准方法(包括 GEDM)进行对比. 表 2 给出了联邦场景(F) FeLeDetect 与其他 5 种基准方法以及 GEDM 的错误检测结果对比, 并用粗体表示最优 *F*1 分数.

在精度方面, 由表 2 可以看出, FeLeDetect 的 *F*1 分数优于本地场景下的任何方法(包括 GEDM). 与本地场景下检测精度最佳的方法 GEDM 相比, FeLeDetect 的 *F*1 分数在此基础上进一步提高了平均 23.2%. 这是因为本地场景下, 所有错误检测方法仅使用本地数据信息, 缺少其他来源数据信息的支持. 相反, FeLeDetect 利用联邦学习机制充分捕获跨源数据特征. 这证明跨源数据确实能够提升错误检测的精度. 由于真实错误数据集 Flights1 中简单错误如缺失、格式错误占全部错误类型的 85%以上, 困难错误类型如替换错误、违反依赖关系占比较小, 因此所提方法没有明显的精度提升(但依旧打败了其余方法或与之持平). 以上实验现象表明, 对于较容易检测的数据错误, 本文方法能够超过或与最先进的方法持平; 对于较难检测的数据错误, 本文方法能够远超现有方法. 此外, 在 3 个数据集上, FeLeDetect 的 *F*1 分数均达到了与集中场景下 GEDM 的 *F*1 分数相同的水平. 这也从实验结果角度验证了本文设计的联邦协同训练算法 FCTA 的信息无损性, 其能够达到与集中场景下 GEDM 相当的错误检测效果.

在运行效率方面, 从表3可以看出, 相较于联邦场景与集中场景, 所有检测方法在本地场景下的运行时间最短. 这是因为在本地场景下, 数据集规模更小, 且没有联邦通信的时间开销. 尽管如此, FeLeDetect 的错误检测依旧是比较高效的. 尤其是在较大规模数据集, 如接近 100 000 条元组的 Adult 数据集上, FeLeDetect 仅在 6 分钟左右即可完成错误检测任务, 其运行效率相较于本地场景下精度最高的基准方法 Raha 有超过 60%的提升.

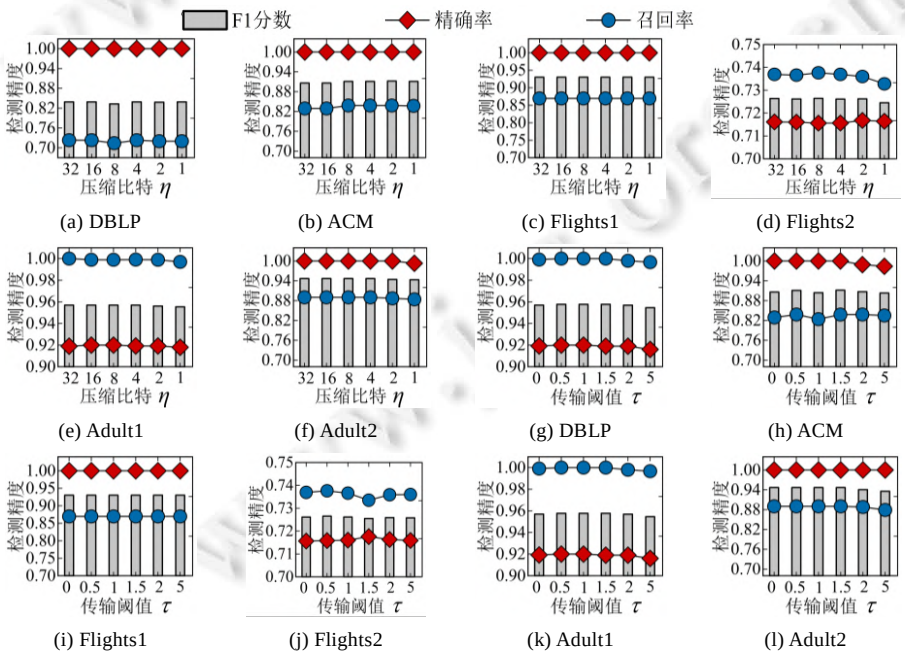
由于数据去重可以在不影响检测效果的前提下减少联邦训练过程中数据传输的通信量，因此，这里只探究量化压缩与降频传输对联邦通信量以及检测效果的影响.

首先，实验在{32,16,8,4,2,1}之间变化压缩参数η，以探究压缩比特数对通信量及错误检测精度的影响.注意，η=32 表示不进行量化压缩优化.如图 6(a)所示，通信量随着η的减小显著降低.这是因为压缩后的比特数η越小，传输矩阵 $\boldsymbol{M}_V$ 中每个数值的编码长度就越短，数据传输量也越小.另一方面，错误检测精度受压缩比特数影响较小.如图 7(a)–(f)所示，当η变化时，检测结果的精确率曲线、召回率曲线以及 $F1$ 分数曲线都较为平稳.当压缩比特数降为 1 时，精度才有较明显的降低.这是因为多关系图(MRG)上，节点与边的初始特征都是经过随机初始化，并通过图神经网络特征聚合得到最终特征表示的.压缩神经网络中间结果相当于在不同层的图神经网络间增加了一个激活层，以压缩特征向量的编码空间.由于初始特征向量是随机生成的，压缩中间层特征向量的值相当于简化其编码空间，尽管量化压缩会导致参与双方在交换神经网络中间结果时存在一定误差(误差上限为 $2^{1-\eta}$)，但这样微小的中间结果误差对精度的影响是微乎其微的，却能大大降低联邦训练的通信代价.



(a) 通信量随不同压缩比特数变化　　(b) 通信量随不同传输阈值变化

图 6　通信量随不同压缩比特数/传输阈值的变化



(a) DBLP　(b) ACM　(c) Flights1　(d) Flights2

(e) Adult1　(f) Adult2　(g) DBLP　(h) ACM

(i) Flights1　(j) Flights2　(k) Adult1　(l) Adult2

图 7　错误检测精度随不同压缩比特数/传输阈值变化

- FeLeDetect 通信优化实验

接着，将压缩参数η固定为 4，并在{0,0.5,1,1.5,2,5}之间变化频率阈值τ，以探究频率阈值对通信量及错误检测精度的影响.注意，τ=0 表示不进行任何降频优化.通信量随τ的变化曲线如图 6(b)所示.可以看出，当传

输阈值由 0 增加到 0.5 时, 通信量显著降低. 这是因为随着传输阈值的增加, 更多的中间结果被过滤, 因而需要传输的数据量显著减少. 但随着τ继续增加, 通信量的降低速率逐渐变缓, 通信量趋于一个稳定值. 具体来说, 当τ由 0.5 增加至 5 时, 通信量几乎不再降低. 这是由于在训练后期, 模型逐渐收敛, 神经网络中待学习参数的更新速度减缓且趋于稳定, 因而相邻两次产生的中间结果矩阵差异很小, 且很小的传输阈值(如 0.5)足以将某些中间结果过滤掉. 而在训练初期, 模型参数更新较快, 即使将传输阈值设置为 5, 也无法将训练初期产生的中间结果过滤掉. 因此, 通信量也不再随阈值增大而显著减小. 另一方面, 如图 7(g)–(l)所示, 在阈值τ较小时, 检测精度几乎不受影响; 当传输阈值由 2 增大至 5 时, 精度开始下降. 这是因为当传输阈值较小时, 降频传输选择性过滤神经网络产生的某些信息量较低的中间结果, 并复用信息量较高的中间结果, 因此不会影响关键信息交换, 不会对检测结果造成显著影响; 然而当阈值增大到一定值时, 某些关键的特征向量被过滤(没有被传输), 因此造成模型训练效果变差且精度下降. 较小的阈值设定既能保证结果精度, 又可以大幅度降低通信量, 这进一步表明了降频传输的优势.

- 补充实验

接下来, (1) 进一步给出了 GEDM 以及联邦错误检测方法 FeLeDetect 在不同数据集下的训练时间 *TT*; (2) 验证了 FeLeDetect 在不同错误率及错误类型比的数据集下的通用性; 以及 (3) FeLeDetect 的可扩展性.

首先, 表 4 给出了 GEDM 在本地和集中场景下的模型训练时间 *TT*, 以及 FeLeDetect 在联邦场景下的协同训练时间 *TT*. 注意: 如第 5.3 节所述, 模型训练指在训练数据集上以给定的批大小进行 300 个 epoch 的训练; 错误检测指在完整数据集上, 利用训练好的模型进行错误检测. 对比不同的数据集, 显然大数据集 Adult 需要更长的运行时间. 同时, 训练时间还与批大小紧密相关. 批大小设置较大的数据集(如 Flights)所需的训练时间 *TT* 较小. 这是因为每个 epoch 中迭代次数(iteration)等于训练数据集规模与批大小的比值, 当训练集规模相似时, 较大的批大小意味着更少的迭代次数, 因而神经网络进行前向传播与反向传播的次数更少, 所需的训练时间自然更短. 另外, 结合表 3 的错误检测时间可以看出, GEDM/FeLeDetect 的错误检测时间要比训练时间快若干个数量级. 即使是在大规模数据集(如 Adult)上, FeLeDetect 的错误检测过程仅耗时 386.97 s, 远低于 Raha 的运行时间.

表 4 GEDM 在本地场景(L)和集中场景(C)/FeLeDetect 在联邦场景(F)的训练时间 *TT* (s)

数据集		GEDM (L)	GEDM (C)	FeLeDetect (F)
D-A	DBLP ACM	270.28 286.64	633.75	1 443.04
Flights	Flights1 Flights2	67.02 61.69	65.86	525.84
Adult	Adult1 Adult2	1 023.32 1 005.63	1 711.02	4 474.41

接着, 以数据集 D-A 为例, 验证错误检测方法 FeLeDetect 在不同错误率及不同错误类型比的数据集下的通用性. 正如第 1 节所述, 现实世界中数据错误是稀疏的. 因此, 在默认错误类型比(80%)条件下, 将数据集的错误率从 3%变化至 9%, 并测试 FeLeDetect 的 *F*1 分数. 此外, 在默认错误率(5%)条件下, 将错误类型比由 20%变化至 80%, 并测试 FeLeDetect 的 *F*1 分数. 由于各数据集上的结果类似, 这里只报告数据集 D-A 上的错误检测结果. 表 5 给出了实验结果. 从表 5 可以看出, *F*1 分数随着错误率的增加有轻微的提升. 这是因为错误的稀疏性加剧了错误检测的难度. 其次, 可以看到, *F*1 分数随着错误类型比的降低呈现轻微的提升. 这是因为错误类型比越低, 意味着缺失错误或格式错误的占比越大, 而替换错误占比越小. 相较于替换错误, 数据缺失和格式错误更容易被检测出来, 因而检测结果越精确. 然而, 即使在替换错误占主导地位(80%)时, FeLeDetect 仍能获得较高的 *F*1 分数. 这是因为 GEDM 模型能够有效地捕获每一个数据源的特征, 联邦训练 FCTA 保证了跨源数据信息的无损性. 所以, FeLeDetect 适用于不同的错误率及不同错误类型比的数据集.

最后验证 FeLeDetect 的可扩展性. 本实验选取数据集 Adult, 并变化其数据集规模. 表 6 给出了在不同规模的数据下, FeLeDetect 的训练时间 *TT* 和错误检测时间 *DT*, 并分别计算训练时间与数据规模、错误检测时间

与数据规模的皮尔逊相关系数, 以评估其之间的线性相关性.

表 5 FeLeDetect 在不同错误率/错误类型比下的错误检测 *F*1 分数

数据集		错误率				错误类型比			
		3%	5%	7%	9%	80%	60%	40%	20%
D-A	DBLP	0.78	0.84	0.83	0.85	0.83	0.88	0.89	0.89
	ACM	0.89	0.91	0.93	0.97	0.90	0.91	0.94	0.97

从表 6 可以看出, FeLeDetect 的训练时间和错误检测时间都随着数据集规模成近似线性增长(皮尔逊相关系数接近 1). 这证明了 FeLeDetect 具有良好的可扩展性.

表 6 FeLeDetect 的训练时间 *TT* 及错误检测时间 *DT* 随数据集规模变化对比

数据规模|*D*|	*TT*	*DT*
20 000	818.25	39.45
40 000	1 518.05	99.66
60 000	2 237.33	187.97
80 000	3 407.48	299.29
97 864	4 369.12	386.97
皮尔逊相关系数	0.99	0.99

6 总 结

本文提出了一种基于联邦学习的跨源数据错误检测方法 FeLeDetect.

- 首先, 本文设计了一种基于图的错误检测模型 GEDM 以充分捕获每个数据源不同粒度的数据特征.
- 其次, 本文提出了一种信息无损的联邦协同训练算法 FCTA, 在保证数据隐私的前提下, 协同训练部署在不同数据源的错误检测模型 GEDM; 并在此基础上进一步设计了一系列优化方法, 以降低联邦训练过程中的通信开销以及人工标注成本.
- 最后, 在公开数据集上进行了充分的实验, 验证了: (1) GEDM 在本地场景和集中场景下的优越性; (2) FeLeDetect 在 GEDM 的基础上进一步提高了错误检测的精度; (3) 本文提出的通信优化算法大大降低了联邦训练过程中的通信代价.

由于真实的错误数据集往往缺少真实值(ground truth), 因此难以评估错误检测算法的有效性. 未来, 我们计划收集、标注并公开含真实错误的基准数据集, 以便于错误检测任务的进一步研究.

References:

[1] Ilyas IF, Chu X. Trends in cleaning relational data: Consistency and deduplication. Foundations and Trends in Databases, 2015, 5(4): 281−393.

[2] Heidari A, McGrath J, Ilyas IF, Rekatsinas T. Holodetect: Few-shot learning for error detection. In: Proc. of the ACM SIGMOD Int'l Conf. on Management of Data. Amsterdam: ACM, 2019. 829−846.

[3] Guo ZM, Zhou AY. A survey of data quality and data cleaning research. Ruan Jian Xue Bao/Journal of Software, 2002, 13(11): 2076−2082 (in Chinese with English abstract). http://www.jos.org.cn/jos/article/abstract/20021103?st=search

[4] Wang P, He Y. Uni-Detect: A unified approach to automated error detection in tables. In: Proc. of the ACM SIGMOD Int'l Conf. on Management of Data. Amsterdam: ACM, 2019. 811−828.

[5] Dasu T, Loh JM. Statistical distortion: Consequences of data cleaning. Proc. of the VLDB Endowment, 2012, 5(11): 1674−1683.

[6] Wu E, Madden S. Scorpion: Explaining away outliers in aggregate queries. Proc. of the VLDB Endowment, 2013, 6(8): 553−564.

[7] Prokoshyna N, Szlichta J, Chiang F, Miller RJ, Srivastava D. Combining quantitative and logical data cleaning. Proc. of the VLDB Endowment, 2015, 9(4): 300−311.

[8] Elmagarmid AK, Ipeirotis PG, Verykios VS. Duplicate record detection: A survey. IEEE Trans. on Knowledge and Data Engineering, 2006, 19(1): 1−16.

[9] Naumann F, Herschel M. An introduction to duplicate detection. Synthesis Lectures on Data Management, 2010, 2(1): 1−87.

[10] Kandel S, Paepcke A, Hellerstein J, Heer J. Wrangler: Interactive visual specification of data transformation scripts. In: Proc. of the Int'l Conf. on Human Factors in Computing Systems. Vancouver: ACM, 2011. 3363−3372.

[11] Dallachiesa M, Ebaid A, Eldawy A, Elmagarmid A, Ilyas, IF, Ouzzani M, Tang N. NADEEF: A commodity data cleaning system. In: Proc. of the ACM SIGMOD Int'l Conf. on Management of Data. New York: ACM, 2013. 541−552.

[12] Khayyat Z, Ilyas IF, Jindal A, Madden S, Ouzzani M, Papotti P, Quiané-Ruiz JA, Tang N, Yin S. Bigdansing: A system for big data cleansing. In: Proc. of the ACM Int'l Conf. on Management of Data. Victoria: ACM, 2015. 1215−1230.

[13] Fan W, Li J, Ma S, Tang N, Yu W. Towards certain fixes with editing rules and master data. The VLDB Journal, 2012, 21(2): 213−238.

[14] Chu X, Morcos J, Ilyas IF, Ouzzani M, Papotti P, Tang N, Ye Y. Katara: A data cleaning system powered by knowledge bases and crowdsourcing. In: Proc. of the ACM SIGMOD Int'l Conf. on Management of Data. Victoria: ACM, 2015. 1247−1261.

[15] Panko RR. What we know about spreadsheet errors. Journal of Organizational and End User Computing, 1998, 10(2): 15−21.

[16] Regulation GDP. Regulation (EU) 2016/679 of the European parliament and of the council. Regulation (EU), No.679, 2016.

[17] Cappuzzo R, Papotti P, Thirumuruganathan S. Creating embeddings of heterogeneous relational datasets for data integration tasks. In: Proc. of the ACM SIGMOD Int'l Conf. on Management of Data. Portland: ACM, 2020. 1335−1349.

[18] Bonawitz K, Eichner H, Grieskamp W, Huba D, Ingerman A, Ivanov V, Kiddon C, Konecny J, Mazzocchi S, McMahan HB, Overveldt TV, Petrou D, Ramage D, Roselander J. Towards federated learning at scale: System design. In: Proc. of the Machine Learning and Systems. 2019. 374−388.

[19] Abedjan Z, Morcos J, Ilyas IF, Ouzzani M, Papotti P, Stonebraker M. Dataxformer: A robust transformation discovery system. In: Proc. of the IEEE Int'l Conf. on Data Engineering. Helsinki: IEEE, 2016. 1134−1145.

[20] Mahdavi M, Abedjan Z, Fernandez RC, Madden S, Ouzzani M, Stonebraker M, Tang N. Raha: A configuration-free error detection system. In: Proc. of the ACM SIGMOD Int'l Conf. on Management of Data. Amsterdam: ACM, 2019. 865−882.

[21] Krishnan S, Wang J, Wu E, Franklin MJ, Goldberg K. Activeclean: Interactive data cleaning for statistical modeling. Proc. of the VLDB Endowment, 2016, 9(12): 948−959.

[22] Visengeriyeva L, Abedjan Z. Metadata-driven error detection. In: Proc. of the 30th Int'l Conf. on Scientific and Statistical Database Management. Bozen-Bolzano: ACM, 2018. 1−12.

[23] Huang Z, He Y. Auto-Detect: Data-driven error detection in tables. In: Proc. of the ACM SIGMOD Int'l Conf. on Management of Data. Houston: ACM, 2018. 1377−1392.

[24] McMahan HB, Moore E, Ramage D, Hampson S, Agüera y Arcas B. Federated learning of deep networks using model averaging. arXiv:1602.05629, 2016.

[25] Muhammad K, Wang Q, O'Reilly-Morgan D, Tragos E, Smyth B, Hurley N, Geraci J, Lawlor A. Fedfast: Going beyond average for faster training of federated recommender systems. In: Proc. of the 26th ACM SIGKDD Conf. on Knowledge Discovery and Data Mining. CA: ACM, 2020. 1234−1242.

[26] Chen Y, Qin X, Wang J, Yu C, Gao W. Fedhealth: A federated transfer learning framework for wearable healthcare. IEEE Intelligent Systems, 2020, 35(4): 83−93.

[27] Suzumura T, Zhou Y, Baracaldo N, Ye G, Houck K, Kawahara R, Anwar A, Stavarache LL, Watanabe Y, Loyola P, Klyashtorny D, Ludwig H, Bhaskaran K. Towards federated graph learning for collaborative financial crimes detection. arXiv:1909.12946, 2019.

[28] Ji S, Pan S, Long G, Li X, Jiang J, Huang Z. Learning private neural language modeling with attentive aggregation. In: Proc. of the Int'l Joint Conf. on Neural Networks. Budapest: IEEE, 2019. 1−8.

[29] Li T, Sahu AK, Zaheer M, Sanjabi M, Talwalkar A, Smith V. Federated optimization in heterogeneous networks. In: Proc. of the Conf. on Machine Learning and Systems. Austin, 2020. 429−450.

[30] Wu C, Wu F, Lyu L, Huang Y, Xie X. Communication-efficient federated learning via knowledge distillation. Nature Communications, 2022, 13(1): 1−8.

[31] Liu Y, Kang Y, Zhang X, Li L, Cheng Y, Chen T, Hong M, Yang Q. A communication efficient collaborative learning framework for distributed features. arXiv:1912.11187, 2019.

[32] Horváth S, Ho CY, Horvath L, Sahu AN, Canini M, Richtárik P. Natural compression for distributed deep learning. arXiv:1905.10988, 2019.

[33] Goetz J, Malik K, Bui D, Moon S, Liu H, Kumar A. Active federated learning. arXiv:1909.12641, 2019.

[34] Yang Q, Liu Y, Chen TJ, Tong YX. Federated machine learning: Concept and applications. ACM Trans. on Intelligent Systems and Technology, 2019, 10(2): 1−19.

[35] Liang G, Chawathe SS. Privacy-preserving inter-database operations. In Proc. of the Int'l Conf. on Intelligence and Security Informatics. Tucson: Springer, 2004. 66−82.

[36] Hu YC, Niu D, Yang JM, Zhou SP. FDML: A collaborative machine learning framework for distributed features. In: Proc. of the 25th ACM SIGKDD Conf. on Knowledge Discovery and Data Mining. Anchorage: ACM, 2019. 2232−2240.

[37] Fu FC, Shao YX, Yu LL, Jiang JW, Xue HR, Tao YY, Cui B. VF2Boost: Very fast vertical federated gradient boosting for cross-enterprise learning. In: Proc. of the ACM SIGMOD Int'l Conf. on Management of Data. ACM, 2021. 563−576.

[38] Ge CC, Wang PF, Chen L, Liu XZ, Zheng BH, Gao YJ. CollaborER: A self-supervised entity resolution framework using multi-features collaboration. arXiv:2108.08090, 2021.

[39] Wu ZH, Pan SR, Chen FW, Long GD, Zhang CQ, Philip SY. A comprehensive survey on graph neural networks. IEEE Trans. on Neural Networks and Learning Systems, 2020, 32(1): 4−24.

[40] Yao L, Mao CS, Luo Y. Graph convolutional networks for text classification. In: Proc. of the AAAI Conf. on Artificial Intelligence. Honolulu: AAAI, 2019. 7370−7377.

[41] Do K, Tran T, Venkatesh S. Graph transformation policy network for chemical reaction prediction. In: Proc. of the 25th ACM SIGKDD Conf. on Knowledge Discovery and Data Mining. Anchorage: ACM, 2019. 750−760.

[42] Wu CH, Wu FZ, Cao Y, Huang YF, Xie X. FedGNN: Federated graph neural network for privacy-preserving recommendation. arXiv:2102.04925, 2021.

[43] Hamilton W, Ying Z, Leskovec J. Inductive representation learning on large graphs. In: Proc. of the Annual Conf. on Neural Information Processing Systems. Long Beach: NIPS, 2017. 1024−1034.

[44] Zhu L, Liu Z, Han S. Deep leakage from gradients. In: Proc. of the Conf. on Advances in Neural Information Processing Systems. Vancouver: NIPS, 2019. 14747−14756.

[45] Uddin MF, Youssef AM. Cryptanalysis of simple substitution ciphers using particle swarm optimization. In: Proc. of the IEEE Int'l Conf. on Evolutionary Computation. Vancouver: IEEE, 2006. 677−680.

[46] Mudgal S, Li H, Rekatsinas T, Doan A, Park Y, Krishnan G, Deep R, Arcaute E, Raghavendra V. Deep learning for entity matching: A design space exploration. In: Proc. of the ACM SIGMOD Int'l Conf. on Management of Data. Houston: ACM, 2018. 19−34.

[47] Li X, Dong XL, Lyons K, Meng W, Srivastava D. Truth finding on the deep Web: Is the problem solved? Proc. of the VLDB Endowment, 2012, 6(2): 97−108.

[48] Abedjan Z, Chu X, Deng D, Fernandez RC, Ilyas IF, Ouzzani M, Papotti P, Stonebraker M, Tang N. Detecting data errors: Where are we and what needs to be done? Proc. of the VLDB Endowment, 2016, 9(12): 993−1004.

[49] Yan JN, Schulte O, Zhang MH, Wang JN, Cheng R. SCODED: Statistical constraint oriented data error detection. In: Proc. of the ACM SIGMOD Int'l Conf. on Management of Data. Portland: ACM, 2020. 845−860.

[50] Ge CC, Gao YJ, Miao XY, Yao B, Wang HB. A hybrid data cleaning framework using markov logic networks. IEEE Trans. on Knowledge and Data Engineering, 2022, 34(5): 2048−2062.

[51] Powell SG, Baker KR, Lawson B. Errors in operational spreadsheets: A review of the state of the art. In: Proc. of the 42nd Hawaii Int'l Conf. on System Sciences. Waikoloa: IEEE, 2009. 1−8.

[52] Mariet Z, Harding R, Madden S. Outlier detection in heterogeneous datasets using automatic tuple expansion. Technical Report, MIT-CSAIL-TR-2016-002, Cambridge: Computer Science and Artificial Intelligence Laboratory, Massachusetts Institute of Technology, 2016.

[53] Lahijani MM. Semi-supervised data cleaning [Ph.D. Thesis]. Berlin: Technical University of Berlin, 2020.

[54] Paszke A, Gross S, Massa F, Lerer A, Bradbury J, Chanan G, Killeen T, Lin Z, Gimelshein N, Antiga L, Desmaison A, Köpf A, Yang E, DeVito Z, Raison M, Tejani A, Chilamkurthy S, Steiner B, Fang L, Bai J, Chintala S. Pytorch: An imperative style, high-performance deep learning library. In: Proc. of the Annual Conf. on Neural Information Processing Systems. Vancouver: NIPS, 2019. 8024−8035.

附中文参考文献:

[3] 郭志懋, 周傲英. 数据质量和数据清洗研究综述. 软件学报, 2002, 13(11): 2076−2082. http://www.jos.org.cn/jos/article/abstract/20021103?st=search

陈璐(1989－), 女, 博士, 教授, 博士生导师, CCF 专业会员, 主要研究领域为度量空间数据管理, 时空数据管理, 查询可用性分析.

郭宇翔(1998－), 男, 博士生, 主要研究领域为数据集成, 数据准备.

葛丛丛(1995－), 女, 博士, 主要研究领域为数据集成, 数据清洗.

郑白桦(1977－), 女, 博士, 教授, 博士生导师, 主要研究领域为移动/普适计算, 空间数据库, 大数据分析.

高云君(1977－), 男, 博士, 教授, 博士生导师, CCF 高级会员, 主要研究领域为数据库, 大数据管理与分析, DB 与 AI 融合.