

考虑属性和空间逻辑的地震灾情数据 清洗器设计与实现*

张莹¹, 郭红梅^{1♣}, 尹文刚², 赵真¹, 胡斌¹

(1. 四川省地震局, 四川 成都 610041; 2. 武警警官学院, 四川 成都 610213)

摘要: 针对目前地震灾情信息普遍存在的质量问题, 根据灾情数据的属性逻辑, 采用逻辑分离的清洗框架中基于规则引擎的数据清洗方法进行初步清洗; 再从灾情数据的空间逻辑关系出发, 通过聚类分析进一步修正数据中存在的错误, 弥补基于逻辑分离的清洗框架在孤立点异常检测方面的不足。基于上述规则引擎与空间聚类相结合的清洗技术, 设计并实现了一种考虑属性和空间逻辑的地震灾情数据清洗器, 并应用于2017年“8·8”九寨沟7.0级地震, 分析四川省地震灾情快速上报接收处理系统接收到的客观震感灾情信息。结果表明: 地震灾情数据清洗器可快速检测并修正异常的灾情信息, 有效提高接收到的灾情数据质量。

关键词: 地震灾情; 数据清洗; 属性和空间逻辑; 规则引擎; 空间聚类

中图分类号: P315.94

文献标识码: A

文章编号: 1000-0666(2021)01-0078-09

0 引言

地震发生后, 快速获取准确可靠的灾情信息是综合研判受灾情况、进行应急指挥决策的基础。信息化技术的不断发展极大地丰富了地震灾情信息的来源, 使震后通过多种手段更加快速获取灾情信息成为可能(张方浩等, 2016)。信息类型的多样性, 对开展地震应急工作具有重要意义, 但在实际应用中, 在获取到的原始地震灾情信息中, 包含了很多重复、不完整、错误等“脏数据”。用传统的人工方法手动处理“脏数据”效率低下, 如何快速检测出灾情数据中存在的质量问题, 及时剔除或修正“脏数据”, 使获取到的灾情信息准确可靠, 是现阶段地震应急工作中亟待解决的关键问题之一。

作为数据质量研究的基础, 数据清洗基于不同的数据逻辑类型, 是改善数据质量的重要方法(任明仑, 宋月丽, 2019)。数据的属性逻辑是数据本身及与其他数据间在性质上的一种逻辑关系。在目前常用的数据清洗技术中, 针对数据属性逻

辑, 将数据清洗框架分为基于一般数据抽取转换的清洗框架和基于逻辑分离的清洗框架, 如基于规则引擎的数据清洗方法等就属于后一种清洗框架(曹建军等, 2012)。基于逻辑分离的清洗框架将定义的清洗规则和对应的编码进行分离, 按照清洗需求可智能化地调整清洗过程, 在减少人工干预和编程工作量的同时实现了低耦合的数据清洗(王江, 2016), 避免了基于一般数据抽取转换的清洗框架将数据清洗过程进行分解时, 修改处理逻辑需回到需求分析阶段重新进行系统编译的不足。同时, 实验证明由于不易受数据分布的影响, 通过基于逻辑分离的清洗框架虽能检测出大部分的异常数据, 但却可能遗漏一些孤立点存在的异常(封富君等, 2017)。而灾情数据除属性逻辑外, 还存在因空间相关性表现在性质上的空间逻辑关系。对于具有空间逻辑关系的数据, 通过聚类分析可有效检测出异常的孤立点, 从而进一步修正数据中存在的错误(施伟锋等, 2019), 弥补基于属性逻辑分离清洗框架的缺陷。

因此, 为解决目前地震灾情信息普遍存在的

* 收稿日期: 2020-04-02.

基金项目: “十三五”国家重点研发计划项目课题(2017YFC1500905)、中国地震局地震工程与工程振动重点实验室重点专项(2019EEEEVL0102)和四川地震科技创新团队专项(201901)联合资助。

第一作者简介: 张莹(1989-), 工程师, 主要从事地震灾害风险及信息处理研究. E-mail: 115453242@qq.com.

♣ 通讯作者简介: 郭红梅(1984-), 高级工程师, 主要从事地震风险和地震灾情信息处理研究.
E-mail: 115453242@qq.com.

质量问题，本文采用逻辑分离的清洗框架中基于规则引擎的方法与空间聚类分析相结合的技术，设计并实现了一种考虑属性和空间逻辑的地震灾情数据清洗器，并以2017年“8·8”九寨沟7.0级强烈地震为例，分析四川省地震灾情快速上报接收处理系统以多种灾情获取手段接收到的客观震感灾情信息。

1 系统设计

1.1 总体架构设计

为满足灾情数据清洗时在安全性和时效性方

面的需求，系统采用3层式C/S架构，总体架构如图1所示。从逻辑层面分为数据层、应用层和用户层。其中，数据层由数据访问和数据存储两部分构成，通过数据访问实现对地震灾情快速上报接收处理系统前置库和正式库的访问，由数据存储将清洗完成后的数据存入灾情上报正式库中。考虑属性和空间逻辑的数据清洗，应用层提供基于规则引擎与空间聚类分析方法的清洗功能，还包括空间展示、系统管理等业务功能，实现对不同来源地震灾情数据的清洗、展示和管理。在用户层中由管理人员对系统进行运维，保障业务人员的正常操作。

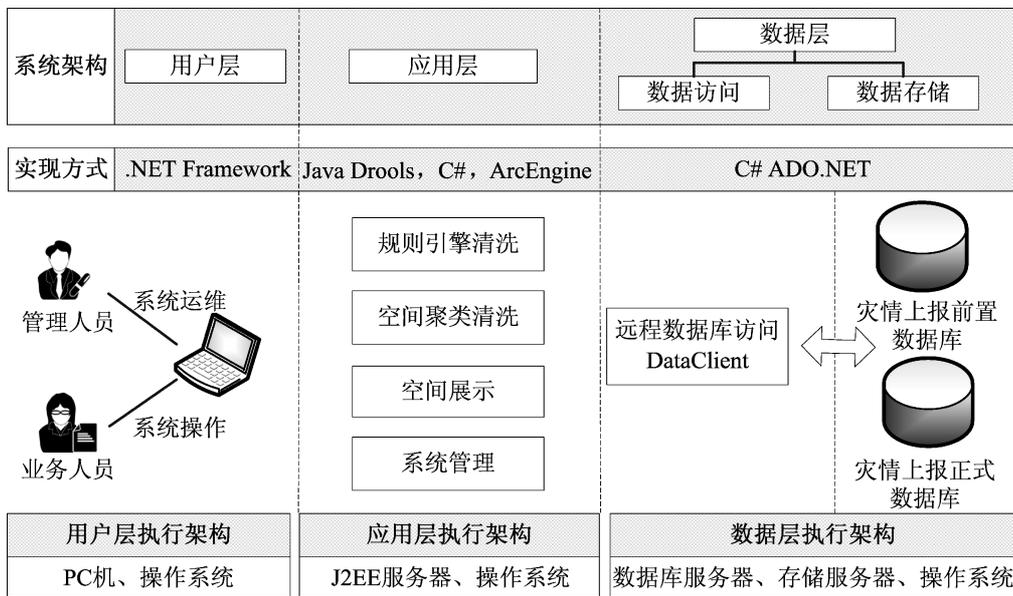


图1 地震灾情数据清洗器总体架构

Fig. 1 Overall architecture diagram of earthquake disaster data cleaner

1.2 业务流程设计

根据地震灾情数据清洗器的总体架构，在对灾情数据清洗时，各逻辑层之间需按数据流向和一定的流程进行相互调用完成数据清洗，笔者按照此思路设计其业务流程。由图2可见，首先从数据层的灾情上报前置库中导出待清洗的灾情数据，然后针对数据属性逻辑，在应用层中通过执行规则引擎实现对灾情数据的初步清洗，再考虑数据空间逻辑，应用空间聚类对经规则引擎清洗后的灾情数据进行聚类分析，进一步检测并修正异常的孤立点数据，最后将清洗完成后的灾情数据导

入数据层的灾情上报正式库中存储。

2 系统实现

2.1 考虑属性和空间逻辑的灾情数据清洗实现

通过整理不同来源的地震灾情数据，可将地震灾情信息分为震感、建筑物破坏、人员影响、居民生活影响、交通系统破坏、地面破坏和次生灾害7大类（陈维锋等，2014），各大类又可细分为对应灾情下的25小类，灾情信息构成及分类如图3所示。

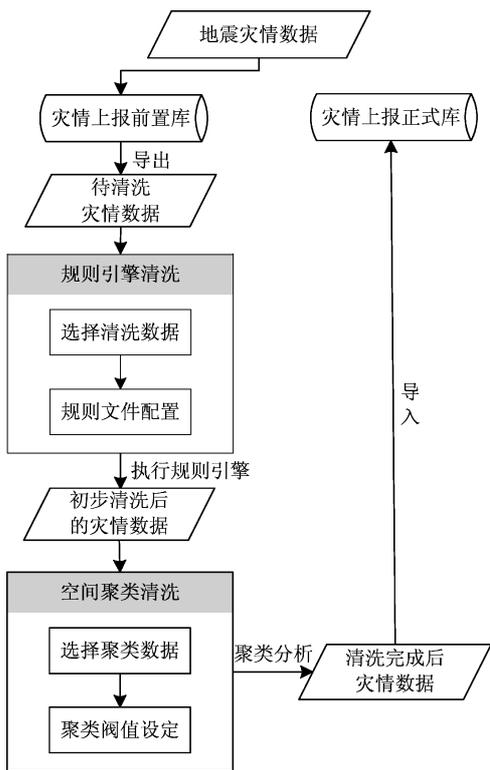


图2 地震灾情数据清洗器业务流程
Fig. 2 The processing flowchart of the earthquake disaster data cleaner

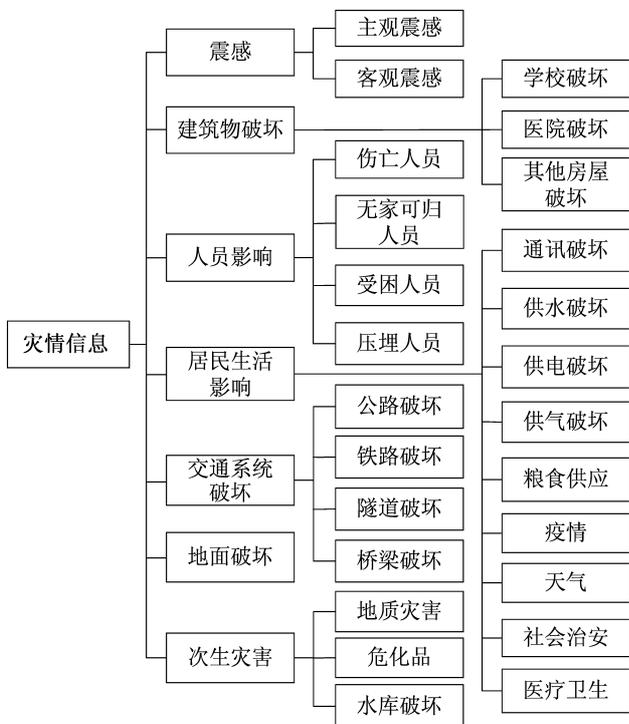


图3 地震灾情信息构成及分类
Fig. 3 Composition and classification of the earthquake disaster information

2.1.1 考虑属性逻辑

针对数据属性逻辑，采用逻辑分离的清洗框架中基于规则引擎的清洗方法，将逻辑关系通过一系列规则表达后，实现数据清洗，清洗流程如图4所示。首先对源数据集进行质量分析，找出数据存在的质量问题，针对不同的质量问题按照标准的规则编码规范定义数据清洗规则（赵志伟，2018），然后应用规则引擎组合、编排及解析定义的规则，再通过样本数据集校验经规则引擎处理后的清洗规则，根据校验发现的问题对规则进行反复修改，最终形成规则库，通过规则执行实现地震灾情“脏数据”的清洗转换。

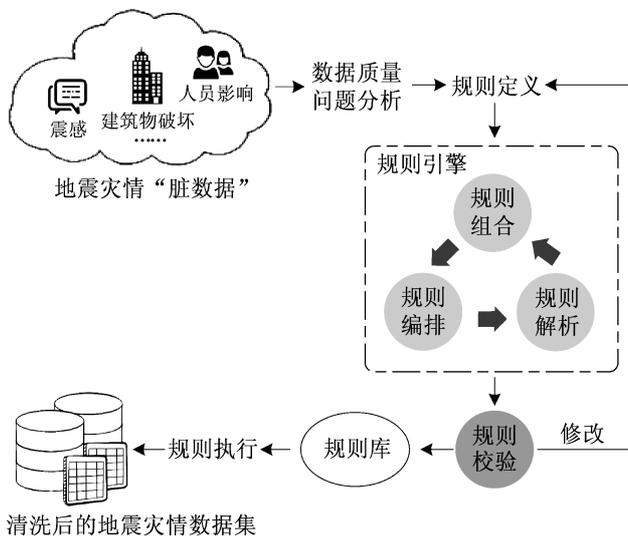


图4 基于规则引擎的地震灾情数据清洗流程
Fig. 4 Cleaning process of the earthquake disaster data based on rule engine

通过分析历次地震中不同来源的地震灾情信息，笔者发现灾情数据存在的质量问题主要包括关键属性值缺省、字段记录重复、灾情属性值与实际受灾情况不符等，再基于规则引擎的规则编码规范，分别制定数据清洗规则。

(1) 关键属性值缺省

应用于地震应急中的灾情信息应具备完整的上报时间、地点、受灾情况等关键属性，若关键属性值缺省则可能失去实际应用的价值，关键属性值缺省的记录经数据清洗后将被剔除。清洗规则如下：

< rule name = " Attribute default " >/关键属性值缺省规则集

```

    < parameter identifier = " Eqdata " > < class >
Data </class > </parameter >
    < java:condition > Eqdata. getByTime( " Time " )
= null </java:condition >/时间字段为空
    < java:condition > Eqdata. getByLocation( " Loca-
tion " ) = null </java:condition >/地点字段为空
    < java:condition > Eqdata. getByDisaster( " Disas-
ter " ) = null </java:condition >/灾情字段为空
    < java:consequence > Eqdata. deleteDta( ); </
java:consequence >/执行剔除操作
</rule >

```

(2) 字段记录重复

字段记录重复主要出现在震感类灾情信息中。因部分灾报员在上报震感信息时,除主震震感外,还持续上报了同一地点的余震震感,若两者相同将使记录重复,造成数据冗余的同时,还可能影响对主震震感的总体判断。因此按照时间顺序,对同一次震害事件,按照只保留先上报的主震震感,剔除余震震感的规则进行字段记录重复数据清洗。清洗规则如下:

< rule name = " Record repeated " >/字段记录重复规则集

```

    < parameter identifier = " Eqdata. SubShock " >
< class > Data </class > </parameter >
    < java:condition > Eqdata. SubShock. getByLoca-
tion( " Location " ) = Eqdata. SubShock. getByLocation
( " Location " ) </java:condition >/地点字段相同
    < java:condition > Eqdata. SubShock. getByDisas-
ter( " Disaster " ) = Eqdata. SubShock. getByDisaster
( " Disaster " ) </java:condition >/震感灾情字段
相同
    < java:condition > Eqdata. SubShock. getByTime
( " Time " ) = time1 </java:condition >/读取时间
字段
    < java:condition > Eqdata. SubShock. getByTime
( " Time " ) = time2 </java:condition > < java:condi-
tion > time1. before( time2 ) </java:condition >/比较
时间先后
    < java:consequence > Eqdata. SubShock. delet-
eDta( time2 ); </java:consequence >/执行剔除操作
</rule >

```

(3) 灾情属性值与实际受灾情况不符

在接收到的震感、建筑物破坏、交通系统破坏等灾情信息中,存在灾情属性值与实际受灾情况不符的情况,例如同一地点的震感信息为无震感,但建筑物破坏信息中的建筑物破坏属性却为部分破坏,灾情信息间相互矛盾。针对这一问题,郭红梅等(2015)根据中国地震烈度表对人的感觉、房屋破坏及其他震害现象描述建立了地震灾情信息对应关系,如表1所示。表中不同灾情信息对应的灾情属性设计如下:主观震感灾情属性:Ⅰ级:无震感;Ⅱ级:仅仅有感;Ⅲ级:震感强,可行走;Ⅳ级:站立不稳,行走困难;Ⅴ级:被地震摔倒。客观震感灾情属性:Ⅰ级:房屋破坏不可见;Ⅱ级:房屋破坏不易见;Ⅲ级:少数房屋轻微破坏,房屋破坏可见;Ⅳ级:多数房屋轻微破坏,房屋破坏易见;Ⅴ级:多数房屋破坏;Ⅵ级:多数房屋毁坏。建筑物破坏灾情属性:Ⅰ级:完好;Ⅱ级:部分破坏;Ⅲ级:毁坏。交通系统破坏灾情属性:Ⅰ级:通行;Ⅱ级:通行困难;Ⅲ级:中断。

表1 地震灾情信息对应关系

Tab. 1 Correspondence relationship of earthquake disaster information

灾情信息	灾情属性					
主观震感	Ⅱ级	Ⅲ级	Ⅲ级	Ⅳ级	Ⅳ级	Ⅴ级
客观震感	Ⅰ级	Ⅱ级	Ⅲ级	Ⅳ级	Ⅴ级	Ⅵ级
建筑物破坏	Ⅰ级	Ⅰ级	Ⅰ级或 Ⅱ级	Ⅱ级	Ⅱ级或 Ⅲ级	Ⅲ级
交通系统破坏	Ⅰ级	Ⅰ级	Ⅰ级	Ⅱ级	Ⅱ级	Ⅱ级或 Ⅲ级

对灾情数据进行融合,即当同一地点的灾情属性值满足表1中某一列的条件时,可通过制定规则检测并修正灾情属性值与实际受灾情况不符的记录。以建筑物破坏灾情信息为例的部分清洗规则如下:

< rule name = " Attribute error " >/属性值与实际不符规则集

```

    < parameter identifier = " Eqdata " > < class >
Data </class > </parameter >
    < java:condition > Eqdata. SubShock. getByDisas-
ter( " Disaster " ) = " Ⅱ级 " </java:condition >/主观

```

震感灾情属性为 II 级

```
<java:condition> Eqdata. ObjShock. getByDisaster("Disaster") = " I 级" </java:condition>/客观震感灾情属性为 I 级
```

```
<java:condition> Eqdata. Traffic. getByDisaster("Disaster") = " I 级" </java:condition>/交通系统破坏灾情属性为 I 级
```

```
<java:condition> Eqdata. Building. getByDisaster("Disaster") != " I 级" </java:condition>/建筑物破坏灾情属性不为 I 级
```

```
<java:condition> Eqdata. Building. modifyDisaster("Disaster") = " I 级" </java:condition>/执行修正操作
```

```
</rule>
```

2.1.2 考虑空间逻辑

对于具有空间逻辑关系的数据,空间聚类分析方法根据数据的空间分布,将相同或相似属性的数据归入同一集合中,未被归入任何集合的数据为空间离群点,即数据异常点(李明等,2019),从而进一步发现基于规则引擎的方法在进行数据清洗后遗漏的异常孤立点数据。

首先从地震灾情数据中选定需要清洗的灾情数据集 $X = (x_1, x_2, \dots, x_n)$, 其中 n 为灾情数据记录条数,再从中随机抽取 k 个元组作为 k 个初始聚类中心点 $v^{(0)} = \{v_1^{(0)}, v_2^{(0)}, \dots, v_k^{(0)}\}$, 然后通过公式(1)计算其他数据到初始聚类中心点的欧氏距离(杨俊闯,赵超,2019):

$$d_{ij} = \sqrt{\sum_{i=1}^n \sum_{j=1}^k [x_i - v_j^{(0)}]^2} \quad (1)$$

根据最短距离将数据划归到距离最近的聚类中心所在集合中,将通过计算但未被归入任何集合的数据暂时标记为异常点。在全部数据计算完成后,重新迭代计算聚类中心 $v^{(l)}$,对数据再次进行归类,直至相邻两次计算的聚类中心间差距达到设定的阈值 ε 或没有变化为止,将仍未被归入任何集合的数据确定为异常点。算法实现部分关键代码如下:

```
public class DatacleanCluster {
    private int k;//元组个数
    private int num;//迭代次数
    private List<Double> X;//原始灾情数据集
```

```
private List<Point> X = new ArrayList<Point>();
public double getDis(Point x, Point v) { return
Math.sqrt(Math.pow(x.getX() - v.getX(),
Math.pow(x.getY() - v.getY(), 2));} };//计算
其他点到聚类中心点的欧氏距离
public List<DatacleanCluster> cluster(List
<DatacleanCluster> cluster) {
for(int i=0;i<num;i++){
Set<Point> center = new HashSet<Point>();
//迭代计算聚类中心
for(int j=0;j<k;j++){
List<Point>
Center = DatacleanCluster.get(j).getMembers();
int size = Center.size();
if(size<3){
Center.add(cluster.get(j).getCenter());
//得到新的中心点
continue;}
if(lastCenter.containsAll(center)){
lastCenter = center;
break;}} //迭代计算的聚类中心不再变化,退出迭代
for(int ns=0;ns<k;ns++){
noise += DatacleanCluster.get(ns).getError();} //得到异常点集合
return DatacleanCluster;}
}
```

2.2 主要功能实现

根据系统总体架构及业务流程,笔者设计并实现了考虑属性和空间逻辑的灾情数据清洗器。地震灾情数据清洗器由灾情数据、规则清洗、数据融合、空间聚类、空间展示和系统管理 6 大功能模块构成。

数据清洗器采用统一的控制界面将各功能模块集成,实现系统主界面菜单、工具栏、数据选项卡、地图浏览窗口等构件的统一布局。系统主要功能实现如下:

(1) 规则清洗模块

规则清洗模块采用规则引擎制定数据清洗规则,通过清洗规则设置,可对各类灾情数据中存在的键属性值缺省、字段记录重复等问题进行检测与修正。在处理结果显示和图形输出选项卡中可查看数据清洗条数等详细记录及数据的空间

分布情况，操作界面如图 5a 所示。

(2) 数据融合模块

数据融合模块基于中国地震烈度表对人的感觉、房屋破坏及其他震害现象描述所建立的地震灾情信息对应关系，制定融合规则，将经过规则清洗的地震灾情数据进行融合，修正灾情属性值与实际受灾情况不符的记录，经过修正的灾情数据以黄色高亮显示，操作界面如图 5b 所示。

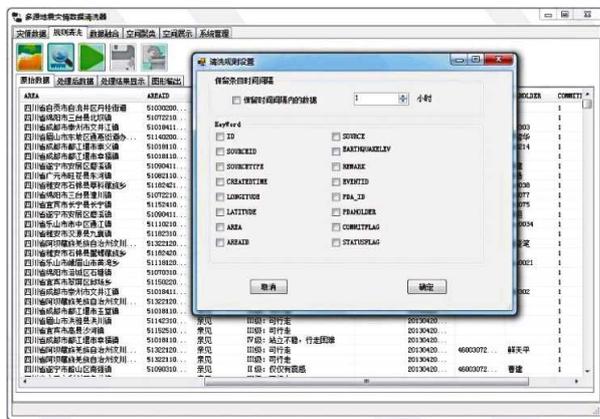
(3) 空间聚类模块

空间聚类模块应用空间聚类分析方法，对通过规则清洗及数据融合初步清洗后的灾情数据进行聚类分析，进一步检测并修正异常的孤立点数据。

在聚类文件中选择需要清洗的灾情数据，为使处理结果更符合实际情况，可针对不同的灾情数据设置不同的空间聚类半径。点击清洗按钮后，在处理结果选项卡中可查看聚类分析后的灾情数据，其中对经修正处理的灾情数据进行了标记，操作界面如图 5c 所示。

(4) 空间展示模块

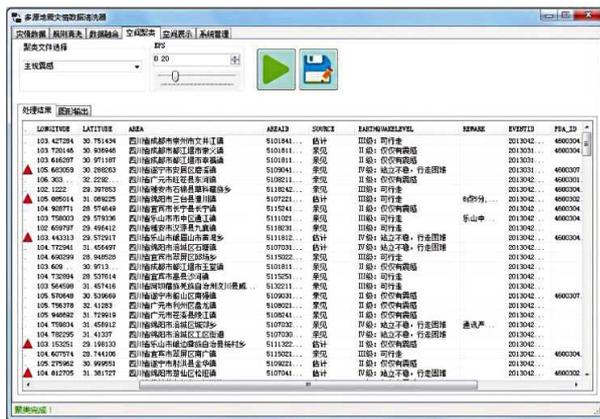
空间展示模块可分级展示选定灾情数据的总体空间分布情况，与导入的烈度圈等矢量数据进行对比，制作输出专题图，并提供基础的矢量计算及栅格分析等空间分析功能，实现灾情信息的可视化，操作界面如图 5d 所示。



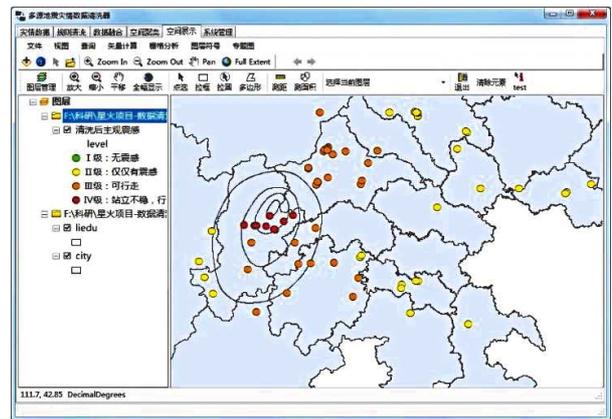
(a) 规则清洗模块界面



(b) 数据融合模块界面



(c) 空间聚类模块界面



(d) 空间展示模块界面

图 5 地震灾情数据清洗器操作界面

Fig. 5 Operation interfaces of the seismic disaster data cleaner

3 应用实例分析

笔者以四川省地震灾情快速上报接收处理系统在 2017 年“8·8”九寨沟 7.0 级强烈地震中接

收到的灾情信息为例，对系统进行实例应用分析。在九寨沟 7.0 级强烈地震中，系统共收到 151 个受灾点上报的主观震感、客观震感、人员伤亡、房屋破坏等灾情信息共 542 条。为便于与实际地震烈度进行对比，科学评价清洗效果，同时考虑接收

到的各类灾情信息数量,笔者选取与地震烈度评定最相关的共 138 条客观震感灾情信息,应用地震灾情数据清洗器进行数据清洗。

首先在规则清洗模块中设置清洗规则,共检测并剔除客观震感灾情信息中的重复记录 30 条,关键属性值缺省记录 5 条,经规则清洗的信息示例如表 2 所示。再通过数据融合模块,将同一地点上报的主观震感、客观震感、建筑物破坏、交通系

统破坏灾情信息按照规则进行融合,共检测并修正客观震感灾情信息中灾情属性值与实际受灾情况不符的记录 21 条,经数据融合清洗的信息示例如表 3 所示。最后,在空间聚类模块中设置聚类半径,对经过规则清洗及数据融合后的灾情信息进行聚类分析,再次检测并修正客观震感灾情信息中灾情属性值与实际受灾情况不符的记录共 17 条,经空间聚类清洗的信息示例如表 4 所示。

表 2 经过规则清洗的灾情信息示例

Tab. 2 Examples of disaster information after rule cleaning

客观震感灾情信息关键字段			质量问题类型
上报时间	地点	客观震感级别	
2017-08-08 21:40	九寨沟县大录乡	Ⅳ级:多数房屋轻微破坏,房屋破坏易见	字段记录重复
2017-08-08 22:00	九寨沟县大录乡	Ⅳ级:多数房屋轻微破坏,房屋破坏易见	
2017-08-08 22:00	九寨沟县马家乡	—	关键属性值缺省

表 3 经过数据融合清洗的灾情信息示例

Tab. 3 Examples of disaster information after data fusion cleaning

灾情信息	上报时间	地点	主观震感级别	客观震感级别	建筑物破坏等级	交通系统破坏等级
	2017-08-08 22:27	若尔盖县唐克乡	Ⅲ级:震感强,可行走	Ⅴ级:多数房屋破坏	Ⅱ级:部分破坏	Ⅰ级:通行
数据融合修正			无误,不修正	Ⅲ级:少数房屋轻微破坏,房屋破坏可见	无误,不修正	无误,不修正

表 4 经过空间聚类清洗的灾情信息示例

Tab. 4 Examples of disaster information after spatial cluster cleaning

上报时间	地点	位置		客观震感级别	空间聚类修正
		$\varphi_N/(\circ)$	$\lambda_E/(\circ)$		
2017-08-08 21:58	九寨沟县漳扎镇	33.25	103.79	Ⅵ级:多数房屋毁坏	无误,不修正
2017-08-08 22:10		33.22	103.84	Ⅵ级:多数房屋毁坏	无误,不修正
2017-08-08 22:21		33.12	103.75	Ⅵ级:多数房屋毁坏	无误,不修正
2017-08-08 22:17		33.18	103.78	Ⅲ级:少数房屋轻微破坏,房屋破坏可见	Ⅵ级:多数房屋毁坏

经过上述数据清洗,应用空间展示模块分级展示清洗前后的客观震感总体空间分布情况,并导入实际地震烈度圈等矢量数据进行对比,制作输出专题图,如图 6 所示。由图可见,在清洗前,部分灾情点的客观震感信息与实际地震烈度分布不符。例如,Ⅷ度区内存在个别客观震感为Ⅲ级:少数房屋轻微破坏,房屋破坏可见和Ⅳ级:多数房屋轻微破坏,房屋破坏易见的灾情点,与《中国地震烈度表》(GB/T 17742—2008)

对烈度为Ⅷ度时房屋多有损坏、少数破坏的描述不符。在清洗后,检测并修正了客观震感中灾情属性值与实际受灾情况不符的记录,其空间分布与地震现场实际调查形成的烈度圈分布吻合良好,表明灾情数据质量较清洗前有了较大提高。同时,执行各清洗步骤用时均在 1~2 s,即可快速检测并修正异常的灾情信息,相比传统的人工手动方法,极大提高了灾情数据质量问题处理的效率。

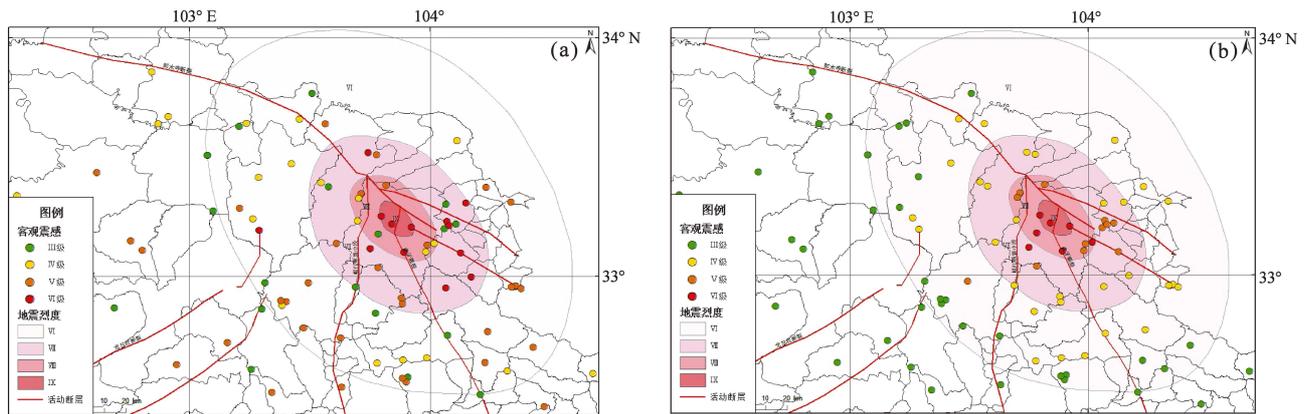


图6 九寨沟7.0级地震客观震感数据清洗前(a)、后(b)空间分布对比

Fig. 6 Spatial distribution comparison of objective seismic data of the Jiuzhaigou M7.0 earthquake before (a) and after (b) cleaning

4 结论

本文基于规则引擎与空间聚类分析相结合的清洗技术,设计并实现了一种考虑属性和空间逻辑的地震灾情数据清洗器,为解决地震灾情数据质量问题提供了技术平台。系统对2017年“8·8”九寨沟7.0级强烈地震客观震感灾情信息清洗后的结果表明,系统能快速处理多种数据质量问题,有效提高接收到的地震灾情数据质量。但随着信息化技术的不断发展,未来地震灾情信息的来源将更加多样化,信息量也将不断增大,可能出现新的数据质量问题,对数据清洗的性能等提出了更高的要求。因此,在后续工作中需不断对系统进行优化,例如考虑将规则引擎的匹配方式采用RETE等模式匹配算法替代目前的表达式解析,改进基于规则的清洗逻辑对不同数据质量问题的适用能力等(何群,2017)。采用负载均衡的方法将数据清洗流程进一步细化,提高并发处理能力,避免因灾情数据量激增导致的系统运行缓慢等问题,使地震灾情数据清洗器不断适应灾情数据质量的需求,成为更实用化的数据清洗工具。

参考文献:

- 曹建军,刁兴春,陈爽,等. 2012. 数据清洗及其一般性系统框架[J]. 计算机科学,39(S3):207-211.
- 陈维锋,郭红梅,张翼,等. 2014. 四川省地震灾情快速上报接收处理系统[J]. 灾害学,29(2):116-122.
- 封富君,姚俊萍,李新社,等. 2017. 大数据环境下的数据清洗框架研究[J]. 软件,38(12):193-196.
- 郭红梅,陈维锋,张莹,等. 2015. 多源地震灾情信息融合的烈度判定方法研究[J]. 华北地震科学,33(3):43-49.
- 何群. 2017. 基于关系逻辑的大数据处理研究[J]. 微电子学与计算机,34(8):1-5.
- 李明,王贵文,柴旭荣,等. 2019. 基于空间聚类的中国东北气候分区及其气象干旱时间变化特征[J]. 自然资源学报,34(8):1682-1693.
- 任明仑,宋月丽. 2019. 大数据:数据驱动的过程质量控制与改进新视角[J]. 计算机集成制造系统,25(11):2731-2742.
- 施伟锋,卓金宝,兰莹,等. 2019. 一种基于属性空间相似性的模糊聚类算法[J]. 电子与信息学报,41(11):2722-2728.
- 王江. 2016. 数据清洗技术研究及清洗框架的设计与实现[D]. 内蒙古:内蒙古大学.
- 杨俊闯,赵超. 2019. K-Means聚类算法研究综述[J]. 计算机工程与应用,55(23):7-14,63.
- 张方浩,和仕芳,吕佳丽,等. 2016. 基于互联网的地震灾情信息分类编码与初步应用研究[J]. 地震研究,39(4):664-672.
- 赵志伟. 2018. 智能化规则引擎技术研究[J]. 软件,39(8):65-69.
- GB/T17742—2008,中国地震烈度表[S].

Design and Implementation of Seismic Disaster Data Cleaner Considering Attributes and Spatial Logic

ZHANG Ying¹, GUO Hongmei¹, YIN Wengang², ZHAO Zhen¹, HU Bin¹

(1. *Sichuan Earthquake Agency, Chengdu 610041, Sichuan, China*)

(2. *College of Armed Police Officer, Chengdu 610213, Sichuan, China*)

Abstract

Aiming at the general quality problems of the current earthquake disaster information, we use the rules engine – based data cleaning method in the logically separated cleaning framework to perform preliminary cleaning of the disaster data according to the attribute logic of disaster data. Then starting from the spatial logical relationship of disaster data, we further correct the errors by cluster analysis, which can make up for the shortcomings of outlier detection based on logical separation cleaning framework. By the cleaning technology combining the above rule engine and spatial clustering, we designed and implemented an earthquake disaster data cleaner that considers attributes and spatial logic, and applied it to the 2017 “Agu. 8” Jiuzhaigou $M7.0$ strong earthquake to analyze the objective seismic sensation disaster information received by the Sichuan earthquake rapid report receiving and processing system. The results show that the earthquake disaster data cleaner can quickly detect and correct abnormal disaster information, and effectively improve the quality of the received disaster data.

Keywords: earthquake disaster; data cleaning; attributes and spatial logic; rule engine; spatial clustering