

**大数据治理**

***理论与方法***

王宏志李默涵◎编著

中国工信出版集团



電子工靠出**版** **社**.

BtI:/OwUS . LOi

**大数据治理**

**理论与方法**

王宏志 李默涵 **编著**

**電** **子** **工** **業** **出** **版** **社**

**Publishing House of Electronics Industry**

北京·BEIJING

**内** **容** **简** **介**

大数据治理是传统信息治理的延续和扩展，其涉及的内容非常广泛。大数据治理确保以正确的方式 对数据和信息进行管理，为大数据的有效应用保驾护航，使得数据成为一个有机整体而不是各自为政。 大数据治理所需的技术包括大数据管理、存储、质量、共享与开放、安全与隐私保护等多个方面。本书 首先对大数据治理的背景和基本概念进行简要介绍，为读者提供大数据治理的背景知识；然后从政策、 管理和技术等方面对大数据治理相关的概念和方法加以介绍，对数据架构管理、元数据管理、主数据管 理、数据集成、数据质量管理、数据的标准化、数据资产化、数据安全与隐私保护等进行深入探讨，以 期为读者提供一个比较全面的大数据治理的场景。

本书可作为高等院校“数据科学与大数据技术”专业的本科生、研究生的教材，也可供大数据领域 的从业人员阅读，还可为对大数据治理感兴趣的广大读者提供参考。

未经许可，不得以任何方式复制或抄袭本书之部分或全部内容。

版权所有，侵权必究。

**图书在版编目** **(CIP)** **数据**

大数据治理：理论与方法/王宏志，李默涵编著.一北京：电子工业出版社，2021.10 ISBN 978-7-121-42181-5

I.① 大… Ⅱ.①王…②李 … Ⅲ.①数据管理一高等学校一教材 IV.①TP274

中国版本图书馆 CIP 数据核字(2021)第204000号

责任编辑：刘 瑀

印 刷：天津千鹤文化传播有限公司 装 订：天津千鹤文化传播有限公司 出版发行：电子工业出版社

北京市海淀区万寿路173信箱

邮编：100036

开 本：787×1092 1/16 印张：13.25 字数：339千字

版 次：2021年10月第1版

印 次：2021年10月第1次印刷

定 价：58.00元

凡所购买电子工业出版社图书有缺损问题，请向购买书店调换。若书店售缺，请与本社发行部联 系，联系及邮购电话：(010)88254888,88258888。

质量投诉请发邮件至 zlts@phei.com.cn, 盗版侵权举报请发邮件至 dbqq@phei.com.cn。 本书咨询联系方式： liuy01@phei.com.cn。



**王宏志**

哈尔滨工业大学计算学部教授、博士生导师，英

才学院副院长，海量数据计算研究中心主任，数据科

学与大数据技术专业负责人，美国加州大学欧文分校

博士后，微软亚洲研究院铸星计划访问学者，青年龙江

学者。主要研究方向为大数据治理、大数据管理与分

析、数据库系统、工业大数据等。在VLDB、

SIGMOD 等国内外重要会议和期刊上发表学术论文

300余篇，出版学术专著三本，其论文被SCl 收录90余

次，他引3000余次，授权发明专利29项。获得微软学

者、中国优秀数据库工程师、IBM 博士英才等称号，

获得黑龙江省自然科学一等奖和教育部高等学校科技

进步一等奖各一项，获得黑龙江省青年科技奖、宝钢

优秀教师奖、CSC-IBM 奖教金。先后主持国家自然

科学基金重点项目、国家支撑计划课题、国家博土后

特别资助等10余项。其主讲的MOOC 课程“大数据算

法”获批第一批全国精品在线开放课程。

**李默涵**

1987年10月出生，博土，副教授，广州大学“百

人计划”引进人才，广州大学-任子行工控网络安全态

势感知联合实验室常务副主任。主要研究方向为数据

管理、数据质量、数据安全等。于哈尔滨工业大学取

得学士、硕士、博士学位。目前是CCF 数据库专委会

通信委员， CCF 女计算机工作者委员会委员，广东省

计算机学会大数据专委会和人工智能专委会委员，是

多个国内外知名期刊的审稿人。主持国家自然科学基

金、广东省自然科学基金等项目，先后参与973项

目、国家重点研发计划、广东省重点领域研发计划等

项目10余项。



**Preface**

在数字经济时代，大数据的价值及应用受到了越来越多数据拥有者和使用者的重 视。由于真实数据经常来自独立自治的数据源，而这些数据源存在着缺少面向具体应 用的顶层设计、缺少质量保障等机制问题，因此很多大数据分析模型和算法达不到预 期的效果。这些问题催生了大数据治理领域的研究。大数据治理是连接大数据科学和 应用的桥梁，要实现大数据的变现，就离不开科学的大数据治理，离不开与时俱进的 管理革新。目前，大数据治理的必要性已得到较为广泛的认可，也已有许多成功案例。

本书全面、系统地向读者解读大数据治理这一 复杂主题。对数据科学与大数据专 业的本科生、研究生和大数据领域的从业人员来说，本书能够帮助读者理解大数据治 理的框架、流程、技术和典型应用等，具有一 定的可参照性、可操作性和可读性。本 书为读者应对大数据治理领域的挑战，提供了理论和实践参考。

本书共分为9章，第1章对大数据治理的背景进行了详细的介绍，然后清晰地界 定了相关概念；在之后的章节中，对数据架构管理、元数据管理、主数据管理、数据 集成、数据质量管理、数据标准化、数据资产化及数据安全与隐私保护进行了详细的 探究。

本书基于作者在大数据治理与安全方面的研究经验和成果编写，对大数据治理进 行了系统、全面的介绍。本书的主要特点如下：

本书的内容十分丰富、翔实。

作者对大数据治理方面的基础和实践应用进行了系统而完整的介绍，同时，基于 大数据治理在典型行业的实践，深入浅出地介绍了当今主流的大数据技术与平台，在 相关章节的内容中加入了大数据治理领域前沿的学术研究成果概述，这样一来，不但 可以让读者掌握大数据治理各个板块的基础知识，同时能对读者之后在该方面的进一 步研究工作起到指导作用。

本书在结构上非常严谨。

本书9个章节之间安排得十分紧凑，章节内容环环相扣、层层递进，具备深度和 广度，对于读者把握全书的主要内容十分有利。

本书图文并茂，图表和公式清晰，案例丰富。

大数据治理： 理论与方法………………

本书以实用性为导向，对大数据治理进行了全方位的解构，精心设计图表、案例 适合作为高等院校“数据科学与大数据技术”专业的核心课程教材。

在本书的编写过程中，哈尔滨工业大学的赵子天、刘畅、陈翔、张于舒晴等同学 和广州大学的周琥晨、彭铭锐、王晟、蔡寅胤、张宏妞、蒋进、王泽世、江钦辉等同 学在资料搜集、整理，图文后期处理等多个方面为本书提供了帮助和支持，在此表示 感谢。最后，作者关于大数据治理方面的研究和本书的写作还得到了国家自然科学基 金项目 (U1866602)、 广东省自然科学基金面上项目“开放域数据投毒攻击检测与防 御关键技术研究”(2021A1515012307)、 广州市科技计划一般项目“知识受限场景下 数据投毒攻防理论及技术研究”(202102021207)、广东省高校创新团队项目 (2020KCXTD007)、 广州市高校创新团队项目(202032854)和工信学术出版基金项目 的支持，在此表示感谢。

由于编写时间有限，书中难免有错误之处，请读者批评指正。

作 者



**Contents**

**第1章** **大数据治理的背景和基本概念** [1](#bookmark2)

1.1 大数据治理的定义 [2](#bookmark3)

1.2 大数据治理的应用 [3](#bookmark4)

1.2.1 大数据治理的任务 [3](#bookmark5)

1.2.2 数据治理与数据管理的区别 [4](#bookmark6)

1.2.3 大数据治理的典型案例 [4](#bookmark7)

1.3 大数据治理的挑战 [7](#bookmark1)

1.3.1 大数据的发展和现状 [7](#bookmark1)

1.3.2 当下面临的挑战 [8](#bookmark8)

1.4 本书的主要内容 [9](#bookmark9)

**第2章** **数据架构管理** [11](#bookmark10)

2.1 数据架构概述 [12](#bookmark11)

2.2 IBM数据架构参考模型 [13](#bookmark12)

2.2.1 逻辑层 [14](#bookmark13)

2.2.2 垂直层 [16](#bookmark14)

2.3 企业数据架构参考模型 [17](#bookmark15)

2.4 CESI 大数据参考架构模型 [18](#bookmark16)

2.5 大数据技术参考架构 [19](#bookmark17)

2.6 数据湖(Data Lake) [20](#bookmark18)

2.7 面向大数据的数据架构实现 [21](#bookmark19)

2.7.1 Hadoop [21](#bookmark20)

2.7.2 Storm [23](#bookmark21)

2.7.3 Spark [24](#bookmark22)

2.7.4 三种架构的比较分析 [25](#bookmark23)

2.8 数据架构设计原则 [25](#bookmark24)

**第3章** **元数据管理** [27](#bookmark25)

3.1 元数据概述 [28](#bookmark26)

大数据治理：理论与方法

3.1.1 定义 [28](#bookmark27)

3.1.2 组织方式 [29](#bookmark28)

3.1.3 作用和意义 [31](#bookmark29)

3.2 业 务 元 数 据 [32](#bookmark30)

3.2.1 意义 [32](#bookmark31)

3.2.2 概念 [33](#bookmark32)

3.2.3 实践要点 [34](#bookmark33)

3.3 技术元数据 [34](#bookmark34)

3.3.1 意义 [34](#bookmark35)

3.3.2 概念 [35](#bookmark36)

3.3.3 实践要点 [36](#bookmark37)

3.4 元 数 据 管 理 [36](#bookmark38)

3.4.1 元数据管理方案 [37](#bookmark39)

3.4.2 元数据标准和规范 [39](#bookmark40)

3.4.3 元数据管理的成熟度 [40](#bookmark41)

**第** **4** **章** **主** **数** **据** **管** **理** [42](#bookmark42)

4.1 主 数 据 概 述 [43](#bookmark43)

4.1.1 主数据的概念 [43](#bookmark44)

4.1.2 主数据管理的类型 [44](#bookmark45)

4.1.3 主数据管理的基本思路 [44](#bookmark46)

4.2 主数据管理系统 [45](#bookmark47)

4.2.1 主数据管理系统的架构设计 [46](#bookmark48)

4.2.2 主数据管理系统的核心功能 [49](#bookmark49)

4.2.3 主数据管理系统的实现风格 [50](#bookmark50)

4.3 主数据管理的成熟度 [52](#bookmark51)

**第** **5** **章** **数** **据** **集** **成** [56](#bookmark52)

5:1 数据集成的基本概念 [57](#bookmark53)

5.1.1 数据集成的定义 [57](#bookmark54)

5.1.2 数据集成的分类 [58](#bookmark55)

5.1.3 数据集成的难点 [59](#bookmark56)

5.2 传统数据集成 [60](#bookmark57)

5.2.1 联邦数据库系统 [61](#bookmark58)

5.2.2 中间件集成 [62](#bookmark59)

5.2.3 数据仓库 [63](#bookmark60)

5.3 传统数据集成的关键技术 [65](#bookmark61)

5.3.1 模式匹配 [65](#bookmark62)

5.3.2 数据映射 [69](#bookmark63)

5.4 跨界数据集成 [70](#bookmark64)

5.4.1 基于阶段的集成 [70](#bookmark65)

**目** **录** **|**

5.4.2 基于特征的集成 [71](#bookmark66)

5.4.3 基于语义的集成 [71](#bookmark67)

5.5 智能化时代的数据集成 [73](#bookmark68)

5.5.1 知识图谱融合 [73](#bookmark69)

5.5.2 联邦学习 [79](#bookmark70)

**第** **6** **章** **数** **据** **质** **量** **管** **理** [83](#bookmark71)

6.1 概 述 [84](#bookmark72)

6.1.1 数据质量的定义 [84](#bookmark73)

6.1.2 数据质量问题 [86](#bookmark74)

6.2 数据质量评估 [88](#bookmark75)

6.2.1 数据一致性评估 [88](#bookmark76)

6.2.2 数据完整性评估 [89](#bookmark77)

6.2.3 数据时效性评估 [91](#bookmark78)

6.2.4 数据精确性评估 [94](#bookmark79)

6.2.5 实体同一性评估 [94](#bookmark80)

6.3 缺 失 值 填 充 [95](#bookmark81)

6.3.1 什么是缺失值 [95](#bookmark82)

6.3.2 缺失值处理方法 [96](#bookmark83)

6.3.3 缺失值处理例析 [98](#bookmark84)

6.4 实体识别与真值发现 [100](#bookmark85)

6.4.1 什么是实体识别 [100](#bookmark86)

6.4.2 基于规则的实体识别方法 [101](#bookmark87)

6.4.3 什么是真值发现 [103](#bookmark88)

6.4.4 真值发现方法 [104](#bookmark89)

6.4.5 真值发现的关键技术 [105](#bookmark90)

6.5 错误检测与修复 [112](#bookmark91)

6.5.1 格式内容清洗 [112](#bookmark92)

6.5.2 逻辑错误清洗 [113](#bookmark93)

6.5.3 非需求数据清洗 [114](#bookmark94)

6.5.4 面向图数据的错误检测技术 [114](#bookmark95)

6.6 面向大数据的数据清洗 [121](#bookmark96)

6.6.1 大数据清洗的探索 [121](#bookmark97)

6.6.2 时间序列错误清洗 [121](#bookmark98)

6.6.3 基于众包的数据清洗 [126](#bookmark99)

6.6.4 基于用户反馈的数据清洗 [128](#bookmark100)

6.6.5 面向半结构化数据、图数据的数据清洗 [133](#bookmark101)

6.7 数据质量标准 [141](#bookmark102)

6.7.1 ISO 8000 [141](#bookmark103)

6.7.2 《信息技术 数据质量评价指标》 [142](#bookmark104)

大数据治理：理论与方法

6.7.3 《工业数据质量 通用技术规范》 [144](#bookmark105)

第 7 章 数 据 标 准 化 [145](#bookmark106)

7.1 数据标准化概述 [146](#bookmark107)

7.1.1 数据标准 [146](#bookmark108)

7.1.2 数据标准管理 [147](#bookmark109)

7.1.3 企业数据标准化 [148](#bookmark110)

7.2 数据标准化例析 [148](#bookmark111)

7.2.1 工业大数据 [148](#bookmark112)

7.2.2 信息技术元数据注册系统 [150](#bookmark113)

**第** **8** **章** **数** **据** **资** **产** **化** [151](#bookmark114)

8.1 数据资产管理概述 [152](#bookmark115)

8.1.1 数据资产 [152](#bookmark116)

8.1.2 数据资产管理 [153](#bookmark117)

8.2 数据资产发现与评估 [154](#bookmark118)

8.2.1 数据资产发现 [154](#bookmark119)

8.2.2 数据资产评估 [155](#bookmark120)

8.3 数据交易与数据定价 [156](#bookmark121)

8.3.1 数据交易 [156](#bookmark122)

8.3.2 数据定价 [159](#bookmark123)

8.4 拓展：大数据拍卖模型 [160](#bookmark124)

8.4.1 问题背景 [160](#bookmark125)

8.4.2 基本拍卖模型 [160](#bookmark126)

8.4.3 扩展的Vickrey 拍卖模型 [161](#bookmark127)

8.4.4 扩展的序贯拍卖模型 [164](#bookmark128)

**第** **9** **章** **数** **据** **安** **全** **与** **隐** **私** **保** **护** [167](#bookmark129)

9.1 概 述 [168](#bookmark130)

9.1.1 基本概念 [168](#bookmark131)

9.1.2 数据安全与数据管理生命周期 [169](#bookmark132)

9.2 数据安全存储、传输与访问 [170](#bookmark133)

9.2.1 加密存储和传输 [170](#bookmark134)

9.2.2 访问控制 [173](#bookmark135)

9.3 数据安全检索与处理 [174](#bookmark136)

9.3.1 安全检索 [174](#bookmark137)

9.3.2 安全处理 [177](#bookmark138)

9.4 隐 私 保 护 [178](#bookmark139)

9.4.1 基本概念 [178](#bookmark140)

9.4.2 数据脱敏 [180](#bookmark141)

9.4.3 k-匿名 [181](#bookmark142)

9.4.4 差分隐私 [183](#bookmark143)

**目** **录** **|**

9.4.5 图数据隐私保护 [184](#bookmark144)

9.5 智能化时代的数据安全 [190](#bookmark145)

9.5.1 推荐系统中的数据投毒攻击 [193](#bookmark146)

9.5.2 众包环境下的数据投毒攻击 [195](#bookmark147)

9.5.3 数据投毒攻击的防御 [198](#bookmark148)



**第1** **章** **大数据治理的**

**背景和基本概念**

当前，大数据在各行各业广泛产生，其价值也得到了许多数据拥有者的重视，用 大数据来优化决策、营销、产品、流程、服务等方面的需求也日益旺盛。然而，在实 际应用中常见的问题是，大数据分析模型建立得很完美，算法设计得也很漂亮，但将 模型和算法应用于真实数据时，却并没有达到预期的效果。究其原因，是因为真实数 据经常来自独立自治的数据源，而这些数据源存在着缺少面向具体应用的顶层设计、 缺少质量保障机制等问题。这些问题催生了大数据治理这一研究领域。大数据治理涉 及组织、行业、国家三个层面，在这三个层面定义、构建一套完整的体系，不仅需要 成熟的模型和算法，还需要完善的法律法规、全面的标准体系等。目前，大数据治理 的必要性已得到较为广泛的认可，也已有不少成功实践，但当前大数据治理的技术研 究仍有待完善。

大数据治理是传统信息治理的延续和扩展，其涉及的内容非常广泛，包括数据架 构管理、元数据管理、主数据管理、数据质量管理及数据标准化和资产化等。所需的 技术支撑需要涵盖大数据管理、存储、质量、共享与开放、安全与隐私保护等多个方 面。大数据治理的目标是确保以正确的方式对数据和信息进行管理，为大数据的有效

**大数据治理：** 理论与方法………

应用保驾护航。可以说，在大数据战略从顶层设计到底层实现的“落地”过程中，治 理是基础，技术是承载，分析是手段，应用是目的。

大数据治理包含很多内容，包括政策层面、管理层面和技术方面层面的内容。本 章将对大数据治理的背景和基本概念做简要介绍，尝试使读者建立对大数据治理的基 本认识。后续章节将会从政策、管理和技术等多个方面对大数据治理相关的概念和方 法加以介绍，并对大数据治理中的主要方面，即数据架构管理、元数据管理、主数据 管理、数据集成、数据质量管理及数据标准化和资产化等进行深入探讨，以期为读者 提供一个比较全面的大数据治理图景。

1 **.1** **大数据治理的定义**

“治理”源于拉丁语“掌舵” 一词。在宏观层面上，从体系框架角度定义，大数 据治理框架是对大数据的管理和利用进行评估、指导和监督的体系框架，它通过制定 战略方针、建立组织架构、明确职责分工等，实现大数据的风险可控、安全合规、绩 效提升和价值创造，并提供不断创新的大数据服务。从信息治理计划和策略角度定义， 大数据治理是广义信息治理计划的一部分，即制定与大数据有关的数据优化、隐私保 护与数据变现政策。从部署与管理角度定义，大数据治理是企业数据可获性、可用性、 完整性和安全性的部署及全面管理。在微观层面上，从策略或程序角度定义，大数据 治理是描述数据该如何在其全生命周期内使用和管理的组织策略或程序。

如图1- 1所示，大数据治理是数据管理框架的核心，指导其他数据管理方法的执 行。大数据治理通过制定正确的政策、操作规程，确保以正确的方式对数据和信息进

行管理。

大数据治理是广义信息治理计划的 一 部分，即制定与大数据有关的数据优化、 隐私保护与数据变现的政策。

具体来说，上述定义包括以下内涵。

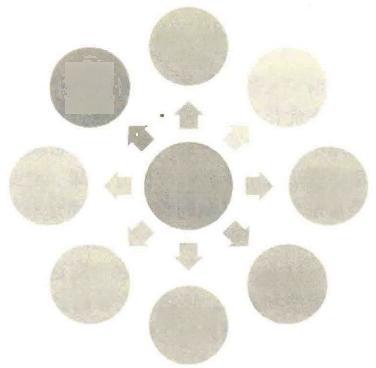
(1)大数据治理是广义信息治理①计

划的 一 部分。信息治理机构必须将大数据 整合到既有的信息治理框架中。

(2) 大数据治理关乎政策制定。这里

的政策是指人们在特定情形下采取的措施。 如大数据治理政策可能申明：“未经顾客知 情并同意，组织不得将顾客的 Facebook 资 料整合到其主数据记录中”。

(3) 大 数据必 须优 化。与企业对实物 资产的优化管理类似，组织必须对大数据 进行优化，包括元数据管理、数据质量管



数据架构

管理

元数据 管理

大数据

治理

数据

标准化

数据质量

管理

数据安 全与隐 私保护

数据 资产化

主数据 管理

数据集成

**图1-1** **数据管理框架**

① 信息治理 (Information Governance) 即领导、指导、控制、提供保障的行为或过程，通过这些行为或过程， 信息被当作贯穿于整个企业的资源得以有效管理，其中包括解决信息冲突问题方面的管理。

**第1章** **|大数据治理的背景和基本概念**

理、信息生命周期管理等。

(4) 大数据必须变现。变现的方式既可以是直接将数据卖给第三方，也可以是利 用数据开发新的服务。

(5)大数据的安全隐私至关重要。在处理社交媒体、地理定位、生物计量学和其 他形式的个人可识别信息 (Personally Identifiable Information,PI) 时，组织必须制定 适当的政策，以防止大数据误用带来的声誉、法律等方面的各种风险。

(6)大数据治理必须对各种冲突进行协调。基于不同目标，大数据往往会带来多

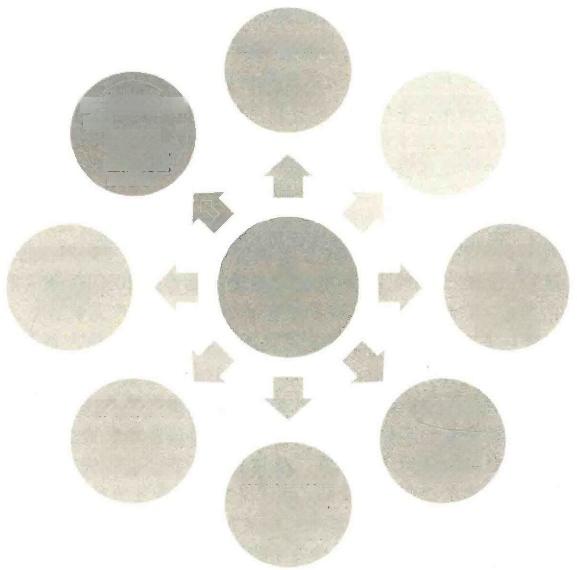
种冲突，如客户隐私与企业利益之间的冲突、计算代价和服务质量之间的冲突等。

**1.2** **大数据治理的应用**

**1.2.1** **大数据治理的任务**

大数据治理是一个系统的、大型的、长期的工程。大数据治理机制是技术与管理 相结合的一套持续改善管理机制，贯穿在数据管理的整个过程中，通常包括组织架构、 政策制度、技术工具、数据标准、流程规范、监督及考核等方面，将其他几个数据管 理职能贯穿、协同在一起，使得数据成为一个有机整体，而不是各自为政。

我们可以将图1- 1扩充为图1-2的形式，进一步细化大数据治理的任务。



数 据 架 构 管 理

架构设计、

实现、修正

安全存储、

传输、访问、

检索、处理，

隐私保护

**主** **数** **据** **管** **理**

架构设计、 管理模式、 使用模式、

实现风格等

**数** **据** **标** **准** **化**

国际标准、

国家标准、

行业标准等

**数** **据** **质** **量** **管** **理**

缺失值填充、

实体识别、

错误检测与

修复等

元 数 据 管 理

业务元数据、 技术元数据

**数** **据** **集** **成**

传统数据集 成、跨界数 据集成等

**数** **据** **资** **产** **化**

发现与评估、 交易与定价

数 据 安 全 与 隐 私 保护

大数据 治理

图1-2 大数据治理的任务

大数据治理： 理论与方法……………………

**数据治理与数据管理的区别**

数据管理和数据治理有很多地方是互相重叠的，它们都围绕数据这个领域展开， 因此这两个术语经常被混为一谈。此外，每当人们提起数据管理和数据治理的时候， 还有一对类似的术语叫信息管理和信息治理，更混淆了人们对它们的理解。因此，我 们在这里对数据治理和数据管理做一个区分。

数据治理 (Data Governance,DG): 对数据资产管理 (The Management of Data As- sets) 行使权力和控制的活动集合(规划、监控和执行)。数据治理职能指导其他数据 管理职能如何执行。数据治理的目标是制定正确的政策、操作规程，确保以正确的方 式对数据和信息进行管理。

数据管理 (Data Management,DM): 规划、控制和提供数据与信息资产的业务职 能，发挥数据和信息资产的价值。

当然，不可否认的一点是，数据治理是数据管理的一部分。这个概念目前已经得 到了业界的广泛认同。数据管理包含多个不同的子领域，数据治理是其最重要的子领 域之一。

**1.2.3** **大数据治理的典型案例**

下面介绍大数据治理的一些典型案例，说明其应用和价值。

**案例1** **通过大数据治理，提高运营效率，降低运营成本**

公路、铁路和航线上的交通堵塞，浪费时间、增加污染，造成极高的社会成本。 2013年，麦肯锡全球研究所公布的一份研究报告表明，通过改良需求管理和维护从而 增强对现有基础设施的利用，全球每年可节省4000亿美元。针对此类情况，以大数据 分析和挖掘为代表的数字化技术能够大展身手。信息收集及战略运用能够改善预测准 确率，进而帮助提高交通基础设施的可靠性，并增加其效率和利用率。事实上，已经 有一些相关案例。

**案例研究1** **巴西利用** **GPS 数据优化可用空域使用率系统**

在巴西，航空流量在过去的10年中增长迅速，预计到2030年每年乘客数量将翻 番，达到3.1亿多人次。航空拥堵问题也顺理成章地受到了广泛关注。为了应对该问 题，巴西开始引进一套利用 GPS 数据优化可用空域使用率的系统，用于缩短飞机之间 的距离和缩短航线。

常规的方法是让准备降落的飞机在空中排队。使用新系统后，每架飞机将拥有自己 的航道。这听起来可能简单，但是需要大量的数据及对这些数据快速而复杂的分析作为 支撑。系统需要处理每架飞机的飞行距离、速度和载荷能力，以使得航线最短。飞机可 以在距离机场近得多的地方“曲线”着陆，而不是在即将着陆的时候在空中排队。

巴西利亚国际机场首次采用了这套系统，使每架飞机每次着陆节省了7分半的时 间和约300升燃油，同时每架飞机平均少飞了约40千米。巴西计划在全国的10家最繁 忙的机场部署这套系统。据初步估计，在北美机场部署这套系统可以使机场的运力提 升16%～59%,具体视机场实际条件将有所不同。

第1**章|大数据治理的背景和基本概念**

**案** **例** **2** **通过大数据生命周期治理，降低IT** **成** **本**

大数据的归档需要一些代价，但大数据治理可以降低IT 成本。

案例研究2 欧洲某公用服务公司实施大数据生命周期治理，节约运营成本

表1-1归纳了欧洲某公用服务公司在部署电气智能仪表时，包括大数据归档和压缩 在内的总体运营成本节约情况。其他额外收益还包括客户服务、应用绩效、雇员生产 率和系统正常运营时间等方面的改进，但这些收益并未体现在下表中。

**表1-** **1** **针对电气智能仪表的大数据治理所带来的运营成本节约**

|  |  |  |
| --- | --- | --- |
| A | 单个智能电表每年的存储容量要求 | 4.5MB |
| B | 单个天然气智能仪表每年的存储容量要求 | 4.0MB |
| C. | 预计的智能电表数量 | 2700万块 |
| D | 预计的天然气智能仪表数量 | 2000万块 |
| E | 用于备份、测试和灾难恢复的生产系统的常规克隆数 | 4 |
| F | 智能电表所需的总存储量(A\*C\*E) | 486TB |
| G | 天然气智能仪表所需的总存储量(B\*D\*E) | 320TB |
| H | 每年每TB数据的平均运营成本 | 25000美元 |
| 1 | 智能电表的年存储成本(F\*H) | 1215万美元 |
| J | 天然气智能仪表的年存储成本(G\*H) | 800万美元 |
| K | 实施大数据生命周期治理带来的节约 | 60% |
| 1 | 年度节约资金[(I+J)\*K] | 1209万美元 |

电气智能仪表产生的数据量，因记录频率而有较大差异。典型的频率是每15分钟 读数一次，每笔交易至少占用128字节。报告故障、极端电压报警和配置消息等额外 事务，会增加电气智能仪表的数据量。 一些专家估计，为支持某些智能电网功能，每 台电气智能仪表每年产生的数据量可能高达1.7GB。本案例使用了一个保守数字，即 每台电气智能仪表每年产生的数据量为4.5MB。

**案** **例** **3** **通过大数据治理，提高医疗效率**

生命科学与信息科学是21世纪两个重要的前沿科学。医疗健康已成为大数据应用 的重要领域。医疗健康大数据的应用不仅可为人类带来更好的医疗健康服务，同时利 用大数据方法可不断发现新的知识内容，促进医学技术的进步。医疗健康大数据的类 型复杂多样，对大数据治理提出了更高要求。

**案例研究3** **早产儿病情诊断**

安大略理工大学的卡罗琳·麦格雷戈 (Carolyn McGregor) 博士及其研究团队与 IBM 一起与很多医院合作，用一个软件来监测处理即时的病人信息，然后把它用于早 产儿的病情诊断。他们使用了个人体征数据，包括心率、呼吸、体温、血压和血氧含

大数据治理： 理论与方法……………………

量，和一些其他的数据(如孕妇产检数据、电子病历、遗传数据等)。

软件会监控16类数据，这些数据每秒可以达到1260个数据点之多。在明显感染 症状出现的24小时之前，软件就能监测到早产儿细微的身体变化所发出的感染信号， 尽早预测、控制早产儿的病情，从而提高早产儿的存活率。

早产儿的稳定未必是病情好转的标志，如果可以从海量的数据中找出隐含的相关 性从而提早知道病情，那么医生就能够提早治疗，也能更早地知道某种疗法是否有效， 这一切都有利于病人的康复。

**案例4** **大数据治理中的大数据质量问题**

大数据给数据质量管理带来了新的挑战。如果能得到恰当的治理和管理，那么内 部数据的质量就可被测量和控制。尽管商业服务提供商 (CSP) 已对外部数据实现了 有限控制，但必须评估其价值和质量。基于对外部数据的理解和对融合数据使用情况 的监测，可以小心翼翼地应对蜂拥而至的内外部数据。

**案例研究4** **Twitter 数据的代表性如何?**

某CSP 在全国范围内推出一项新产品，其采集与产品销售、故障通知单、网络使 用和 Twitter 有关的数据。许多推文 (Tweet) 一 致表明，Twitter 用户对产品抱负面情 绪。营销团队担心，这些数据是异常的，因为产品销售形势喜人，故障通知单也并未 出现显著增长。

为什么 Twitter数据如此不正常呢?进一步的分析显示， 一方面，年龄较大的用户 对产品相对更满意，这些用户一般通过调查问卷和故障通知单等传统方式反馈意见。 另一方面，对较为年轻的用户来说，该产品的表现却差强人意。这些年轻用户不太使 用传统的反馈方式，而更常在 Twitter上以负面方式讨论产品。

由于社交媒体信息大多数是自我报告式的，因此，此类信息在某种程度上更倾向 于有偏抽样。在数据融合时， CSP 使用了一个流程来解决大数据质量问题。由于社交 媒体的数据代表用户目标总体 (Population),所以需要分析Twitter 数据的总体置信度。

**案例5** **大数据治理与隐私安全**

在大数据高速发展的时代，隐私似乎正在成为一个敏感的话题，正如网上一个很 著名的说法——大数据时代，人人都在“裸奔”。但许多国家和地区已经逐渐开始对大 数据治理中隐私问题的研究，并开始提供国家层面的保障和法律层面的措施。

**案例研究5** **加利福尼亚州的智能电表隐私**

加利福尼亚州公用事业委员会采取措施保护公用事业客户电力使用数据的隐私和 安全。2012年3月，圣地亚哥天然气与电力公司 (SDG&E) 与加拿大安大略省的信息 和隐私专员，发布了一份题为“从设计着手保护隐私： SDG&E 智能定价计划的最佳实 践 (Applying Privacy by Design:Best Practices to SDG&E's)”的白皮书，详细地介绍 如何从设计着手，保护用户电量使用数据隐私的实践。

除成立监督隐私合规性的工作组外，SDG&E 还设置了首席客户隐私官。首席客户 隐私官也是分管客户服务的副总裁和高管团队的成员，负责完成隐私影响评估。

6

**第1章|大数据治理的背景和基本概念**

SDG&E 智能定价计划的关键点是“隐私静默”。换言之， SDG&E 的政策是，客户

只需选择加入而不是退订“默认隐私”设置。

**案例6** **开放数据服务，探索大数据战略治理**

在如今的大数据时代,随着信息技术的不断发展，数据资源作为信息的基础和来 源，其重要性也逐渐受到重视。而政府作为公共服务的提供者，在履行相应职责的过 程中获得、收集和保存了大量与人民生活息息相关的数据信息，这些数据的公开对于 推动大数据战略具有重大的意义。

**案例研究6** **上海政府数据资源向社会开放**

2012年6月，上海正式启动了政府数据资源向社会开放试点工作，建立了国内首 个政府数据开放网站 “上海政府数据服务网”(一期),为公众提供了“一站式” 的数据检索、浏览和下载服务。截至2017年10月19日的统计数据表明，上海政府数 据服务网的开放数据项总量已达54145条、开放数据资源已达1422个、开放数据部门 已有41个，数据内容涵盖了经济建设、资源环境、教育科技、道路交通、社会发展、 公共安全、文化休闲、卫生健康、民生服务、机构团体、城市建设、信用服务共12类 社会领域，几乎覆盖到了公民生活的方方面面。

1**.3** **大数据治理的挑战**

**1.3.1** **大数据的发展和现状**

2018年4月21 日，在苏州举行的第十三届中国电子信息技术年会上，梅宏院士应 邀作题为“大数据治理体系建设若干思考”的大会报告。在报告中，梅宏院士回顾了 大数据技术与产业生态的发展历程，指出大数据治理成为当前发展的热点方向；剖析 了大数据治理相关研究与实践的现状，分析了尚存在的问题，并介绍了一种多层次、 多维度的大数据治理体系框架。下面我们借梅宏院士的发言来回顾一下大数据的发展 和现状。

全球大数据产业的生态在2012年基本形成雏形，2013年达到大数据宣传的顶峰。 回顾历年的 Gartner曲线图，在2012年、2013年基本达到一个顶点，2014年基本上趋 于理性、成型，2015年、2016年的宣传热点不在曲线中出现。我想，这在某种意义上 表明大数据进入理性、健康的发展期。

2012年，大数据基础设施是研发重点。基础设施相关的业务和企业发展是比较快 的，生态系统正在逐步地形成，同时也在快速的演化过程中。2014年，大数据分析就 成了生态系统里面最火热的部分。分析成为热点，有大批的公司成立并快速发展，风 险投资大量涌入。应该说这个时候，生态系统还处于初级阶段。2016年，大数据应用 成为发展的重点，面向行业和领域的应用型企业发展迅猛，生态系统逐步形成，向更 为成熟的程度发展。2017年又出现一个新的重点，是数据治理、数据安全，这里面的 数据治理还是相对狭义的治理，数据治理、数据安全首次被纳入基础设施的范畴。数 据资源、数据安全成为重要的基础设施，整个生态系统发展全速推进，并将研发资源

大数据治理： 理论与方法 …

当成重要的生态系统的环节出现。

2015年，我们国家的大数据生态状况，在三个技术层次上主要被5～8家国际厂 商主导，国内厂商多出现在数据使用层当中的数据可视化领域，极少数涉及数据的升 级和管理。国内的大数据生态系统，也在向着细分的大数据服务方面发展，可以看到， 基础技术和系统方面缺乏原创，分析方法与算法应用牵引不足，研究实用性和易用性 偏弱是主要问题，虽然互联网大数据的应用水平和效果接近国际先进水平，但是其他 行业和企业的大数据应用水平和效果则明显落后。

2016年，我们可以看到大数据领域在国际上仍然占主导，基于物联网的数据采集 与数据管理初现，各内容类互联服务竞相成为数据源，垂直化与行业化应用发展迅猛， 应该说，我们的水平开始提升，开始积累大量的原始数据，数据的收集和管理已经成 为我们国家大数据产业发展的关键点。习近平总书记讲数据经济，数据是关键的资源， 所以大家对数据的重视度日益增高。

从国家总量来看，中国大数据市场规模2015年达到115.9亿元，同比增长38%; 2016年是168亿元，同比增长45%。预计2022年将达到735亿元。

前面我们主要从产业角度回顾了大数据的发展。应该可以看到，产业地图展现和 技术研发走向有这样一个趋势和阶段，从基础技术和系统，以及大数据基础设施的构 建、分析方法和算法，到领域行业应用。现在大数据资源管理安全成为一个新的热点， 由于大数据作为战略资源的地位越来越重要，大数据的管理、安全隐私、开放共享成 为当前的重点，因此建立大数据的治理体系，成为当前一项紧迫的任务。

**1.3.2** **当下面临的挑战**

大数据治理体系是涉及国家实施大数据战略的重要基础和保障，也是发挥大数据 作用、做大做强大数据产业的重要因素。大数据治理体系建设，已经成为大数据相关 管理规则、相关技术和产品研发的重点。因而需要分层次多维度推进大数据治理体系 的建设，给予大数据治理体系足够的重视。

目前，大数据治理的理论与技术仍处于早期发展阶段，这可以从以下几个方面来 理解。

(1)大数据治理体系至少涉及组织、行业、国家这三个层面。

(2)大数据治理体系需要完善的法律法规、全面的标准体系支撑，这涉及是否需 要围绕“数据”为主体来制定制度法规和标准规范。

(3)大数据治理的重要性已得到较为广泛的认同，也已有不少成功实践，但我们 可以看到，现在的实践尚不足以支撑标准形成。

(4)大数据治理体系技术支撑需要涵盖大数据管理、存储、质量、共享与开放、 安全与隐私保护等多个方面，当前相应的技术研究关联性和系统性还存在不足，将技 术整体整合起来还有不少问题。

(5)大数据资产地位得到广泛认同，但是如果没有有效的管理和应用，那么这些 数据反而会成为负担，这对任何层面都是一样的。

(6)大数据管理的相关方法与技术层面已有不少成熟产品与技术，但还缺少完善 的多层级管理体制和高效管理机制。单有技术，没有管理体制和管理机制也完不成大

8

第1章 |大数据治理的背景和基本概念

数据管理。

(7)大数据共享与开放已经成大数据成功应用的关键，需要将技术和标准有机结 合，才能够建立在不同层级上的良好的大数据的共享与开放环境。

我们可以基于大数据的特点，总结出以下大数据面临的挑战和治理的要点

政策/流程 大数据处理流程复杂，每个过程的问题都有可能影响大数据的应用， 因而大数据治理应覆盖大数据的获取、处理、存储、安全等各个环节。

数据管理专员制度 大数据成为企业的重要战略资源，因而需要在企业中为大数 据治理设置数据管理专员。

数据生命周期管理 大数据的有效使用需要对数据的全生命周期进行管理，包括 存储、保留、归档、处置等步骤，在数据生命周期管理的过程中需要有效平衡时间与 存储空间。

数据架构管理 数据作为重要的资源和服务，必须要配以精心设计的数据架构， 数据架构作为整个大数据治理的骨架，在保证治理任务顺利实施中扮演着重要的角色。

元数据管理 大数据需要与内容相关的元数据及传统数据定义标准均保持一致； 术语字典应包含大数据的术语；需要为非结构化数据提供分类、语义支持，Hadoop、 NoSQL 数据库等面向大数据技术的元数据需要纳入元数据存储库管理。

主数据管理 主数据是所有数据中最具价值的且被多个部门反复使用的数据，它 们是公司的基本业务数据，良好的主数据管理可以为企业或组织节省大量数据整理的 时间，并且能够提高数据质量。

数据集成大数据时代，很难保证企业或组织只需要处理单一来源的数据，大数 据多源异构的特点使得其需要进行有效集成才能够得以协同工作，而数据集成需要统 一元数据标准，对大数据做统一定义。

数据质量 大数据规模大、变化快、多源异构等特点导致其有极大可能存在数据 质量问题，因此应识别对业务有关键影响的数据元素，检查和保证数据质量。

数据标准化 在大数据时代，数据的交换、传递、共享非常重要，数据标准化能 够使各个应用系统对客观实体的分类和描述手段一致，或者提供相应的转换接口。为 了能够更好地建立良好的大数据共享与开放环境，数据标准化势在必行。

数据资产化 数据作为一种新型资产，如果不能被良好管理，也可能变成一种 “负债”。如何管理数据资产，“盘活”数据以充分释放其附加价值变得非常重要。

数据安全和隐私保护 作为以互联网为依托的大数据，它将面临网络带来的各种 安全风险，这些风险威胁到大数据的安全，并可能给用户造成利益损失。在科学研究、 产品开发、数据公开的过程中，算法需要收集、使用用户数据，因此， 一些敏感数据 就不可避免地面临安全风险。如何保护数据安全和用户隐私是顺利实施大数据治理的 最基础的问题之一。

**1. 4 本书的主要内容**

本书的后续章节，我们将对数据架构管理、元数据管理、主数据管理、数据集成、 数据质量管理、数据标准化、数据资产化、数据安全与隐私保护这几个方面进行深入

大数据治理： 理论与方法

的讨论。

在第2章，我们将讨论数据架构管理，此章将分析几种经典的数据架构，并给出 数据架构设计的一些基本原则。

在第3章，我们将讨论元数据管理，此章将给出元数据的定义，并讨论如何管理 业务元数据、技术元数据。

在第4章，我们将讨论主数据管理，此章给出了主数据的定义，并介绍了主数据 管理系统的架构、功能、使用和实现，同时，也介绍了主数据管理成熟度的模型。

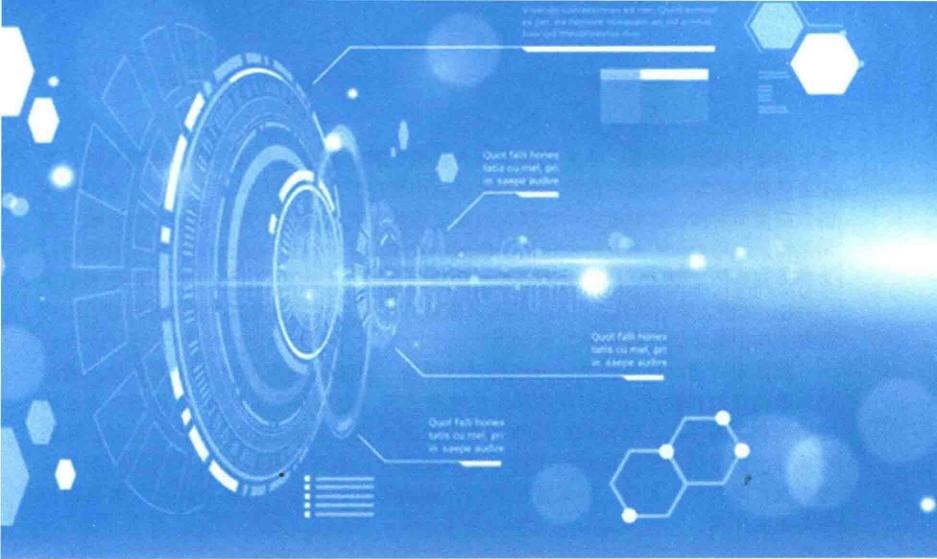
在第5章，我们将讨论数据集成，此章介绍了传统数据集成和跨界数据集成的一 些主要技术。

在第6章，我们将讨论数据质量管理，此章给出了数据质量的定义，并介绍了缺 失值填充、实体识别与真值发现、错误检测与修复等方面的技术。

在第7章，我们将讨论数据标准化，此章介绍了数据标准化的基本概念和实际 案例。

在第8章，我们将讨论数据资产化，此章给出了数据资产化的基本概念，并讨论 了数据资产发现、评估、交易、定价、拍卖等问题。

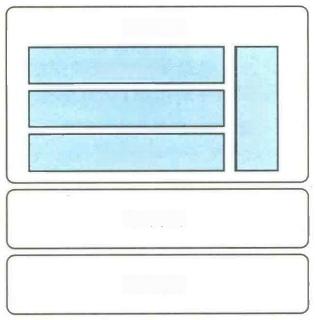
在第9章，我们将讨论数据安全与隐私保护，此章给出了数据安全与隐私保护的 相关概念，并概述了数据安全存储、传输、访问、检索、处理以及隐私保护的常见 技术。



数 据 治 理

**第** **2** **章** **数据架构管理**

我们通常所说的“数据架构”与“应用架构”和“技术架构”并列，三者共同组 成IT 架构(如图2-1所示)。IT 架构由业务架构驱动，从业务架构出发分析业务流程、 定义数据架构，流程和数据结合定义应用架构，根据数据架构和应用架构设计技术 架构。

**数据架构**

**数据标准**

**数据模型**

**数据分布**

应用架构

技术架构

图2-1 IT架构

值得注意的是，业务架构和应用架构均包含数据架构的内容，业务架构中数据架 构即数据概念模型，分析重点是数据领域、主数据和核心业务对象。业务运营的两条

**大数据治理：** 理论与方法……

重要线索是流程和数据，业务流程离不开数据流转，业务运营状况通过数据反映，基 于业务架构的端到端流程建模过程中会衍生出对应的业务数据对象，需要与数据架构 的数据模型对接。流程模型和数据模型对接后落实到应用(系统)层面，就形成了应 用架构。应用架构将业务对象转换为数据对象或具体的数据库表对象，数据模型进一 步转换到具体应用(系统)的逻辑模型和物理模型，在此基础上分析数据对象和应用 (系统)功能之间的创建、引用、修改或删除 CRUD关系，以明确功能边界划分，对应 数据架构中最终的数据分布。

数据架构通过数据分类、分层部署等手段，从多重视角将数据合理布局。通过 整体架构管控和设计，支持业务操作和管理分析，满足业务发展及IT 转型中涉及的 与数据相关的需求，架构的扩展性和适应性能够提升数据分析的及时性、灵活性和 准确性。数据治理是数据架构中的重要组成部分，本章将从数据架构概述和参考模 型讲起，并介绍主流的三种大数据架构，以助于读者理解大数据治理在大数据架构 中的重要作用。

**2.1** **数据架构概述**

数据架构 (Data Architecture) 是一套整体组件规范，其提供了一种正式的方法， 作用在于：

(1)管理和定义收集的数据类型；

(2)设计、定义和指导数据的使用、存储、管理和集成方式，以便服务于相应的 组织机构、数据库系统、IT 服务、业务程序等。

数据架构是一个广义术语，它用于定义数据需求，指导数据资产的整合和控制， 使数据投资和业务战略相匹配。数据架构包括正式的数据命名、全面的数据定义、 有效的数据结构、精确的数据完整性规则和健全的数据文档等。数据架构管理是定 义和维护如下规范的过程：提供标准的、通用的业务术语/辞典；表达战略性的数据 需求；为满足上述需求，概述高层次的整合设计；使企业战略和相关业务架构相 一致。

根据《DAMA 国际数据管理知识体系指南》,数据架构的组成如表2-1所示。

**表2-1** **数据架构的组成**

|  |  |
| --- | --- |
| **名** **称** | **含** **义** |
| 数据架构结果 | 各个级别的模型、定义和数据流，通常称为数据架构组件 |
| 数据架构活动 | 表单、部署和实现数据架构意图 |
| 数据架构行为 | 影响企业数据架构的各种角色之间的协作、思维模式和技能 |

企业中与数据相关的应用需要企业数据架构来支撑。企业数据架构是企业架构的 一部分，与其他业务、技术架构相整合，其是一套规范和文档的集合，由表2-2所示的 三个不同层次组成。

第2章 |数据架构管理

**表2-2** **三个层次**

|  |  |
| --- | --- |
| **层** **次** | **含** **义** |
| 概念/业务模型 | 包括所有数据实体，并提供概念或语义数据模型 |
| 逻辑/系统模型 | 定义数据实体的链接方式并提供逻辑数据模型 |
| 物理/技术模型 | 为特定过程和功能提供数据机制，指导如何在底层技术基础架构上实现实 际数据架构 |

企业数据架构主要包括三类规范，如表2-3所示。

**表2-3** **三类规范**

|  |  |
| --- | --- |
| **规** **范** | **含** **义** |
| 企业数据模型 | 企业数据架构的核心 |
| 信息的价值链分析 | 使数据与业务流程及其他企业架构的组件相一致 |
| 相关数据交付架构 | 包括数据库架构、数据整合架构、数据仓库/商务智能架构、文档和内容架 构，以及元数据架构 |

除了上述定义，还有不同的组织和研究人员提出了一些其他的数据架构定义， 包括：

“通用的词汇表示并集成需求，确保数据资产在系统中的存储、治理、管理和使 用，以支持组织战略。”——Peter Aiken

“一组规则、策略和模型，用于确定收集哪些数据，以及如何在数据库系统中使 用，处理和存储数据。”——Keith Foote

“有效地使用数据并建立在业务需求的基础之上。”——Sven Blumberg,et.al.

“描述如何收集、存储、转换、分发和使用数据。它包括管理结构化格式的规则， 例如数据库和文件系统，以及将数据与使用它的业务流程连接起来的系统。” Dal- leMule and Davenport

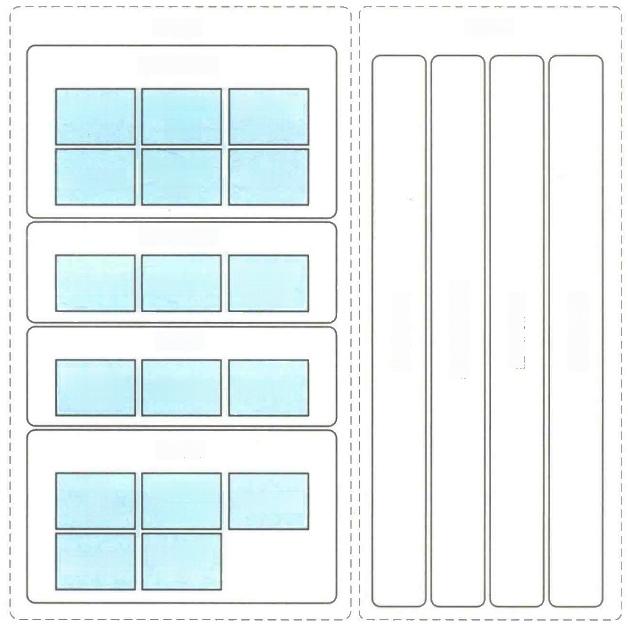
“模型、策略、规则或标准来管理收集的数据，以及如何存储、治理和使用数据库 系统或组织中的数据”——Business Dictionary

2 **IBM** **数据架构参考模型**

我们可以将数据架构简单分解为数据分布、数据模型、数据标准和数据治理。 数据架构为数据资产的管理和应用奠定基础，支撑数据的存储、访问、整合和分 析，既包含相对静态的部分，如元数据、业务对象数据模型、主数据、共享数据 等，也包含相对动态的部分，如数据流转、ETL(Extract-Transform-Load) 、 整合、 访问应用和数据全生命周期管控治理等。

本节将对 IBM 数据架构模型做一个介绍，其模型比较简洁易懂，可以为读者提供 一个全局的概念和理解。在此模型中，数据架构由一个个组件构成，这些组件划分为 逻辑层和垂直层，如图2-2所示。

大数据治理： 理论与方法 ·



逻辑层

数据来源层

数据管理

系统

数据发布

程序

整理存储层

数据整理

分析层

分析引擎

使用层

业务流程

管理流程

可视化和

发现

系 统 管 理

大 数 据 治 理

事务 拦截器

报告引擎

企业遗留

系统

智能设备

数据存储 系统

分析层实

体识别

分布式数

据存储

服 务 质 量

信 息 集 成

模型管理

数据获取

实时监视

垂直层

其他

图2-2 数据架构参考模型

**2.2.1** **逻辑层**

逻辑层提供组织执行特定功能组件的方法。被称为逻辑层，就意味着每层的功能 没有必要在独立的机器或进程上运行。逻辑层通常包括数据来源层、整理存储层、分 析层、使用层这4个主要层次。

**1.数据来源层**

本层包含所有必要的数据源，负责提供解决业务问题所需的数据基础。数据可以 高度异构的，其格式可以是结构化、半结构化和非结构化的；数据的获取速率和传输 速率可以不同；收集方式可以是实时或者批量式的；数据源可能位于企业内部，也可 能位于企业外部；数据的访问权限也可以不同。

数据的主要来源可以大致分为以下几种。

(1)企业遗留系统——企业应用程序，执行业务需要的分析并获取需要的结果 包括客户关系管理系统、结算操作、大型机应用程序、企业资源规划、Web 应用程序 开发等。

(2)数据管理系统——存储逻辑数据、流程、策略和各种其他类型的文档，如电 子表格或者 Word 文档。如果可以，这些文档会被转换为可用于分析的结构化数据。

**第2章** |数据架构管理

(3)数据存储系统——数据存储系统包含企业数据仓库、操作数据库和事务数据库。 通过此数据源获取到的通常是结构化数据，可以直接使用或通过简单地转换来满足需求。

(4)智能设备 如智能手机、智能仪表和智能医疗设备等，能够捕获、处理和 传输使用最广泛的协议和格式的信息。这些设备可用于执行各种类型的分析。它们中 的绝大多数都可以在本地执行实时分析，对其传输的数据也可进行批量分析。

(5)数据发布程序——这些程序本身拥有数据或有能力获取数据，并以复杂的格 式和特定的频率通过特定的过滤器发布数据。它们每天都会产生海量的数据，这些数 据具有不同的格式、以不同的速度生成而且通过各种渠道被获取。

(6)其他——还有许多数据来自其自动化的发布者，包括地图、地区详细信息、 位置详细信息、矿井详细信息等地理信息，社交媒体、电子邮件、博客、在线论坛等 个人信息，温度、湿度、光照、加速度等传感器数据等。

**2.** **整理存储层**

因为传入的数据可能具有不同的特征，所以整理存储层中的组件必须能够以各种 频率、格式、大小在各种通信渠道上读取数据，这一层的主要组件包括：

(1)数据获取 从各种数据源获取数据、存储数据，或将其发送给负责数据整 理的组件。这类组件必须足够智能，要能够判断是否和在何处存储所获取的数据，数 据在存储前是否需要改动，以及数据是否能够直接发送给业务分析层。

(2)数据整理 将数据修改为所需格式以实现分析用途。这类组件既可能使用 简单的转换逻辑，也可能使用复杂的统计算法，由分析引擎负责确定所需的数据格式。 这些组件面临的主要挑战是如何处理非结构化的数据，如图像、音频、视频或其他二 进制格式数据。

(3)分布式数据存储——存储来自各数据源的数据。通常，这类组件会提供多个 数据存储选项，如分布式文件存储 (DFS) 、 云、结构化数据存储、NoSQL等。

**3.** **分析层**

分析层读取整理存储层处理之后的数据。设计分析层需要事先认真规划，以便完 成分析层的主要任务，包括从数据中获取业务洞察、生成数据、分析结果、确定所需 实体、定位可提供所需实体的数据源、理解数据分析所需的算法和工具等。这些任务 通常由下述组件共同完成：

(1)分析层实体识别——负责识别和补全上下文实体。这是一个复杂的任务，性 能要求很高。这类组件向下连接了数据整理组件，以获取特定格式的数据；向上则连 接了分析引擎，向其提供所需的上下文实体以执行分析。

(2)分析引擎 使用其他组件来处理和执行分析。具体来讲，所谓的“其他组 件”包括实体识别、模型管理和分析算法。分析引擎通常支持不同的、可并行处理的 各种工作流、算法和工具。

(3)模型管理——负责维护、验证和检验各种数据分析模型，并通过持续训练来 提高模型准确性。如果模型已经较为成熟，那么模型管理组件会将它们推荐给实体识 别或分析引擎使用。

**4.使用层**

这一层负责尽可能充分地使用分析层的输出结果，该结果由企业或组织内部的各

大数据治理： 理论与方法…………………

用户和外部的若干实体(如客户、供应商、合作伙伴等)使用。

对于外部实体来说，分析层的输出结果可以用来提供产品营销信息。例如，基于 所获取的业务洞察，公司可以使用客户的偏好数据和感知到的位置信息，在客户经过 店铺时向他们提供个性化的营销信息。同时，也可以基于业务洞察检测欺诈，实时拦 截异常交易，并将相关数据提交给企业基于已有数据训练好的模型。在欺诈发生时， 提醒客户采取防范措施。

对于内部用户来说，分析层的输出结果提供了理解、寻找和导航企业内外信息的 能力。对于内部使用者，为业务用户构建报告和报表使得利益相关者更好地制定决策 和战略；从数据中生成实时业务警告；监视关键绩效指标。

(1)事务拦截器——实时拦截高容量事务，将它们转换为一种容易被分析层理解 的实时格式，以便执行实时分析。事务拦截器应能够集成并处理来自各种来源(如传 感器、智能仪表、麦克风、摄像头、GPS 设备、ATM 和图像扫描仪等)的数据；可以 使用各种类型的适配器和API 来连接到数据源，也可以使用各种加速器来简化开发， 如实时优化和流分析、视频分析、领域相关的加速器、社交媒体分析以及情感分析。

(2)业务流程管理流程——来自分析层的业务洞察可供业务流程执行语言 (BPEL) 流程、API 或其他业务流程使用，通过将上下游应用程序、人员和流程的功能 自动化，进一步获取业务价值。

(3)实时监视——使用分析结果来生成实时警告。将警告发送给感兴趣的使用者 和设备，如智能电话和平板电脑。使用分析组件的输出结果，定义并监视关键指标， 以便确定操作有效性。实时数据可从各种来源以仪表盘的形式向业务用户公开，以便 监测系统的健康程度，或度量营销活动的有效性。

(4)报告引擎——生成与传统商业智能报告类似的报告。用户可基于分析层的输 出结果，创建临时报告、定制报告或自定义查询和分析。

(5)可视化和发现——给出直观易理解的数据展示，并引导用户从数据中得出想 要的结果。这些组件需要组合不同内容和格式的数据，来给出综合的可视化展示。同 时，这项能力也可以使得企业或组织能够将传统内容(包含在企业内容管理系统和数 据仓库中),与新的社交内容(如博客文章)组合到单个用户界面中，从而引导用户得 到更多的发现。

2.2.2 垂直层

影响逻辑层(数据来源层、整理存储层、分析层和使用层)的所有组件的各个方 面都包含在垂直层中，包括信息集成、大数据治理、系统管理、服务质量。下面我们 对这些垂直层的重要内容加以简要介绍。

**1.** **信息集成**

大数据应用程序从各种数据源获取数据，并存储在 HDFS、NoSQL 和 MongoDB 等 数据存储系统中。这个垂直层可供各种组件使用(如数据获取、数据整理、模型管理 和事务拦截器),并负责连接各种数据源。信息集成负责整合具有不同特征(例如协议 和连接性)的数据源的信息，因此需要高质量的连接器与适配器来连接和适配不同的 数据源。各种组件还可以使用信息集成的功能在大数据应用程序中存储信息、检索信

**第2章** |数据架构管理

息，以便处理这些信息。

**2.** **大数据治理**

大数据治理涉及定义指南来帮助企业制定相关数据的正确决策。大数据治理有助 于处理企业内外部的复杂数据。在将数据传入企业进行处理、存储、分析、清除、归 档时，需要强有力的指南和流程来监视、构建、存储与保护数据。具体包括管理各种 格式的数据，训练和维护统计模型，设置外部数据存储、使用的策略和机制，定义数 据归档和清除策略，定义跨系统复制数据的方法，配置数据加密策略。

**3.** **服务质量**

这一层定义数据质量、安全性策略、数据获取策略。

数据质量，包括识别必要数据，保证数据时效性，验证数据准确性，验证数据一 致性，验证数据真实性，定义通用业务语言，验证业务和应用场景相关的特性等。

安全性策略，包括数据可用性及关键性识别，数据访问约束，数据共享和发布机 制，数据存储和保留策略，数据源约束，社交媒体使用条款等。

数据获取策略，包括数据采集量，数据采集频率，数据采集的周期(按需、连续 或离线),数据过滤器等。

**4.** **系统管理**

系统管理对大数据至关重要，因为它涉及跨企业集群和边界的许多系统。包括： 管理系统日志、虚拟机、应用程序和其他设备；关联各种日志，帮助调查和监视具体 情形；监视实时警告和通知；使用实时仪表板显示各种参数；引用有关系统的报告和 详细分析；设定和遵守服务水平协议；管理存储和容量；归档和管理归档检索；执行 系统恢复、集群管理和网络管理；执行策略管理。

2.3 企业数据架构参考模型

企业数据架构参考模型主要包括数据架构、流程架构、业务架构、应用架构、技 术架构及价值链分析等。

(1)数据架构包括企业数据架构和其他数据架构。其中企业数据架构包括主题域、 实体层级、概念视图、企业逻辑视图、企业数据字典、业务术语、实体生命周期/状 态、参考数据值、数据质量规则；其他数据架构包括数据交付架构、数据仓库架构、 数据集成架构、内容管理架构、元数据架构、数据模型标准、系统开发生命周期 (SDLC) 模板等。

(2)流程架构包括职能分解、流程工作流、信息产品、事件和业务周期、程序规则等。

(3)业务架构包括目标和战略、组织架构、角色和职责、地点位置等。

(4) 应用架构包括应用系统组合、实施项目组合、软件组件架构、SOA 等。 (5)技术架构包括平台、网络拓扑、标准和协议、软件工具组合。

(6)价值链分析包括数据、业务流程、组织、角色、应用、地点、目标、项目和 技术平台之间的关系。

**大数据治理：**理论与方法………

口价值链

信息交互通信椐架

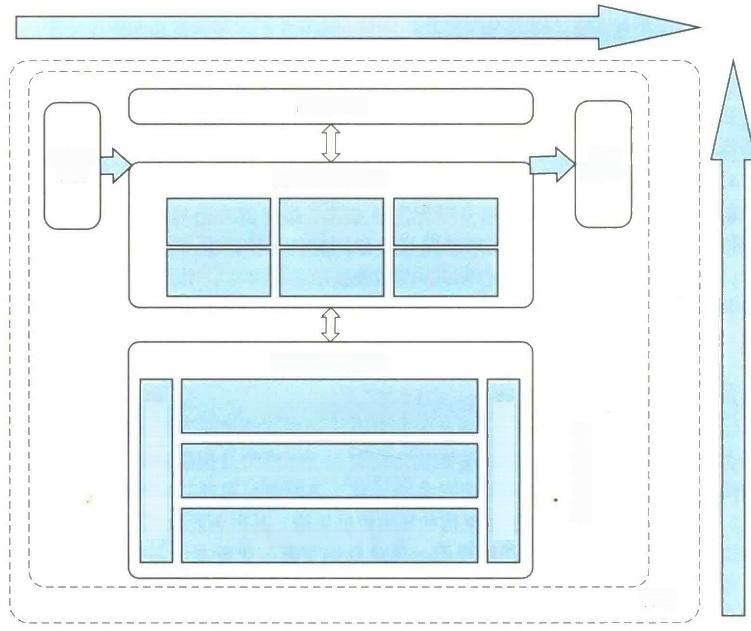
信息交互通信框架

安 全 与 隐 私

**2.4** **CESI大数据参考架构模型**

美国国家标准与技术研究院 (National Institute of Standards and Technology,NIST) 对9种大数据解决方案的体系架构进行了详细剖析和对比分析，确定了大数据体系架 构的共性部分，在此基础上按照逻辑角色和商业应用的目的给出了大数据参考架构①②。

中国电子技术标准化研究院 (China Electronics Standardization Institute,CESI) 对 NIST 研究成果进行了丰富和完善③,在原有架构的基础上细化出活动和组件的概念，明确了 角色的行为动作和行为环境支撑及相互之间的逻辑关系，使得整个架构更加具体形象， 如 图 2 - 3 所 示 。

信息价值链

数据提

供者

系统协调者

大数据应用提供者

采集

预处理

访问

分析

其他

可视化

大数据框架提供者

处理框架：提供计算和分析

平台：提供数据组织与分布

基础设施：提供网络、计算和存储

图2- 3 CESI 大数据参考架构

数据

消费者

管理

① NIST Big Data Public Working Group(NBD-PWG).NIST big data interoperability framework:volume5,archi-

tectures white paper survey [R/OL].(2015-08-25)[2016-04-10].

② NIST Big Data Public Working Group(NBD-PWG).NIST big data interoperability framework:volume6,refe-

rence architecture [R/OL].(2015-08-25)[2016-04-10].

③ 全国信息技术标准化技术委员会大数据标准工作组.大数据标准化白皮书(2018版) [R]. 北京：中国电

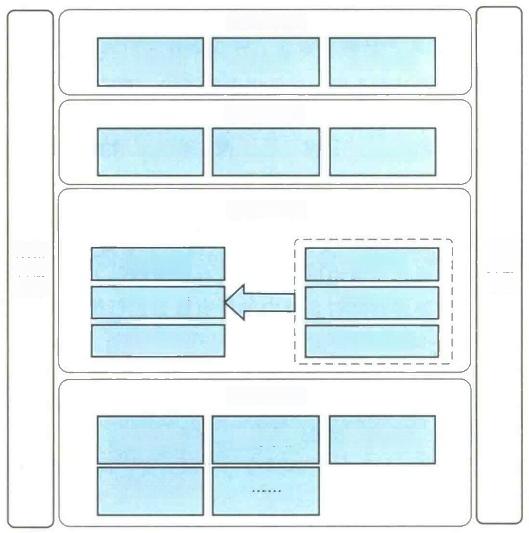
子技术标准化研究院，2018:1-129.

第2章|数据架构管理

整个参考架构围绕两个价值链进行构建 ： 横向为信息价值链 ， 通过数据收集 、 集 成、分析、应用分析结果创造价值；纵向为信息技术 (IT) 价值链，通过提供网络、 基础设施、平台、应用工具及其他服务创造价值。架构定义了5个逻辑角色：数据提 供者、大数据应用提供者、大数据框架提供者、系统协调者和数据消费者，整个架构 以 大 数 据 应 用 提 供 者 为 中 心 提 供 了 连 通 其 他 4 个 角 色 的 接 口 。 架 构 包 含 两 种 服 务 和 功 能保障构件：安全与隐私和管理，分别对各接口和大数据框架提供者内部进行安全与 隐私监管及对全系统各要素进行统一管理 ， 从而构成大数据应用的完整体系 。

) ) **大数据技术参考架构**

《 大 数 据 发 展 研 究 报 告 》 编 写 组 ① 借 鉴 了 ISO/IECJTC1/SC32 ( 数 据 管 理 和 交 换 分 技 术委员会)对大数据标准概念模型的研究成果，提出了大数据技术参考架构。其分层 视图如图2-4所示。按照分层视图的惯例，横向层次的上层依赖于下层，纵向层次贯穿 横向每一层，为横向层次提供约束和支撑。简单来说，该架构横向包括数据采集层、 数 据 支 撑 层 、 数 据 服 务 层 和 共 性 应 用 层 4 个 层 次 ， 纵 向 包 括 数 据 传 输 技 术 和 数 据 安 全 技术两类贯穿数据生命周期的关键技术 。



共性应用层

业务决策

数据服务层

数据分析

数据支撑层

流程管理

技术

数据质量

数据可视化

数据采集层

业务数据

外部来源数据

数据平台

数据存储

数据整理

数据 传输 技术

互联网数据

数据

安全

数据挖掘

感知数据

科学研究

信息服务

趋势预测

图2-4 大数据技术参考架构

① 《大数据发展研究报告》编写组.综合分析冷静看待大数据标准化渐行渐近(下) [J]. 信息技术与标准 化，2013(10):17-20.

大数据治理：理论与方法…………………

横向各层次具体功能如下：

数据采集层完成数据源的汇聚，形成大数据的初级资源池。

数据支撑层将收集上来的多样化、快速化和大规模的数据进行标准化处理，通过 数据平台集中向外发布。通过流程管理、质量管理和数据可视化等方法，为数据的业 务处理和应用提供支撑。

数据服务层根据不同类型的需求，提供数据挖掘、数据分析与信息服务等内容。

共性应用层是用户参考或者直接使用服务层提供的相关数据结果，为具体应用提 供支持，如科学研究、业务决策和趋势分析等。

纵向各层次具体功能如下：

数据传输技术为数据的采集、汇聚、交换、存储、应用等各阶段提供通道，使得 数据本身和其各个阶段的产物能够通过这一通道快速地传输，最终传递到用户手中。 网络传输通信贯穿大数据的各个层次，将需要和产生的数据进行传输交换。

数据安全技术为大数据提供可靠的信息安全保障环境，提升数据传输、处理和利 用的安全可控水平。从技术角度看，涉及加密存储技术与存储策略、链路的安全、数 据安全管理、安全审计技术机制等内容。

2**.6** **数据湖** **(Data Lake)**

数据湖是大数据时代发展起来的一种数据存储架构，其定义是由Pentaho 公司的创 始人兼首席技术官詹姆斯·狄克逊 (James Dixon) 给出的：“有人问数据湖是什么时， 我告诉他们，它就是你以前在磁带上拥有的东西。拿来你在磁带上的东西，把它倒入 数据湖，然后开始应用该数据。我们的看法是，只把需要的数据导入 Hadoop, 如果你 想结合来自数据湖的信息和客户关系管理 (CRM) 系统里面的信息，我们就进行连接， 只有需要时才执行数据结合。”

数据湖是一种在系统或存储库中以自然格式存储数据的方法，它支持以各种模式 和结构形式配置数据，主要思想是对企业中的所有数据进行统一存储，并将原始数据 (源系统数据的精确副本)转换为可用于报告、可视化、分析、机器学习等各种任务的 转换数据。数据湖整体形成一个集中式数据存储，容纳所有形式的数据：结构化数据 (关系数据库)、半结构化数据 (CSV 、XML 、JSON) 、非结构化数据(电子邮件、文 档、PDF) 和二进制数据(图像、音频、视频)。

数据湖的核心思想是把不同结构的数据统一存储，使不同数据有一致的存储方式， 在使用时方便连接，真正解决数据集成问题。其使用的方式为 Schema On Read, 即在 数据访问时，由数据使用者来解析和确定数据的格式，按需进行数据探索和处理，原 始数据写入者不关心其是否有统一的数据格式，不预设表结构以接入数据(对应的技 术为 Schema On Write)。可以看出，数据湖对终端用户的经验和能力提出了很高的要 求。数据湖在带来数据处理方面的优势的同时，也带来了一些新的风险。数据湖的优 势和风险如表2-4所示。

第2章 |数据架构管理

**表2-4** **数据湖的优势和风险**

|  |  |
| --- | --- |
| **优** **势** | **风** **险** |
| 无须建模定义数据结构即可保存，允许用户保存非结构 化、半结构化的数据。降低数据保存的成本，同时降低数 据产生和使用之间的延迟 | 有些数据如果在写入的时候不遵循一定的格式， 在使用时不一定能够解析其格式，若解析错误，使 用数据的结果将与其预期差异很大 |
| 给予终端用户最大的灵活度来处理数据，不同用户可以 对数据有不同理解，对于现在不需要处理或者无法处理的 数据、也可以保留原始数据供未来使用 | 用户在使用时，不得不先花时间去解析数据的格 式，不同用户多次解析数据造成计算资源浪费 |

**面向大数据的数据架构实现**

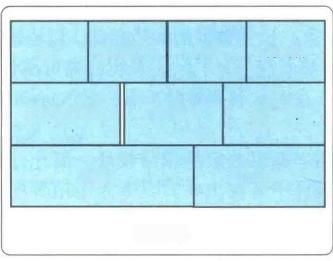
顾名思义，大数据是指规模庞大的数据集合。通常，对大数据来说，单台计算机 难以对其进行获取、存储、管理和分析。因此，大数据和云计算、分布式集群密不可 分。由于大数据涉及的数据类型复杂、数据量庞大，所以必须依赖于云计算的分布式 处理和存储技术。

实现大数据的数据架构离不开大数据框架，它是实现大数据架构的技术，且可以 协助我们在分布式环境中进行大数据处理，完成计算任务划分、集群负载均衡、数据 迁移转化等工作。

本节将简要介绍三大主流的大数据框架的数据架构。

**2.7.1** **Hadoop**

Hadoop 通过大量高效的硬件集群和标准接口构建大规模分布式计算系统，为大数 据提供存储和计算。Hadoop 的核心组件是 HDFS(Hadoop Distributed File System) 和 Hadoop MapReduce,其他组件为核心组件提供配套和补充性服务，其基本体系架构如 图2-5所示。



Pig Chukwa Hive HBase

Zoo

Keeper

Avro

Hadoop

MapReduce

HDFS

Core

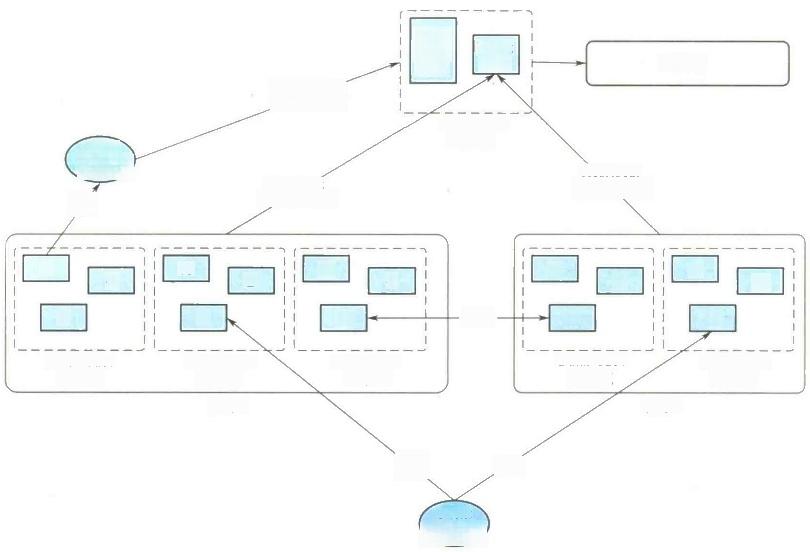
图2-5 Hadoop的基本体系架构

1)HDFS

其思想来源于Google文件系统 (Google File System,GFS), 是 GFS 的开源实现，

大数据治理： 理论与方法

以流式数据访问模式实现超大规模数据集存储。HDFS采取数据集“一次写入、多次读 取”方式，实现了分布式环境下流式访问数据的能力，保证了数据的大吞吐量。HDFS 的基本体系架构如图2-6所示，总体采用主从式结构模式，主要由 Client、NameNode 和 DateNode 几个组件构成。

Client

客户端

读出

Name

Space

Block

Map

State

元数据操作

NameNode

主节点

Heartbea

BlockReport

元数据

Heartbeat

BiockReport

A

C

E

DataNodel

从节点

A

B

E

DataNode2

从节点

机架1

B

C

D

DataNode3

从节点

复制

|  |  |
| --- | --- |
| A  C  D  DataNode4  从节点 | D  B  E  DataNode5 从节点 |

机架2

写入 写入

Client

(客户端

图2 - 6 HDFS 的基本体系架构

Client 是客户端，主要功能是为用户提供访问文件系统的接口，通过 NameNode 和 DataNode 交互访问HDFS 中的文件； NameNode 是主节点，负责协调 Client 对文件系统 的访问，管理文件系统的命名空间、文件目录树和元数据信息，同时负责监控和调度 DataNode;DataNode 是从节点，负责数据的实际存储，以心跳 (Heartbeat) 的方式周期 性地向 NameNode 报告其健康状况。为了保证系统的高可用性，还设有 SecondaryName- Node, 其是负责监控 HDFS 运行状态的辅助节点，是NameNode 的热备份。

2)MapReduce

其主要设计目标是为用户提供抽象的程序模块，简化分布式程序设计，将用户从

烦 琐 的 接 口 和 通 信 等 程 序 设 计 中 解 放 出 来 ， 只 专 注 应 用 程 序 的 设 计 ， 从 而 提 高 开 发 程 序和解决问题的效率。MapReduce 也采取主从式结构模式，基本体系架构如图2-7所 示，主要由 Client、JobTracker、TaskTracker 和几个Task 组件构成。

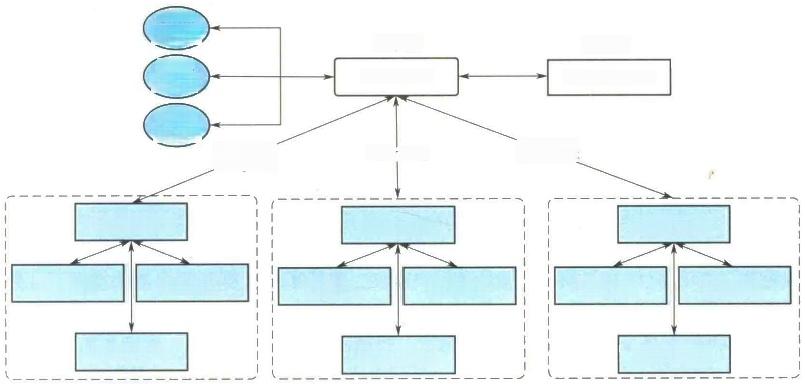
Client 是客户端，负责将用户编写的应用程序提交给 JobTracker, 并提供查看作业 (Job) 运行状态的接口； JobTracker是 MapReduce 的主节点，主要负责监控子节点、 TaskTracker 和作业的运行状况， 一旦子节点出现问题， JobTracker 会将任务转移到其他 子节点执行，同时JobTracker还负责跟踪任务的执行进度和资源的使用情况，发挥任务

第2章 | 数据架构管理

调度的作用。TaskTracker是JobTracker的子节点，主要为任务 (Task) 分配资源和提

供执行环境。Task是任务的具体执行单元，执行Map任务或 Reduce 任务，其中 Map 任 务以<key,value> (键/值对)为输入，经过计算处理后产生以 key值排序的<key,value> 序 列 ， 待 任 务 执 行 完 毕 后 进 入Reduce 任 务 ；Reduce 任 务 以 Map 任 务 产 生 的<key,value>

序列作为输入，进行规约处理，并得到 MapReduce 的最终结果。



Client

客户端

Client

客户端

Client

客户端

Heartbeat

TaskTracker TaskTracker TaskTracker

MapTask MapTask MapTask MapTask MapTask MapTask

ReduceTask

从节点

主节点

TaskScheduler

ReduceTask

从节点

ReduceTask

从节点

主节点

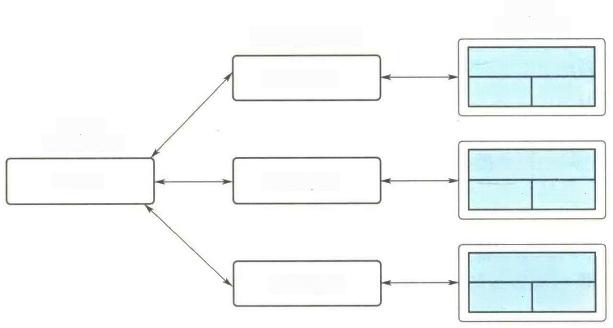
JobTracker

Heart tbeat Heartbea!

图 2 - 7 Hadoop MapReduce 的基本体系架构

2.7.2

Storm是一款开源的分布式实时流处理系统，最早由BackType 公司的 NathanMarz 开发，之后 BackType 公司被Twitter收购，Storm也随之由 Twitter 开源发布，目前 Storm 已成为 Apache软件基金会的孵化器项目之一。Storm 同样也采取主从式结构模式，核 心组件包括3部分： Nimbus 、Supervisor 和Zookeeper, 基本体系架构如图2-8所示。



从节点 Slave Node

Zookeeper节点族

Supervisorl

Zookeeperl

Worker Worker

主节点

Master Node

Supervisor2

Nimbus

Worker Worker

Supervisor3

Zookeeper3

Worker Worker

Zookeeper2

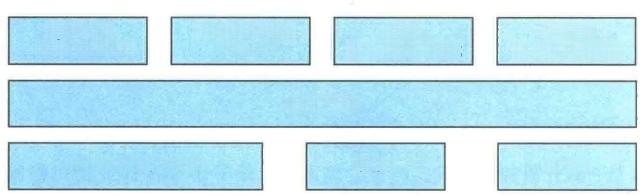
图2 - 8 Storm 的基本体系架构

**大数据治理：** 理论与方法…

Nimbus 是 Storm 集群的主节点，负责向工作节点分发应用代码和分配任务，同时 监控任务的执行状态和工作节点的健康状况。Nimbus 节点被设计成“快速失败 (fail- fast)” 的模式，所有的数据都存储在 Zookeeper上， 一旦节点死掉，会快速重启而不 会对工作节点造成任何影响。Supervisor 是 Storm 集群的从节点，每个节点上运行一个 Supervisor, 负责创建、启动、停止工作进程，控制工作进程执行分配的任务。与Nim- bus 相同， Supervisor也被设计成“快速失败”的模式，所有的状态信息也存储在 Zoo- keeper 上， 一旦节点死掉，会快速重启而不会丢失任何状态信息。Zookeeper 是整个 Storm 集群的桥梁，在整个系统中发挥协调作用，存储着 Nimbus 的数据和 Supervisor 的 状态信息，并负责Nimbus和 Supervisor 的通信。

**2.7.3** **Spark**

Spark 是在 MapReduce 基础上实现的高效迭代计算框架，它的最大特点是支持基于 内存的分布式数据集计算，从而大大提高了运算速度。Spark 最早由美国加州大学伯克 利分校于2009年开发，2010年实现开源发布，2013年由 Apache 软件基金会接管，并 成为其顶级项目。Spark 核心理念是通用和速度，集成了流计算框架、图计算框架、数 据查询引擎、机器学习算法库、分布式文件系统等功能和组件，其基本体系架构如 图2-9所示。



SparkStreaming Real-time

Spark

Standalone Scheduler

GraphX

Graph Processing

MLlib

Machine Learning

SparkSQL

Structured Data

YARN

Mesos

Core

图2-9 Spark 的基本体系架构

Spark Core 是Spark 的核心，实现了Spark 的基本功能，包括任务调度、内存管理、 错误恢复以及和存储系统的交互， Spark Core 定义了一个程序抽象模型 弹性分布式 数据集 (Resilient Distributed Datasets,RDD), 所有的应用程序都被抽象成 RDD 来完成 运算。SparkSQL是处理结构化数据的工具，通过引入RDD数据抽象，能够通过SQL语 言和集成其他SQL 工具实现对结构化数据的高效查询。SparkStreaming 是 Spark 的实时 流数据处理组件，它以时间片对数据进行分割形成 RDD, 能够以相对小的时间间隔对 流数据进行处理，同时提供良好的应用程序接口 (Application Program Interface,API) 和容错机制，能够与其他组件友好合作从而高效完成对流数据的处理。MLlib 是 Spark 的机器学习算法库，可以为处理大数据提供基本的机器学习算法，包括分类、回归 聚类等算法，同时还支持算法模型评估等。GraphX 是 Spark 对图操作和处理大规模并 行图计算的功能库，能够利用 RDDAPI 接口实现对图数据的统一高效处理。YARN、 Mesos 等运行于Spark 架构的底层，负责对集群资源和数据的管理，保证 Spark 集群节 点的扩展和统一高效运行。

**第2章** |数据架构管理

**2.7.4** **三种架构的比较分析**

上述三种开源架构的技术特点各有不同。Hadoop 采用一次写入、多次读取的流式 数据访问方式，更多的是以时间换空间，侧重的是数据吞吐量，不适合迭代式数据处 理，在数据处理的实时响应方面也不占优势，更适合在线下对静态大数据进行处理和 分析。Storm的设计理念是对大数据记录逐条持续进行处理，计算过程非主动结束，同 时容错性较高，更适合对实时流数据的处理。由于集成度相对不高，Storm 对其他类型 的大数据处理性能还有待完善。Spark 的集成程度较高，功能比较强大，能够对不同数 据类型(一般结构化数据，图数据、流数据等非结构化数据)的大数据进行处理。由 于Spark 是基于内存的计算框架，在数据量低于内存容量时计算性能突出，但当数据量 远大于数据容量时存在稳定性问题，更适合进行规模适当的迭代式数据处理。

8 数据架构设计原则

大数据架构从功能模块来看，可划分为数据标准(数据标准字典、数据流程规 范)、数据模型(数据主题域、概念模型、主数据体系、模型选择)、数据管理体系 (管理规范及流程、质量控制、元数据管理、调度管理、日志监控);从业务需求来看， 主要要求有灵活性、简易性、安全性、连续性、成本及时效等。数据架构设计从技术 上看需要遵从以下几项共性的原则：

数据对象统一 良好的顶层设计是保证数据质量的关键，数据对象统一可以避免 大量的不一致、不完整、不精确的劣质数据的产生。

数据和应用分离 这一原则可以说在数据库发展之初就被不断提及，数据应当是 独立存储和管理的，应当尽可能降低数据和应用之间的耦合度。

数据异构 大数据时代，很难保证企业或组织只需要处理单一类型的数据，因此， 在设计数据架构时，必须在设计的开始阶段就将数据的异构性考虑在内。如果这方面 考虑不足，那么当有新类型的数据处理需求时，会花费极大的代价来修改系统。

数据读/写分离 读/写分离有助于提高性能。

完备的数据安全机制 管理良好的数据是财富，无论在企业或组织内部还是在外 部，来自数据安全的威胁都不可忽视。为了避免安全问题导致的灾难性后果，在设计 架构时，应当逐层、逐模块地建立合理完备的安全预警和应对机制。

注重实时性 应当建立一个高效的数据架构，保证可以在适当的时候向决策者传 输和分析数据。另外，也需要重视历史数据的实时访问和流数据的实时处理。同时， 数据架构需要以全速支持高速的数据移动。

制定主数据管理机制 我们在第4章将详细介绍主数据管理。借助主数据管理存 储库，企业拥有单个“黄金副本”,可将数据同步到访问该数据的应用程序。通常，企 业或组织一直在经历着变化，包括增长、重组、合并和收购。主数据管理可以将分布 在各个“信息孤岛”上的重要数据集中起来并保证数据的一致性。

将数据作为服务 许多企业拥有各种数据库和遗留系统，这使得从各种来源获取 信息具有挑战性。可以通过标准化所有数据源的虚拟化数据服务层，来启用设备、系

**大数据治理：** 理论与方法 …………………

统无关的数据访问。从内部看，“数据即服务”可以视为私有云的一种形式，其中数据 通过数据管理平台、各类工具和应用程序作为可重用的标准化服务被提供给企业。从 外部看，“数据即服务”一方面可以为企业或组织的业务提供更丰富的支持，另一方面 也有助于不同企业或组织之间的数据流通。

在管理和执行流程上，在设计数据架构时，企业或组织也有几个共性的原则需要 遵守。

业务导向 应当与业务用户一起确定最有价值的数据类型。良好的数据架构的目 的是将企业的业务和技术方面结合在一起，以确保它们协同工作。

良好的组织结构 设计流程应当使良好的数据治理成为第一要务。与业务部门紧 密合作需要保证数据不但有价值，而且要经过严格审查。识别、提取和构建数据模型 的过程需要确保业务的质量和相关性。架构设计小组应当在个人数据所有者、管理委 员会、业务人员、技术人员之间搭起沟通的桥梁，同时必须确立数据责任。

拥抱变化 设计架构时应当重视可修改性，设计的目标在于建立能够适应业务环 节和技术环境变化。如今，在设计数据架构时，最好不要让数据架构与特定的技术方 案或解决方案过度耦合。进入企业的数据类型可能会发生变化，处理这些数据所使用 的工具和平台也会发生变化。如果市场上出现了一种新的技术或者业务，数据架构都 应该能够迅速变化以适应它。

26



**第** **3** **章** **元数据管理**

元数据 (meta data) 的一个流行而抽象的定义是“关于数据的数据”, “meta” 起 源于亚里士多德的名著《形而上学》中特别创造的词 metaphysics。metaphysics 是由 “meta” 和 “physics” 共同组成的，其中前缀 “meta” 被赋予了“延续与超越、更高抽 象层次”的含义。将 “meta” 与 “data” 结合，其中“数据”(data) 反映存在于真实 世界的交易、事件、对象和关系，而“元数据”则再抽象了一层，反映了数据本身的 交易、事件、对象和关系。中国台湾有学者将 “metadata” 译为“诠释数据”,在字面 上一定程度也反映了元数据的含义。

传统意义上，元数据多用于图书馆的卡片目录，使用元数据主要是为了达到编制 目录，描述图书馆馆藏的内容或特色，进而协助检索的目的。随着数字化逐渐成为存 储数据的普遍方式，图书馆也将其目录数据转换为数字数据库，数字数据库也有相关 的元数据标准。对于数字数据库来说，元数据的主要目的也是帮助组织、归档和查找 相关信息和资源。例如， 一些发达国家的政府会收集通信活动元数据，用于进行流量 分析或大规模监控。

不同行业有不同的元数据标准(例如，博物馆收藏、数字音乐文件、网站等)。根 据针对资源对象的不同，元数据的结构、描述、语义也各不相同。例如，在传统的图 书馆编目中，元数据可能包括主题索引、书评和文摘等； Web 网页的元数据可能包括 页面主题(及哪里有关于主题的更多信息)、页面脚本语言(如 HTML5)、产生页面的 工具等；地图类数据的元数据主要表现为地图类型、地图图例，包括图名、空间参照 系统和图廓坐标、地图内容说明、比例尺和精度、编制出版单位、发布日期或更新日

大数据治理： 理论与方法……………

期、销售信息等；音乐CD 的元数据可提供此专辑的音乐家、歌手和歌曲作者等信息。

详细的、标准化的元数据可以在很大程度上提高数据的可用性，它既可用来识别 某个特定的对象，也可记录该对象的具体特性，如类别、行为、功能、使用方式，或 记录其他对象的关系及该对象的恰当管理方式等。如果没有合适的元数据，数据治理 就缺乏一项重要的基础资源。David Marco 在他的《元数据仓储的构建与管理》 一书中 称元数据是“所有系统、文档和流程中包含的所有数据的语境，是生成数据的知识。” 试想，如果在各类数据驱动的应用中所收集和使用的数据均没有元数据说明(是一堆 不带任何描述的数字和字符串),那么这些数据的业务价值毫无疑问将大大受损。

既然元数据如此重要，那么如何更好地管理它们以发挥其价值呢?需要根据规范 流程、行业标准和实际需求来管理元数据。元数据管理是一项和主数据管理、数据治 理一样重要的功能，因为元数据是基础组件。

为此，本章将从元数据的概念和作用讲起，进而讨论业务元数据和技术元数据管 理当中的一些重要话题。

3.1 元数据概述

**3.1.1** **定义**

元数据最基本的含义是“描述数据的数据”。这一定义简洁而富有广泛的含义，因此 也延伸出了多种分类方法。例如，1994年，在 VLDB 会议文章 Metadata:A User's View 中，元数据被分成两类：结构性元数据和指南性元数据。其中，结构性元数据描述了诸 如表格、栏、密钥和索引等数据库对象的结构；指南性元数据帮助人们找到特定的物品， 而且经常被压缩为一系列自然语言中的关键字。2004年，在美国国家信息标准组织 (NI- SO) 的 Understanding Metadata 中，将元数据分成描述性元数据、结构性元数据和管理性 元数据，其中描述性元数据通常用于发现和识别数据对象，如标题、作者、科目、关键 字、出版商等；结构性元数据描述对象的构成物是如何组织起来的，如索引；管理性元 数据提供有助于管理资源的信息，如文件类型、文件创建时间和文件创建方式等。

元数据根据分类方法不同，有很多种定义和概念，难以一一尽述，本章的后续部 分将从数据仓库的角度展开，讨论元数据的分类法和其中一些重要概念。

在数据仓库中，数据来源于许多业务处理系统，其结构和语义均较为复杂，为了 管理这些复杂数据，需要有效维护和管理元数据。按照元数据的基本含义，数据仓库 的元数据应是“描述数据仓库中数据的数据”。从这个角度讲，可以将元数据类比为数 据库管理系统中的数据字典，其保存了数据的逻辑结构、文件组织、地址、索引等信 息。同时，由于数据仓库的复杂性较高，元数据还需要描述不同维度、不同语义、不 同领域数据的构建、管理方法和规则。由此可见，元数据对于数据仓库管理系统来说， 是非常重要的组成部分，其使用和管理是否安全有效，直接影响到数据仓库管理系统 的安全性和有效性。我们可以从如下几个方面来分析。

对于数据仓库的构建者来说，数据仓库中的数据往往来自不同的数据源，数据仓 库对这些异构数据进行复制、预处理、集成、汇总等操作。因此，元数据记录的信息

28

**第3章**|元数据管理

非常丰富，包括元数据系统到数据仓库的映射、数据转换的规则、数据仓库的逻辑结 构、数据更新的规则、数据导入日志及装载周期等信息。

对于数据仓库的管理者来说，其需要关于数据本身和系统的大量信息，如数据迁 移日志、备份日志、数据变换规则、仓库使用统计量、错误报告和审计跟踪等，这些 信息都需要依赖元数据来记录和管理。

对于数据仓库的使用者来说，其需要查询定制报表，并进行一些数据分析和挖掘 的任务，良好管理的元数据有助于使用者明确报表应该包含哪些内容、分析和挖掘应 该针对哪些数据。

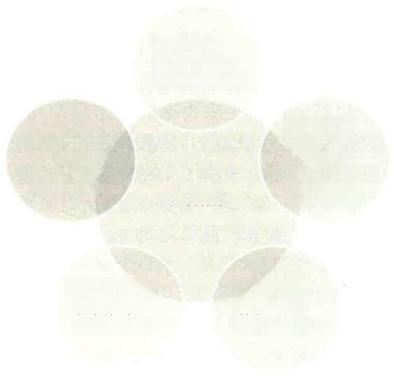
在数据仓库中，根据常用的元数据分类法，将元数据分为技术元数据 (Technical Metadata) 和业务元数据 (Business Metadata)。技术元数据为开发和管理数据仓库的技 术人员服务，业务元数据为管理层和业务分析人员服务。

技术元数据存储技术角度的数据描述，提供了数据管理和处理中的关键技术细节， 以便相关技术人员更好地掌控技术细节，包括结构信息描述(如接口信息、表信息等) 和数据处理的相关描述(如程序信息、存储过程、函数、序列等)。

业务元数据存储业务角度的数据描述，提供了沟通终端用户和实际系统之间的语 义层，帮助不了解系统技术细节的业务人员顺利使用数据，包括业务规则、业务术语、 指标定义、系统使用业务语言等。

元数据在数据仓库中扮演着重要的角色，其记录技术人员和业务人员需要的重要 信息，是整个数据仓库顺利运行的重要基础之一。元数据贯穿存在于整个数据仓库中。

图3-1中给出了元数据可能涉及的范围，但需要注意，这些只是元数据的一些可能 的举例，随着技术和业务的发展，元数据的外延可以非常丰富，难以一一列举。



**资源描述：**

主题、内容、

属性、特点等

**业务规则：**

清洗规则、 抽取规则、 转换规则、 汇总规则等

**“血缘”分析：**

数据来源、

数据关系等

**影响分析：**

使用状况、 评价分析、 改进建议等

**检索分析：**

数据组织、 数据定位等

元数据

图3-1 元数据

**3.1.2 组织方式**

可以从两种视角出发设计元数据的组织方式

大数据治理：理论与方法……

第一种是从数据集的视角出发，为每个数据集存储对应的元数据文件，文件中包 含与对应的数据集相关的全部元数据。这种方法在数据集数量不多时具有比较好的效 果，元数据相对比较独立，既可以被单独传输，也可以被用户单独访问以了解对应数 据集的概况。其不足之处在于：

(1)当数据集非常多时，可能要存储大量的元数据文件，这给元数据的索引、管 理和关联操作分析带来了较大不便；

(2)当数据不断演变时，元数据也可能发生变化，多个独立的元数据文件会给数 据演变时的维护工作造成困难。

第二种是从元数据管理和可扩展性的视角出发，建立元数据库，专门存储和管理 元数据。这种思想又可以进一步衍生出两条路线：

第一条路线是将元数据构建成结构化数据库表，例如，数据库表的一列代表元数 据的一个要素，数据库表的一行代表一条元数据内容。这种方法可以较为方便地实现 对元数据的统一管理，可扩展性也较高。增加数据集时只需要在元数据表中增加一行， 增加新的元数据要素时，只需要在元数据表中增加一列。其缺点主要有三个：

(1)与使用简单的元数据文件相比，建立元数据库的代价通常更高，包括前期的 设计和建立代价，以及后续的维护和更新代价；

(2)灵活性受限，如果元数据异构性较强(不同数据集的元数据要素种类差别比 较大)或结构比较复杂，构建元数据表比较困难；

(3)元数据本质上是结构化数据库表的一行，也需要用户能够接受和使用这种形 式。例如，图3-2就是本书的在结构化的元数据表中的记录。

|  |  |  |
| --- | --- | --- |
| 标题 | 作者 | 出版年份 |
| 大数据治理：理论与方法 | 王宏志，李默涵 | 2021 |

图3-2 本书在结构化的元数据表中的记录

为了解决结构化数据库表灵活性受限的问题出现了第二条路线，它使用资源描述 框架 (Resource Description Framework,RDF) 。RDF 是万维网联盟 (W3C) 提出的一组 标记语言的技术规范，采用“主—谓—宾”的形式表达资源描述和网络资源的内容与 结构。经常使用XML文件的形式表示。图3-2中的元数据如果写成(比较简陋的) RDF 形式，会是这样一个XML 文件。

<?xml version="1.0"?>

<RDF>

<Description about="<http://www.xxxx.com/>">

<标题>大数据治理：理论与方法</标题>

<作者>王宏志，李默涵</作者>

<出版年份>2021</出版年份>

</Deseription>

</RDF>

不过，由于当前企业和组织要处理的数据量越来越大，元数据库易于管理和高可

30

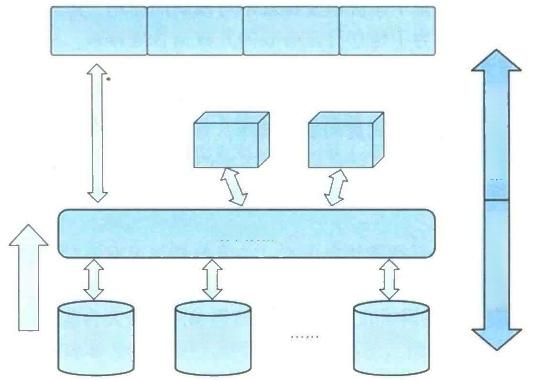
第3章 |元数据管理

EL

扩展性的特点更适合这种情况，因此更推荐使用元数据库。元数据库既可以通过统一 的格式和规则提供对数据的重要信息的一致描述，方便企业的统计和管理；也可以作 为一类数据索引，将分布在各处的数据集市(将在第5章介绍)集成起来。基于元数 据库，关联分析不同数据集市也比简单使用文件要方便。

**3.1.3** **作用和意义**

图3-3展示了数据仓库和元数据的关系，从图中可以看出，元数据库贯穿了整个数 据仓库，其在整个数据仓库中发挥的作用可以从以下几方面详述。



数据分析 数据挖掘 报表展示 ……

细节数据

数据源1

技术元数据

业务元数据

数据源2

数据源n

图3-3 数据仓库和元数据的关系

对底层的数据集成来说，元数据定义了多源异构集成所必需的关键信息。

数据仓库的基础是仓库数据库服务器，其通过后端工具和实用程序从操作数据库 或外部数据源提取数据。其数据很可能来自多个异构的数据集市，而由于数据是增长 和演变的，所以数据仓库也可能是以递增、进化的方式构建的。换言之，数据集市的 数量和结构也会随着时间发生增加和演变。在这个增量的数据集成过程中，仓库的模 式、维、分层结构，数据集市的位置和内容，数据的“血统”(数据的迁移历史和演变 序列等)等都要通过元数据来记录。如果没有维护良好的元数据，那么数据仓库在建 立和演变的过程中很有可能偏离设计初衷，甚至无法使用。

对模型设计构建者来说，元数据提供了沟通上下层次的关键信息。

模型构建是为终端用户服务的。因此，对处于中间层次的模型设计者和构建者来 说，元数据的存在就显得至关重要了。面向下层，元数据记录了数据结构、模式、抽 取规则等数据本身的重要信息；面向上层，元数据记录了流程、逻辑、规则等业务相 关的重要信息。只有维护结构良好、语义完备的元数据，才能指导模型的设计者和构 建者正确完成模型的构建，确保模型不会偏移目标功能。

对数据使用者来说，元数据提供了帮助理解数据的关键信息。

上层的数据使用者可能并不理解数据仓库的技术细节，此时，元数据所提供的易 于被人类理解的语义信息就显得尤为重要了，数据使用者可以通过元数据快速找到所

大数据治理： 理论与方法…

需要的数据，并理解数据的使用方式。

对随时间演变的数据来说，元数据有助于维护良好的数据质量。

优质的元数据记录了数据“血统”、数据转换规则、数据的结构和模式，以及数据 和业务逻辑的关系。这使得任意一个层次的操作者都能够从元数据中获取自己需要的 信息，进而保证在数据演变的过程中，对数据仓库的维护尽可能正确。这种正确性最 终会反映在数据仓库中数据的质量上。同时，成功的元数据管理系统可以把整个数据 仓库的演变有效地组织起来，使得系统的业务或功能发生变化时，不需要依赖特定的 开发人员，从而也保证了演变过程中的数据的优质性。

综上所述，元数据在整个系统中发挥着不可或缺的作用，因此对其进行有效的管 理也自然显得非常重要。为了建立行之有效的元数据管理体系，从企业或组织层面来 说，应该重视以下几点。

(1)确立清晰完善的元数据管理和维护策略，基于科学的方法和专业的工具来管 理元数据。应建立专业的技术词典和业务词库，方便查阅和规范元数据定义。对上述 词典、词库应进行分类和标记，以支持不同类型的数据访问和处理。同时，还应依赖 现有的成熟工具和编码体系，并在策略中预留出一定的富余量，以应对可能发生的 演变。

(2)维护清晰一致的元数据指向关系，将元数据与正确的对象链接在一起。元数 据不能独立存在，其描述的是具体的数据的相关信息。因此，能否简洁有效地获取元 数据到目标数据集/目标数据源的指向关系就显得尤为重要。为此，组织或企业应当建 立清晰一致的“元数据地图”,以便在有需要时，可以很容易地将元数据与目标数据 集/目标数据源关联起来。

(3)建立增量式的元数据构建和维护体系，可以基于多种方式采集和更新元数据。 元数据管理一方面需要明确当前的目标、需求、约束和策略等，另一方面还需要规划 组织或企业的未来愿景和需求。针对大数据异构性强、体量大、变化频繁等特点，设 计能满足不同场景下企业和组织需求的元数据采集和更新策略。

3.2 **·业务元数据**

对于不了解技术细节的业务人员来说，直接将数据仓库中存储的数据拿给他看， 无异于将一本“天书”交给他。业务人员如果不能读懂数据，那么让其基于数据来提 升业务能力就是强人所难了。而业务元数据恰好能够帮助业务人员更好地理解数据 语义。

**3.2.1** **意义**

大数据时代，数据量已经增长到很多普通机器都无法处理的程度。这些数据结构 多样、语义各异，即使对数据管理的专业人士来说，都是很难理解的，更遑论不了解 技术细节的业务人员了。但同时，从企业和组织的角度出发，肯定会希望数据能够被 最大限度地利用起来，以提高业务的质量。因此，需要有中间媒介来沟通数据和业务 人员——业务元数据正是一种很好的中间媒介。

**第3章** | **元数据管理**

业务元数据可以帮助企业或组织解决如下3个问题。

**1.** **理解偏差**

理解偏差包括很多方面：不同业务人员对业务理解不一致、业务人员和技术人员 对系统不同部分的理解不一致、技术人员对底层的数据库系统理解不一致等。这些理 解不一致在系统的构建和使用过程中非常常见。我们前面提到过，可以建立专业的技 术词库和业务词库，这些技术词库和业务词库中应该包含技术和业务术语的明确定义 和适用场景，同时进行准确的分类和标记。这样，业务人员和技术人员可以通过查阅 词库来获取业务和技术元素的精确定义，从而更好地理解系统中的各类元素及其联系， 从而较大程度改善理解偏差问题。

**2.** **导航困难**

对于大数据来说，线性扫描一遍全部数据的时间是难以接受的。如何从海量数据 中快速找到所需要的内容，很多情况下依赖于构造精巧的索引结构。而元数据就可以 视为一类实用的索引结构。业务元数据可以为业务人员提供较完备的导航功能，让业 务人员可以快速找到自己需要的内容。

**3.** **员工依赖**

业务元数据作为一种易于查询和访问的文档，可以帮助业务人员完成工作的交接。 当老员工离职的时候，新员工可以通过了解元数据来熟悉业务和系统，这样一来，就 降低了企业或组织对业务人员的依赖性，也降低了老员工离职时给企业带来的损失。

由上可知，元数据可以作为业务人员和计算机系统之间的“翻译接口”之一，在 帮助业务人员理解系统的同时，也帮助系统更好地为企业或组织提供服务。

**3.2.2** **概念**

对于大数据来说，业务驱动的数据定义和组织非常重要。这驱使业务元数据的设 计者从业务的角度出发，理解需求并设计数据的传递、存储和分析。目前业务驱动可 以基于一些现有的标准(如ISO/IEC11179-4) 来创建。大体来说，业务元数据中需要 明确定义几项内容，包括名称、定义、应用场景、相关数据源等。

例 关于点击率的元数据。

名称：点击率。

定义：网站页面上某内容被点击的次数与被显示次数之比。

应用场景：在网络广告中，点击率是在网页上的一条广告打开后被点击次数的百 分比。例如，如果该网页被打开了1000次，而该网页上某一广告被点击了10次，那么 该广告的点击率为1%。

相关数据源： S135 、S136 、S137。

业务元数据必须真正代表业务层对数据的理解，其关系到企业对数据的需求是否 准确地被实现，以及数据架构是否产生漂移。业务元数据中的很多术语在技术元数据 中描述角度不同，甚至很难体现。其来源可能也比技术元数据要多样化，包括 ERP (企业资源计划)系统中的业务逻辑、业务规则，各类业务报表，数据模式(如 Excel 表格的表头、计算公式、填写说明等),产品说明书中的各项(包括产品型号、等级

大数据治理： 理论与方法

价格、成分等)。系统的需求文档中往往也包含了大量的元数据。

目前，很多企业只关注技术元数据，忽略了业务元数据，其往往喜欢从数据源头 开始，自下而上地建立自己的元数据体系。单纯地自下而上构建元数据体系，会使得 从上层业务角度来看的数据令人费解，其中充满着让人难以理解的符号，如“S135、 S136、S137”等。这些符号蕴含的语义是什么、与哪些具体业务相关联都不清楚。与 技术元数据相比，业务元数据的来源更复杂，分散在企业环境的各个角落。为实现业 务元数据的管理，企业需要有效的方法和手段。

**3.2.3** **实践要点**

业务元数据本质上还是一种数据，所以，只要应用场景合适，通用的数据管理 手段都可以用来管理业务元数据，如结构化数据库、XML 数据库、知识图谱等。但 由于业务元数据是从业务的角度定义和组织的，所以在实践中有一些需要格外注意 的要点。

**1.** **自上而下地建立业务术语体系**

这包括企业数据模型的高层信息、整个企业的业务概念、不同粒度概念的层级从 属关系、不同类别信息间的相互联系等。这些信息应当共同构成业务元数据所涉及的 方方面面。

**2.在自动化抽取元数据的同时也需要人工保证元数据的准确性**

大数据时代，完全手工梳理元数据渐渐变得不可行，为此需要设计自动化抽取工 具。不过，与技术元数据不同，业务元数据往往存在于非结构化数据中，如保险业的 保单、制药业的药品说明、医疗行业的病例、商业合约、法律条文等。这些非结构化 数据中包含的语义信息往往比较复杂、专业，自动化抽取很难保证完全准确，因此， 需要专业人员予以修正和确认。如何在自动化的效率和人工的准确性之间找到恰当的 平衡点，是实践中需要特别考虑的问题。

**3.** **为业务元数据做好恰当的标记**

标记业务元数据和技术元数据的关联关系，可以将业务元数据链接到相关的技术 元数据和原始数据中去，达到索引数据的目的。

**4.必要时设置专门的管理者**

由于元数据的重要地位，在必要时，企业和组织应当设置专门的管理者，负责识 别和管理业务元数据。

3.3 技术元数据

**3.3.1** **意义**

技术元数据是关于数据仓库技术细节的数据，是开发和管理数据仓库所需要的重 要信息。对于大多数非技术人员来说，其可能都不会意识到技术元数据的存在。但是， 维护良好的技术元数据对于数据仓库系统的开发非常重要，这主要表现在以下4方面。

**第3章**|元数据管理

安全层

**1.技术需要**

技术元数据存在的一大原因就是，在构建和维护系统的时候，需要元数据来记录 一些关键信息，如数据源的地址、数据源的大小、数据格式、数据仓库的模式、系统 层次结构、各模块的输入/输出等。这些元数据必须被记录下来，并且在系统演变时得 到及时的维护，这样才能保证系统的开发、演变、维护过程的正确性。研发人员在研 发新的算法时，也往往离不开技术元数据。

**2.** **审计跟踪**

如前文所述，数据仓库的数据源可能时刻处于演变的过程中，需要技术元数据来 记录数据“从哪里来”“到哪里去”“有什么用”“如何演变”等信息，这些信息对欺 诈分析和风险评估等应用非常关键。

**3.** **权限管理**

数据仓库中融合了许多异构数据源，这些数据源的所有者可能是独立的；同时， 数据也不总能共享给其他的“同事”们。出于安全、版权、隐私等原因，通常要求严 格的权限管理，这些与权限管理的信息都存在技术元数据中，在数据读、写、传等操 作之前，需要先基于权限元数据验证。

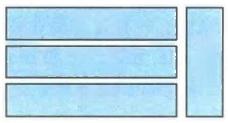
**4.** **促进沟通**

和业务元数据类似，技术元数据也承担着沟通的责任。例如，系统中不同模块的开发 和维护，需要靠技术元数据来定义和管理接口，新入职的技术人员可以通过阅读技术元数据 来理解系统的整体结构和技术细节，系统管理员也可以基于技术元数据进行正确的管理。

**3.3.2** **概念**

技术元数据的定义多种多样，但“万变不离其宗”。各种定义的基本思想均是“技术 元数据是系统设计与管理人员在开发和日常管理数据仓库时需要的元数据”。其既包括一 些静态的技术参数，如技术术语定义、对象和数据结构的定义、设备参数、统计指标等， 也包括一些动态参数，如数据流中的事件、实时访问记录、实时查询统计数据等。

技术元数据可能以形式各异的结构存在于系统各处。例如，表示系统分层结构 (如图3-4 (a) 所示)的元数据可能以文档中的箱图形式存在。 一些安全性检查策略 也可能直接存在于代码中，如图3-4 (b) 所示。

业务层

逻辑层

物理层

(a) 系统分层结构

if((wouldToggleZenMode(ringermode)&&checkCallerSystemOrSamePackage(caller)&& checkAceessPolicy(caller)){

throw new SecurityException(…)

(b) 安全性检查策略

图3-4 技术元数据

大数据治理： 理论与方法…

**3.3.3** **实践要点**

从概念可知，技术元数据往往会覆盖数据源接口、数据仓库与数据集市存储、 ETL 、OLAP、数据封装和前端展现等全部环节，其可能用来表达技术规则、数据结构 属性、数据映射关系和数据“血缘”、统计指标等一种或者多种语义。为此，在实践中 需要有良好的技术元数据收集和管理机制。

(1)有计划地收集和维护分散在各处的技术元数据。技术元数据往往分散在系统 各处，想要有效管理这些元数据，首先要明确元数据放置在哪儿，以及如何收集这些 元数据。 一个可行的建议是从系统架构入手。通常一个设计良好的系统具有较完备的 架构文档，里面明确了各利益方的需求、各模块的定义和关系，通过阅读架构文档， 可以自上而下地对系统有一个全面的认识，进而也更容易梳理出应该从何处收集何种 元数据。

(2)设计有效的存储和管理技术元数据的机制。如前文所述，技术元数据的来源 虽然未必像业务元数据那么广泛，但形式各异，可能存在于系统的各个部分。为此， 需要建立元数据存储和管理系统，设计简洁一致的数据转换接口，根据应用特性存储 和管理元数据。具体的存储和管理策略我们会在后面的章节稍加讨论。

(3)恰当的连接有相关性的技术元数据。和业务元数据一样，技术元数据也不应 当是独立存在的，需要将技术元数据与其相关的数据、模块、外部组织等元素准确连 接起来，这样才能发挥元数据的作用，并且保证元数据的一致性。

(4)以易于理解的方式表达技术元数据。元数据的存在形式多种多样，但为了满 足其沟通各方的要求，元数据应当以易于理解的方式被表达。图3-4 (b) 所示的安全 检查策略，就应该被抽取和表达成人类更容易理解的形式。

**3.4** **元数据管理**

前文已经反复强调过元数据对于系统、企业和组织的重要性，那么如何有效地对 元数据进行管理呢?我们在前面已经讨论了一些方法学层面的要点，在本节，我们来 讨论元数据管理中的一些细节。

在明确“怎么管理元数据”之前，企业或组织应该首先回答“我们需要的元数据 是什么”,以及“为什么需要这些元数据”。为此，可以根据系统本身和系统各利益方 的类别，对元数据进行分类。举例来说， 一种针对商业智能 (Business Intelligence,BI) 元数据的分类如下：数据源元数据、ETL 元数据、数据仓库元数据、报表元数据、业 务元数据。这几类元数据每一类都对应着 BI的一个层次，每一类元数据也都有自己更 精细的定义。

可以参考系统架构设计的方式，让各利益方列举自己的元数据需求，回答“是什 么”和“为什么”这两个问题。在每个利益方提供了完备的元数据需求后，再由组织 多次讨论，迭代得到最终的元数据需求。在本节的后续部分，我们将讨论如何管理这 些元数据

 36 

**第3章** |元数据管理

**3.4.1** **元数据管理方案**

元数据广泛存在于各处，在应用程序、配置文件、IoT 、云、媒体和数据模型中都 有分布。因此，为了管理元数据，需要有成形的管理框架。目前，很多ETL 厂商、数 据库厂商或开源组织都提供了一些元数据管理方案或框架，如Ascential 的 DataStage 、 Informix 的 PowerCenter 、Oracle Metadata Management 、Apache Allas 、Cloudera Navigator、 MetaStage 、MetaCenter。

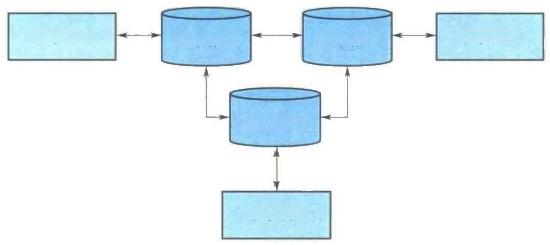
根据元数据的管理方式，可以将元数据的管理方案分类成三类：分布式、集中式、 联邦式。

**1.** **分布式**

分布式元数据管理的整体结构如图3-5所示。由于系统的各种历史遗留问题，在很 多实际系统中，子系统之间的连接往往不是一开始就建立的，而是随着应用的发展需 要被逐步建立的。因为早期各系统互相独立，所以各子系统的元数据往往具有较高的 异构性、自治性、冗余性。在连接子系统时，将元数据集成到一起往往会导致很高的 开销且易出错，因此，很多企业或组织会选择仍旧由各子系统各自维护自身的元数据， 并提供必要的共享机制。这种方法的优缺点如下。

优点：开销相对较小；可以用比较自然的方式管理元数据，元数据能够和其关联 的数据源或数据集一起管理；不易因元数据集成而导致错误。

缺点：元数据异构性强，因此元数据共享通道设计较为复杂；元数据冗余性高， 更新时易导致元数据不一致；需要很多的元数据交换接口，维护不易。



**子系统A** 元数据A 元数据C 子系统C

元数据B

子系统B

图3-5 分布式元数据管理

**2.** **集中式**

集中式元数据管理可以克服分布式元数据管理的一些缺点，其设计了一个集中式 的数据管理组件，专门用于管理元数据，如图3-6所示。该组件通常称为中心元数据 库，负责管理和发布所有子系统的相关元数据，提供元数据的一致表达，每个子系统 直接从中心元数据库获取元数据。

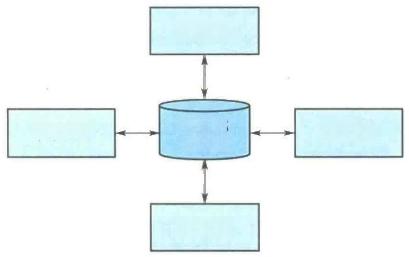
这样做的优点和缺点都很明显。

优点：数据通道的数量较分布式大为减少，只需要维护每个子系统到中心元数据 库的数据通道，元数据本身的共享在元数据库内部即可完成；降低了元数据的冗余性；

大数据治理： 理论与方法……………………

可以提供元数据的标准的、 一致的表达。

缺点：对于中等规模的企业而言，中心元数据库足够维护和管理其涉及的所有元 数据，但对规模较小的企业(小企业)和规模较大的企业(大企业)而言，集中式则 存在一些问题。



子系统A

中心元

数据库

子系统B

子系统D

子系统C

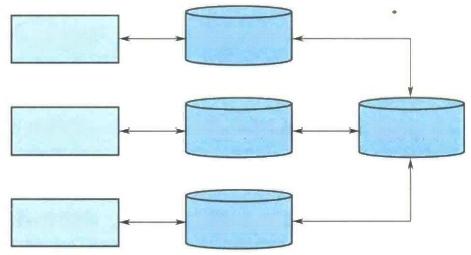
图3-6 集中式元数据管理

(1)对于小企业来说，集中式元数据管理可能会显得“过度设计”,因为这些企业 中的元数据并没有多到必须去单独管理的程度，设计集中式元数据库往往会增加不必 要的开销。

(2)对于大企业来说，集中式元数据管理则可能显得“设计不足”,大企业中的元 数据量大且复杂，涉及的子系统也非常多，单一的中心式元数据库可能无法负担整个 系统的元数据管理，从而成为系统性能和数据质量的瓶颈。

**3.** **联邦式**

如前文所述，分布式和集中式元数据管理各有优缺点，联邦式元数据管理则尝试 综合这两类方案的优点。图3-7所示为简单的两层级联式联邦元数据管理的整体结构， 如果有需要，也可以建立多层级联式的结构。联邦式元数据管理同时保留了本地元数 据和共享(中心)元数据机制，建立了层级式的元数据管理方案。



本地元数据A

本地元数据B

本地元数据C

中央元数据

子系统A

子系统B

子系统C

图3-7 联邦式元数据管理

联邦式元数据管理的优缺点如下。

优点：保留了分布式元数据管理的一部分特点，使得不同类别的元数据可以相对

**第3章** | **元数据管理**

独立、自主地被管理；共享元数据提供了对元数据的规范、 一致的表示；元数据通道 相对比较简单(相对分布式而言)。

缺点：系统相对前两种方案更为复杂，需要考虑的细节也更多，因此对系统的设 计要求更高，成本也相应会有所增加。

**3.4.2** **元数据标准和规范**

在前文中我们已经反复提到，元数据的交互非常重要。随着元数据的种类、来源、 形式越来越多，不同系统、组织或企业对元数据交互的需求日益旺盛，人们不得不开 始考虑设计元数据标准。如果各类系统、组织或企业都能够基于统一的标准来管理元 数据的表达和交互，那么可以极大地降低元数据异构性及交互开销，并提高系统稳定 性和元数据质量。

目前，国际上已经有许多成型的元数据标准，包括用于各种网络数据资源的英国 Dublin 元数据核心元素标准、美国联邦地理数据委员会 (Federal Geographical Data Committee,FGDC) 的地理空间数据的元数据内容标准、美国国家航空航天局 (NASA) 的 DIF 标准等。

对数据仓库来说，有元数据联盟 (Metadata Coalition,MDC) 的开放式信息模型 OIM(Open Information Model) 和对象管理组织 (Object Management Group,OMG) 的 公共仓库元模型 CWM(Common Warehouse Model)。

元数据联盟建于1995年，目的是提供标准化的元数据交互。MDC 于1996年发布 了MDIS(Metadata Interchange Specification),1997年接受了微软的建议，对OIM的技 术进行评审。

对象管理组织在1995年采用了MOF(Meta Object Facility),1997 年采用了 UML (Unified Modeling Language),2001 年 ，OMG 颁布了CWM。这三个标准形成了OMG 建 模和元数据管理、交换结构的基础，推动了元数据标准化的快速发展。

目 前 ，MDC 和 OMG已经合并，今后将遵循统一的 CWM 标准。

CWM 的主要目的是让人们在异构环境下进行规范的元数据存储和交换，交换的双 方涉及不同的数据仓库工具、平台、子系统和元数据库。其主要包含四方面的规范。

CWM 元模型：用于描述数据仓库系统。

CWM XML:CWM 元模型的XML表示。

CWM DTD: 共享元数据的交换格式。

CWM IDL: 共享元数据的应用程序访问接口。

CWM 基 于 UML(Unified Modeling Language) 、MOF 和 XMI(XML Metadata Inter- change), 这三部分分别实现了对元数据的设计、存储和交互，为元数据管理提供全面能 力支持。其中UML 对 CWM 模型进行建模；MOF是元模型和元数据的存储标准，提供在 异构环境下对元数据知识库的访问接口； XMI使元数据以XML文件流的方式进行交换。

目前，绝大多数数据仓库和元数据管理工具已经支持CWM, 或已经宣布在下一版 本的产品中支持CWM。其也已经被Java 标准化组织着手扩展到J2EE 体系结构当中， 形成JMI(JAVA Metadata Interchange) 规范、用于OLAP 分析的JOLAP 规范和用于数 据挖掘的JDMAPI 规范。

**大数据治理：** 理论与方法

**3.4.3** **元数据管理的成熟度**

在实施元数据管理的过程中，可以参照元数据管理的成熟度模型确定企业当前元 数据管理所在层次，并根据业务需要制定路线图实现元数据管理水平的提升。元数据 管理成熟度模型代表了一种从组织结构出发，基于知识管理氛围去理解元数据的视角。 为了给元数据管理的成熟度树立一个基准，我们需要分析它与人、流程、技术之间的 相互联系。

目前，元数据管理的成熟度有多种分类方法。在 Metadata Maturity Helps You Main- tain Business Relevance 中，将其分为 Ad Hoc、组织、度量、分析、优化5级成熟度。

Ad Hoc: 开始接触元数据的应用，包括管理内容和内容工作流。

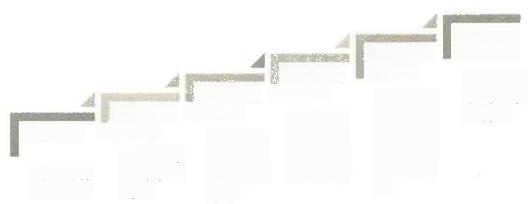
组织：对内容技术的常规理解，通常以内容管理系统和集中的共享文档存储库的 形式开始

度量：展示内容管理系统和核心竞争力方面的经验，如获取、编目、转换、转码， 分发等。

分析：通过有组织的知识传递来管理对业务至关重要的存储库和工作流系统。

优化：了解和预测企业需求，以准备未来的业务需求。

IBM 针对元数据管理提出如图3-8所示的成熟度模型，将元数据管理的成熟度分 成6个级别。



元模型驱 动管理

构建元模 型和元元 模型来进 一步优化 元数据管 理

元数据统 一存储

由中央元 数据库负 责存储元 数据

元数据集 中管理

不仅集中 存储，而 且集中规 划和控制

从属于业 . 务系统

元数据与 业务系统 绑定

元数据管 理自动化

高度自动 化的元数 据管理

初始状态

基本没有 管理可言

图3-8 元数据管理的成熟度模型

初始状态：基本没有管理可言。元数据分散于日常的业务和职能管理中，业务和 职能相关的人员局部产生，元数据传递依赖人员的交流，系统设计时不考虑元数据

*交互。*

从属于业务系统：元数据与业务系统绑定。元数据随业务系统的构建被设计、获 取、维护，并随业务系统孤立地全部或部分管理。业务元数据和技术元数据基本仍处 于分散状态，未得到统一管理，元数据之间互通互联困难。

元数据统一存储：由中央元数据库负责存储元数据。元数据依然在局部产生和获 取，但会集中到中央元数据库进行存储，由中央元数据库统一管理业务元数据和技术 元数据。并且，业务元数据和技术元数据之间全部或部分通过手工方式做了关联。这 种方式会使得元数据在整个企业层面可被感知和搜索，极大地方便了企业获取和查找 元数据。不过，在产生和获取元数据时，由于缺乏统一的规划，元数据中容易存在冗 余、不一致、二义性等问题。

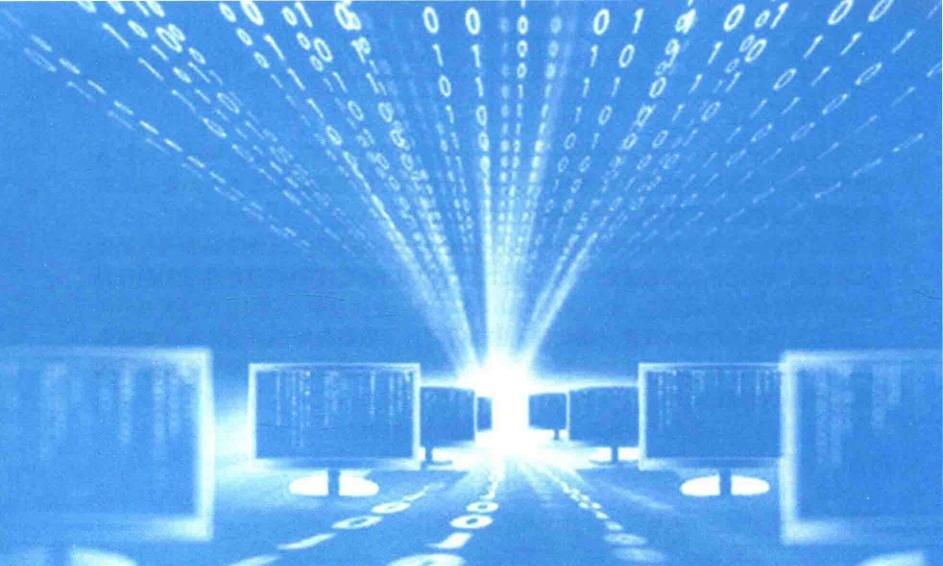
40

**第3章**|元数据管理

元数据集中管理：不仅集中存储，而且集中规划和控制。这一层次增强了元数据 的集中控制，元数据无法孤立地被修改，在其修改前后都需要通知其他人。业务元数 据和技术元数据之间还是通过手工方式进行映射的。

元模型驱动管理：构建元模型和元元模型来进一步优化元数据管理。基于元模型 及元元模型可以优化元数据中的不一致和冗余，创建、管理和共享元数据词库和对元 数据进行归类的分类系统。

元数据管理自动化： 高度自动化的元数据管理。元数据中的任何变化将自动触发 业务工作流，以便其他业务系统进行相应的修改，基于元模型和元元模型，可以完成 元数据的自动推断和映射。



**第** **4** **章** **主数据管理**

直观地讲，主数据 (Master Data) 就是所有数据中最具价值的那部分数据，它们 是公司的基本业务数据，通常是长期使用的，且可用于多个应用程序。例如，企业的 核心业务实体，包括客户、员工、产品、物料、厂商、账户等多方面的数据，都属于 主数据。近些年，主数据的重要地位逐渐被人们所认可。学术界和产业界关于主数据 的研究日益增多。

和元数据类似，主数据也不是一个新的概念，但也在近些年才得到人们的重视。 早期，人们对信息系统架构的设计不够重视，同一企业或组织中的各子系统各自独立 产生和维护自己的数据。但在实际的业务和技术流程中，各子系统的数据往往会互相 重叠，此时，孤立产生和维护数据的弊端就会显现出来——当数据在局部发生更新时， 这种更新很难被传播到其他子系统中。如果被更新的数据存在于大量的子系统中，就 会使得数据中存在大量不一致、过时或缺失等质量问题。例如，对政府来说，很多部 门或者职能机构的子系统中都存有本地居民的个人信息，这些个人信息对于相关业务 的正常运转是非常重要的。理想情况下，这些个人信息在不同的子系统应该都是完整、 一致且不过时的。但如果没有良好的管理和规划，很难想象如何能够保证这些重要数 据不会随着时间的推移变得混乱不堪。近些年，随着业务需求和监管需要，人们不得 不拿出精力来解决这些分散在各处的主数据带来的问题，因此“主数据管理” (Master Data Management,MDM) 的概念及重要性也被业界逐渐接受。

主数据和上一章的元数据是不同的两个概念。在这里可以回想一下上一章的内容： 元数据是指“描述数据的数据”;而主数据则是指实体数据，如客户信息、厂商信息

**第4章** | 主 数 据 管 理

员工信息等。举例来说，同一个系统可能在不同的企业或组织中被多次部署(如同一 套产品信息管理系统),在不考虑个性化定制的情况下，每套系统的元数据都基本是- 样的，但不同企业或组织中所维护的主数据(如核心产品信息)可能是各不相同的。 主数据管理的重点在于如何将不同职能部门中共享的那些重要的实体数据统一地管理 起来，进而提高数据质量并发现重要的业务洞察。

本章我们将从主数据的概念讲起，逐步探讨主数据管理的架构设计、核心功能及 实现风格。

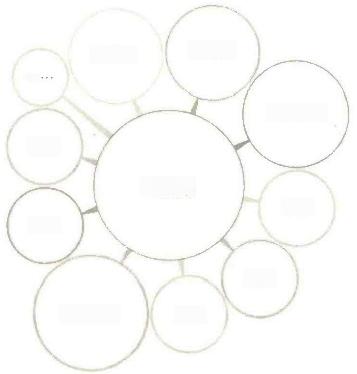
4.1 主数据概述

4.1.1 **主数据的概念**

主数据是整个企业范围内各个系统(包括操作或事务型应用系统及分析型系统) 间要共享的业务实体的数据，如客户、产品、厂商、账户等信息。主数据是关于关键 业务实体的权威的、最准确的数据。

通过分析主数据域间的关系，可以获得对企业或组织来说非常重要的业务洞察。 例如，银行维护的主数据可能包括客户基本信息、客户账号信息、理财产品信息等。 如果能够将客户的基本情况和客户使用其账号购买的理财产品信息关联起来进行分析， 银行就能获得客户的储蓄和理财习惯，进而能够为客户提供更好的个性化服务和推荐。 又或者，执法机构需要维护犯罪嫌疑人、证人、受害人等当事人主数据，以及案件、 判决相关的主数据，这些主数据如果能够进行关联分析，就能够为执法机关的案件追 踪和案件研判提供良好的数据辅助。

需要注意的是，主数据指的不是企业内所有的业务数据，只有那些有必要在各个 系统间共享的重要数据才是主数据，如大部分的交易数据、账单数据等都不是主数据， 而像描述核心业务实体的数据，如客户、供应商、员工、合作伙伴、产品、原料等都 是主数据(或者更严格地说，是主数据的一部分),如图4-1所示。



零售商

人力资源

主数据

仓库

客户

员工

供应商

\*

原料

产品

合作伙伴

图4- 1 主数据包括的内容

**大数据治理：**理论与方法…

我们可以尝试整理出对主数据的一个相对严格的定义。

主数据是企业内部数据的一个子集，是跨越多个部门或多种服务被重复使用的高

价值核心数据，并具有一致、准确、完整、可控的性质。

主数据一致性的管理主要是为了保证多个主数据实体或数据表之间的一致性，例 如，同一类原料的计量单位在不同的实体或数据表中应当保持一致。

主数据的准确性和完整性是为了保证数据如实地反映真实世界的情况，例如，如 果有客户的地址发生变化，主数据应当及时更新以保持准确和完整。

主数据的可控性是为了保证主数据不会发生非预期的变化和使用，即其更新、读/ 写、运算都是受监管、可追溯的。

为了保证主数据的这4个性质，需要有系统的主数据管理思路和方法。

**4.1.2** **主数据管理的类型**

不同的行业或不同的机构可能需要管理不同类型的主数据。例如，政府部门的政 务管理系统涉及的主数据可能是居民、城市、社会民生方面的基本信息，企业的主数 据则包括客户、员工、原料等基本信息。在确定如何管理主数据之前，需要明确被管 理的主数据的类型。

从顶层设计的角度来看，理想的主数据管理应当是与领域无关的，即能够直接 (或经过微调后)适用于各种主数据类型。不过，在实际应用中，主数据管理经历了漫 长的发展阶段，且并非每个企业或组织都需要如此高度泛化的主数据管理。因此目前 主数据管理根据所涉及的领域可以分为单领域主数据管理 (Single-domain MDM) 和多 领域主数据管理 (Multi-domain MDM)。

单领域主数据管理局限在某个特定领域。如果主数据管理关注产品或者服务的定 义和生命周期，此类主数据管理也称为产品信息管理 (Product Information Management, PIM) 。如果主数据管理关注管理客户相关的信息，此类主数据管理也经常称为客户 数据集成 (Customer Data Integration,CDI)。针对单一领域的专用解决方案可以解决 基本的主数据管理问题，但可扩展性可能不够高。例如，如果某公司制定了产品信 息管理方案来管理其产品数据，那么当公司发现自己也需要管理客户数据时，之前 制定的主数据管理方案就可能会受到很大挑战。因为针对产品数据设计的主数据结 构可能很难适用于客户数据。产品实体之间的关系和客户实体间的关系管理方式可 能也存在差异。

多领域主数据管理提供了较高的可扩展性。随着数据的积累，以及企业或组织规 模的扩大，不同新领域的数据会不断加入，单领域主数据管理会逐渐发展为多领域主 数据管理。

**4.1.3** **主数据管理的基本思路**

主数据管理是指一整套规范、技术和方案，用于生成和维护企业主数据，并保证 主数据的4个重要性质。针对不同性质的特性，我们可以整理出三个维度的主数据管 理的基本思路： 一致性对应主数据的标准化管理，准确性和完整性对应主数据的质量 管理，可控性对应主数据的安全性管理。

第4章丨主数据管理

**1.标准化管理**

正如我们在主数据定义中已经讲过的，主数据是跨越多个部门或多种服务被重复 使用的高价值核心数据，而这种重复使用需要建立在标准化的数据管理之上，因此， 为了实现跨部门、跨服务的数据集成和共享，主数据的标准化管理应当先于其他维度 的管理。主数据标准包括管理标准、业务标准和模型标准。

准：广泛意义上的管理标准是指对企业标准化领域中需要协调统一的管理 事项所制定的标准。针对主数据管理来说，管理标准应当包括主数据获取和组织的方 法、程序和规程，以及涉及主数据获取、维护、使用的各个部门的工作内容、职责范 围、工作程序、工作方法、工作质量、考核奖惩办法等。

业务标准：业务标准是对主数据业务含义的统一解释及要求，包括来源、管理级 别、业务环境相关的描述及含义解释、主数据相关的业务规则等。编码规则、分类规 则、描述规则都属于主数据的业务标准。

模型标准；和传统数据库类似，主数据的模型标准包括逻辑模型和物理模型。其 中，逻辑模型主要完成从业务场景、业务概念、业务规则到主数据实体、模式、属性、 规则的逻辑层映射；物理模型则负责制定主数据的实际数据结构、存储结构、存储方 式、读/写模式、底层数据库选取等物理层设计。

**2.** **质量管理**

广义来说，数据质量的维度不仅包括准确性和完整性，还包括一致性、可控性、 及时性、最小性等。但我们为了使论述更清晰，在这里采用了狭义的数据质量定义， 即准确性和完整性。

因为主数据是基础的、核心的重要数据，所以保证其准确性和完整性非常重要。 为了保证主数据质量，首先要遵循严格的标准化管理，除此之外，还需要专门针对主 数据质量控制进行统一的规划和监管，保证可实时跟踪和及时反馈。

具体的质量管理手段我们将在第6章“数据质量管理”中讨论，这些手段对主数 据的质量管理会有所帮助，在此不做赘述。

**3.** **安全性管理**

数据安全性管理包括制度安全、存储安全、运算安全、传输安全、服务安全、隐 私保护等方面。对于主数据来说，安全性管理是保证主数据可控性的一个重要的手段。 通过对数据安全的多个维度做严格把关，我们可以掌控数据怎样存储、运算、传输、 使用，也可以约束什么样的人、在什么时间、通过什么方式、获取怎样的数据。在本 书的第9章我们会专门讨论如何保证数据安全，相关细节在此不做赘述。

**主数据管理系统**

为了解决众多数据问题，许多组织正在使用或考虑使用主数据管理系统。主数据 管理系统为企业数据创建单一的事实来源，帮助打破数据孤岛，改善大数据管理。构 造精良、适合应用场景的主数据管理系统是实现高效主数据管理的基础。

主数据管理大致可以分为三类：

大数据治理： 理论与方法…………………

(1)主数据管理作为某个套装软件的一部分，为套装软件的其他模块提供服务。 在这种方案中，由于主数据管理作为附属于整个软件的一个组件，其功能的完备性往 往取决于套装软件本身所面向的需求。如果套装软件只面向某种特定的应用场景，那 么主数据管理的功能往往也不够完善。

(2)主数据管理主要面向分析应用。这种类型和上面的类型相似，其完整性受制 于所面向的应用。

(3)主数据管理本身自成一套独立、完整的解决方案。这种方案突出了主数据本 身的重要性，因此从主数据管理的角度，其整体架构和功能方面的完整性和先进性都 比前两种方案要好。

不同类型适用不同的管理方式和主数据，其实现和使用方式也各不相同。选择主 数据管理方案时，企业或组织可能有数十种不同的供应商可供选择。有些是纯粹提供 服务的初创公司，仅提供主数据管理或类似的服务；有些是规模较大的公司，提供各 种数据管理和集成工具；还有一些是大型的企业软件供应商，能够提供更广泛的产品 组合。举例来说， Informatica MDM 、Semarchy xDM 、PiLog MDRM 、SAP NetWeaver MDM 、IBM InfoSphere Master Data Management 、Oracle Data Relationship Management 等 都属于较为知名的主数据管理系统。

由于主数据管理的需求广泛存在于各类企业中，而各类企业的需求并不相同，所 以，和其他软件系统一样，主数据管理系统的架构也可以分为多种类型。基于企业的 不同层面的需求，我们可以对主数据管理系统的架构从多个角度进行分类。例如，根 据技术环境，我们可以将主数据管理系统分为集中式的主数据管理系统和分布式的主 数据管理系统；根据企业整体系统规模及主数据组件和系统中其他组件的关系，我们 可以将主数据管理系统分为本地模式、联邦模式和混合模式；根据主数据获取和管理 的策略，可以将主数据管理分为整合、协调和下发等模式。

**4.2.1** **主数据管理系统的架构设计**

如前文所述，主数据管理的目的是对企业中的那些重要、稳定、共享需求高的数 据进行统一管理。因此，在企业的系统中往往需要设计专门的主数据管理组件，用于 提供主数据相关的专门服务。在逻辑功能上，主数据管理系统在整个系统生态中处于 核心地位，其功能是将正确的、完整的、 一致的主数据及时同步分发到各子系统中， 如图4-2所示。

从系统的分层 (layer) 视图上来看主数据管理，可以认为主数据管理系统位于应 用系统层和业务层之间，起协调承接作用，如图4-3所示。

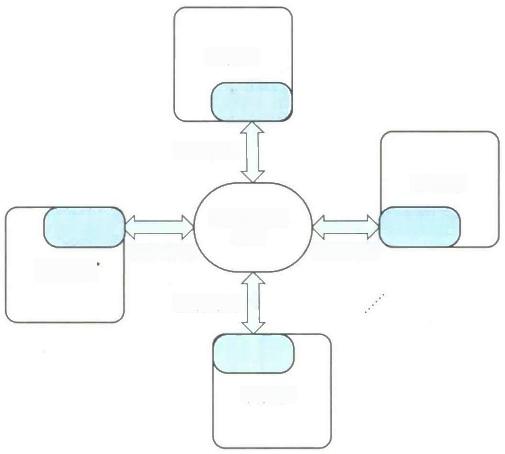
在物理上，主数据管理既可以采用集中式的管理，也可以采用分布式的管理。集 中式主数据管理使用全局唯一的主数据存储，分布式主数据管理则使用超过一个主数 据存储。我们以本地模式、联邦模式和混合模式这三种主数据管理系统架构模式来进 一步说明。

**1.** **本地模式**

本地模式的基本结构如图4-4所示，这是三种模式中最简单的模式。通常，在企业 规模较小、系统组成较为简单的情况下可以直接使用本地模式，其只需要单独维护一

**第4章**| **主数据管理**

份主数据存储，并进行相应的管理即可。在非常简单的情况下，企业可以只维护一张 或数张主数据表。



子系统1

主数据1

收集/分发

子系统n

主数据n 收集/分发’

子系统2

收集/分发

主数据3

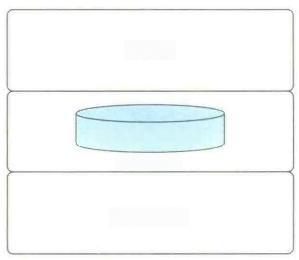
子系统3

主数据管理 系统

收集/分发

主数据2

图4 - 2 主数据管理系统的核心地位

业务层

**主数据管理系统**

数据源层

应用系统层

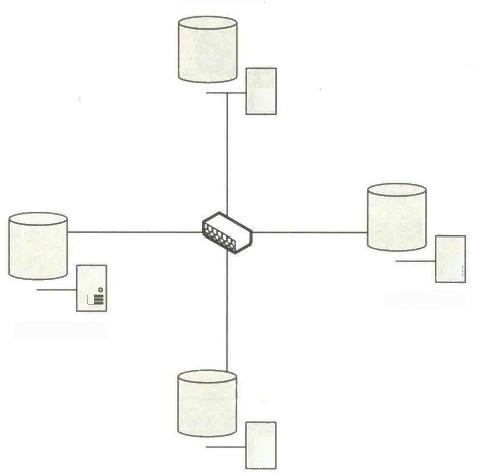
图4- 3 主数据管理系统所处的层

**2.联邦模式**

联邦模式的基本结构如图4-5所示，其仍然设计了全局唯一的主数据管理，但在整 体结构上借鉴使用了联邦数据库的结构，也就是说，允许不同的部门和系统使用异构 的数据管理系统。这种模式适合于稍大规模的企业，在这种企业中，主数据管理未必 是一开始就被考虑到的，各个部门的系统之间可能也存在着固有的异构性。消除这种 异构性代价很高且在很多情况下并无必要。 一种更理性的方案是，保留这种异构性， 将不同的异构系统当成不同的“联邦”个体。主数据管理也仅作为“联邦”个体之 一 ， 维 护 全 局 一 致 的 主 数 据 ， 并 提 供 适 当 的 主 数 据 服 务 。 因 此 ， 联 邦 数 据 库 在 较 高 程 度上保留了复杂性的同时，也带来了较大的灵活性提升。这种模式让一些企业实施迭

大数据治理： 理论与方法……

代渐进的主数据管理成为可能 企业可能在各个子系统都发展到一定规模之后才开 始重视主数据管理，此时可以尝试使用联邦模式，在增加主数据管理功能的同时，尽 量少地影响其他子系统。



企业数据仓库

三

三 主数据管理

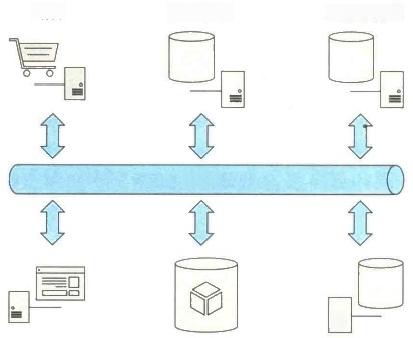
企业资源计划

三



客户关系管理

图4-4 本地模式

活动数据

销售

主数据管理

总线

=

OLAP

人力资源

审计数据

图4-5 联邦模式

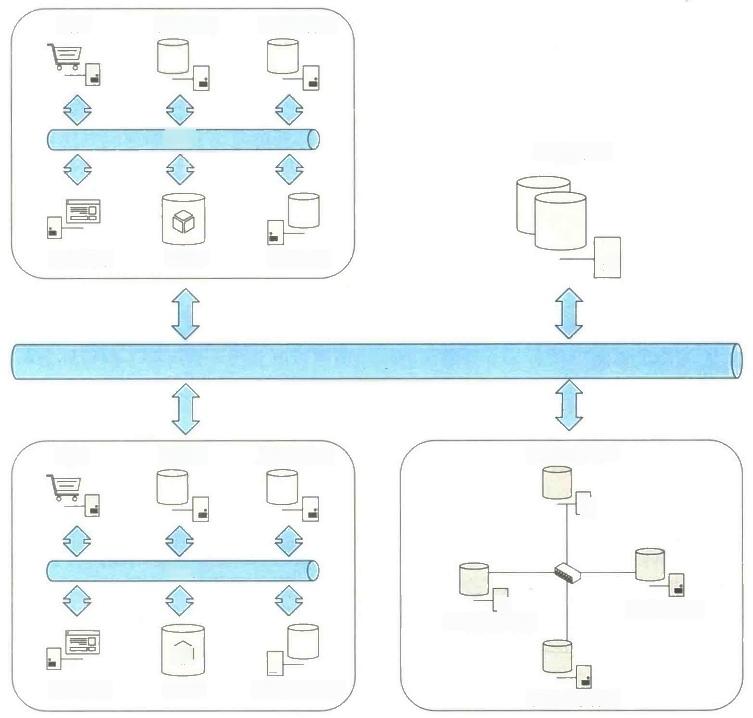
**3.** **混合模式**

在混合模式中，前两种模式可以共存，如图4-6所示。不过，这种模式并不假设存

48

**第4章** | 主数据管理

在全局唯一的主数据管理，而允许存在多个主数据管理组件或子系统。这种模式被设 计出来应付更复杂的情况，如果企业大到了一定程度，或不同的企业之间存在共享主 数据的需求，那么就可以选择这种模式。 一些主数据管理组件和子系统可能是以前的 系统中遗留下来的，我们不必也很难完全从系统中移除它们。更理性的策略是，我们 增加上层的主数据管理，这些主数据可以包含子系统中的全部主数据，也可以只按共 享的需求包含一个子集。这种模式提供了更高的自由性，因为我们不再强制全局唯一 的主数据管理，而允许按照具体需求，定制层次化的主数据管理。这种模式也可以避 免主数据成为整个系统的性能瓶颈 当对主数据的访问特别频繁时，全局唯一的主 数据管理可能难以应付高频的访问请求。但是，任何特性的提高都是要付出代价的。 我们可以认为联邦模式是一种特殊的混合模式，但为了更清楚地说明问题，我们仍将 它们区别开来讨论 。



销售 活动数据 主数据管理

主数据管理

审计数据

总线

企业数据仓库



总线

主数据管理

企业资源计划



客户关系管理，

客户关系管理

主数据管理

人力资源

人力资源

元数据

销售

OLAP

总线

OLAP



三



图4-6 混合模式

**4.2.2** **主数据管理系统的核心功能**

设计主数据管理系统解决方案时，不仅要考虑主数据管理的核心组件如何设计，

大数据治理： 理论与方法

还需要考虑其他要与之交互的重要组件，如元数据管理、客户关系管理、数据仓库等。 为了能够更有效地管理主数据和为重要组件提供相应的主数据服务，主数据管理系统 必须包含以下功能。

模型管理：该模块负责定义实体和属性，并维护实体、属性之间的关系和层次， 包括实体模型、属性模型、业务规则、存储模型等关于主数据逻辑模型和物理模型方 面的设计、管理和更新。例如，如果使用传统的关系数据库来存储主数据，那么主数 据库包括的表格、每张表格的数据模式、表格之间的外键关系、相同或不同表格之间 的属性关系都属于需要管理的范畴。

数据整合和维护：该模块负责主数据相关的增、删、改、查的业务逻辑，包括主 数据收集、清洗、融合、审核、校验等功能。在这个模块中，数据质量管理是非常重 要的一个环节。当从各个业务系统中收集到主数据的原始数据之后，首先要对这些原 始数据进行清洗、融合和验证。例如，如果重要客户的基本信息是主数据，那么有几 个步骤是不可或缺的：

(1)需要对数据进行清洗，保证其姓名、电话、地址、邮编等信息的正确和完整；

(2)如果对从不同渠道收集到了同一个客户的多份信息，还需要对这些信息进行 融合，包括客户的识别、冲突值的消解等工作；

(3)在得到了完整一致的客户信息之后，还需要周期性地对客户信息进行验证， 确保信息不会过时失效。

数据服务：该模块负责向所有相关业务系统提供其需要的主数据服务，包括主数 据查询、分发、订阅等。各个业务系统需要像主数据管理系统请求各种类型的服务， 如销售部门可能会请求客户信息主数据；人力资源部门可能会请求员工信息主数据； 子系统在创建和更新数据时可能还会请求配置型主数据，如国家、民族、性别等信息 的标准表达。这些服务接口需要被良好定义，并能够尽可能适应业务系统中可能发生 的各类变化。

系统管理：该模块提供基本的安全管理、系统检测、报表分析等功能，包括用户 权限管理、参数设置、系统日志管理、系统状态监控等。这部分和大部分的系统管理 的功能基本类似。

**4.2.3** **主数据管理系统的实现风格**

主数据管理系统的实现风格有很多，针对具体的业务场景和使用模式，会有不同 的实现风格。本节选取其中具有代表性的4种详细讲解：合并风格 (Consolidation Style) 、登记风格 (Registry Style)、共存风格 (Coexistence Style)、集中/事务风格 (Centralized/Transaction Style)。

**1.** **合并风格**

这种风格非常适用于分析型主数据管理。顾名思义，合并风格是将不同来源的主 数据经过转换、清理、匹配和集成后，合并为一致、准确、完整的唯一视图。得到的 唯一视图中的记录被称为黄金记录 (Golden Record)。这种风格从不同来源获取主数 据，经过处理之后，将黄金记录存储于中央的主数据hub 中。中央的主数据 hub 将存 储的黄金记录提供给下游的报表或分析应用，或者作为其他操作的参考点。关于主数

 50 

**第4章** |主数据管理

据的任何更新都会发布给主数据的来源。

这种风格的优点在于，其允许企业或组织从现有的系统中提取主数据，并进行中 央集成式管理。企业或组织可以从许多现有系统中提取主数据并将其进一步引流到单 个可控的主数据 hub 中。合并后的主数据 hub 负责管理主数据，并快速、低开销地提 供业务范围报告和分析。不过这种风格主要用于分析型应用，目的是提供值得信赖的 报告和分析数据源。

**2.** **登记风格**

这种风格假设提供主数据的数据源，能保证其数据质量，其主要通过对来自各种 源系统的数据运行清洗和匹配算法来发现匹配的重复记录，通过为匹配的重复记录分 配唯一的全局标识符 (identity), 该风格的主数据管理可以识别主数据的唯一版本。主 数据管理系统只管理最小量的信息，用来唯一标识主数据，并提供在其他系统和数据 库中详细数据的引用。

这种风格不会将数据发送回源系统，因此对主数据的更改将继续通过现有的源系 统进行。其只负责清除或匹配主数据标识的引用信息，并假定源系统提供的数据不存 在数据质量问题。

主数据管理系统存储了主数据的匹配结果和标识符，并提供引用链接信息，用于 访问详细数据。这种风格提供了低成本、高效、 一定程度实时的主数据集成，同时有 助于解决某些复杂环境中的应用。注册表样式提供了一个只读型数据视图，而无须修 改主数据，这是一种保持对主数据的一致访问的有用方法。但是，因为其对于数据质 量的管理完全依赖于源系统，所以其对于对现有系统的性能和可用性特别敏感。同时， 还需要严格管理主数据管理系统和源系统之间的数据交换，以防止互相污染。

请注意，登记风格和合并风格的区别在于，合并风格真正地存储了黄金记录，但 登记风格只存储了标识符，因此相对更加轻量级。如果企业或组织拥有大量的源系统， 就可以使用注册表样式的方法来分析主数据，同时避免在源系统中覆盖信息的风险。

**3.** **共存风格**

共存风格允许通过与合并风格相同的方式构造黄金记录，但是主数据存储在中央 主数据管理系统中，并在其源系统中进行更新。顾名思义，共存风格允许在多处创建 和存储主数据。其与合并风格类似，但部署成本可能比合并风格高，因为主数据更改 可能发生在多处，即在主数据管理系统及应用程序系统中均可能发生。将主数据模型 的所有属性上传到主数据管理系统之前，必须经过数据清洗，以保持一致和完整。

这种风格的主要优点是，数据在源系统中被控制，然后与主数据 hub 同步，因此 数据可以在和谐共存的同时仍然提供真实的单一版本。除此之外，还可以提高主数据 的质量，并且令访问速度更快。由于主数据单一存储，因此形成分析报告也更加容易。

**4.** **集中/事务风格**

集中/事务风格使用链接、清洗、匹配和丰富算法来存储和维护主数据属性，以管 理主数据。可以将丰富后的主数据发布回其各自的源系统。主数据 hub 支持主数据记 录的合并，并且源系统可以订阅 (subscribe) 由中央主数据管理系统发布的主数据更 新，通过从服务接口获取的数据来更新主数据，从而保证完全的一致性。但是，为了

大数据治理： 理论与方法 ·

保持顺利的双向交互，这种风格要求主数据管理系统能够一定程度上操纵源系统，这 可能会增加系统设计的复杂性。

这种风格的优点在于，其维护的主数据始终准确且完整，中心主数据管理系统可 以支持数据属性级别的安全性和可见性策略。这种风格对协同型、操作型和分析型的 使用模式均适用，如果有必要，也可以从合并风格或者共存风格演变到集中/事务风 格。不过，由于其能够支持更丰富的应用场景，这种风格往往也会带来高成本和高复 杂度，其甚至可能要求调整现存应用、商业流程或机构结构。

4.3 **主数据管理的成熟度**

对企业来说，有必要提供主数据全生命周期管理。这涉及如下几个关键步骤：主 数据建模、主数据建立及共享、主数据维护、副本监管、系统管理。

主数据建模：从业务、物理、UI 、模板角度创建主数据模型，并为主数据制定标 准的编码规则。

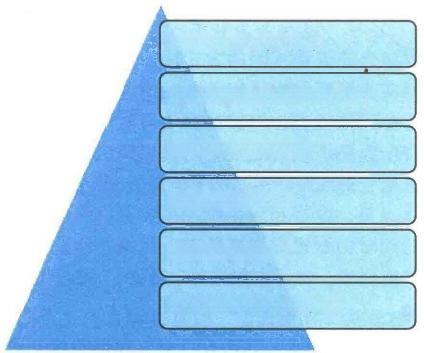
主数据建立及共享：设置加载、分发的数据权限，对对象、属性、记录等三种级 别配置数据权限。

主数据维护：基础维护(增加、删除、修改、查询、详细、下载、封存、解封)、 版本管理(查看、回退、对比)、审批流程定义执行、查重及合并、主数据日志、数据 溯源等。

副本监管：副本模型的建立和维护，包括数据转换、消息处理、副本日志管理、 副本分发、副本监控等。

系统管理：支持系统服务注册、业务系统注册、系统参数设置、多数据库等。

根据上述几个步骤实施的复杂程度，参照Jill Dyche 和 Evan Levy 的观点①大体可以把主 数据管理可以分为6个层次，从低到高反映了主数据管理的不同成熟度，如图4-7所示。



Level5: 企业数据集中

Level4: 业务规则和政策支持

Level 3: 集中总线处理

Level2: 同等访问

Level 1: 提供列表

Level0: 没有主数据管理

**图4-7** **主数据管理成熟度**

① The Baseline on Master Data Management:Five Levels of Maturity for Master Data Management.

**第4章** |主数据管理

我们简单介绍一下这6个层次。

**Level 0: 没有主数据管理。**

Level 0 意味着企业的各个应用之间没有任何的数据共享，在不同系统、部门、组 织之间没有建立起关于关键数据的共识。例如， 一个企业的不同分公司销售不同产品， 对这些产品的生产和销售由多个独立的系统来处理，各个系统独立处理产品数据并拥 有自己独立的产品列表，各个系统之间不共享产品数据。每个系统负责管理和维护自 己的关键数据(如产品列表、客户信息等),各个系统间不共享这些信息，这些数据是 不连通的。不存在跨组织、跨系统的统一数据定义。同时，也不存在系统之间的数据 迁移标准，不需要跨系统的数据校准，各系统独立维护自己的数据质量。

**Level** **1:** **提供列表。**

无论公司规模多大，列表管理都是非常常见的一种方式。在公司内部，会维护一 些逻辑或物理的列表。不同系统和用户在需要某些数据时都可以访问这些列表。Level 1在 Level 0的基础上稍微提升了一些主数据管理的成熟度：各个部门虽然还是独立维 护各自的关键数据的，但会通过列表管理维护一个松散的主数据列表，从而向其他各 个部门提供其需要的数据。Level 1 中，数据变更及数据变更操作都是由人来决定的， 因此，只有人完成数据变更决定后才会变更数据。依靠这种方式维护主数据列表在主 数据数量比较少时是可行的，但是当需要维护的主数据超过人力能够维护的规模之后， 列表管理就显得不够了。

如果主数据构成比较复杂，如存在层次或分组，那么很难仅依赖列表进行管理。 单纯使用列表管理存在的问题包括：没有系统和严格的方法来确保更改主数据列表； 定义和维护主列表涉及大量会议和人员参与；如果有数据冲突，删除和更改是手动处 理的；各个应用程序必须了解如何导航到主列表；上述步骤并不总是对外部数据用户 可见的。

**Level 2: 同等访问**

Level 2与 Level 1 相比，引入了对主数据的自动管理。在 Level 2 的主数据管理中， 企业或组织会建立数据标准，定义主数据模型以唯一地标识每个主记录。在Level2 中， 存在一个中央存储库 (Central Repository), 也称为“主数据主机”(Master Data Host), 其是一个数据库或应用系统，为各个系统间在线访问和共享数据提供了严密的支持。 举例来说， 一个销售自动化系统对外部应用提供数据访问功能。当呼叫中心应用作为 外部应用需要增加一个客户时，其会提交一个事务，请求数据所有者增加一个客户条 目。主数据主机将增加数据并告知外部应用。

Level 2引入了“同等访问”(Peer-Based Access),也就是说， 一个应用可以调用 另一个应用来更新或刷新需要的数据。当创建、读取、更新和删除处理规则定义完成 后，还需要保证每个应用“同等”理解基本的业务规则，以便访问中央存储库并与之 进行交互。因此，每个同等应用必须正确恰当地创建、增加、更新和删除数据。Level 2中存在数据访问和完整性规则，但必须由各个应用程序系统进行管理。集成新的应用 程序系统通常需要将所有数据和处理完整性逻辑复制到每个使用的应用程序中。在 Level1 中，数据变更是基于手工方式的。在 Level 2,数据变更是自动完成的——通过 由具体技术实现标准流程，允许多应用系统修改数据。Level 2可以支持使用不同的应

大数据治理： 理论与方法………………

用和变更单一、共享的数据知识库。在 Level 2中，由于需要支持同等访问，其要求各 个拥有(部分拥有)主数据的子系统之间具有良好的互操作性，以便支持其彼此访问 和更新数据。

Level 3: 集中总线处理。

Level 3较 Level 2的升级之处在于，其打破了各个独立应用的组织边界，使用各个 系统都能接受的数据标准统一建立和维护主数据，而Level 2的主数据还是按照各个系 统分开存储的，没有真正整合在一起。

集中处理意味着为主数据管理构建了一个通用的、基于目标的平台。随着企业或 组织规模的增大，缺乏完善的主数据管理会导致其系统架构面临极大挑战，每个子系 统独立处理主数据将导致数据规则和数据访问的复杂性急剧上升。Level 3 的主数据管 理就提供集中数据访问，并对跨应用和系统的数据使用进行了控制。Level 3 的管理可 以使得应用访问数据的复杂性降低，与分散环境相比，主数据管理有更多的可控性： 首先，面向业务的数据规则和相关的处理在Level3 中被集中化了；其次，存在明确的 主数据管理者，Level3 提供专门的数据管理过程并负责解决数据冲突；再次，数据清 洗和匹配是集中化的，对于应用程序来说是透明的；最后，数据的准确性和一致性由 集中的主数据hub 来保证。

可以看出，Level 3 所采用的集中式主数据管理降低了分散情况下各个子系统的耦 合度，与Level2 相比，子系统之间不需要较高的互操作性。从主数据的角度来看，各 个子系统都只需要和中心化的主数据管理系统交互，彼此之前不再有强烈的同等访问 需求。中心化的主数据管理通过一个公共的平台作为hub, 让各个子系统在主数据的定 义和管理上都达成共识，并以标准化的方法转换异构数据，对数据的共识打破了各个 独立应用的组织边界。

不过，集中化的主数据管理要求中心主数据定义和管理的共识要经过更仔细的设 计，需要了解各个子系统，以保证中心主数据管理系统相对稳定，否则一旦需要修改 中心主数据管理系统，就可能会影响大面积的子系统。同样，在设计不合理时，子系 统的(可能是完全合理的)升级或改动也会影响中心主数据的管理，进而影响其他子 系统。

Level 4: 业务规则和政策支持

在 Level 3中，已经实现主数据的集中管理，但是，如前文所述，中心主数据管理 在从不同的子系统中收集到主数据后，对其进行清洗和匹配，形成全局唯一的单一主 数据版本。然而，由于主数据在业务和决策中往往扮演着极其重要的角色，因此，主 数据的一致性、准确性、完整性、可控性需要被特别关注。如何保证数据是正确的? 这在前三个层次中并没有被明确考虑。Level 4增加了数据质量层面的考量。Level 4对 主数据hub 和其他外部系统进行数据质量检查。

在公司组织比较复杂时，影响业务数据访问和操作的规则 (rule) 及策略 (policy) 相对也比较复杂。主数据管理需要制定更加成熟的变更管理流程，涉及数据和业务 流程：

(1)系统应当支持集中式的规则应用和分布式的规则处理，换言之，不需要保证 所有的规则都在主数据 hub 中，但需要保证规则是集中应用的；

**第** **4** **章**丨主数据管理

(2)在中心主数据管理组件或系统内进行数据清洗和匹配；

(3)在系统架构上最好支持 SOA, 因为 SOA 非常适合跨应用程序服务和数据 交换；

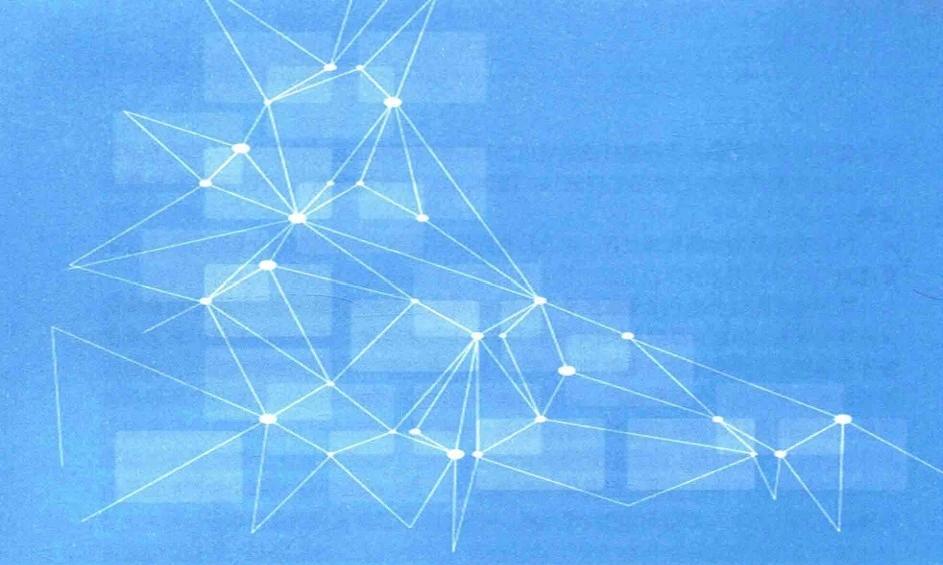
(4)有成熟的数据管理过程，以支持将数据问题传达给责任方(无论是个人还是 系统)。

需要注意的一点是，完全指望单一系统完成整个企业范围内的主数据管理和质量 保证并不现实。因此，虽然 Level4 假定了中心主数据管理，其还是需要辅以整个企业 范围内的相关工作流整合。举例来说，在一套产品的整个生产和销售流程中，不同的 部门负责不同的生产和销售流程，主数据虽然可以唯一表示某类甚至某个产品的基本 信息，但不同部门的业务规则是各不相同的。如何将这些结构和语义异构的规则整合 起来是非常关键的问题。主数据管理需要保证当一个产品的关键信息需要被查询或者 更新时能够提供正确的结果，但具体如何保证这一点，还需要各个部门的业务流程作 为辅助。同时，为了支持部门的合理调整，对规则和策略扩展性的支持也必不可少。

**Level 5: 企业数据集中。**

在 Level 5中，总线和相关的主数据被集成到独立的应用中。主数据和应用数据之 间没有明显的分隔。当主数据记录的详细资料被修改后，所有应用的相关数据元素都 将被更新。这意味着对于所有的应用或子系统来说，其访问的是相同的数据实例。所 有的应用系统通过统一管理的主数据集成在一起。应用系统和主数据管理的内容是同 步的，所以当变更发生时，应用系统都将更新。Level 5提供一个集成的、同步的架构， 当一个有权限的系统更新一个数据时，公司内所有的系统将同步这次更新。系统更新 完数据后，不必单独通知其他系统做相应值的更新，主数据管理将使这种更新变得 透明。

Level 5 和 Level 4 的不同之处在于，主数据管理功能不需要源系统做专门的开发或 支持。所有的应用都清楚地知道它们对主数据仅有使用权，而没有所有权或控制权。 软件架构在设计和实现时应保证所有的应用可以访问主数据。所有应用对主数据的访 问都是透明的，主数据在一处更新后，会被同步广播到所有的应用程序中，以保持一 致性。例如， 一个客户在他的个人信息页面中更新了电话，这个更新会被广播到所有 涉及客户电话的系统中，并保持一致。Level 5将数据作为一种服务，提供统一的数据 定义、授权使用和变更传播方式。



**第** **5** **章** **数据集成**

“数据”一词由来已久。在日常的生活和工作中，无论是外部环境的客观运行状 态，还是人类主观的生理和心理感受，只要是能被记录下来的就是数据，其内容形式 包括数字、图片、时间、矢量坐标等结构化或非结构化的数据。数据是人类对世界最 忠实、原始的记录。

不过，未被加工的原始数据往往不能用来解释任何问题。而且，数据如果彼此孤 立，并未进行有效的关联，那么也难以全面覆盖要研究的问题的各个方面。当前，人 类探索客观世界运行规律的需求日益强烈，这些记录了各类信号的数据成为人类开展 分析的原始材料。通过对数据进行处理，人类可以建立数据之间的联系，从数据中对 比、总结出规律性的结论，这些结论被称为“信息”。在从数据中获取信息的基础上， 可以选择那些能够积极指导任务执行和管理的信息，借助它们解决问题或是做出决策， 这些信息被称为“知识”。从数据到信息再到知识的转化链反映了人类记录世界、认识 世界、探索世界的过程。

近几十年来，科学技术的迅猛发展和信息化的推进，使得人类社会所积累的数据 量已经超过了过去5000年的总和，被采集、存储、处理和传播的数据量也与日俱增。 许多企业或组织机构在管理运作中积累了大量数据，包括业务运作、客户、产品和人 员相关的数据等。因为数据和信息对运营和管理等方面非常重要，所以大量的信息管 理系统被广泛应用于各个部门。这些业务子系统提高了局部的效率和应用性。然而， 在更高的层面上， 一类时常发生的情况是，这些业务子系统是彼此分离的。它们采用 了不同的数据库系统，或使用了不同的开发技术。各系统的业务逻辑不同，在数据库

**第5章** |数据集成

的建设上也没有遵循统一的标准。随着时间的推移，每个部门都成为一个独立的数据 源，它们各自独立地累积了一批格式和语义不同的异构数据，使得这些独立的数据源 构成了一个巨大的异构数据环境。在这样的异构数据环境下，那些对某些部门的管理 决策者有重要意义的数据往往在产生时就混杂于其他数据中，因此难以直接被获得和 利用。

企业实现数据共享，可以使更多的人更充分地使用已有数据资源，减少资料收集、 数据采集的劳动和花费。然而，在实施数据共享的过程当中，由于不同用户提供的数 据来源可能不同，其数据的内容、格式和质量千差万别，有时会遇到数据格式不能转 换或格式转换后丢失信息等棘手问题，严重阻碍了数据在各个部门和各个系统中的流 动与共享。这种分散性和不兼容性，以及部门间的障碍，共同阻碍了对信息的进一步 整合。为了打破障碍，使信息可以在整个企业和机构范围内以协调一致的方式收集， 整理、分析和共享，建立行之有效的数据集成管理机制已成为增强企业商业竞争力的 必然选择。

此外，由于现代企业的飞速发展，可以毫不夸张地说：“数据是业务的命脉。”在 数字业务时代，如何最有效利用数据是一个非常关键的问题。企业逐渐从一个孤立节 点发展成不断与网络交换信息并进行商务事务的实体，企业数据交换也从企业内部走 向了企业之间。越来越多的数据都必须及时地进入企业系统，进行存储并投入使用， 以实现业务战略。同时，数据中存在着大量的不确定性和易变性，系统在实现技术和 物理数据上也存在着紧耦合关系，这些因素使得一旦上层应用或底层物理数据发生变 化，整个体系将不得不随之修改。

如前文所述，进行数据集成将面临许多问题，包括如何适应现代社会发展的复杂 性、如何有效扩展应用领域、如何分离应用需求和实现技术、如何准确描述各种数据 源格式，以及如何有效发布和交换数据等。本章的后续部分将对现有的数据集成技术 进行探讨，并简要讨论相关问题的发展现状。

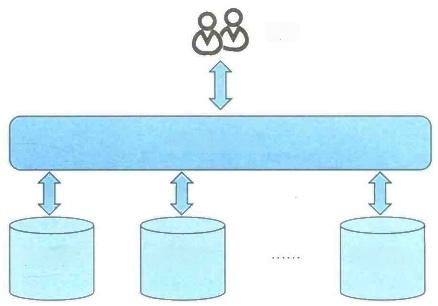
5.1 **数据集成的基本概念**

**5.1.1** **数据集成的定义**

数据集成就是将若干个分散的数据源中的数据在逻辑或物理层面集成到统一的数 据集合中。具体来说，就是将不同来源、格式、性质的数据在逻辑或物理层面上有机 地集成，通过一种一致的、精确的、可用的表示法，整合描述同一现实实体的不同数 据，进而提供全面的数据共享，并经过数据分析、挖掘，产生有价值的信息。

数据集成的核心任务是要将互相关联的分布式异构数据源整合到一起，使用户能 够以透明的方式访问这些数据源。以上表述中，整合是指维护数据源整体上的一致性、 提高信息共享利用的效率；透明的方式是指用户可以通过统一的方式完成对异构数据 源的访问，而无须关心具体的底层实现。实现数据集成的系统称为数据集成系统(见 图5-1),它为用户提供统一的数据源访问接口，执行用户对数据源的访问请求。

**大数据治理：** 理论与方法

用户

**数据集成系统**

数据源2

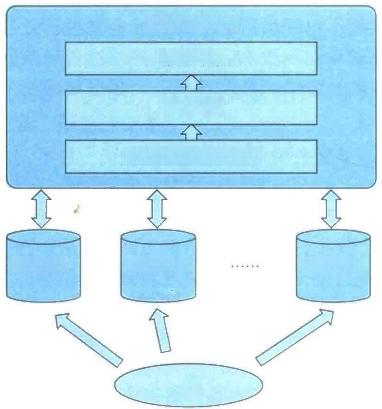
数据源1

数据源n

图5-1 数据集成系统

**5.1.2** **数据集成的分类**

根据数据集成方式的不同，其可以分为传统数据集成和跨界数据集成，如图5-2所 示。传统数据集成通过模式匹配、数据映射，冗余检测，数据合并等技术，通过统一 模式访问将多个数据源集成起来。传统数据集成方式在商务智能等领域得到了广泛的 应用，对于大数据而言，传统数据集成仍然发挥着作用，我们将在5.2节介绍传统数 据集成的概念，在5.3节介绍其关键技术。



数据集成系统

数据合并

冗余检测

模式匹配、数据映射

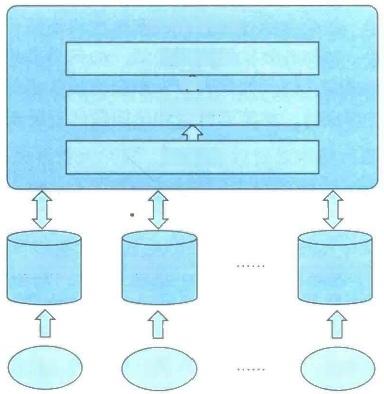
数据源2

域

数据源1

数据源n

(a)传统数据集成



数据集成系统

知识验证

分

知识融合

知识抽取

数据源2

域1 域2 域n

数据源n

数据源1

(b)跨界数据集成

图5-2 传统数据集成和跨界数据集成

在大数据时代，不同领域产生的多个数据集隐含地与某些对象之间存在关联性。 例如，尽管一个地区的气象数据和交通数据来自不同的数据域，但是这些数据会显示 这个地区的某些相关状况，这时候，对来自不同数据域的数据集成很难简单地通过模

58

第5章|数据集成

式匹配、数据映射来实现，而需要用不同的方法从每个数据集中提取信息(通常是知 识),然后把从不同数据集提取的知识有机融合在一起，从而感知这一区域的有效信 息。除模式映射之外，还涉及多种知识融合的方法，这与传统数据集成有很大的不同， 面向这类需求的数据集成称为跨界数据集成。我们将在5.4节介绍跨界数据集成。

我们用两个例子来说明二者的异同。

一个公司想对三个不同子公司的同一年的客户订单进行集成，假设公司对于订单 数据有相对统一的规范、不同子公司的客户和产品也有一定的重合度或相似度，那么 这些订单信息可以视为来自同一个域、但模式和内容有所不同的三个数据集。数据集 成系统首先需要将不同数据集的模式通过模式映射统一为一致的数据模式，接着通过 冗余检测发现不同数据集的重叠，最后根据前两步的结果对数据做正确的合并。这样 的过程就是传统数据集成的过程。

有组织想集成三个不同的电商同一年的客户信息，假设不同电商主营领域不同、 客户信息的表示方式不同，同一个客户在不同电商平台使用的账户也不同，那么这三 个电商平台的数据可以认为是来自不同域的，传统的模式匹配可能不足以将其很好地 对应起来， 一种比较常见的做法是从这些数据集中抽取知识，再通过知识融合和验证， 将数据或知识在更高的维度上集成起来。

**数据集成的难点**

数据集成的数据源狭义上主要指关系型数据库，广义上也包括各类 XML 文档、 HTML文档、知识图谱、电子邮件、普通文件等结构化、半结构化、非结构化的数据 源。数据集成是信息系统集成的基础和关键。如我们在上一章中讨论的，成熟的主数 据管理是离不开有效的数据集成的。好的数据集成系统要保证用户以低代价、高效率 集成使用异构的数据。要实现这个目标，必须解决数据集成中的一些难题。

数据集成的难点可以归纳为以下方面。

**1.** **异构性**

被集成的数据源通常是独立开发的，各数据源固有的异构性给集成带来很大困难。 这些异构性主要表现在语法异构和语义异构上。

语法异构一般指源数据和目的数据之间命名规则及数据类型上的异构性。如果数 据源是关系型数据库，那么命名规则通常指模式名(表名)和属性名(字段名)。举 例来说，存储员工信息的表可能叫“员工表”,也可能叫“人员表”。在不同的表中， “姓名”可能作为一个属性存在，也可能分成“姓”和“名”作为两个不同的属性存 在；“出生日期”可能是一个日期类型的属性，也可能被拆分为多个数值型的属性。语法 异构解决起来相对简单，其无须关心每条记录的内容和含义，只要对数据模式有了解， 能够找到表、属性的恰当映射，就可以把不同表中的数据统一起来。这种映射关系通常 数量较少，很容易找全。不过，在实际应用中也存在着一些不常见或不易被发现的语法 异构，如某些领域相关的数据约束。举例来说，在我国，“邮编”这个属性独立决定“城 市名”,如果在集成的时候不注意，就可能会出现邮编相同，城市名却不同的情况。

语义异构很难单纯从分析数据结构入手，其涉及对数据内容和含义的理解。例如， 两条来源不同的记录的“姓名”属性相同，但未必指代同一个人。如果不加判定就盲

大数据治理： 理论与方法………………………………

目融合，很可能由于融合错误而为后续的应用带来困难。语义异构的处理难度要比语 法异构大得多，其需要对数据含义做进一步分析，而且经常需要额外的领域知识作为 辅助。解决语义异构问题在数据集成中是一个重要又困难的问题。

**2.** **分布性**

数据源往往是异地分布的，数据集成的过程依赖于网络传输，如何保证传输过程 的性能和安全性也是集成过程的难点。这其中有一些问题需要考虑。

能不能传?出于某些安全性的考虑，并非所有的数据都适合通过网络进行传输， 因此，如果需要使用网络传输，首先需要考虑的就是这些数据是否适合被网络传输。

传哪些数据?在确定了是否可以传输之后，接下来要考虑的问题是哪些数据需要 通过网络传输，并非所有数据都需要参与数据集成过程。例如，如果只需要简单的模 式映射就可以完成数据集成，那就没有必要将整个数据库都通过网络传输。在批处理 数据集成中，源系统往往会周期性地(每天、每周或每年)传输一批数据到目标系统， 目标系统对数据采用集成所需的必要操作并进行存储；而在实时数据集成中，由于其 面向的应用对处理速度的要求极高，所以一次只会传输少量数据，通常称为“消息”, 多个系统通过互传“消息”来完成数据集成任务。

传输速度能否满足要求?如果传输的数据量太大，还需要考虑传输速度是否能够 满足要求，在数据量大的情况下，可能有一些加速的方法，例如，如果数据有多个副 本，就可以考虑是否有可能将整个数据集合拆解进行传输。

**3.** **自治性**

在一些应用中，不同的数据源有很强的自治性，数据源的拥有者有增加、修改、 删除数据的权利，其可以直接在本地修改数据而不通知数据集成系统。如果修改影响 到了某些关键记录，或修改了关键数据模式的定义(比如将“电话号码”属性拆分为 “固话”和“手机”),那么就有可能导致数据集成出错。同时，增加新的数据可能引 入某些原集成系统中没有考虑过的新的数据约束。这些都会给数据集成系统的鲁棒性 带来挑战。

5.2 传统数据集成

传统数据集成的主要目的是数据共享。如前文所述，数据可能来自不同的数据源 系统，但均来自同一个域，传统数据集成最重要的一点就是将来自不同数据源系统的 数据统一匹配到目标模式下。

形式化地，令1表示数据集成系统，则I=<G,S,M>, 其中G是全局模式， S 是 数据源模式， M 为全局模式和数据源模式之间的映射。在数据集成中，在一些情况下 数据经过转换后可以统一存储。但在另一些情况下，全局模式只是一种“虚拟模式”, 系统仅需要记录全局模式和源模式的映射关系，当查询请求出现时，需要通过重写查 询将针对目标模式的查询请求转化为针对源模式的查询请求，发送到源系统中执行， 再将结果整合返回给用户。

数据集成已经被学术界和企业界研究了很多年，产生了大量的成熟方法，本节将 介绍三种非常重要的方法： 联邦数据库系统、中间件集成和数据仓库。

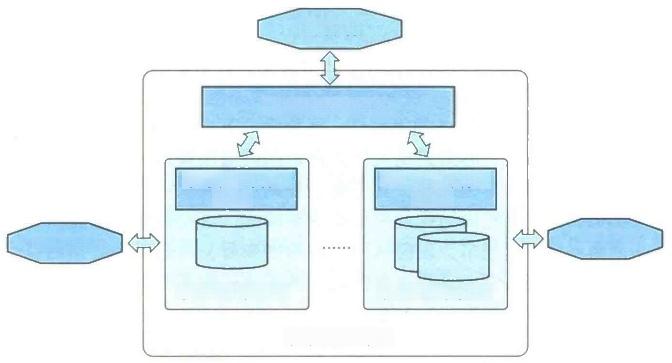
60

**第5章** |数据集成

**5.2.1** **联邦数据库系统**

联邦数据库系统 (Federated Database System,FDBS) 是多数据库系统中的一种。 其概念大约在20世纪80年代提出，主要目标是以透明的方式将多个自治数据库系统映 射到单个联邦数据库中。构成联邦数据库系统的自治数据库系统称为单元数据库系统 (Component Database System,CDBS)。单元数据库系统通过计算机网络互连，并且可以 在地理上分散。联邦数据库系统将单元数据库系统按不同程度进行集成，并从整体上 提供控制和协同操作。

联邦数据库系统的基本结构如图5 - 3所示，在多个单元数据库系统之上，构建 一 个 联邦数据库管理系统，该管理系统提供全局的数据服务，而各个单元数据库系统依然 可以提供本地的数据服务，不受联邦数据库影响。详细来说，单元数据库系统可以是 异构的，其既可以是集中式的，也可以是分布式的。在构建集成系统时，将各数据源 的数据集成为全局模式，使用户能够按照全局模式透明地访问各数据源的数据。全局 模式描述了数据源共享数据的结构、语义及操作等。用户直接在全局模式的基础上提 交数据服务请求，由联邦数据库管理系统将这些请求转换成各个数据源在本地能够执 行的版本。可以说，联邦数据库或虚拟数据库是联邦数据库系统中所有组成数据库的 组合，其只在本地的数据库上层搭了一层联邦数据库管理系统，即只进行了模式集成， 并没有进行实际的数据集成。



全局服务

联邦数据库管理系统

本地数据库管理

系统1

本地服务

单元数据库系统1 单元数据库系统n

联邦数据库系统

本地数据库管理

系统n

本地服务

图5-3 联邦数据库系统的基本结构

在建立联邦数据库系统时，需要明确其控制及操作方式，即需要明确谁管理联邦 数据库系统，以及单元数据库系统如何集成。基于这层考虑，可以将联邦数据库分为 松耦合联邦数据库和紧耦合联邦数据库。

松耦合联邦数据库要求用户创建和维护联邦数据库，对联邦数据库系统不做硬性 限制。松耦合联邦数据库需要在单元数据库之间两两构造模式之间的翻译器，用户通 过特定的多数据库操作语言及自己对联邦数据库的定义、维护来访问和使用数据，在 这种情况下，单元数据库更具有自治性。松耦合联邦数据库提供统 一 的查询语言，将

大数据治理： 理论与方法

很多异构性问题交给用户自己去解决，其对数据的集成度不高，但其数据源的自治性 强、动态性能好，集成系统不需要维护一个全局模式。不过这种方式的问题也很明显： 极端情况下，对于任意数据库系统D, 和D₂, 这种模式都允许D, 以 D₂ 能理解的术语查 询D₂, 所以如果n 个数据库中的每个都需要与其他n-1 个数据库进行交互，则我们需 要定义n(n-1) 个数据接口以支持查询。

紧耦合联邦数据库由联邦数据库系统提供统一的全局模式和数据访问接口，由全 局数据库管理员创建并维护该模式。这种方式的好处是，集成的过程几乎不需要终端 用户参与， 一旦全局模式设计好了，整个系统管理也相对简单易理解。根据全局模式 的数量，可以将紧耦合联邦数据库分为单联邦数据库和多联邦数据库，这也提供了一 定的灵活性。但其缺点也同样明显，首先，由于数据固有的异构性和自治性，自动化 地构建全局模式的算法可能会很复杂；其次，即便复杂的算法可用，我们也很难保证 构建的结果一定是准确的，因此需要领域专家决定或确认数据库模式间的对应关系， 这带来了较大的人力开销；最后，任一数据库一旦发生变化，单靠变化的数据库本地 的修改并不能保证系统的可靠性，还需要重新确认全局模式，这会影响联邦数据库的 可扩展性。

联邦数据库集成的特点是，数据源独立存在，但一个数据源可以访问其他数据源 提供的信息。但正如我们之前看到的，联邦数据库中间有很多设计细节是难以实现的， 需要慎重考虑。

**5.2.2** **中间件集成**

中间件 (Mediator) 在软件架构设计中非常常见。数据集成中的应用和其他场景下 的中间件的应用方式非常类似 中间件位于数据源系统(异构的数据库、遗留系统、 Web资源等)和应用程序(服务调用接口、终端用户等)之间，向下协调各数据源系 统，向上为应用层提供统一的数据模式和数据访问的通用接口。中间件的存在几乎不 会影响各数据源系统提供本地服务，其主要是在高层次提供一个检索服务。和紧耦合 联邦数据库类似，中间件集成同样使用全局数据模式，通过在中间层提供一个统一的 全局模式来隐藏底层数据细节，使得从用户的角度来看，数据层的数据源整个构成了 统一整体。因此，如何构造上层的全局模式并保证底层数据源的本地模式能正确映射 为全局模式，是实现中间件集成的关键。

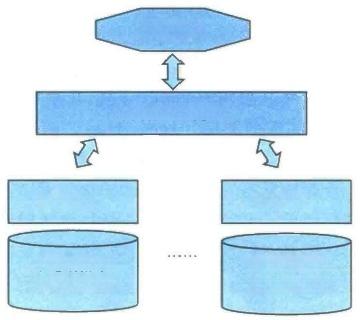
G.Wiederhold 最早给出了中间件集成方法的架构。与联邦数据库相比，中间件数 据集成可以提供更丰富的可能性，其不仅支持集成结构化数据源，也支持集成半结构 化和非结构化数据源。中间件数据集成系统也经历了很长的发展阶段，1994年斯坦福 大学推出的 TSIMMIS 系统就是一个典型的中间件集成系统。

中间件数据集成系统的基本架构如图5-4所示，其核心在于中间件和封装器的设 计。其中，每个数据源都对应一个与之匹配的封装器，作为每个数据源的门面，中间 件通过与封装器交互来访问各个数据源。用户基于全局数据模式向中间件发送查询请 求。中间件收到用户请求之后，将其转换成各个数据源能够处理的子查询请求，并对 此过程进行优化，以提高查询处理性能。封装器对特定数据源进行封装，将其数据模 型转换为系统所采用的通用模型，并提供一致的访问机制。中间件将各个子查询请求

 62

**第5章** |数据集成

发送给封装器，由封装器和其封装的数据源交互，执行子查询请求，并将结果返回给 中间件。



应用程序

中间件(全局模式)

封装器1

·本地数据库1

(本地模式1)

本地数据库n (本地模式n)

封装器n

图5-4 中间件数据集成系统

中间件注重全局查询的处理和优化，其相对于联邦数据库系统的优势在于，能够 集成非数据库形式的数据源，有很好的查询性能，自治性强；但很多中间件集成是只 读的，而不像联邦数据库系统那样对读/写都支持。

**5.2.3** **数据仓库**

数据仓库 (Data Warehouse) 是用于生成报表和数据分析的系统，其被视为商业智 能的核心组成部分。数据仓库是一个中央数据存储库，其集成了来自一个或多个不同 来源的数据。通过将当前和历史数据存储在同一个地方，可以很容易地为整个企业创 建分析报告。

数据仓库中存储的数据是从下层的不同数据源或系统(如市场部门或销售部门) 中获取的。数据在最终存储之前，可能需要一些预先的处理。提取、转换、加载 (Extract-Transform-Load,ETL) 是将数据从一个或多个源转移存储到目标系统的通用过 程，其中目标系统和源系统之间存在着一定差异，表现为数据表示不同，或应用上下 文不同等。ETL 在20世纪70年代成为流行的概念，后被经常用于数据仓库中。

提取： ETL过程的第一个(很多情况下也是最重要的)步骤是从源系统中提取数 据。由于我们之前讨论过的那些多源数据中固有的异构性，是否能正确提取数据很大 程度上决定了后续步骤的正确性。大多数数据仓库项目会整合来自不同源系统的数据， 这些数据可能使用了不同的组织形式，如关系数据库、XML 、JSON、文本文档等。提 取阶段旨在将数据转换为适合处理的单一格式，并验证提取结果的正确性。可以使用 一些规则来验证数据的正确性，如果数据未通过规则的正确性验证，那么可以报告源 系统并进行进一步处理。

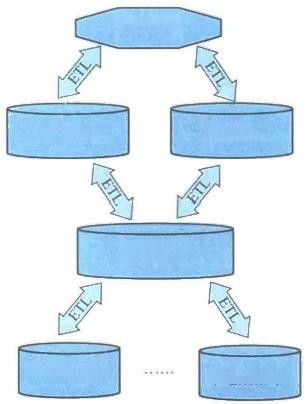
转换：在数据转换阶段，可以将一系列规则或函数应用于提取得到的数据中，以 准备将其加载到最终目标系统中。转换的一个重要功能是数据清洗，其目的是仅将 “适当的”数据传递给目标。不同系统交互时可能需要多种转换类型才能满足业务和技

大数据治理： 理论与方法……

术需求。

装载：装载阶段将数据装载到最终目标系统中，原则上，该最终目标系统可以是 任何数据存储系统，数据仓库只是目标系统的一种。由于需求不同，此过程的差异可 能很大。数据仓库可能只记录当前数据和某个时间段的历史数据等，装载的方式不尽 相同。更复杂的系统还会维护被装载的数据的所有更改的历史记录和审计跟踪。

数据存储在数据仓库中后可以被用户查询。ETL 几乎贯穿整个数据仓库构建和使 用过程，如图5-5所示。图中的数据集市 (Data Mart),有时也称为数据市场，包含为 了满足特定需求存储的多维数据。数据集市中的数据是从数据仓库中提取出来的一个 子集，目标是为了某些(通常是重要的)应用提供便捷、高效的服务。



应用程序

数据集市 数据集市

数据仓库

本地数据库1 本地数据库n

图5-5 数据仓库构建和使用过程

除了ETL, 数据仓库的构建可能还涉及 ELT 或 CDC 。ELT 是提取、装载、转换 (Extract-Load-Transform) 的缩写，其适用于数据结构化比较好、集成工作量不大的情 况，主要思想是，在数据提取之后不立刻转换，而装入暂存区进行清洗，最后在数据 仓库中完成转换。CDC则是变化数据抓取 (Change Data Capture) 的缩写，通常通过在 源系统上安装第三方应用程序来采集源系统的数据变化，多用于对实时性要求非常高 的场景。

从数据仓库的定义和结构可知，其是在数据源已经大量存在的前提下，为了进一 步挖掘数据潜在价值并制定重要决策而构建的。即使有些数据仓库还是用关系数据库 管理系统来管理的，但它和“大型数据库”还有很显著的区别。简而言之，从内容和 设计的原则来讲，传统的操作型数据库是面向事务设计的，数据库中通常存储在线交 易数据，设计时尽量避免冗余， 一般采用符合范式的规则来设计。而数据仓库是面向 主题设计的，为了更好地为主题服务，会在设计时有意识地引入一些冗余，采用反范 式的方式来设计，从而提供更便捷、高效的查询分析服务。

第5章 | **数据集成**

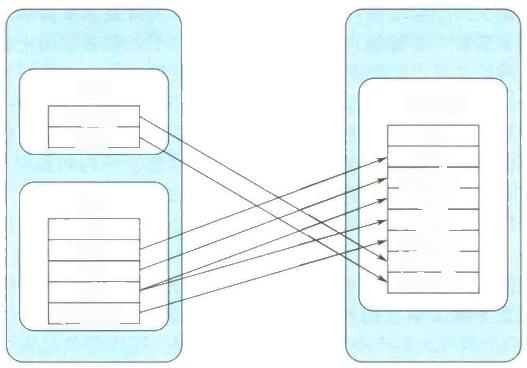
5.3 **传统数据集成的关键技术**

传统数据集成经过多年研究，相关的技术很多，从数据管理的角度而言包括模式 匹配、数据映射等。

**5.3.1** **模式匹配**

对于数据模式 (schema), 可以将其视为元数据的 一 种。其标示了数据的各部分分 别表示什么语义，例如，传统数据库表的表头就是数据模式。在数据集成中，模式匹 配的主要目标是完成不同数据模式之间的语义映射。这个过程需要两个主要步骤：首 先需要识别数据对象的语义相关性，如包含关系或等价关系；其次，需要完成语义相 关的对象之间的映射，这些映射可能是组合的，并不一定是一一映射的。通常，很难 完全自动地完成这两个主要步骤，但如何尽可能提高自动化的程度是这个领域的主要 问题。

模式匹配的困难主要来源于数据的多源异构性：不同的模式可能使用不同的标识 来指代相同的信息；或者使用同一个标识来指代不同的信息；又或者对同一个信息的 描述精度不一致。例如，在图5-6所示的模式匹配的例子中，源模式中有“公司”和 “员工”两张表，其中数据需要被集成到“个人信息”表中。源模式到目标模式的映 射如图中箭头所示。源模式和目标模式的异构性表现在不同方面，例如，“公司”表的 “名称”属性应当映射到目标模式的“工作单位”;“员工”表的“地址”属性应该被 拆分为目标模式的“城市”和“街道”两个属性；“员工”表的 “ID” 属性和“个人 信息”表的 “ID” 属性指代的数据元素并不相同。这些映射关系对于一个非常熟悉数 据的专家来说，可能是比较容易得到的，但是想自动发现这些映射关系并不容易。当 数据源数量增加到 一 定程度，完全靠人工来完成模式映射变得不可行，因此必须要尽 可能地自动化。



**源模式**

公司 名称 地址

员工

ID

姓名

生日

地址 电子邮箱

**目标模式**

个人信息

ID

姓名 出生日期

城市

街道 电子邮箱 工作单位 办公地址

图5-6 模式匹配示例

**大数据治理：** 理论与方法………

**1.** **模式匹配的方法**

模式匹配的方法可以大致分为模式级的匹配、实例级的匹配、混合匹配，它们都 是利用模式信息或模式和实例级别信息的方法。其中模式级的匹配仅考虑模式信息， 而实例级的匹配还考虑具体的数据实例。

模式级的匹配考虑抽象的模式元信息，包括列名(对于关系模式而言)、元素名 (对于XML 等半结构化数据而言)等，如名称、描述、数据类型、关系类型(部分， 继承等)、约束和模式结构。具有相似属性的模式元素通常被看作匹配的元素。基于约 束的匹配除了考虑模式元素的属性，还考虑包含在模式中的约束。这样的约束用来定 义数据类型和取值范围，唯一性、强制性、关系类型和基数等。两个输入模式中的约 束相匹配，则可以确定和约束相关模式元素的相似性。例如，基于邮编的语义，当关 系中两列数据都是6位数的时候，可以猜测其是否都是邮编。

实例级的匹配通过观察数据实例来确认模式元素的含义。在模式级可用的信息不 足时，实例级的匹配可以增强对模式级匹配结果的置信度。例如，为了确定 DName 和 EName究竟哪个应该匹配到 DeptName, 可以通过查看 DName 和 EName 的数据实例来 确认；固定电话号码可能有8位和7位两种，遇到7位数字时，可以通过查看数据实例 来确定其是否就是电话号码。

混合匹配直接结合多种匹配方法，将几个模式匹配的方法加以合并来基于多准则 或信息源确定候选匹配。有些额外的信息可以用在混合匹配中，如词典或用户自定义 规则等。

历史匹配信息可以加以重用来辅助新的匹配，其动机是模式中的结构和子结构经 常被重复使用。例如，在电子商务领域中，不同的信息系统中经常有着类似的模式。 但是，这样的重用通常被限定在某个特定领域中，例如，工资和收入在工资单应用程 序中可以被视为相同的，但在纳税申报应用程序中则不相同。

**2.** **模式匹配的关键技术**

目前学术界和工业界对于模式匹配已经展开了广泛的研究。由于准确的模式匹配 要基于对于数据语义的准确理解，这种理解对于机器来讲是非常困难的任务，这使得 模式匹配任务通常需要一定量的人工操作，因此，对于大数据应用来说，如何提高模 式匹配的自动化程度是模式匹配在技术上面临的主要难点之一。我们可以将自动化的 模式匹配技术分成几个特定阶段，包括经典的模式自动匹配、复杂的模式自动匹配、 针对特定领域的模式自动匹配及使用半自动的模式匹配工具等。这些模式匹配技术需 要理解模式的不同侧面，包括文本相似性、语义相似性、结构相似性等，如图5-7所 示，经典的模式自动匹配方法通常将目光放在某个特定侧面上。

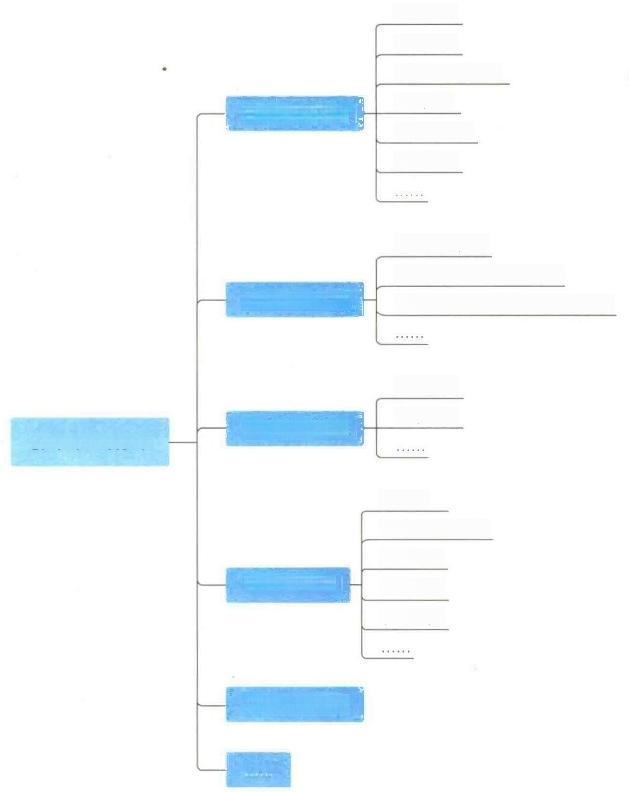
下面简要介绍基于文本相似性的匹配、基于语义相似性的匹配和基于结构相似性 的匹配。

基于文本相似性的匹配通常基于名称、描述、分隔符、字符串、子字符串进行匹 配。这种技术主要考虑文本上的相似性，例如，来自不同数据表的“毕业学校”和 “毕业院校”这两个属性的四个字中仅有一个字不同，可以简单地认为文本相似度为 0.75。从文本上看相似性较高，那么基于文本相似性的匹配就可能会将这两个属性映 射为同一个。基于文本相似性的匹配技术主要依赖于底层的文本相似度度量，常用的

 66 

**第5章**|数据集成

字符串相似度度量方法都可以用在这里，包括编辑距离 (Edit Distance,也称为 Leven- shtein距离)、海明距离、Jaccard Distance 、J-W 距 离 (Jaro-Winkler Distance)、余弦相 似性 (Cosine Similarity)、欧氏距离 (Euclidean Distance) 等。这种方法的优点是相对 简单，易于实现和理解，由于只需要将模式元素当成字符串来比较文本层面的相似程 度，因此自动化程度也较高。但是，由于其简单性，很多涉及数据语义层面的匹配可 能会出现遗漏或错误。例如，“最高学位”和“最高学历”两个属性的相似度同样很 高，但并不是同一个属性，这种语义上的细微区别通过文本相似性从语义上很难区分， 其是否匹配需要进一步从语义层面进行处理和判断。



编辑距离

海明距离

Jaccard Distance

J-W距离

基于文本相似性

余弦相似性

欧氏距离

基于辅助信息

基于模式元素的特定性质

基于自然语言处理/深度学习技术

位置相似

联系相似

模式匹配关键技术

工作流

早搜索空间剪枝

分区比较

并行匹配

优化匹配

基于交五式手段

基于语义相似性

基于结构相似性

基于策略组合

图5-7 模式匹配的关键技术

基于语义相似性的匹配通常要利用一些自动化的语义理解技术来比较两个模式实 例内容的语义相似性，从而推断两个模式元素的相似性。语义相似性在很多场景下比

大数据治理： 理论与方法…………………

文本相似性更能反映两个模式元素是不是“真正相似”。但是，从自动化的角度来讲， 文本相似性由于可以通过简单地比较字符匹配数量来完成，因此通常具有较高的自动 化程度，而语义相似性想要同时提升准确度和自动化程度则并非易事。简单地，可以 通过一些辅助信息来帮助计算语义相似度。常用的辅助信息包括叙词表、首字母缩略 词、字典、近/同义词表、不匹配列表、规则表等。例如，可以通过查询近/同义词表 来确定哪些词语的意思是相近或相同的，如果我们知道“学校”和“院校”是近义 词，那么即使这两个词的文本相似性只有0.5,我们也可以将其匹配起来。此外，也可 以通过模式元素的特定性质来帮助判断语义是否一致。例如，可以根据两个模式的数 据类型、取值的类型范围、独一性要求、空值性要求或者外键要求来检测两个模式之 间元素相似性；可以通过解析数据库中查询日志，从用户使用这两个关系模式的关系 中推断这两个模式的相似性；可以由了解数据的专家或用户给出一阶逻辑规则集合来 帮助判断什么情况下模式元素可以匹配。如果已经构建了完备的辅助信息，那么基于 语义相似性的匹配可以借助辅助信息，将语义相似性的判断转化为文本匹配度的判断， 从而获得较高的自动化程度。然而，辅助信息的获取、维护、更新往往很难自动完成。 例如，构建并维护一个字典可能需要大量的人工操作，对于专业程度较高的字典来说， 如医学、机械、IT 技术等领域的专业字典，构建和维护字典往往还需要专门的领域专 家参与其中，这又进一步降低了其自动化程度。 一些自然语言处理技术或许可以解决 辅助信息依赖的问题。例如，可以利用信息检索和文本挖掘中常用的TF-IDF(Term Frequency-Inverse Document Frequency), 来比较两个模式实例内容的语义相似性，从而 推断这两个模式元素的相似性。随着深度学习技术的发展，人们开始尝试构建深度神 经网络来进行自然语言理解，进而完成语义匹配任务。例如， BERT(Bidirectional En- coder Representations from Transformers) 就已经被研究人员用来完成模式匹配任务。此 外，也可以针对某些特定领域建立模式匹配相关的语料库，来提升该领域的模式匹配 效率和精度。

基于结构相似性的匹配通过分析模式元素的结构特征来判断其是否相似或匹配。 如果两个模式元素出现在相似的组里(位置相似),或者有相同的联系、与相似的元素 有关(联系相似),就可以认为这两个模式元素是相似的。与基于语义相似性的匹配类 似，基于结构相似性的匹配方法也需要额外的辅助信息，用来指明哪些元素是相关的。 基于辅助信息，往往可以将元素建模为图，将模式元素建模为图中的节点，将元素之 间的联系建模为图中的边，进而将模式匹配问题转化为图中的问题，再利用图算法来 解决问题。

基于策略组合的匹配将目光专注于模式的某个侧面通常有助于设计出小而精的算 法，但往往会忽视模式其他侧面提供的对于匹配有利的信息。因此，可以使用一些策 略来组合不同技术以期得到更优的匹配结果。现有的组合策略有工作流、早搜索空间 剪枝、分区比较、并行匹配、优化匹配等。工作流策略首先按照一定的顺序独立的执 行基础匹配算法，然后将基础匹配算法的输出结果进行组合得到最终输出。早搜索空 间剪枝策略的基本假设是大部分模式元素能够使用相对简单高效的算法来得到准确匹 配结果。这样一来，就可以先使用一个执行效率较高的基础匹配算法来获得大部分元 素的准确结果。对于剩余的难度较高的匹配任务，则选用更加昂贵和精确的匹配算法。

 68

第5章 |数据集成

分区比较策略将输入/输出模式分成互不相交的区域，针对不同的分区使用不同的匹配 算法。并行匹配针对数据规模大、重复工作多的场景，通过使用多个匹配器并行匹配 来加速模式匹配过程。优化匹配是指通过采用一些预处理技术来寻找更快的搜索策略， 避免重复遍历的一种匹配策略。策略组合因为涉及多个侧重点不同的基础算法，所以 其在技术上需要解决的共性问题之一是如何在策略约束的框架下，自适应地调整基础 算法的组合。所谓自适应调整， 一方面要根据任务自动选择要使用哪些基础算法，另 一方面要设置合适的参数来整合各基础算法的结果来得到最终输出。这种自适应性方 面的要求使得组合策略相对复杂很多，但只要配置得当，组合策略通常能获得比基础 算法更优的模式匹配结果。

由于完全自动化的模式匹配很难保证匹配的准确度，因此基于交互式手段将用户 的意见引入模式匹配任务也是较为常见的解决思路。例如，系统可以提供用户图形界 面来展示匹配的结果，并提供更正接口让用户修正不恰当的匹配结果；或者系统可以 支持增量式匹配，先生成k 个最佳匹配由用户来进行选择恰当的匹配结果，再基于用 户的选择进一步修正其他匹配结果；也可以提供top-k 匹配，为模式中的每个元素都生 成k 个最佳匹配元素，由用户决定最终选择哪一个。用户做出的选择可以视为比机器 自动生成的匹配建议更加权威、可信的匹配建议，这些更权威可信的匹配建议除了可 以在增量匹配中为后续匹配提供参考，也可以作为主动学习模型的训练数据，帮助机 器建立更优质的匹配模型。

**5.3.2** **数据映射**

在数据集成中，数据映射的目标是在给定两个数据模型之间建立起数据元素的对 应关系。数据映射在数据集成中，不仅用于完成数据源和目标之间的数据转换，还用 于识别数据关系、发现隐藏的敏感数据和冗余数据等。例如，在图5-6所示的例子中， 完成了模式匹配，我们还需要将源模式数据表中的每一行具体地填充到目标模式数据 表中，这就涉及一些具体的数据映射任务。

(1)从源模式数据表的数据表示到目标模式数据表的数据表示的转换，例如，源 模式数据表中的生日的表示可能是“××年××月××日”形式的字符串，而目标模 式数据表中的生日可能为“××××-××-××”形式的字符串，二者之间需要定义 正确的转换。

(2)源模式数据表中的数据对于目标模式数据表来说可能冗余，例如，如果在姓 名为“张三”的人的信息已经存在于目标模式数据表中，就需要首先识别出源模式数 据表中的“张三”和目标模式数据表中的“张三”是否是同一个人，如果是，就需要 判断目标模式数据表中是否存在需要更新的数据；如果不是，在存储时要对姓名相同 的两个人做出明确标识，以便区分。

(3)源模式数据表中的数据可能需要进一步提取后才能集成到目标模式数据表中， 例如，由于源模式数据表的“地址”属性映射到了目标“个人信息”表的“城市”和 “街道”属性，就需要正确地从“地址”中提取出“城市”和“街道”,再填入目标模 式数据表。

对于不同的应用场景，映射任务的复杂度有所不同，具体取决于要映射的数据的

**大数据治理：** 理论与方法……

层次结构及源数据库和目标数据库的结构之间的差异。较简单的场景可能可以完全自 动化完成数据映射，但在较为复杂的情况下， 一些手工的工作往往不可避免。在很多 分类法中，模式匹配或模式映射也作为数据映射的一部分讨论，但我们为了强调匹配 和映射在抽象(模式)和具体(数据)层面的不同，还是将二者分开讨论。

为了完成数据映射，可能的思路大致有两类：字符(串)层面映射和语义层面 映射。

所谓字符(串)层面映射，就是通过观察数据在字符上的相似度来判断映射关系 是否存在。例如，可以通过比较字符串的编辑距离来判定两个字符串是否指代同一个 数据对象。具体地说，如果我们在“地址”属性上发现“哈尔滨南岗”和“哈尔滨南 岗区”仅有一个字的差别，就可以判断二者很有可能是指代的地址相同，从而完成二 者的映射。这种匹配相对比较容易完成，自动化程度较高，因此在实际往往首先会从 字符串层面考虑是否匹配。

所谓语义层面映射，则需要考虑数据背后所表达的语义是否是一致的。很多在字 符层面很相似的数据未必表示同一含义，例如，“哈尔滨工业大学”和“哈尔滨工程大 学”仅有一个字不同，但是指代的是两所不同的学校；“广州”和 “Guangzhou” 指代 的城市相同，但字符完全不同；“苹果”既可以指代水果和也可以指代苹果公司，需要 根据上下文语境来确认具体含义。为了完成语义层面的映射，需要判定不同的概念或 实例的等价性，常见的等价性如下。

类别等价：表明类别或概念是相同的，如“人员”与“个人”相同。

属性等价：表明两个属性是相同的，如“家庭住址”与“家庭地址”是相同的 属性。

实例等价：表示两个具体的实例是相同的，如“广州”与 “Guangzhou” 是同一个 城市。

上述等价性的判定任务是语义映射任务的重要组成部分，在很多场景下，我们很 难百分之百地判定等价性，因此也有一些研究关注于如何判定近似等价性。目前，已 经有一些方法可以辅助完成语义映射，如同义词表、以 Word2Vec为代表的词嵌入、潜 在语义分析等方法。

5.4 跨界数据集成

前面我们的讨论更多集中于传统数据集成，在传统数据集成中，基本假设数据来 自同一个域。跨界数据集成要处理的任务更难，其要对不同域相关联的数据进行集成， 基于不同域产生的多个数据集中数据对象的隐含关联性融合数据，协同发现新知识。 跨界数据集成的难度在于必须理解来自不同域的多模态数据。

按照集成的方法不同，跨界数据集成可以被分为三类，分别是基于阶段的集成、 基于特征的集成和基于语义的集成。

**5.4.1** **基于阶段的集成**

基于阶段的集成在数据分析、挖掘的不同阶段使用不同的数据集合。由于数据集

 70 

**第5章**| 数据集成

是异步使用的，所以用在不同阶段的不同数据集可以是低耦合的，并不强求数据形式 必须一致。

例如，基于汽车的GPS 导航数据或交通网数据检测交通异常。汽车在道路上的异 常事故可以通过汽车在道路上的行为与平时明显不同来表示。用检测到异常的时间跨 度和该地点汽车行为落入异常范围为条件，可以初步发现异常，从而可以进一步检索 在这一路段发生事故的时候，人们在这里发布的相关媒体信息，从这些社交媒体信息 中通过挖掘代表行的术语(如灾难、防御等)来描述检测到的异常，检测到的这种异 常在正常时间段很少发生但是在这段异常时间内发生得很频繁。第一步是缩小被检测 的社交媒体的规模，第二步是使第一步获取到的结果的语义更加丰富。在这两步中使 用不同的数据集，这些数据集并不需要经过传统数据集成的过程，而在知识层面上实 现了集成。

**5.4.2** **基于特征的集成**

基于特征的集成方法是指基于表征学习等方法，从不同数据集合中提取出来的原 始特征中学习出新的特征，把这种新的特征应用于分类、预测等数据分析挖掘任务。 这种方法的实现可以有如下途径。

**1.** **直接关联**

这种方法对从不同的数据集中提取的特征平等看待，把它们直接连接成一个特征 向量，这个特征向量被用于数据分析、挖掘任务。因为不同数据集的表示、分布和规 模不同，这种方法存在一定的局限性。第一，在少量的训练样本中，这种相互关联有 可能导致过拟合，并且每个样本的具体特征可能被忽略；第二，从不同形式的数据中 发现关联性不强的数据之间的高度非线性关系是很困难的；第三，从存在相互关联的 不同数据集中提取的数据特征可能存在冗余和依赖。

**2.** **基于深度神经网络** **(Deep** **Neural** **Network,DNN)** **的方法**

这种方法利用DNN技术从不同的数据集学习出具有强大表达能力的特征。利用这 种方法可以有效实现：

(1)在其他形式的数据的帮助下，形成更好的单一形式的特征；

(2)这些共享的特征显示出不同数据形式间的相关性。

在实践中，基于DNN 方法的效果通常取决于我们能否调整好参数。找到一系列合 适的参数可以使得到的效果更好。然而，尽管很多场景下我们可以获取大量用于训练 的样本集合，找到最理想的参数仍然是非常耗费时间的过程，这需要大量的实验。另 外，我们很难解释中间层象征着什么,也不能理解用DNN 获得更好特征的过程。

**5.4.3** **基于语义的集成**

基于特征的数据集成方法不关心每个特征的含义，仅仅把这种特征视为一个真实 值或者绝对值。与基于特征的数据集成不同，基于语义的集成方法需要清晰地理解每 个数据集语义。我们要知道每个数据集代表着什么、为什么不同的数据集可以融合 它们之间怎样相互增强特征。

在数据集成的过程中，可从多种多样的数据集中提取出许多能够帮助人们解决问



大数据治理： 理论与方法

题的特征。因此，这些特征是可以识别的且有价值的。关于同一个实体的不同数据集 和不同特征的子集可以被视为一个实体的不同视图。例如，从不同数据源训练之后的 信息可以鉴定一个人的一些信息，如脸、指纹或签名等； 一幅图可以通过不同的特征 集合(如颜色等)表现。 一方面，因为这些数据集描述相同的实体，它们之间存在潜 在的相似性。另一方面，由于这些数据集是互不相同的，因此它们分别包含着独有的 信息。所以，整合不同的视图可以更加准确、全面地描述一个物体。

基于语义的数据集成方法又可以分为基于多视图的方法、基于相似性的方法、基 于概率依赖的方法和基于迁移学习的方法。下面对这些方法加以简述。

**1.** **基于多视图的方法**

基于多视图的方法可以分成三类：共同训练、多核学习和子空间学习。共同训练 算法可选择最大化双方在不同视图中数据达成一致的可能性。多核学习算法利用自然 对不同的视图做出反应的核，把无论线性还是非线性的特征整合在一起，从而改进学 习的效果。子空间学习算法旨在获得一个可以供不同视图共享的子空间，前提是假设 输入的视图是从这个潜在的子空间中产生的。

**2.** **基于相似性的方法**

相似性存在于两个不同的物体之间。如果我们知道两个物体 (X 、Y) 在某种维度 上存在相似性， Y的信息可以暂时使用X 替代，当Y 的信息缺失的时候，且当X 和 Y 分别有不同的数据集的时候，我们可以学习到这两个物体之间的多个相似的性质，X 和Y 的数据将会基于另一个数据集的相应数据进行计算。合并相关的两个对象，这些 相似性可以相互增强。例如，从稠密数据集中学习到的相似性可以增强其他来自稀疏 数据集中的相似性，因此帮助填补了后者的缺失数据。我们还可以通过结合两个对象 的多个数据集更加精确地预测两个对象之间的相似性。因此，不同的数据集可以基于 相似性相互混合使用。耦合矩阵分解和流形对齐是这类方法中的两个典型的方法。

**3.** **基于概率依赖的方法**

概率图模型是利用图模型表达随机变量间条件依赖关系的概率模型。通常来说， 该方法使用基于图结构的表达方式对多维空间上分布进行编码。两种典型的图表是贝 叶斯网络和马尔科夫网络。这种方法通过概率依赖弥补不同数据集合间的差异性，这 种概率依赖强调对象间的交互而不是相似性。例如，将从不同数据集中提取出的特征 (在图模型中称为变量)表达为图结构中的结点，特征之间的概率依赖用图中的边描 述。这样的图结构可以从数据中自动学习出来，也可以利用人的先验知识生成。图模 型中的变量可以通过推理得到。图模型的学习过程是，根据观察到的数据观察其中变 量间的概率依赖关系。最大期望 (Expectation Maximization,EM) 算法是常用的基于概 率依赖的算法。推理过程是根据观察到的变量值和学习到的参数预测因变量状态的过 程。推理算法包括以变分法为代表的确定性算法和以吉布斯采样为代表的随机化方法。

**4.** **基于迁移学习的方法**

机器学习和数据挖掘的一个假设是训练集和数据集有相同的分布和特征空间，但 是，在现实世界的应用中，这个假设未必成立。例如，我们有时候有一个 A 域上的分 类任务，但是只有在B 域上的数据，后者和前者的特征空间和数据分布可能完全不同。

 72 

**第5章** |数据集成

和需要假设标记数据和未标记数据相同的半监督学习不同，迁移学习允许训练集和测 试集的域、任务和分布不相同。在现实世界中有很多迁移学习的实例，例如，学习骑 自行车的经验会对学习骑摩托车有帮助。

利用迁移学习技术，可以将从一个领域上学习到的知识迁移到另一个领域上，从 而能够有效地应对数据稀疏的问题，实现跨域的数据集成。迁移学习甚至可以在不同 的学习任务之间迁移，例如，从图书推荐迁移到旅行推荐。

5.5 **智能化时代的数据集成**

大数据时代，数据的丰富极大地促进了各类智能化方法的发展。 一方面，人们希 望从海量数据中提取信息、提炼知识，并不断扩大知识的边界；另一方面，人们也希 望能够在扩大数据量的同时为数据提供必要的保护。智能化时代的数据集成涉及的方 面很多，受篇幅所限，本节选取两类比较典型的数据集成中涉及的问题做简要介绍。

**5.5.1** **知识图谱融合**

智能化时代，人们期望机器具备理解世界知识的能力，也就是说，希望机器能够 识别知识之间的复杂关联关系，以及能够理解知识的潜在语义信息。知识图谱是由实 体和关系组成的多关系图，通过图结构来描述实体具有的各类属性和实体之间的多种 联系，是一种结构化的知识表示方法。构建知识图谱的目的，就在于为人工智能技术 赋能，让机器学习到知识的更多含义，从而具备一定的认知能力。目前，知识图谱已 经被广泛应用于大数据决策分析、搜索引擎、推荐系统和智能问答等众多领域。

知识图谱通过规范化语义来汇聚知识，进而支持复杂关系数据的挖掘分析和推理。 一方面，知识图谱可以视为“大数据的精炼”,其记录了人们利用手工方法或自动化方 法从数据中提炼出来的相对静态、稳定的知识；另一方面，知识图谱本身也可以视为 一类图结构数据，对其高效地应用离不开各类图数据处理算法。知识图谱的有效性会 受其数据量和覆盖面的影响，数据量越大、知识的覆盖面越完整，为下游应用提供的 信息就越充足，反之则可能使得应用的可用性大打折扣。在实际使用中，人们往往不 能一次性将所有知识构建到一个图谱中，而常常基于不同的原始数据集构建出多个异 构知识图谱。不同组织会针对同一领域构建不同的知识图谱，同一个组织也可能会根 据不同需求构建不同领域的知识图谱。随着下游应用的复杂化，人们往往需要扩大知 识的边界，即集成多个异构图谱来更好地支持下游应用。

直觉上，集成多个不同来源的异构图谱有两种思路。

第一种思路是，先集成用于构建图谱的底层数据，将不同图谱的底层数据集成为 一个大数据集合之后，再在新的数据集合上重新构建更完善的图谱。这种思路在底层 原始数据完整，且在图谱异构性较强而底层数据异构性较弱时比较适用，此时集成原 始数据的难度要低于直接集成图谱，可以利用我们前面介绍过的传统数据集成的方法 来完成底层原始数据的集成。不过在一些场景中，如果用于构成知识图谱的底层数据 不完整/不可用，或者数据比较杂乱、异构性过强，那么就很难采用这种思路来完成图 谱的集成。

大数据治理： 理论与方法 ……………………

第二种思路是，不考虑底层原始数据，直接将多个异构图谱集成为一个更大、更 完善的知识图谱。这种思路适用于底层数据集成难度较高或图谱异构性相对较弱的场 景。这种思路需要处理的是图谱的异构性。如何将不同知识图谱中语义相同的知识关 联并融合起来，就成了一个核心问题。由于知识图谱表达的是实体间的联系，因此第 一步就是要明确，哪些来自不同图谱的实体能够“对齐”,即表达同样的语义。如果能 够明确知道两个知识图谱中哪些实体可以“对齐”,我们就能够以对齐实体为基准，将 两个知识图谱“拼接”起来。我们将知识图谱形式化表示为 G=(E,R,T), 其中E 为 实体，R 为关系，T 为(头实体，关系，尾实体)三元组，对于两个不同的知识图谱 G₁=(E₁,R₁,T₁) 和 G₂=(E₂,R₂,T₂), 实体对齐的任务是要在两个知识图谱中找到并统 一实际含义相同的实体，即发现e₁ ∈E₁,e₂∈E₂,e₁ 与 e₂ 的含义相同。

针对实体对齐问题，传统的方法是通过比较名称、文本描述和属性值等符号特征 的相似度来对齐实体。这种特征相似度的计算往往会受到不同图谱之间语义异质性的 影响，当要集成的图谱异构性强、噪声多、完整度低时，对齐效果不佳。近年来，基 于深度学习的知识向量表示方法引起了广泛的关注，这种方法保留了知识潜在语义信 息，提升了实体对齐的效果。当前的方法有两大类主要的对齐思路，分别是基于平移 模型的实体对齐和基于图神经网络的实体对齐。其中，基于平移模型的实体对齐方法 的核心思想是基于实体和关系的向量关系来学习图谱的向量表示(嵌入),并根据嵌入 的距离来找到实体对齐，相较于基于图神经网络的实体对齐方法，其更适合大规模的 知识图谱对齐任务。运用图神经网络的方法，可以学习知识图谱中实体在图上的结构 特征，进而比对不同图谱中两两实体的特征相似度，可以取得比较好的对齐效果。因 此，下面我们将对这两种思路做简单介绍。

**1.基于平移模型的实体对齐**

该方法对知识图谱中的结构化信息进行编码，将实体和关系嵌入n 维向量空间， 那么每个实体都可以表示为向量空间中的向量 (a,…,a,), 通过计算向量的相关性， 如欧氏距离、余弦距离等，就可以得到实体之间的相似度，进而判定实体是否可以 对齐。

当前，基于嵌入的实体对齐任务大多是在平移模型 (TransE①) 及其拓展模型(如 PTransE 、MTransE 、IPTransE等)的基础上实现对齐操作的。TransE 是 Bordes 等研究 者在2013年提出的知识表示模型，其将实体与实体间的关系表示为实体向量间的平移 操作。平移模型的思想源于 Mikolov 等研究者于2013年在word2vec模型中发现的词向 量之间的语义平移现象，例如：

v(中国) -v( 北 京 ) =v(日本)- (东京)

其 中v(w) 表示w 对应的词向量。上式的直观含义是，“中国”的词向量与“北 京”的词向量做向量减法，约等于“日本”的词向量和“东京”的词向量做向量减 法。再深入探究一层，“中国”和“北京”、“日本”和“东京”的共同点在哪里呢? 如果我们将这些概念都构建为知识图谱中的实体，会很容易得知“中国”和“北京”、

① Bordes A,Usunier N,Garcia-Duran A,et al.Translating embeddings for modeling multi-relational data [C]// Neural Information Processing Systems(NIPS).2013:1-9.

**第5章**|数据集成

“日本”和“东京”都具有关系“首都”。换言之，如果我们将关系“首都”也表示为 向量，那么这个向量应当约等于v(中国) -v(北京)和v(日本) -v (东京)。这种语义 平移现象表明了词与词之间的隐含语义关系被有效编码进了词向量。TransE 模型受此 启发，对于给定的三元组的向量表示(h,r,t), 其 中h,r ,t 分别表示头实体、关系和 尾实体。TransE 将关系r 视为从头实体h 到尾实体t 的平移向量，即 h+r=t 。 其形式化 的评分函数为

E(T)=d(h+r,t)=|lh+r-tl

基于上述思想，我们可以通过向量运算来推断概念或实体之间的关联关系，例如， 如果我们对齐了两个图谱中的实体“中国”和 “China”, 同时也对齐了关系“首都” 和 “Capital”, 那么当我们发现 v( 中 国 ) + v( 首 都 ) = v( 北 京 ) , 且 v(China)+

v(Capital)=v(Peking) 时，就可以直接对齐“北京”和 “Peking” 。 基于嵌入的实体对 齐模型对知识图谱中的结构化信息进行编码，生成了实体的向量表示，并通过计算实 体与实体间向量表示的相似度来识别相同的实体，即完成实体对齐。现有实体对齐模

型往往需要利用已知的对齐种子集合L={( e,,e)le,∈E 作为标签数据，通过训练

来实现对知识图谱实体的有效对齐。

目前知识图谱实体对齐领域尚在发展当中， 一些新的思路和想法也层出不穷。从 模型嵌入表示和实体对齐的顺序的角度来看，基于嵌入的实体对齐大体上可分为两种 技术路线。

第一种技术路线是“先合并，后嵌入”。这个路线把一些预先匹配好的实体对齐种 子合并，使用平移模型将两个图谱直接训练到同一个向量空间中，训练使得实体对齐 种子有相似的嵌入表示，进而可进行实体对齐任务。例如，可以通过最小化预先设计 好的评分函数将使得两个知识图谱G, 和 G₂ 之间的对齐实体/关系的距离最近，进而使 得两个知识图谱嵌入一个统一的向量空间中。

第二种技术路线是“先嵌入，后合并”。这类方法先用平移模型分别独立地训练两 个图谱的嵌入表示，然后用一些预先匹配好的实体对齐种子训练一个线性变换，从而 对齐两个向量空间。

不过， 一些方法往往过度依赖于实体对齐种子的数量与质量，而获取大量的高质 量的预先对齐的种子实体在知识图谱来源广泛、异构性较强、甚至包含噪声的场景下 并不现实。此外， 一些方法假设两个独立构建的知识图谱对应的向量空间能够简单地 通过一个线性变换完成合并，这种硬变换实际上丢失了许多有用的信息。因此，如何 应对种子的有限性，以及尽可能多地保留并利用实体特征成为了一些技术的主要攻克 方向。由于知识图谱除实体和关系之外，还有属性，因此很多前沿的方法会从实体属 性的角度出发，完成实体对齐。也有方法从实体、关系、属性等多个视角同时出发， 完成实体对齐任务。

整体来说，当前基于嵌入的实体对齐大多还是基于TransE 生成结构嵌入的基础上， 通过对三元组提取更多特征来优化实体的嵌入表示，使得最终的嵌入表示有利于完成 实体对齐任务。

**2.** **基于图神经网络的实体对齐**

传统的平移嵌入方法十分强调实体的嵌入，未充分考虑到实体对齐中的关系或属



大数据治理：理论与方法

……………………

性嵌入，在理想化的平移对齐前提下，极大地限制了实体对齐的有效性。因此，基于 嵌入实体对齐方法还有着很大的改进空间 。 由于知识图谱本身可以视为图数据 ， 因此 一些处理图数据的前沿方法也被尝试用来处理知识图谱中的一些问题 。 随着深度学习 的广泛应用，针对图数据的深度学习方法也得到了全面的发展，其中一些方法也被用 来完成实体对齐任务 。

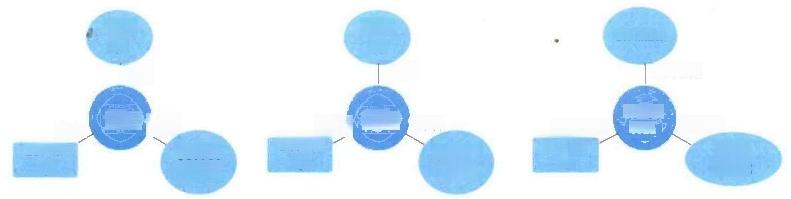
基于卷积神经网络、循环网络和深度自动编码器的思想，研究人员设计了图神经 网络 (Graph Neural Network,GNN)① 对复杂图数据进行处理。图卷积神经网络 (Graph Convolutional Network,GCN)② 是最有代表性的图神经网络之一，其借鉴了 一 维、二维数据结构上的卷积神经网络 (Convolutional Neural Network,CNN)③ 的思想， 尝试利用神经网络提取图数据的空间结构特征 。 其核心思想为通过聚合图结构中的邻 居信息来生成节点的向量表示 。 由于在知识图谱中 ， 实体被表示成图中的节点 ， 所以 不难看出，如果可以利用图神经网络学习生成知识图谱中实体的向量表示，那么就也 有可能利用向量的相似度或关系计算来完成实体对齐任务 。

基于图神经网络的实体对齐方法 ， 就是使用图神经网络生成实体的向量表示 ， 再 通 过 向 量 计 算 来 完 成 实 体 对 齐 。 基 本 的 做 法 是 先 使 用 GCN 将 实 体 投 影 到 低 维 向 量 空 间 中，再基于现有的已经对齐好的种子实体或关系找到新的对齐实体或关系。由于实体、 关系 、 属性在知识图谱中都具有很重要的作用 ， 所以各类方法的侧重点也有所不同 。

很多方法关注实体本身在实体对齐中的作用 。 典型的思路是对实体提取两方面的 特征：结构特征和属性特征。其中结构特征反映实体与知识图谱中其他元素之间结构 上的关联关系，属性特征则反映实体自身具备的属性层面的特征。这类方法假设应当 对齐的实体具有相似的结构和属性 ， 通过结合结构特征和属性特征来发现这些可以对 齐的实体 。 下面的例子简要说明了上述思路的大致执行过程 。

例5-1 如图5 - 8所示，实体“华山”和“西岳”都具有相同的属性“海拔 2154.9米”,并且都与实体“陕西省”存在“位于”的关联关系，与实体“沉香劈山

救 母 ” 存 在 “ 典 故 ” 的 关 联 关 系 ， 那 么 “ 陕 西 省 ” 和 “ 沉 香 劈 山 救 母 ” 的 特 征 向 量 会



陕西

省

位于

西 岳 典故

沉香劈

山救母

elevation l

2.154

m

沉香劈

山救母

2154.9

米

2154.9

米

陕西

省

Mount Hua

Chen

Xiang

华 山

legend

海拔

Shaanxi

location

位于

海拔

*典故*

(a) (b) (c)

**图5** **-** **8** **实体对齐示例**

① Franco Scarselli,Marco Gori,Ah Chung Tsoi,Markus Hagenbuchner,and Gabriele Monfardini.The graph neural network model.IEEE Transactions on Neural Networks,2009.

② Thomas N.Kipf and Max Welling.Semi-supervised classification with graph convolutional networks.In ICLR,2017. ③ LeCun,Y.,Bottou,L.,Bengio,Y.,&Haffner,P.(1998a).Gradient-based learning applied to document

recognition.Proceedings of the IEEE,86,2278-2324,1998.

 **76**

第5章| 数据集成

在聚合的过程中被分别聚合进“华山”和“西岳”的结构特征向量，从而使得“华 山”和“西岳”有相似的结构向量，同时，二者共同的属性“海拔2154.9米”也会生 成相似的属性特征向量。通过以某种方式(如简单拼接)合并结构特征向量和属性特 征向量，我们会得到两个相似的实体向量表示，分别表示“华山”和“西岳”,再通 过简单的向量相似度计算即可得知这两个实体可以对齐。

上述方法在“对齐实体具有相似的结构和属性”这一假设成立的前提下，往往会 取得还不错的对齐效果，不过在知识图谱异构性较强或种子实体数量有限的情况下， 可能不能达到想要的效果。下面的例子说明了这种情况。

例5-2 假设我们要对齐的是中文知识图谱和英文知识图谱，如图5-8 (a) 和 图5-8 (c) 所示，那么我们想要使用与例5-1中的同样流程时，必须先获取两对预先 对齐的种子实体，即“陕西省”和“Shaanxi”,“沉香劈山救母”和“ChenXiang”,同 时我们还需要对齐关系“位于”和“location”、 “典故”和“legend”。也许机器翻译 有助于我们自动化地完成一些对齐任务，但在异构性较强或需要背景知识时，这些预 先对齐的种子实体往往并不容易获得。例如，机器翻译可能会将“沉香”翻译为 “Agarwood”。在种子实体缺乏的情况下，图5-8 (a) 和图5-8 (c) 中的“华山”和 “Mount Hua” 想要对齐并不容易。

尝试将传统方法和图神经网络结合起来或许是一条可行的路线。例如，在生成向 量表示的过程中引入规则或词典，可以基于规则或者词典先对实体进行预处理，完成 传统方法能较容易完成的部分，再利用图神经网络去完成后续的对齐步骤。

近年来，基于图神经网络的实体对齐研究层出不穷。在研究中主要关注实体本身 在实体对齐中作用的方法包括 CCN-Align①、MuGNN② 和 GMNN③等。其中 GCN-Align 为每个实体多分配两个向量： H, 实体结构向量和H, 实体特征向量，利用GCN 将属性 信息和结构信息能够结合在一起；MuGNN 对每个GNN 通道上的进行不同的关系加权， 以补全缺失的关系和剔除唯一的实体；GMNN 认为以往的知识图谱跨语言对齐研究依 赖于仅从单语知识图结构信息中导出的实体嵌入，没有考虑到仅有少量邻居的实体和 缺乏足够结构信息的实体，由此引入主题实体图和局部实体子图，利用上下文信息表 示实体。

经 典 的 图 神 经 网 络 很 多 并 不 专 门 对 图 中 的 关 系 ( 边 标 签 ) 加 以 区 分 ， 而 知 识 图 谱 中的实体、关系、属性都是有明确的区分的，因此一些注重处理关系信息的方法也被 提出。例如，R-GCN④、 RDGCN⑤等 。R-GCN 定义了在一个关系多图的传播模型，将

① Wang Z,Lv Q,Lan X,Zhang Y.Cross-lingual knowledge graph alignment via graph convolutional networks.In: EMNLP 2018,2018.

② Cao Y,Liu Z,Li C,Liu Z,LiJ,Chua TS.Muli-chanmel graph neural network for entity alignment.In:ACL. 2019,2019.

③ Roth A E.Deferred acceptance algorithms;history,theory,practice,and open questions.International Journal of

Game Theory,36(3-4):537-569,2008.

④ M.Schlichtkrull,T.N.Kipf,P.Bloem,R.van den Berg,I.Titov,and M.Welling,"Modeling relational data

with graph convolutional networks,"in ESWC.Springer,2018,pp.593-607.

⑤ Wu Y,Liu X,Feng Y,Wang Z,Yan R,Zhao D.Relation-aware entity alignment for heterogeneous knowledge graphs.In:IJCAl 2019,2019.

大数据治理： 理论与方法

不同关系边类型所连接的邻居节点进行一个线性转化，通过聚合邻居节点的信息来更 新自身节点的表示，由此每个节点的信息聚合是根据不同边类型节点聚合得到的；但 由于 R-GCN 对每个关系都要使用一个权重矩阵，因此其缺点在对于包含数千个关系的 真知识图谱将需要过多的参数集。RDGCN 扩展关系构造对偶关系图，直接对关系信息 进行建模。通过原始实体图与其对应的对偶关系图之间多重交互作用来整合关系信息， 使模型能够有效地将更复杂的关系信息融入实体表示中。

此外，在基于图神经网络完成实体对齐任务时，还有许多因素需要考虑，例如， 如何选取邻居、如何利用注意力机制让对齐的重点聚焦于重要的实体和关系上、如何 处理种子实体对缺失的情况，以及如何结合图神经网络和平移模型来取得更好的效果 等。当前，这方面的研究还在积极发展中，受本书篇幅所限，我们在此不再对这些方 法进行深入讨论。

**3.** **实体对齐效果评价**

衡量实体对齐方法的优劣需要规范的数据集和评价标准，本节我们总结了一些常 用的数据集和评价标准，供读者参考。

国外的知识图谱项目与实验数据集包括 Freebase 、WikiData 、DBpedia 、WordNet、 YAGO 等，国内的则包括 OpenKG 、CN-DBpedia 、XLore 等。

(1)Freebase: 由美国软件公司Metaweb开发，于2007年公开运营，于2010年被 谷歌收购。Freebase 集成了许多网上的资源，被描述为“一个开放共享的世界知识数据 库”,以及“一个庞大的、协作编辑的交叉链接数据数据库”。谷歌于2015年已经正式 宣布用知识图谱API 替代 Freebase API。不过 Freebase 的数据集仍是知识图谱项目及实 验中常用的数据集，其包含超过1.25亿个关系元组、超过4000种类别、超过7000种 属性。由于 Freebase本身数据量非常庞大，不易使用，所以在许多研究中会使用 FB15k 和FB15k-237 这两个子集。

(2)WikiData: 由维基媒体基金会 (Wikimedia Foundation) 运转的协作编辑的多 语言知识图谱。它是开放数据的常见来源，继承了Wikipedia 的众包协作的机制，支持 以三元组为基础的知识条目自由编辑。WikiData 基于声明 (Statement), 以三元组的形 式描述了每项的详细特征。每条声明包含项的一个属性和对应的属性值。例如， 一条 声明可能陈述了项“中国”的属性“首都”的值是“北京”。截止到本书成书之时， WikiData 已经被编辑了14亿多次，包含超过9390多万项。

(3)DBpedia: 从维基百科 (Wikipedia) 的词条中提取结构化数据，并将其他数据 集连接到维基百科中。DBpedia 是一个大规模的多语言百科知识图谱，可以使用标准的 Web 浏览器、自动爬虫或使用类似 SQL 的查询语言(如 SPARQL) 进行复杂的查询 来。截止到成书之时， DBpedia已经支持超过140种语言，包含数十亿个 RDF 三元组， 上述数量仍在稳步增长。

(4)WordNet: 一个大型英语词汇数据库，由普林斯顿大学认识科学实验室建立和 维护。WordNet 描述了英文词汇之间关联特点，名词、动词、形容词和副词被分为一组 认知同义词 (Synsets), 每个都表达一个不同的概念。WordNet可以免费公开下载，其 结构使其成为计算语言学和自然语言处理的有用工具。

(5)YAGO:“Yet Another Great Ontology” 的缩写，是2007年由德国马普研究所开

第5章|数据集成

发的链接数据库，主要集成了Wikipedia 、WordNet 和 GeoNames 三个来源的数据，包含 关于人、城市、国家、电影和组织等方面的知识。目前，YAGO 的最新版本是 YAGO 4,其是一个基于Wikidata 的 RDFS知识库，包含超过5000 万个实体和20亿个事实。

(6)OpenKG: 中国中文信息学会语言与知识计算专业委员会所倡导的开放知识图 谱社区项目，旨在推动以中文为基础的知识图谱数据、算法和工具的开源和开放工作， 由来自浙江大学、东南大学、同济大学等多个单位的知识图谱团队提供底层技术支持 和系统维护。截止到本书成书之时，OpenKG 共发布了近200个不同领域的知识图谱。

(7)CN-DBPeidia: 由复旦大学知识工场实验室研发并维护的大规模通用领域结构 化百科。主要从中文百科类网站(如百度百科、互动百科、中文维基百科等)的纯文 本页面中提取信息，经过滤、融合、推断等操作后，最终形成高质量的结构化数据， 供机器和人使用。CN-DBpedia 提供全套API, 并且免费开放使用，自2015年12月份 发布以来已经在问答机器人、智能玩具、智慧医疗、智慧软件等领域产生数亿次 API 调用量。目前， CN-DBPeidia包含超过900万的实体以及超过6700万的三元组关系。

(8)XLore: 中英文跨语言知识图谱，融合了中英文维基和百度百科的数据，对百 科知识进行结构化，并建立了中英文实体间的跨语言链接。目前XLore 拥有超过2000 万个实体和超过200万个概念。

除了准确率和召回率，实体对齐还有一些其他的评价标准，如 Hits@k、Mean Rank 、Mean Reciprocal Rank 等。

(1)Hits@k: 令关系三元组 (h,r, t) 表示头实体h 和尾实体t 具有关系r, 我们 保持头实体h 或尾实体t不变，将三元组中另一个实体替换成任意一种其他的实体。假 设共有n 个实体，那么这样的替换一共有n-1 个(注意，我们保持其中一个实体及关 系不变，只改变剩余的一个实体)。因此，我们会得到n-1 个新的关系三元组，然后我 们对这些三元组计算实体关系距离，并按照实体关系距离从小到大排列得到三元组有 序列表。接着，我们从头开始遍历这个长度为n-1 的有序列表，看前k(top-k) 个三 元组是否能够命中有真实关系的实体，若命中，则 Hits@k 分值加1。k 通常会取一个 比较小的值，如10或者100。

(2)Mean Rank(MR): 平均排名，MR 计算在测试集合中平均到第几个才能命中 到正确结果，值越小，说明越早命中正确结果，方法性能越好。

(3)Mean Rtetpeal Rank(MRR):平均倒数排名  ,是第 一

个正确答案的倒数积。在实体对齐中，正确匹配的分值是其排名的倒数(第1个结果 匹配则得1分，第2个结果匹配则得1/2分，以此类推，第n 个结果匹配则得1/n 分)。 所有的分值相加之后得到最终的 MRR 分值，分值越高，说明算法效果越好。

5.5.2 联邦学习

学术界和工业界的几个典型大数据驱动应用的成功使得人们希望能尽可能多的扩 大数据的边界、融合更多的数据，从而进一步提升各行各业中数据驱动服务的品质。 集成不同部门、不同组织、不同行业的数据是增加数据量的重要手段之一，然而在很 多行业中，数据集成面临的困难不只是技术层面的。例如，不同公司可能因为行业竞

大数据治理： 理论与方法……

争而不愿直接交换底层数据，不同的医院可能因为隐私而无法共享病患信息，不同的 部门可能因为数据管理困难而难以集成部门的商业或技术数据。此外，从本章的前面 几节内容也可以看出，在数据异构性非常强的场景中，想要准确地完成数据集成任务 会消耗巨大的人力、财力和时间。因此，想通过数据集成解决数据孤岛问题并非易事。

同时，数据安全和数据隐私保护在国内外也受到越来越多的重视。2021年6月10 日，十三届全国人大常委会第二十九次会议表决通过了《中华人民共和国数据安全法》 (简称《数据安全法》)。《数据安全法》中提到“国家建立数据分类分级保护制度， 根据数据在经济社会发展中的重要程度，以及一旦遭到篡改、破坏、泄露或者非法获 取、非法利用，对国家安全、公共利益或者个人、组织合法权益造成的危害程度，对 数据实行分类分级保护。国家数据安全工作协调机制统筹协调有关部门制定重要数据 目录，加强对重要数据的保护。”“各地区、各部门对本地区、本部门工作中收集和产 生的数据及数据安全负责。”

由此可见，将不同来源的数据库集成一个大的数据库并非易事。那么,是否有一 种可能——我们不直接集成数据，但能打破数据孤岛，达到数据集成的效果呢?联邦 学习或许会成为成功的解决方案之一。

联邦学习 (Federated Learning) 的概念于2016年由谷歌提出，其尝试不集成“源 头”而直接集成“结果”。具体地，联邦学习不直接集成数据，而是尝试基于加密机 制，让参与方能够安全地交换模型训练的某些中间结果，不断迭代，最终完成在整个 大数据集上的模型训练。换言之，联邦学习让参与各参与方基于自己所有的数据训练 本地模型，再基于这些本地模型和专门提供信息与模型参数的加密交换功能的模块， 通过加密机制下的参数交换建立一个虚拟的共有模型，让这个虚拟的共有模型为各方 提供服务，同时，为了保护数据，联邦学习还要保证任何一个参与方都不能基于公有 模型反推出其他参与方的特征。基于上述思想，联邦学习可以保证训练的过程中各参 与方都不暴露原始数据，但能基于各参与方提供的数据训练出比单一数据集上更强大 的模型。

为什么需要这么做呢?我们可以考虑这样一个场景：几家业务相关的企业想要合 作，每家企业通过共享自己拥有的一些信息来达成业务上的提升。最直接的想法是各 企业拿出自己可共享的数据，将所有企业的可共享数据集成为一个大数据集合，再在 这个大数据集合上进行分析，将分析结果分发给参与的各企业。这样，各企业获得的 数据分析结果是在集成了其他相关企业的数据之后得到的，比企业自己仅基于本地数 据获取的分析结果更加精准，因此整体上看，所有企业都是这次数据共享的受益者。 不过，这样的数据共享真正实施起来可能会遇到很多困难，其中一个很重要的问题在 于，各企业可能并不想就这样直接开放自己的原始数据给其他企业，因为这些数据可 能涉及商业机密或者个人隐私。如果希望使用更多的数据完成机器学习，但数据集成 在源头就受到了阻碍，就可以考虑使用联邦学习。

由于各参与方通常是业务关联的，因此其数据集之间可能有重叠的样本或重叠的 特征。我们可以用一个数据矩阵来表示样本和特征，如图5-9所示，数据矩阵中的每一 行表示一条样本(如数据集中的一个用户或一个对象),每一列表示一维特征，不同的 参与方可能会在行或列上有一些重叠度。根据适用场景的不同和行列重叠的程度不同

 80 

**第5章**|数据集成

联邦学习利用重叠样本或重叠特征的侧重各有不同，据此可以进一步分为横向联邦学 习 (Horizontal Federated Learning)、纵向联邦学习 (Vertical Federated Learning) 与联邦 迁移学习 (Federated Transfer Learning)。三种联邦学习的特点如表5-1所示。

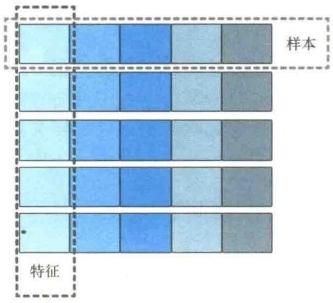


图5-9 数据矩阵

**表5-1** **三种联邦学习的特点**

|  |  |  |  |
| --- | --- | --- | --- |
| **种** **类** | **适** **用** **场** **景** | **集** **成** **方** **式** | **目** **标** |
| 横向联邦学习 | 特征重叠度较高而样本 重叠度较低 | 横向切分，特征取交集， 样本取并集 | 在保护数据的前提下 扩大样本量 |
| 纵向联邦学习 | 样本重叠度较高而特征 重叠度较低 | 纵向切分，样本取交集， 特征取并集 | 在保护数据的前提下 增加特征维度 |
| 联邦迁移学习 | 样本和特征重叠度都 较低 | 引入迁移学习 | 在保护数据的前提下 将源任务上的结果迁移 到目标任务 |

横向联邦学习关注各参与方的特征重叠度较高而样本重叠度较低的情况。例如， 哈尔滨工业大学提供的学生(样本)信息集合和广州大学提供的学生信息集合中重叠 的学生很少，但由于都是高校学生信息集合，因此样本的特征重叠度是较高的。在这 种情况下，可以将数据集按照数据矩阵的横向进行切分，取出那些特征相同而样本不 完全相同的数据进行训练。可以看出，横向联邦学习的目标是在保护数据的前提下扩 大样本量。通过“特征取交集，样本取并集”的方式来集成各参与方的数据，通过增 加样本量来取得更好的学习效果。具体地，可以有一个中心服务器为各参与方提供参 数交换服务，各参与方本地训练模型后加密梯度上传给中心服务器，中心服务器聚合 各参与方提供的信息后更新模型，并分发更新后的信息给各参与方，如此迭代进行直 到完成训练。

纵向联邦学习关注样本重叠度较高而特征重叠度低的情况。例如，医学专业的学 生可能会同时在学校附属医院工作，学校和医院关于这些学生的样本重叠度是很高的， 但由于双方职责不同，在特征上的重叠并不多。在这种情况下，可以将数据集按照数 据矩阵的纵向进行切分，取出那些样本重叠而特征不完全相同的数据进行训练。可以

大数据治理： 理论与方法

看出，纵向联邦学习的目标是在保护数据的前提下增加特征维度。通过“样本取交集， 特征取并集”的方式来集成各参与方的数据，通过增加特征维度来取得更好的学习效 果。具体地，各参与方可以对自己的数据进行加密，并由一个协作者来帮助个参与方 进行样本对齐(确认双方的重叠样本),帮助更新和交换参与方训练过程中的训练信 息，完成加密模型训练。

联邦迁移学习关注样本和特征重叠度都较低的情况。此时，由于在样本和特征上 都重叠较少，所以很难使用横向或纵向联邦学习的方法进行数据切分，而引入迁移学 习来完成联邦学习，目标是利用任务之间的相似性或相关性，在保护数据的前提下将 源任务上的学习结果迁移到目标任务。



**第** **6** **章** **数据质量管理**

如前文所述，在元数据管理、主数据管理和数据集成中，数据质量管理都扮演着 不可或缺的角色。随着大数据时代的到来，数据源日益丰富，但由于对其中很多数据 源缺乏掌控，因此很难保证这些来自不同数据源的海量信息是正确、 一致的。当前， 低质量数据及其所引发的知识和决策错误已经在全球范围内造成了恶劣的后果。举例 来说，美国的零售业中，每年由于错误标价造成的低质量数据就导致了25亿美元的损 失；2006年由于数据不一致而导致的信用卡欺诈失察给美国银行业造成了48亿美元的 损失；在数据仓库开发过程中，30%～80%的开发时间和开发预算花费在清理数据错 误方面；数据质量问题给每个企业增加的平均成本是产值的10%～20%。随着大数据 的广泛应用，数据质量管理越来越多地受到重视，同时也面临着诸多挑战。

数据质量的问题可能发生在数据生产和使用的各个环节。数据的真实性、准确性 完整性、时效性都会影响数据质量。除此之外，数据的加工、存储过程都有可能涉及 对原始数据的修改，从而引发数据的质量问题。所以，技术、流程、管理等多方面的 因素都有可能会影响到数据质量。

数据质量管理是一项长期复杂的过程。因为它不仅贯穿了数据采集、整理、加工、 存储、使用、分析、共享、交换等纵向多个环节，还涉及标准的制定、规范的落地、 全生命周期管理等横向多个领域。为此，需要投入大量的人员、时间、软硬件成本。

目前，数据质量管理的重要性已经得到了数据生产和使用的各环节认可。数据质 量管理问题需要同时通过制度和技术两个手段去解决，其中，制度手段包括制定数据 质量度量标准、数据质量监管体系和数据质量管理制度等；技术手段包括缺失值填充、

大数据治理： 理论与方法…………………

实体识别、真值发现、错误检测与修复等。本章我们将从数据质量开始，着重针对技 术手段，探讨各类数据质量管理问题。

**O.|** 一 **概述**

6.1.1 **数据质量的定义**

数据质量的定义有许多种。有观点认为，只有数据正确地描述了其对应的真实世 界的对象，才是高质量的；也有观点认为，只要数据满足了预期用途，就是高质量的。 针对不同的应用场景，数据质量的定义可能并不相同，同时，也可以在不同场景下展 开描述数据质量。例如，我们可以从以下几个方面来定义数据质量。

基于用户需求定义的数据质量：满足特定用户预期需要的程度。

基于数据本身定义数据质量：包括真实性、完备性、自治性等。

基于约束关系定义数据质量：包括数据的原子性、关联性、满足约束的程度等。

基于生产和使用过程定义数据质量：包括数据能被正确使用、存储、传输、共享 的程度。

数据质量有多个角度的描述方法，通常包括以下角度。

**1.数据一致性**

所谓数据一致性，是指数据集中每个数据在语义表达上是一致的，即不存在语义 错误或矛盾。例如，记录(公司=“先导”,国码=“86”,区号=“010”,城市= “上海”)就含有一致性错误，因为“区号=‘010’”的潜在语义是城市是北京，而 “城市=‘上海’”则指明了城市是上海，二者出现了语义上的不一致。

**2.数据精确性**

所谓数据精确性，是指数据集中的每个数据都能准确地描述现实世界中的实体， 即不存在模糊或近似。例如，某城市人口数量为4130465人，而数据库中记载的为400 万人。从宏观角度来看，该数据是合理的，但从微观角度来看，该数据并不精确。容 易看出，数据精确性是相对的，在很多场景下很难做到绝对精确。

**3.** **数据完整性**

所谓数据完整性，是指数据集中包含足够的数据来支持各种查询和计算，即不存 在数据或记录的缺失。例如，某医疗数据库中的数据尽管一致且精确，但缺少了某些 患者的重要病史，这种不完整性就可能导致不正确的诊断甚至严重医疗事故。

**4.数据时效性**

所谓数据时效性，是指数据集中的每个数据都与时俱进，正确描述了当前时刻的 数据对象，即不存在过时、失效的数据。例如，某数据库中的用户地址在2019年是正 确的，但在2020年，该用户搬家，地址未及时更新就会出现数据过时、失效的问题。

**5.** **实体同一性**

所谓实体同一性，就是说同一实体的标识在所有数据集中必须相同，语义表达必 须一致。例如，企业的市场、销售和服务部门可能维护各自的数据库，如果这些数据

 84 

**第6章** **|数据质量管理**

库中的同一个实体没有相同的标识或语义表达不一致，将存在大量具有差异的重复数 据，导致实体表达混乱，进而导致数据分析结果有误。

需要注意的一点是，工业界和学术界对于数据质量各维度的划分和定义并没有统 一的标准，对同一个维度的定义也未必完全相同，表6-1展示了工业界和学术界对数据 质量维度的一些拓展描述。

**表6-1** **数据质量维度(拓展)**

|  |  |  |
| --- | --- | --- |
| **序号** | **维** **度** | **描** **述** |
| 1 | 时效性 | 数据的老化度对当前工作的影响程度  信息系统对现实世界变化做出反应的及时程度 |
| 2 | 流通性 | 数据的实时程度  描述信息何时进入数据仓库 |
| 3 | 一致性 | 当前数据与历史数据有相同的形式且兼容的程度  对数据集定义的语义规则的违反项个数 |
| 4 | 精确性 | 数据库中描述现实世界实体的数据的准确程度  数据正确、可靠、被认证的程度  一个数据v,相对于另一个数据v'来说更正确的程度  在有可参考认证的权威来源情况下，数据的正确程度 |
| 5 | 完整性 | 数据集能够表示现实世界系统中每个有意义状态的能力  对于当前工作，数据的范围、广泛性、深刻性对任务的贡献程度 数据仓库中现实世界信息所占的百分比  非空数据的比率 |
| 6 | 可访问性 | 数据是否可用，或者是否可被快捷地检索到 |
| 7 | 重复率 | (不必要的)重复数据的比率 |
| 8 | 规范性 | 衡量数据标准、数据模型、商业数据和参考数据的存在性、完整性、 质量及文档记录是否合规 |
| 9 | 表现质量 | 数据的表示和收集、使用状态 |
| 10 | 声誉度 | 在存在对数据源和数据本身的额外描述时，数据在来源和内容方面 被重视的程度 |
| 11 | 无害性 | 数据产生的效果对人、过程或者环境的风险等级 |
| 12 | 适量性 | 现有数据量适用于当前任务的程度 |
| 13 | 安全性 | 数据获取权限的限定范围可确保其安全的程度 |
| 14 | 可信性 | 数据真实可信的程度 |
| 15 | 易懂性 | 数据明确、无歧义且易理解的程度 |
| 16 | 客观性 | 数据是否是公正的、公开的、无偏见的 |
| 17 | 切题性&有用性 | 数据对当前工作的适用性和贡献度 |
| 18 | 有效性 | 用户在指定的上下文中能使用数据准确完整地实现目标功能的程度 |

**大数据治理：** 理论与方法……

**(续表)**

|  |  |  |
| --- | --- | --- |
| **序号** | **维** **度** | **描** **述** |
| 19 | 解释性 | 数据有适当且定义明确的语言、符号和单位的程度 |
| 20 | 易操作性 | 数据易被处理且能适用于其他相似格式的程度 |
| 21 | 易用度&可维护性 | 数据易于被获取，以及数据易于被更新、维护、管理的程度 |
| 22 | 可靠性 | 在特定条件下，能保持某个性能等级的程度 |
| 23 | 新鲜度 | 包含一系列的质量因素，其中每个因素代表某方面的新鲜度 |
| 24 | 附加价值性 | 在使用数据的过程中获取收益的程度 |
| 25 | 易学习性 | 数据易于被用户学习的程度 |
| 26 | 衰败率 | 数据发生消极变化的程度 |
| 27 | 简洁度 | 数据被简洁表示的程度(不过分简略，完整且切题) |
| 28 | 同步性 | 在不同应用或系统中数据的等价程度，以及使数据等价化的便捷 程度 |
| 29 | 完备性 | 衡量数据的存在性、有效性、结构良好性及其他基本特性的指标 |
| 30 | 导航度 | 数据易于被发现及连接的程度 |
| 31 | 高效性 | 数据供给当前工作应用的便捷程度 |
| 32 | 可用性 | 在物理上数据可利用的程度 |
| 33 | 覆盖率 | 有效且完整的数据占总体数据的比率 |
| 34 | 交易度 | 数据能够产生商业交易或结果的程度 |
| 35 | 时效度&可用度 | 数据的实时程度及在特定条件下的可用程度 |
| 36 | 易变性 | 数据在现实世界有效的时间段长度 |

**6.1.2** **数据质量问题**

**1.** **数据质量问题的起源**

从前文给出的数据质量的如此多种类的定义我们可以猜想到，引发数据质量问题 的因素也是多种多样的，涉及信息系统、组织架构、制度流程、企业文化等多个方面。 我们梳理了一些较为普遍的原因来帮助读者理解为什么会出现数据质量问题。

来源不一致：大数据时代，同一个现实世界中的对象可能在不同的场景中出现， 每个场景对应的数据源提供的数据未必及时、准确、 一致，这一点我们在数据集成中 已讨论过了。

背景知识不一致：在数据采集和录入的过程中，“人”发挥了很大的作用，参与采 集和录入的人的背景知识强弱会极大地影响到采集和录入的数据质量。

计算和存储资源受限：在计算和存储资源不足的情况下，数据的精确性、完整性、 时效性都会受影响。比如，出于资源方面的考虑，企业可能选择只在每个星期的星期 五存储本星期的数据，这就会导致一部分数据丢失或过时。

安全性要求较高：出于数据安全方面的考虑，精确度的牺牲经常是不可避免的。 例如，差分隐私 (Differential Privacy) 就是通过在可接受的范围内降低数据质量来保护

 86 

**第6章**|数据质量管理

用户隐私的。

数据本身的二义性：在一些应用中，如许多交叉领域的应用，数据本身的二义性 不可避免，描述不同的语义或者同一种语义有不同表达是普遍存在的，在这些情况下， 即使是让人来提高数据质量都是很困难的，更不用提采用机器自动化的手段了。

数据语义的复杂性：很多非结构化数据，如视频、文本、语音等，本身包含复杂 的语义，对这些数据完成语义转换和映射是很困难的，因此在处理的过程中，数据质 量的降低几乎不可避免。

管理手段不恰当：对数据的管理过于严苛或者过于疏松都可能导致数据质量下降， 而如何选择最恰当的管理手段是一件很困难的事情。

**2.** **数据质量问题的防范和修复**

针对上述每一种原因，都有一定的手段可以防范或者修复劣质数据。在本章的后 续部分，我们会介绍几种重要的提升数据质量的技术，包括缺失值填充、实体识别、 真值发现、错误检测与修复等。但读者在阅读后续部分之前，需要明确的一点是，以 目前技术的发展程度，想要完全依赖机器完成自动化的数据质量提升是非常困难的， 很多时候我们不得不选用手工和机器结合的半自动化方式。

我们先简单介绍一下数据质量管理中会用到的基本概念。

数据质量管理是指对数据进行全面质量管理，涉及技术、规则、组织、流程、评 价考核等多个方面，目标是针对一个或多个所需要的维度，及时发现并解决数据质量 问题，提升数据对应维度的数据质量。

请注意我们在本章的定义中强调了数据质量管理必须针对一个或多个所需要的维 度来实施，这是因为指望设计一套普适的、能全面管理和提升数据质量的机制或技术 并不现实，作为数据的拥有者和管理者，必须要在重要、次重要、不重要的维度间有 所权衡。例如，为了消除数据中的不一致，可能不得不将有冲突的数据标记为 “Null” 来等待正确值的填充，这必然影响数据的完整性，此时就要在数据一致性和数据完整 性这两个数据质量维度之间加以权衡。

当前，由于互联网的兴起和产业的数字化，各种数据量急剧增长，数据作为一种 资产，其质量越来越不可忽视。如何有效地实施数据质量管理也越来越受到各行各业 的重视。

数据质量管理的可分为人工管理、半自动化管理、自动化管理三个层次。

人工管理：完全手工实现，管理的效果取决于数据质量管理者的时间、工作态度、 对于数据的熟悉程度，以及背景知识的充分程度等。这对于数据量较小的情况是适用 的，但在大数据时代，完全使用人工管理并不现实。

半自动化管理：通过制定规则、程序比对、统计分析等方法，先完成基本的劣质 数据检测、筛选、处理，将结果转给人工管理者，由人工管理者决定最终要采取的措 施并反馈给自动化管理模块。这种方式虽然还没有脱离人工，但自动化的方式在数据 量很大的时候可以极大地减少工作量。这也是目前比较适合大部分组织的一种方法。

自动化管理：完全依靠自动化和智能化系统完成数据质量的评估和管理。这种方 式是目前技术发展的方向，但由于现有技术仍无法完全保证管理的准确度，所以对于 那些特别重要的数据应该慎用自动化管理。

**大数据治理：**理论与方法

**6.2** **数据质量评估**

近年来，数据质量评估已经成了数据质量管理领域中的研究热点之一。大量国内 外研究机构都对该问题从不同角度进行了研究。我们在本节简单讨论数据质量中比较 常见的五个维度的评估技术，包括数据一致性评估、数据完整性评估、数据时效性评 估、数据精确性评估、实体同一性评估。

**6.2.1** **数据一致性评估**

对数据一致性的研究是为了保证数据不违背特定场景下的语义约束。评估数据一 致性，直观上就是要了解数据集中究竟有多少数据的表述是不存在矛盾的。例如，某 人的婚姻状态一栏的值是未婚，但配偶一栏又不为空，这就存在矛盾，因为按照一般 意义的理解，未婚状态的人是没有(法律意义上的)配偶的，配偶一栏应该为空或填 写“无”。容易看出，数据一致性的评估非常依赖于特定的领域知识，因为如果我们不 具备与“婚姻状态”和“配偶”这两个属性有关的背景知识，就无法判断数据中是否 存在矛盾。

领域知识通常可以用规则或约束的形式表示。数据库中最常见的函数依赖 (Func- tional Dependency) 就是典型的可用于数据一致性评估的规则。

举例来说，邮编和城市之间就存在函数依赖关系“邮编→城市”,也就是说，数据 库中只要两条记录的邮编相同，那么城市就必然相同。在这种约束下，如果数据集中 出现了两条记录t₁ 和 t₂,t₁ [邮编]=t₂ [邮编]=510000,而t₁ [城市]=广州，t₂ [ 城 市]=深圳，那么就出现了数据不一致的质量问题。

基于函数依赖，可以对数据集进行扫描，确定不一致数据项的比例，进而完成数 据一致性评估

不过，函数依赖仅能检测到数据中的部分不一致错误，很多数据一致性问题无法 通过函数依赖表达和检测。因此， 一些其他的一致性约束被提出用于更精确地评估数 据一致性。条件函数依赖 (Conditional Functional Dependency)① 可以表示部分数据上的 数据依赖关系，从而可被用来评估数据一致性以及检测修复不一致数据。

条件函数依赖的定义与函数依赖非常接近，可以视为函数依赖的扩展。基本的形 式为 (X→Y,T,), 其 中X和 Y是两个属性子集，X→Y 是标准的函数依赖，T。是关 于X 和 Y的模式表 (Pattern Tableau), 用于约束X 和Y的取值。图6-1是一个条件函数 依赖的例子。函数依赖φ指明了“员工”“部门”“项目”三个属性之间的关系，其中 [员工，部门] → [项目]是一个普通的函数依赖，模式表 T, 表明了只有在“部门” 为A的时候这三个属性才受到该依赖的约束。也就是说，当“部门”属性值为A 时， 员工属性值才可以决定项目属性值。该约束想要表达的语义是“部门A 的员工都在唯 一的项目中”。换言之，不在部门A的员工存在于两个或两个以上的项目中，如果发现

① Fan W,Geerts F,Jia X,et al.Conditional functional dependencies for capturing data inconsistencies [J].ACM Transactions on Database Systems(TODS),2008,33(2):6.

第6章|数据质量管理

*T,;*

一个员工对应了两个项目，则说明数据中存在不一致。当 T, 比较简单时，也可以将 T。 表示的条件压缩到“→ ”的两端，例如，图6-1所示的条件函数依赖可以改写为φ:(员 工，部门=“A”→ 项目)。

φ:([员工，部门]→ [项目],T)

|  |  |  |
| --- | --- | --- |
| 员工 | 部门 | 项 目 |
| A | |  |

图6-1 条件函数依赖示例

由于条件函数依赖可以表达函数依赖所不能表达的语义，所以可以用来发现函数 依赖不能发现的数据不一致错误，从而更精准地评估数据一致性。基于不一致数据的 比例，可以完成对数据一致性的评估。

6.2.2 数据完整性评估

对数据完整性的研究是为了保证数据中不存在缺失值。在最简单的情况下，可以 直接对数据集进行扫描，检查数据缺失的比例，完成数据完整性评估。例如，在关系

数据库中，可以直接扫描数据集D, 计算NULL的比例，用作为 D的

完整性评估结果，其中|D| 是D 中数据项的个数，count(NULL) 是 D 中为NULL 的数

据项个数。

上述评估方法虽然简单，但过于粗糙，因为有些 NULL 值可能是“伪缺失值”。具 体来说，这些数据项虽然值为 NULL, 但并不一定意味着数据缺失。这可能有下述两种 情况。

(1)数据值本身就应该为空。例如，如果婚姻状态是“未婚”,那么配偶一栏就应 当为空，此时填NULL是完全合理的，并不应该被视为数据缺失。这些 NULL 值如果被 计算进完整性评估结果，则会低估数据的完整性。

(2)数据虽然缺失，但可以通过其他手段正确填补。例如，在中国大陆地区，如 果城市为NULL, 但是邮编不空，就可以根据邮编和城市的对应关系，将城市补全完 整。又或者用户的生日信息为空，但是身份证号是完整的，那么也可以根据身份证号 来补全生日信息。对于这种情况，可以先尝试补全能够补全的数据，再进行完整性评 估，以免低估数据完整性。具体的填补方式，我们在6.3节缺失值填充中会进行讨论。

为了处理上述两种情况，也需要向数据中引入语义规则，我们在数据一致性部分 讨论过的函数依赖或条件函数依赖在此也可以发挥作用。举例来说，在考虑语义规则 和不考虑语义规则的两种情况下，图6-2的数据完整性也会得到两种不同的计算结果。

例6-1 考虑图6-2所示的数据表，其中包含两条记录，共包含14个数据项，其中

有5个数据项为空 ( NULL) 。若. 计算，则数据完整性评估结

果为 



大数据治理：理论与方法

…………………………………

我们知道婚姻状态是“离异”或“未婚”都表示没有配偶，因此配偶应当为空。 同时，如果存在条件函数依赖(邮编=510000 →城市=广州),那么我们可以将图中两 条记录的城市都补充为“广州”。此时图6-2中的数据表中仅有1个缺失的数据项，其

完整度应为

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| ID | 姓名 | 性别 | 城市 | 邮编 | 婚姻状态 | 配偶 |
| 001 | 张三 |  |  | 510000 | 离异 |  |
| 002 | 李四 | 男 |  | 510000 | 未婚 |  |

图6-2 数据完整性评估示例

可以看出，是否适用语义规则可能会很大程度影响数据完整性的评估结果，在使 用了恰当的语义规则后，完整性评估结果的准确度会提升。不过，语义规则需要额外 的获取代价，这些语义规则通常有两种获取途径：

(1)由领域专家给出；

(2)采用类似于挖掘关联规则的方法从高质量的数据集中挖掘得出。如果领域专 家足够专业并了解数据，那么领域专家给出的规则往往会具有较高的质量，能较多地 提升数据完整性的评估准确度。但当数据量非常大时，手工书写规则的工作量很大， 可能很难完整覆盖整个数据集。自动挖掘的方法节省了人力成本，但是存在两个问题： 首先，必须要找到一个“恰当的”高质量数据集用于挖掘规则，再“迁移”到要评估 的数据集上，这样的高质量数据集未必易于获得；其次，挖掘出来的规则可能本身也 存在着质量问题，例如，规则因果关系倒置或规则之间存在冲突。

很多时候，上述方法评估得到的结果还未必准确，因为数据未必能够满足封闭世 界假设。所谓封闭世界假设，是指“非已知的事物都为假”的假设。若将该假定应用 于数据完整性评估这个问题上，则可以解释为“所有应该存在的记录都包含于当前数 据集”。那么,我们只需要考虑当前数据集中已有的记录是否完整，对其做数据完整性 度量即可。例如，在图6-2所示的数据表中，我们只需要检查ID 为001和002的两条 记录是否完整，而无须考虑是否还存在一条完全丢失的ID 为003的记录。与封闭世界 假设相对应的是开放世界假设，其认为“未知的事物就是未知”。那么在评估图6-2所 示的数据集完整性时，我们还会额外思考是否存在一些记录是整条缺失的。

包含依赖指明了多个数据表之间的关系，因此可以被用来度量数据完整性。包含 依赖通常表示关系之间的关系(也可以理解为不同的数据集之间的关系)。例如，我们 可以针对图书采购和展览相关的数据集写出包含依赖φ,其中图书展览表和图书采购 表的模式如图6-3所示。φ的语义是，图书展览表的ISBN 列的所有值都应当出现在图书 采购表的 ISBN 列中。这条规则反映了现实应用场景中的一条规则，在该应用场景中， 图书必须首先被采购，然后才可以被展览。根据φ,如果我们发现某一个 ISBN 出现在 了图书展览表中，但并没有出现在图书采购表中，那么就可以考虑图书采购表中的数 据有缺失。

φ:图书展览.ISBNC 图书采购.ISBN

 90

**第6章**| **数据质量管理**

图书采购表：

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ID | ISRN | 书名 | 出版社 | 作者 | 价格 |
|  |  |  |  |  |  |

图书展览表：

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ID | ISBN | 书名 | 陈列楼层 | 陈列展室 | 行列 |
|  |  |  |  |  |  |

图6-3 包含依赖示例

除包含依赖之外，也可以考虑使用条件包含依赖①来进行数据完整性评估。条件包 含依赖是对包含依赖的扩展，其与条件函数依赖类似，在规则两端增加了取值条件， 因此可以适用于更广阔的场景。其作为数据质量约束可以表示多表之间数据的部分依 赖关系， 一方面可以衡量数据一致性，另一方面，由于其能够指明某些值是必须存在 的，还常被用于度量数据完整性。

**6.2.3** **数据时效性评估**

对数据时效性的研究是为了保证数据不陈旧过时。过时数据是一类比较特别的劣 质数据 它们曾经是优质数据，但随着时间的推移，数据渐渐变得不准确，甚至失 效。因此，在发现过时数据与评估数据时效性时，对于时间戳的利用显得尤为重要。

在理想情况下，数据库中对每个值都存有完整可用的时间戳，这不仅包括数据的 创建时间，也包括数据的截止时间、有效期、生命周期等。在理想情况下，我们可以 基于时间戳来统计数据库中有多少值是已经过时失效的，进而通过过时失效数据的比 例来评估整个数据集的时效性。

不过，在很多情况下，完整、可用、精确的时间戳可能会无法获得，或者获取代 价极高。实际应用中，很多数据库并不会对每个值都存有完整有效的时间戳，即使存 在时间戳，出于数据集成、数据演变、数据格式转换等原因，数据库中的不同值语义 不同，因此其随时间变化的频率科学也不尽相同，这使得同一时刻不同属性的新旧程 度也不相同，时间戳可能缺乏及时有效的维护而变得不可用或不精确。这些问题都使 得对于数据集的数据时效性评估往往要在时间戳不完整或不可用的情况下展开。为此， 应当有不完全依赖于时间戳的数据时效性评估方法，来判定数据时效性。

我们可以将数据时效性划分为两类：绝对时效性和相对时效性。其中，绝对时效 性可以对给定的数据库，形式化地评估单个数据项、元组及数据库整体的时效性；相 对时效性则针对数据库上的特定查询或分析需求，度量数据库相对于查询或特定分析 需求的时效性。以下我们简单讨论两种时效性的评估方法。

**1.** **绝对时效性评估**

绝对时效性的目标是对于给定时刻(通常是当前时刻),判定数据库中的值、记

① Bravo L,Fan W,Ma S.Extending dependencies with conditions [C]//Proceedings of the 33rd international con- ference on Very large data bases.Trondheim,Norway:VLDB Endowment,2007:243-254.

大数据治理： 理论与方法…

录、数据集等是否过时失效，在无法精确判定“过时”或“非过时”的0/1状态时， 尝试量化其过时失效的程度、可能性或确定度。

在数据库中有完整精确的时间戳时，基于时间戳可以完成绝对时效性判定，例如， 可以利用事务时间并结合数值的有效性衰减函数来推断数据的有效时间，进而判定数 据是否过时失效。例如，可以假设数据库中任意值v 的有效性随时间的减弱程度可以用 衰减函数 decline(v)刻画，v 的“年龄”(从v 进入数据库到指定时刻的时长)可以用 age(v) 刻画，那么v 的时效性就可以被定义为exp(-decline(n)xage(n)) 。 这样虽然可 以针对几乎任意指定时刻判定数据是否过时失效，但需要大量的辅助信息，我们需要 保存并维护数据库中每个值的事务时间和有效性衰减函数，这对于数据量大、数据语 义复杂的场景来说并不现实，因此，基于时间戳的绝对时效性判定方法的适用范围非 常有限。

一种可能的方法是基于规则来判定数据时效性，与一致性和完整性类似，如果能 够由领域专家给出或从数据集中挖掘出适用的判定规则，那么就会使所需要的辅助信 息极大减少。规则来源于与领域相关的背景知识。我们可以用下面这个例子来说明这 种方法的大致思想。

例6-2 如果我们知道某公司员工Alice的工资有6000和8000 两个值，这两个值都 并非虚假值，我们只是不知道哪个值陈旧过时了，此时，如果有领域专家给出一条规 则“工资只升不降”,那么当存在多个工资时，我们就可以通过比较大小来判断哪个值 更新。在Alice这个例子中，8000是更高的工资，所以我们就可以知道8000 是较新的 工资。此外，如果我们知道2009年之前入职该公司的员工的工资在2019年都已经涨到 9000以上了，而Alice是2007年入职的，那么又可以进一步推断8000也不是 Alice的 最新工资，而是一个在2019年就已经过时的工资。

我们将上面规则形式化一下，可以得到两条规则：

规则1: Ve,e,(e,[ID]=e,[ID]Ae₁[Salary]<e,[Salary]→e <sam⁶); 规则2:Ve(e₁[EntryTime]<2009 Ae;[Salary]<9000→e,<sshmy2019)。

在上述规则中，符号<₄和>₄定义了属性值之间或属性值和时刻之间的新旧关 系。例如，e,<sum, 表示元组 e, 的 Salary 属性值相较于e, 的 Salary 属性值要旧， e₁<shm,2019 表示元组 e, 的 Salary 属性值相较于时刻2009要旧。

将规则抽象一下，我们可以得到两种时效规则的表示形式。

形式1:Ve,e(ψ→e,<e)。

“→ ”左边表示规则成立的条件，右边则表示条件成立后由规则可以推出的结论。 其中，条件ψ是以下四种谓词的合取形式：

(1)e₁<a⁶₇ ( 或e<ge₁);

(2)e₄[B]=b(>b 或 <b), 其中B∈R,b 是一个常量，k=i 或j;

(3)e₁[B]=e[B](e₁[B]<e;[B] 或e₁[B]<e[B]);

(4)e₄<aT(e₄>aT), 其中k=i 或j,r 是一个表示时刻的常量。

形式2:Ve(ψ→eO₄T)。

条件ψ是以下两种谓词的合取形式：

(1)e[B]=b(>b, 或 <b), 其中B∈R,b 是一个常量；

92

第6章|数据质量管理

(2)e<gr ', 其 中k=i 或j,r '是一个表示时刻的常量。结论中的符号◎₄等于 <4 或 >ac

在某些场景下，用于评估时效性的知识可能并没有那么确定，此时，我们还可以 给规则加上确定度，改写为下述形式：

Ve,e,(ψ→e₁<∈j,ξ(r)) 和 Ve(ψ→e<₄r,ξ(r))

其中，ξ(r) 表示当规则左边条件为真时，右边结论为真的确定度。

基于上述规则，我们可以评估属性值、元组(数据记录)以及数据集的时效性。

对于属性值e,[A] 来说，针对给定时刻θ评估其时效性，就是要判定e,是否在A,属 性上比θ新。我们可以直接利用规则推理，看是否可以推出e;>₄θ 这样的结论，如果 所使用的规则是确定的，那么其时效性要么是0(过时),要么是1(不过时)。如果所 使用的规则是不确定的，那么其时效性等于e,>₄θ 这一结论成立的确定度。

对于元组和规则来说，我们可以使用属性值时效性的平均作为其时效性的评估结 果。元组e 的时效性记为 cur(e), 则其值等于 e 的所有数据项的时效性的平均，即

),其中m 是属性的个数。数据集D 的时效性记为eur(D), 其 值等于D 中所有元组的时效性平均，即, 其 中 |D1 表 示D 中 元组的个数

2. 相对时效性评估

相对时效性的目标的评估数据相对于特定应用场景来说的时效性。例如，用户可 能特别关心某些重要查询的查询结果的时效性，或者用户只想知道其最常使用的数据 集的时效性。此时，绝对时效性可能无法直接判定，又或者判定结果不能有效地表达 用户需求，但我们可以通过某些局部信息，来评估特定集合的数据时效性。

我们可以根据时效性评估要参考的对象对数据时效性做进一步细分。当用户发出 某些特定查询时，由于数据中可能存在过时数据，因此查询结果也可能过时。此时如 果能够将查询结果及相对应的结果时效性一同返回给用户，那么对于用户来说是非常 有帮助的。我们可以将数据相对于查询的时效性称为查询相关时效性。类似地，不同 的用户对于同一批数据的分析和使用也各有侧重。例如，人力资源部门的员工可能会 频繁查询员工的级别和工资信息，后勤部门的员工则更关心员工的办公场所、用地面 积等信息。可以将数据相对于用户的时效性称为用户相关时效性，用户相关时效性反 映了数据集对于特定用户来说，其时效性如何。

评估查询相关时效性首先需要明确查询涉及哪些记录的哪些属性上的时效性评估。 对于查询Q, 在进行时效性评估时，我们可以对齐进行抽象，将Q涉及的实体记为e, Q 涉及的属性集合记为Attr2。对于集合Attr。中的任意属性A, 记录集合T, 称为A,相 对于Q的最新记录集合，对于VteT, 记录t 满足下述两个条件：

(1)t[EID] 等于e 的 ID;

(2)不存在记录s, 使得根据R 在A,上的时效约束能够推出t<₄S。

理想情况下|T₄ |=1, 也就是说，我们能找到唯一的最新值。但实际上，由于知识 不完备、数据不完整数据冗余等原因，我们可能会找到不止一个最新值的候选值，此



大数据治理：理论与方法

……………………………………

时如果没有更进一步的知识指导我们的选择，就无法再区分T 中存在的多个候选值的 时效性了。设T 中的记录共有cnt(T) 种不同的A,值，那么此时查询Q 在A,上的最

新值的确定度就为



我们考虑Attr。中的所有属性，那么Q的时效性是curo=



用户相关时效性的评估可以基于查询相关时效性的评估结果来计算。例如，我们 可以将用户分为3类：第1类用户对查询和属性均没有偏好；第2类用户对特定属性的 时效性要求更高；第3类用户对特定查询的结果时效性要求更高。对于第1类用户， 我们可以尝试直接用绝对时效性作为其用户相关时效性；对于第2类用户，可以采用 查询相关时效性类似的方法来评估其关心的属性集合上的时效性；对于第3类用户， 则可以对其关心的每个查询都评估时效性，最终采用平均值或加权平均值作为其时效 性评估结果。

6.2.4 数据精确性评估

对数据精确性的研究是为了保证数据能够准确地描述对应实体。数据精确性评估 主要考察数据相对于某个标准是否能足够准确地描述对象。例如，某城市的人口数可 以记录为1531万人或1530.59万人，这两个数据在精确性上是有差别的。

和数据时效性类似，数据精确性需要有明确的评价标准。例如，如果在我们的应 用场景中，要求人口数量以“万人”为单位并精确到小数点后2位，那么1531万人显 然是一个不够精确的数据。和前面的数据质量评估类似，我们可以扫描数据集中所有 存在精确度要求的数据，统计不符合精确度要求的数据量，并以符合精确度要求的数 据的百分比作为最后的精确度评估结果。

很多情况下，导致数据不精确的未必是质量问题。在之前的例子中，不知道数据 的具体应用场景时，我们很难说1531万人这个数据是有质量问题的，也未必能保证 1530.59万人这个数据就一定满足精确度要求。所以对于精确性的评估，要放在具体应 用值来看，这更类似与我们前文讨论过的相对时效性。如果数据本身不太精确，而查 询要求返回非常精准的结果，就会导致查询结果不对。

如果数据中存在多个精确度不同的副本，那么在查询精确度要求较高时，我们可 以尝试在数据集中先寻找满足精确度要求的数据副本，再基于这些数据副本计算查询 结果，并度量查询结果的精确性。为了解决这种不精确数据引发的问题，有研究关注 来源不同但语义相同的属性值之间的精确性偏序关系，并给出了数据精确性规则的语 法和语义，用于度量、检测和修复数据精确性问题①。

6.2.5 实体同一性评估

对实体同一性的研究是为了保证描述同一实体的数据是一致的。例如，如图6-4所

①Cao Y,FanWF,Yu WY.Determining the relative accuracy of attributes [C]///Proceedings of SIGMOD.New York,NY,USA:ACM,2013.565-576.

第6章|数据质量管理

示，如果数据中存在两条关于用户张三的记录，两条记录的身份证号相同(因此可以 肯定是同一个人),但性别不同，这就出现了实体同一性问题。

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 姓名 | 身份证号 | 性别 | 生日 |
| 001 | 张三 | 123456200101017890 | 男 | 2001-01-01 |
| · | …… | …… | …… | · |
| 102 | 张三 | 123456200101017890 | 女 | 2001-01-10 |

图6-4 描述同一实体的两条记录

实体同一性问题看起来与数据一致性问题非常类似，但二者的关注点不同。数据 一致性评估关注数据集在整体上是否存在不一致数据，而实体同一性则专门关注同一 实体的描述信息中是否存在矛盾。在图6-4所示的例子中，发现实体同一性问题后，首 先需要判断序号001 和序号102的两条记录是否描述了同一实体。由于身份证号的唯一 性，我们根据身份证号相同判断序号001和102的两条记录描述了同一实体(张三) 的信息。基于这一结论，我们继续检查这两条记录针对张三的描述是否存在矛盾。可 以看出，现在数据集中对张三这个人的描述存在两处矛盾：

(1)性别不同；

(2) 生日不同。

考虑关于张三共有5个属性，其中两个属性存在矛盾，如果我们使用简单的数据

一致性的比例来定义实体同一性，则实体同一性评估结果为 

其他数据质量维度的评估或修复或许会影响实体同一性的评估结果，例如，数据 一致性规则可能指明了如何通过身份证号来判断性别和生日是否正确(例如，我国的 身份证号的第7~13位指明了生日，身份证号的倒数第二位为奇数指明男性，为偶数指 明女性)。那么我们可以根据张三的身份证号获知，张三的生日为2001-01-01,性别为 男。也就是说，编号001记录中的信息是完全正确的，编号102记录中的“性别”和 “生日”这两个属性值应该分别修改为“男”和“2001-01-01”。对数据进行修复之后， 我们再评估实体同一性，得到的结果就是1,因为张三的两条记录完全一致了。

实体同一性的评估效果有很大一部分依赖于我们是否能够准确发现哪些记录描述 了同一实体，这一问题我们在后续的6.4节实体识别与真值发现中会专门讨论，在此 不做赘述。

6.3 缺失值填充

6.3.1 什么是缺失值

在各种实用数据库中，属性值缺失的情况经常发生，甚至是不可避免的。数据采 集时可能漏采、传输时可能丢失、共享时可能被隐去，因此，在大多数情况下，信息 系统中的数据都存在某种程度的缺失。

缺失值的产生的原因多种多样，既可能因为软硬件原因导致数据缺失，如存储器 损坏、传输丢失、软件误删等；也可能因为人的主观失误、历史局限或有意隐瞒造成

大数据治理：理论与方法…………………

数据缺失，如在市场调查中被访人拒绝透露信息或者数据录入人员漏录了数据等。造 成缺失值的原因是多方面的，主要可能有以下几种。

数据难以获取：在数据采集的过程中，很多数据并不是“立等可取”的，采集数 据时往往存在各种原因导致时延；又或者出于权限、隐私保护等原因， 一些数据不能 被采集。这些难以获取的数据体现在信息系统中，就是缺失值。

数据采集时有侧重：出于时间、空间、金钱等方面开销的考虑，在采集数据时会 对数据的重要性有所权衡，那些代价高昂但是作用较小的数据可能会被忽略掉，进而 体现为信息系统中的缺失值。

数据在流动或使用时有丢失：随着信息系统的发展和使用，数据可能会被修改、 转移或共享，在数据流动过程中，出于技术、制度、流程、操作等原因， 一些数据可 能丢失，进而体现为信息系统中的缺失值。

出于上述原因，数据缺失的方式可以大致分为三类。

完全随机缺失：缺失值与当前信息系统中的其他值无关。

完全依赖缺失：缺失值与信息系统中的某些值有完全依赖关系，例如，“邮编”完 全依赖于“城市”和“街道”,当“城市”和“街道”完整时，这种完全依赖缺失值 可以被自动补全。

部分依赖缺失：缺失值与系统中的某些值有依赖关系，但是这种依赖是部分的或 者有概率的，例如，学生的“学号”部分依赖于其“院系”,当“学号”缺失时，如 果“院系”完整，则可以补全其部分学号。

除了上述分类，数据缺失还可以分为随机缺失、不可忽略缺失、单值缺失、组合 缺失等。了解缺失值产生的原因和数据缺失机制后，下一小节我们将讨论缺失值处理 方法。

**6.3.2** **缺失值处理方法**

大致说来，处理缺失值的操作主要有三类：删除、填充、忽略。忽略比较简单， 这里主要讨论如何删除和填充。

**1.** **删除**

由于值本身是缺失的，所以不存在删除“单个值”这种操作，我们在此需要讨论 是删除记录，还是删除属性。对于关系型数据库，记录和属性分别对应着数据表的行 和列。判断如何删除的方法比较简单，基本根据记录或者属性的完整程度来确定。如 果某个属性的大部分值都缺失，那么该属性就可以删除，提高数据集的完整性。对于 记录来说同理，如果某条记录的大部分属性都缺失，那么这条记录可以删除。

删除这种处理方式非常简单，但其存在的问题也很明显。在删除数据集中的数据 的时候往往需要特别谨慎，很多场景下，即使大部分数据是缺失的，记录或属性也不 能随便删除。同时，删除数据可能会丢失数据的隐藏语义，进而使得某些数据分析结 果不准确。在一些场景下，删除记录或属性后还需要对相关的其他记录和属性进行维 护，这可能会导致增大维护的开销。

综上所述，删除适用于数据关键性较低、关联关系不大，且数据缺失较为集中的 简单情况。

第6章|数据质量管理

2. 填 充

比较理想的方式是通过某些流程或者技术对缺失值进行填充，常见的方法如下。

重新采集：所谓重新采集，就是通过之前使用过的数据采集手段重新采集数据。 这种方式适合数据采集代价较低，且可以复现的场景。例如，某些用户的基本信息缺 失时，可以直接要求用户重新补全数据。但在另一些场景下，重新采集则无法补全数 据。例如，如果数据采集是和时间密切相关的，如物联网采集的温度、湿度等信息， 对于这些数据来说， 一旦错过了某个时刻，数据将无法通过重新采集补全，只能尝试 其他的补全方法。

默认值填充：这是一种较为常见的自动填充方法。例如，将缺失的部分填为 “Null”“0”“+x”“-o” 等。这种方式非常简单易用，但是在一些情况下，完全使 用默认值填充可能会造成很严重的问题。美国曾经报道过，有人因为将自己的车牌自 定义为 “Null” (美国允许使用自己定义的车牌)而接到了“天价”罚单，因为在数据 库中，所有缺失的车牌都使用 “Null” 填充了。①

统计填充：通过统计的方法来填充数据。例如，如果我们对某些数据的分布有先 验知识，那么就可以根据分布选择概率较大的值进行填充。简单的填充方法包括使用 平均数、中位数、众数、最大值、最小值等进行填充，也有一些稍微复杂的填充方法。

举例来说，平均数/众数填充的说明如下。对于数值型的属性来说，可以直接使用 未缺失属性的平均值对其进行填充，对于非数值型的属性来说，可根据统计学中的众 数原理，用该属性在其他所有对象出现频率最高的值来补齐缺失的属性值。

热卡填充(或就近补齐):对于一个包含空值的对象o, 在数据库中找到一个与它 最相似的对象o', 然后用o '的值来填充o 的对应位置的值。具体的相似度根据实际应 用场景有不同的定义，表6-2给出了一些常见的相似度定义的例子。我们在这里只给出 了简单的计算公式，如果需要详细了解，读者可能还需要进一步查阅相关资料。这些 相似度定义适用于不同的应用场景，需要根据具体需求选择。容易看出，该方法潜在 地使用了一个假设，即“完整部分相似的对象在缺失部分也相似”,因此，如果要使用 这种方法填充，必须保证该假设是成立的。所以还需要研究属性的语义，综合考虑属 性之间的相关程度，使用相关的属性来辅助完成填充。

表6-2 常见的相似度定义举例

|  |  |  |
| --- | --- | --- |
| 序号 | 相 似 度 | 计 算 公 式 |
|  | 欧几里得距离 |  |
| 2 | 余弦相似度 |  |
| 3 | 相关系数 |  |

① InfoQ网站， 一张 “Null” 车牌引发的天价罚单。

97

大数据治理：理论与方法

(续表)

|  |  |  |
| --- | --- | --- |
| **序号** | **相** **似** **度** | **计** **算** **公** **式** |
| 4 | 指数相似系数 |  |
| 5 | 编辑距离(Levenshtein距离) | 将一个字符串变为另一个字符串所需的最小操作步数，可选的 操作有插入、删除、替换 |
| 6 | 皮尔逊相关系数 |  |

期望最大化 (Expectation Maximization,EM):EM 算法是一种在不完全数据情况下 计算极大似然估计或者后验分布的迭代算法。在每一次迭代循环过程中交替执行两个 步骤——E 步和M 步，算法在E 步和M 步之间不断迭代直至收敛，即在两次迭代之间 的参数变化小于一个预先给定的阈值时结束。该方法可能会陷入局部极值，收敛速度 也不快，并且计算很复杂。E 步和M 步的大致作用如下。

E 步 (Expectation, 期望步):在给定完全数据和前一次迭代所得到的参数估计的 情况下，计算完全数据对应的对数似然函数的条件期望；

M步 (Maximization, 极大化步):用极大化对数似然函数确定参数的值，并用于 下一步的迭代。

预测填充：通过对现有数据建模，预测缺失位置可能的值，用预测结果来填充缺 失值。这种方法比较复杂，但取得的效果也比较好。预测填充和统计填充在某种程度 上存在重叠，因为其同样依赖于数据类型和数据分布，但是这里我们单独将预测填充 作为一类方法。这类方法通常采用机器学习算法，通过对属性的语义或类型进行较为 细致的分析，可以对不同的属性选用合适的方法来学习出可能的值。

对于存在有限取值范围的类别属性，如职业、性别等，可以将属性作为标签，记 录作为训练数据，通过训练分类器进行填充。常用的分类算法如支持向量机 (SVM)、

朴素贝叶斯等。

对于有明确物理意义的数值型属性，如温湿度、电压等，可以采用回归的方法进 行填充，已知值通常作为自变量，待填充的缺失值作为因变量。

对于存在一定关联的属性，还可以建立关联分析模型，通过对相关记录、相关属 性和目标待填充属性进行建模，完成属性的填充。

预测填充对于缺失位置较为集中的情况可能比较有用，例如，在人口数据中，如 果大部分人的“职业”“年龄”“性别”等基本信息是完整的，而“收入”数据出现了 相对集中的缺失，那么就可以对较完整的“职业”“年龄”“性别”和“收入”来建 模，基于“职业”“年龄”“性别”来完成“收入”属性的补全。但是，如果缺失值比 较分散，那么可能要建立许多个模型来完成填充，这可能会带来较大的时间、空间和 金钱的开销，而且预测也容易不准确。

6.3.3 缺失值处理例析

下面给出一些缺失值处理的例子以帮助读者更直观地了解如何处理缺失值。表6-3

98

**第6章** | **数据质量管理**

给出了一些缺失属性及处理方法。

**表6-3** **缺失属性和简单的属性分析及处理方法举例**

|  |  |  |
| --- | --- | --- |
| 序号 | 属 性 | **属性分析及处理方法** |
| 1 | 年收入 | 属性分析：数值型，涉及个人隐私，可基于职业、年龄等信息推断  处理方法：有个人历史信息的场景可使用预测填充(回归模型);有大量相似记录时  可以使用热卡填充或预测填充；商品推荐场景可使用平均值填充；借贷场景可使用最小 值填充；如非必须填充，出于隐私保护的角度考虑，可不处理(忽略) |
| 2 | 运动轨迹坐标 | 属性分析：可能是数值型的二元组/多元组，如经纬度坐标(x,y)或三维地图坐标  (x,y,h)等；也可能是脱敏后的字符串，例如， 一些场景下，运动轨迹可以通过用户接 入的移动基站来表示轨迹坐标  处理方法：可以在数据库中寻找相似轨迹从而使用热卡填充；也可以对坐标划分类别， 使用预测填充 |
| 3 | 人体寿命 | 属性分析：数值型，进行预测评估需要“病史”等隐私数据，进行预测建模相对困难  处理方法：保险费用估计场景下使用最大值填充；人口估计场景下使用平均值填充； 非必须填充的场景可以不处理 |
| 4 | 邮编 | 属性分析：字符串型，与“城市”“街道”属性之间存在依赖  处理方法：以中国地区的邮编为例，在“城市”“街道”已知的场景下，可基于依赖 关系补全详细的邮编；在只知道“城市”的场景下，可以基于属性依赖关系，使用城市 默认的邮编进行填充；在不存在相关属性的情况下，可使用默认值填充 |
| 5 | 地址 | 属性分析：字符串型，可以基于“工作单位”“邮编”等相关属性推断，但由于属性 间不存在绝对的依赖关系，所以只能近似推断  处理方式：在存在“工作单位”属性的场景下，可以使用工作单位的地址填充；在存 在“邮编”时，可以使用邮编对应的城市和街道填充。需要注意的是，以上填充结果均 是近似的，实际的真实值可能和填充结果相差甚远 |

表6-3中的例子都比较简单，在实际应用中，存在着大量更为复杂的场景。我们在 此给出两个更为复杂的例子。

例6-3 某公司有一份数据记录了客户是否最终购买了他们的产品及客户的一些个 人信息，其中属性包括“家庭年收入”“是否有房”“婚姻状况”,最后的类别为“是 否购买”。不过这份数据在“婚姻状况”的属性上存在着缺失值，但是公司需要使用 “婚姻状况”来进行后续的分析，因此必须对缺失值进行填充。

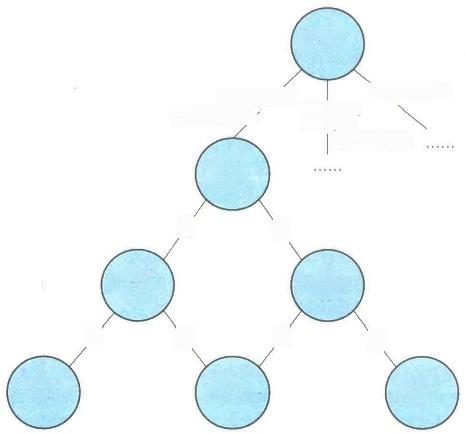
此时， 一种可行的思路是，我们可以将“婚姻状况” (未婚、已婚、离异)作为 分类标签，而将“是否购买”挪作一个已知的属性来构建一棵决策树，如图6-5所示。 也就是说，基于“家庭年收入”“是否有房”“是否购买”来预测“婚姻状况”,从而 对缺失值进行预测填充。这种方法要比我们随机填上“未婚”“已婚”“离异”,或者 填上默认的 “NULL” 效果要好。

如果将这个例子改动一下，假如属性中只在“家庭年收入”上有缺失值，那么还 可以按照表6-3第一行的分析，使用回归模型对数值型属性值进行填充。

例6-4 要填充某个人患有某种疾病的概率。我们已经有了一大批数据，其中测量 的属性包括是否经常锻炼，是否有健康的饮食，是否有心脏病，是否心口痛，是否有

大数据治理：理论与方法…………………

高血压，是否胸痛等，但在是否有心脏病属性上存在着缺失值。可以看出，这些属性 存在着相关性，不能直接使用朴素贝叶斯方法，于是我们可以考虑使用贝叶斯网络来 进行填充。首先构建一个包括上述属性的贝叶斯网络，然后根据其他属性来判断有心 脏病的概率。当概率大于设定的阈值时，我们就认为缺失值是有心脏病。



家庭

年收入

100万元以上 20万元至

*100万元*

(含100万元)

是否

有房

否

是 否

购 买

否

离异

20万元以下 (含20万元)

是否

购买

未婚

已婚

否

是

是

是

图6 - 5 基于决策树填充“婚姻状况”示例

不过，值得提醒的一点是，对于是否患有疾病这种较为敏感且容易出错的属性， 使用自动化填充要特别谨慎。如果有条件，尽量使用重新采集的方法来填充，例如 可以使用问卷询问用户本人或其家属。当不得不采用自动化填充时，在后续的分析步 骤中，可以尽量降低自动化填充的值的可信度，以免出现重大错误。

**6.4** **实体识别与真值发现**

**6.4.1** **什么是实体识别**

在日常生活中，人们每天都要从网络上的不同数据中检索所需要的信息。在检索 过程中会遇到的一个主要问题，就是不同的对象也许会具有相同的名字，或者相同的 对象会具有不同的名字。例如，在DBLP (一个计算机类英文文献的集成数据库系统) 中检索 “Wei Wang” 的文章，会检索到由14个 “Wei Wang” 发表的197篇文章。在 Allmusic中存在72首歌曲及3张专辑的名字为 “Forgotten”。相同的实体可能出现在截 然不同的文本中，而出现时往往会伴有大量的限制干扰信息，因此在上述情况下，人 们往往不能快速地获取他们想要的答案 。

实体识别是数据质量管理中一项重要的技术，实体识别的结果可以在数据质量管 理的各个阶段得到广泛的应用，如真值发现、不一致数据发现，去除冗余数据等。

实体识别是指，在给定的实体对象(包括实体名和各项属性)集合中，正确发现

 100

**第6章** |数据质量管理

不同的实体对象，并将其聚类，使得每个经过实体识别后得到的对象簇在现实世界中 指代的是同一个实体。实体识别要解决的问题主要包括以下两类。

元余问题：同一类实体可能由不同的名字指代，如名字王伟，用英文表示可能是 “Wang Wei”, 也可能是 “Wei Wang”。

重名问题：不同类的实体可能由相同的名字指代，如在 DBLP 中 检 索“Wei Wang”, 会检索到14个不同的作者。

针对不同类型的问题，实体识别中主要有两类技术。

冗余发现：用于处理冗余问题，构造对象名称的相似性函数，并与阈值进行比较， 从而判定对象是否属于同一实体簇。

重名检测：用于处理重名问题，利用基于聚类的技术，通过考察实体属性间的关 联程度判定相同名称的对象是否属于同一实体簇。

**6. 4.2** **基于规则的实体识别方法**

本节将重点介绍基于规则的实体识别方法。

实体识别的目标就是要识别数据集中指代同一实体的元组。传统的实体识别方法 通过比较元组对的相似性来识别实体，它们假设指代相同实体的元组对的相似度比不 指代相同实体的元组对的相似度更高。相似度的定义我们在表6-3中已有举例，在此不 做赘述。

传统的想法非常直观，描述同一个现实世界实体的数据存在较高的相似性也是合 理的。不过，这种假设在实际应用中并不总是成立的，如果假设不成立，那么直接计 算相似度可能就不能被有效地用于实体识别。我们用一个例子来说明这种情况。

例6-5 表6-4中有7个元组，它们是7个名叫“Wei Wang”的论文作者，其属性 包括“姓名(name)” “合作者 (coauthors)” “标题 (title)” “类别 (class)” 。 通 过访问作者的个人主页，我们手动地将这7个元组分成三类。ID 为 on,o₁₂ 和 o₁g的元 组指代一个 UNC 大学的作者实体，记为e₁,ID 为 o₂₁ 和 o₂₂ 的元组指代UNSW 大学的一 个作者实体，记为e₂,ID 为 o₃₁ 和o₂ 的元组指代复旦大学的一个作者实体，记为e₃。

**表6-4** **论文作者元组**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ID | name | coauthors | title | class |
| u | Wei Wang | zhang | inferring… | e₁ |
| Ur2 | Wei Wang | duncan,kum,pei | social… | ei |
| " | Wei Wang | cheng,li,kum | measuring… | e₁ |
| ?H | Wei Wang | lin,pei | threshold… | e₂ |
| “z: | Wei Wang | lin,hua,pei | ranking… | e₂ |
| 0g | Wei Wang | shi,zhang | picturebook… | e₃ |
| e | Wei Wang | pei,shi,xu | utility… | e₃ |

在该例子中，实体识别的任务就是要利用表中的信息来识别实体e₁,e₂ 和 e₃ 。 由 于这些元组，它们的名字都相同，因此不能用名字来区分不同实体；它们的论文标题

大数据治理：理论与方法…………………

彼此都不相似，因此也不能用论文标题来识别指代相同实体的元组。唯有论文的作者 信息可以用于识别实体。因此，对于任意两个元组X 和 Y, 我们用X 和 Y 在属性 coau- thors上的相似度作为X 和 Y 的相似度，记为Sim(X,Y)。 由于Jaccard 相似性①测度常 常被用来测量集合的相似度，因此我们定义两个元组X 和 Y的相似度为：

Sim(X,Y)=ll coauthors(X)∩coauthors(Y)|/ll coauthors(X)U coauthors

(Y)|

故我们有以下的结果：

由于Sim(oμ,O₁₂)=0 且 Sim(on,o₃₁)=12, 所以Sim(on,0₁₂)<Sim(on,O₃₁);

由于Sim(o₁z,Og)=15 且 Sim(o₁₂,O₂)=14, 所以Sim(o₁₂,Og)>Sim(o₁₂,o₂₁)。

我们可以看出，尽管o₁ 和o₁₂ 指代的是同一实体而o₁₁ 和o₂₁ 指代不同实体，但o₁ 和 o₁₂之间的相似度比o₁ 和o₃ 之间的相似度要小。显然，在这个例子中，如果根据这 些元组之间的相似性比较，我们不能得到正确的实体识别结果。和Jaccard 相似性测度 类似，其他相似性函数，如余弦相似度和TF-IDF 测度也都有同样的问题。

基于上面的例子，我们可以得到一些观察结果。

观察1:某些属性值对的存在对识别元组很有用。

以论文作者元组作为例子。属性值对 (coauthors,“lin”) 只出现在指代实体e₂ 的 元组中。因此， (coauthors,“lin”) 的存在能够被用于识别指代e₂ 的元组。相似地， (coauthors,“kum”) 和 (coauthors,“shi”) 的存在分别能够被用于识别指代e₁ 和e₃ 的 元组。

观察2:某些属性值对的不存在也能帮助识别元组。

我们仍以表6-4为例，元组o₁ 的属性 coauthors只包括 “zhang” 。由 于 “zhang” 既出现在o 中也出现在o₃₁ 中 ，(coauthors,“zhang”) 的存在可以被用于区分指代e₁ 或e₃ 的元组和指代其他实体的元组，但是 (coauthors,“zhang”) 的存在并不能用于区 分指代e₁ 的元组和指代e₃ 的元组。由于所有指代e₃ 的元组都包含有 (coauthors, “shi”), 因此 (coauthors,“shi”) 的不存在就可以用于排除o₁₁ 指代e; 的可能性。因 此 (coauthors,“zhang”) 的存在和 (coauthors,“shi”) 的不存在可以一起用于识别指 代e₁ 的元组。

基于以上的观察，我们用下面的规则来识别表中的元组。

R1:Vo, 如果o[name] 是 “Wei Wang” 且 o,[coauthors] 包含 “kum”, 那么 o;指代实体e;

R2:Vo, 如果o,[name] 是 “Wei Wang” 且 o,[coauthors] 包含 “lin”, 那么 o; 指代实体e₂;

R3:Vo, 如果o,[name] 是 “Wei Wang” 且 o,[coauthors] 包含 “shi”, 那么 o,指代实体e₃:

R4:Vo, 如 果o,[name] 是 “Wei Wang” 且 o;[coauthors] 包含 “zhang” 且 不包含 “shi”, 那么o,指代实体e₁。

实体识别规则我们能够看出，在刚才的例子中，规则由两个子句构成：

① 和表6-2中的相似度定义类似，Jaccard 相似性也是一种常用的相似度度量方法。

102

**第6章** |数据质量管理

“如果”子句：包含对元组属性上的约束，例如“如果 [coauthors] 包含kum”; “那么”子句：说明“如果”子句的元组所应指代的实体。

因此，可以用A→B 来表示实体识别规则 “Vo, 如果。满足约束A, 那 么o 指代 实体 B”

为了方便，在下面的定义中，我们令。表示一个元组， U 表示一个数据集， r 表示 一个 ER 规则， R 表示一个 ER 规则集， LHS(r) 和 RHS(r) 分别表示规则r 的左部 和右部。

基于规则的实体识别算法有了有效的规则，基于规则的实体识别就变得非常直接， 也就是通过匹配规则的左部即可以实现有效实体识别。算法伪代码如图6-6所示。

Input: 数据集U, 规则集R, 阈值θ。

Output:U 的划分U ·

1 for E中的每个实体edo

u-②;

3 forU中的每个o,do

4 R(o₁)—FindRules(o₁);

5 for E中的每个实体e,de

6 R(∈)-|rlRHS(r)=∈;);

7 C(o₁. ∈) CompConf(R(o₁)∩R(c));

8 SelEntity(o₁,θc);

9 retum U-{U₁,U₂,…,Uml;

**图6-6** **基于规则的实体识别算法**

其中第4行表示找到所有o, 满足前件的规则，可以通过B+树实现，第7行表示计

算 o, 指代实体e, 的置信度，第8行表示选择置信度最大的实体。

实体识别规则可以由人手工撰写，也可以由程序从数据中自动学习得到。现在已 经有一些工作研究如何基于聚类、频繁项集挖掘等方法来自动化地获得实体识别规则。

**6.4.3** **什么是真值发现**

在经过实体识别之后，描述同一个现实世界实体的不同元组被聚到了一起，这些 对象的相同属性可能包含冲突值。在很多情况下，冲突值来源于数据集成中不同的数 据源。在描述同一实体同一属性冲突值中发现真实值的操作称为真值发现。不同数据 源对同一实体可能提供不同的冲突数据，这就需要在冲突数据中找出真值。真值发现 的目标是，在多个对同一实体的描述信息中，找出最为准确的描述。真值发现是数据 集成中的一个重要部分。

形式化地来说，真值发现问题定义如下。

我们考虑一组数据源V 和一组对象O 。 一个对象代表了一个现实世界实体的某个 特定的方面，如电影的导演；在关系型数据库中， 一个对象对应于一个表中的一个单 元格。对于每个对象O∈O, 数据源 V∈V, 可以(但不一定)提供真值。提供的这 些值被称为“事实”。在为一个对象提供的不同事实中， 一个事实正确地描述了真实世

大数据治理：理论与方法…………………………

界，所以它是真实的，其余的是假的。给定一组数据源V, 我们要为每个满足O∈O 的对象 O 判断其真相。需要注意的是，数据源提供的可以是原子事实的真相，也可以 是原子值集或列表(例如， 一个论文作者名单)。在后一种情况下，如果原子值是正确 的，设置或列表是完整的，我们认为值为真的。每个数据源可以为不同的对象提供不 同数量的事实，但只能为一个对象提供至多一个事实。

基于真值发现问题的输入数据源，我们正式给出以下定义。

定义6-1(对事实的置信度) 一个事实F [ 用C(F) 表示]的置信度是F 是正确的

概率。

定义6-2(数据源的可信性) 数据源V[ 记 为T(V)] 的可信性是V所提供事实的

预期置信度。

相同对象的不同事实可能是相互冲突的，当然也可能相互支持。例如， 一个数据 源声称，成人的平均身高是“175cm”, 而另一个数据源声称为“176cm” 。其中一个是 真的，而另一个也可能是真的。因此，我们有以下事实之间的内涵的定义。

定义6-3(事实之间的内涵) 从事实f₁ ((₂) 到f₂ (f),imp(fi→f₂) 是f;在f₂ 置信 度上的影响力。

定义6-4(数据源之间的关系) 如果两个数据源 v₁ 和 v₂ 直接或间接地从同一来源 得到了相同的部分。我们说两个数据源v₁ 和 v₂ 之间存在着依赖性。相应地，有两种类 型的数据源：独立源和它们的副本。

一个独立源提供独立的值。显然，它可能提供错误的值。良好的独立源更有可能 为每个对象提供真正的值，而不是任何特定的虚假的值。相比于坏的独立源，副本复 制的一部分(或全部)来自其他来源的数据(独立源或副本)。

当我们在真值发现问题中只考虑数据源之间的直接副本时，我们就可以说，如果 v₁直接由v₂ 复制而来，那么v; 就依赖于v₂。

**6.4.4** **真值发现方法**

真值发现中最为直观的方法是投票法，即对同一个对象的不同取值进行投票，将 得票多的描述作为真实值。但显然，投票时不同数据源对应的描述的权值应有所不同， 这取决于数据源的可信度，也称为数据源的精度，如何求解数据源的精度便是问题的 关键。此外，由于现实生活中不同数据源之间存在复制现象，错误数据可能在多个数 据源之间传播，这称为数据源之间的依赖，也是真值发现中需要着重考虑的因素。下 面分别对这两个方面进行简单的介绍。

**1.迭代地计算数据源的精度**

数据源的精度将决定投票时相应描述对应的权值。在这里，计算数据源精度的依 据是， 一个数据源为其他对象提供的真相越多，越有可能为当前的对象提供真相，其 精度越高。首先，我们假定所有数据源的精度都相同，然后模拟投票过程得到每个对 象的真值，进而更新数据源的精度。接下来，重复进行上一步，迭代地计算数据源的 精度。

**2.** **数据源之间的依赖关系**

如果两个数据源之间存在数据复制，则称它们之间存在依赖关系。数据源之间的

**第6章** | **数据质量管理**

依赖会导致实体的某些描述得票偏高，于是我们可以通过减小它们的权值来减少依赖 带来的影响。

计算两个数据源之间依赖程度的一种思路是，统计两个数据源对同一实体所提供 的描述相同的比例。这里需要注意的是，对于某个实体，当两个数据源提供的都是真 值时，它们在这个实体上不应被视为复制关系。

**6.4.5** **真值发现的关键技术**

**1.** **数据冲突**

由于信息错误、记录丢失、输入错误、数据过时等情况可能存在于不同的数据源 中，来自多个数据源的信息常常相互矛盾冲突。在这种情况下，面向多源数据挖掘正 确信息是十分必要的，否则将导致源于数据的知识和决策的严重错误。首先给出三种 典型的多源数据冲突场景。

(1)同一实体信息冲突。在互联网上可以通过多种不同途径获取多个数据源对同 一实体的描述信息。然而，得到的信息并不能保证是一致的。

例如，在搜索引擎中查询“珠穆朗玛峰的高度”,返回包括29035英尺、29002英 尺和29029英尺三种来自不同网站的结果。在这些嘈杂的信息中，哪一个结果更值得 信赖，代表了真实的信息呢?

(2)同类实体信息冲突。在现实世界中，可以收集到多个数据源对同类实体的描 述信息。然而，不同数据源对相同类别的实体描述也可能是不一致的。例如，针对来 自不同医院的中风患者，他们分别接受了“溶栓”“抗凝”“抗血小板”和“使用退烧 药”四种治疗手段。尽管同一种病症通常有多种治疗方案，但上述四种治疗手段是否 都是正确的呢?

(3)同类实体多语义表达。当多源数据以文本形式存在时，描述相同或相似实体 的信息可能以不同词汇和句子结构表达出来。例如，来自三个数据源的有关实体 China 和属性 president 的句子分别为：

·The U.S.President Donald Trump visited China weeks ago.

·Europe is chosen as first overseas trip of China's president Jinping Xi.

·There was a card printed ’China,Jintao Hu'on the table indicated it was the position of Chinese president.

上述的三个句子中哪句表达了China 现阶段的 president 的真实信息呢?

以上三个场景简要说明了数值型数据、文本型数据等不同类型的多源数据组成都 可能在同一实体、同类实体等不同层面上产生冲突。而多源数据的真实信息挖掘，也 就是多源数据的真值发现，正是面向多类型数据、多层面冲突找到粒度更细、内容更 准、价值更高的实体信息的重要技术。当前，面向多源数据的真值发现技术主要面临 两方面的挑战。

(1)结构层面。对于多源数据而言，其结构取决于每个数据源的数据组成。由于 数据源在各自应用中的特点，其表示类型趋向多元化，如表示为关系型数据和文本型 数据。因此，基于多源数据的真值发现需考虑数据组成的不同特点和应用场景，精细 定义不同场合下的真值发现问题。

大数据治理： 理论与方法…

(2)算法层面。由于多源数据中存在实体之间、数据源之间等多个层面的信息冲 突，相比于只考虑单一层面的信息冲突的情况，真值发现问题的固有复杂性大大增加。 因此，基于多源数据的真值发现需仔细考虑不同层面的信息冲突，设计高效的算法， 利用多种线索挖掘更有价值的信息。

**2.数据源与真值**

数据源是获取数据的一个基本单位。 一般来说， 一个数据源包含了其对现实生活 中的一系列客观事物(实体)观察的结果。多源数据，顾名思义，是囊括了多个数据 源，描述大量相同或不同实体信息的冗余数据。根据数据结构的不同，常见的多源数 据可分为多源同构数据、多源异构数据和文本数据。多源同构数据，即来自实体和属 性，能组成相同的多个数据源的集成数据，是现有大部分工作采用的数据类型。

数据的多源性、嘈杂性决定了多源同构数据真值发现在多网页信息筛选、众包平 台多用户反馈质量评估等日常应用中的必要性。多源同构数据并不要求每个数据源都 必须对每个实体提供信息，可以通过多源异构数据对源间相同实体和属性信息进行提 取来获得。在多源同构数据中，对于一个实体集合，每个实体的每个属性都有一定数 量的数据源提供相应的观测信息，现有工作通过不同的聚合方法来挖掘这些实体的真 实属性信息。

该数据具有以下特点。

(1)源内实体唯一性。即每个数据源针对单个实体提供的信息是唯一的。在现实 应用中，单个数据源内可能存在对某个实体的多次描述。例如，维基百科的一个贡献 者可能对某个词条进行多次编辑；众包平台上的一个工人也可能对某个问题提交多个 答案。在这种情况下，通常将时间戳更近的描述作为该数据源的唯一描述。当时间信 息不存在时，用户需要定义一些规则从多个描述中选择一个作为唯一描述。

(2 ) 源间实体同一性。即不同数据源对于同一实体描述的表示统一。这是考虑到 不同的数据源基于各自需求独立开发和维护，可能存在现实生活中的同一实体名字及 其属性值在不同数据源中表示不相同的情况。例如，同一个人的名字可能在不同数据 源以不同格式 (JoeSmith与 Smith,Joe) 表示；同一座山的高度可能在不同数据源以不 同单位(29029英尺与8848米)表示。在这种情况下，多源数据需要实体识别或专家 指导等技术将数据格式统一。否则，原本相同的实体会被判断为不同实体，原本相同 的观测值会被判断为冲突，增加真值发现的复杂程度。

(3) 源间观测冲突性。即不同数据源对于同一实体描述内容不相同。由于真值发 现的必要性立足于多个数据源对同一实体给出了冲突观测值，若每个数据源对实体的 观测值都一致，在没有额外知识指导的情况下，实体的真值将会是该观测值，这就失 去了真值发现的意义。而且，来自不同数据源的大量实体的一致观测值会影响真值发 现算法对数据源可靠程度的评估，从而影响真值发现算法的最终效果，故源间观测一 致的数据需要在真值发现前被妥善检查处理。多源同构数据的以上性质确保了真值发 现的充分必要性。其中，必要性体现在源间同一实体观测冲突需要解决；充分性体现 在真值发现步骤之前不需要源内去重(符合源内实体唯一性)和实体识别(符合源间 实体同一性)。明确了真值发现任务中多源同构数据应具有的性质，我们接下来分析现 有的真值发现研究方法。 一个直观的想法是进行多数投票或平均，这种投票/平均方法

106

**第6章** |数据质量管理

假定所有来源都同样可靠。

然而，这种假设在大多数情况下可能不成立。比如前面给出的例1.1 提到的“珠 穆朗玛峰”:采用多数票表决，结果“29035英尺”的出现次数最多，将被视为正确的 值。然而，在搜索结果中，维基百科提供的“29029英尺”的信息却是事实。这个例 子揭示了不同数据来源的信息质量差异很大，需要设计更加先进的真值发现算法，通 过评估数据源的可靠性来挖掘真正正确的信息。挑战在于，在实践中数据源可靠性通 常是未知的，必须从数据中推断出来。鉴于这一挑战，面向多源同构数据的真值发现 本质上是一种先进的数据聚合技术，通过估计数据源可靠程度和推断真实信息来从多 个数据源提供的冲突数据中发现真值。

**3.** **基于函数依赖的多源同构数据真值发现**

针对一系列实体的正确属性值挖掘，现有的真值发现工作大多利用多个数据源对 同一实体提供的信息冲突来评估不同数据源的可靠程度，通过数据源可靠程度与真值 之间的关系来确定实体真值。也就是说，可靠的数据源提供的实体属性值更可能是正 确的，不可靠的数据源提供的实体属性值倾向是错误的。然而，由于数据分布存在长 尾特性，并不是每个实体都有可靠的数据源提供信息。同时，可靠的数据源也可能提 供错误的信息。在这种情况下，仅考虑数据源的相对可靠程度决定的真值依然可能是 错误的。

考虑有K 个数据源S={S₁,S₂,…,S} 提供的n 个实体e={e₁,e₂,…,e,} 信息，每个 实体存在m 个属性|A₁,A₂,…A 。} 。w₄ 表示S₂ 的可靠分数， w, 的值越高，则代表S, 越 可靠。K 个数据源的可靠分数集表示为 W=|w₁,w₂,…,w₁} 。R 是定义在属性集合Attr (R)={A₁,A₂,…,A 。} 上的一个关系，数据库 D 是关系R 的一个实例，其中存在L 个元 组，这些元组代表来自K 个数据源的n 个实体信息，属性集Attr(R) 包含实体的所有 属性Attr(R)={A₁,A₂,…,A 。} 。 其中，每个实体的属性信息不一定被所有数据源覆盖。 对于两个属性子集X,Y 属于Attr(R), 在 R 上定义的一个函数依赖为F:X→Y, 其中 X代表决定属性集，Y代表依赖属性集，包含F 中所有属性的集合定义为F=XUY。

D 满足F:X→Y, 当且仅当对于每对元组t,t∈D, 若 t₁[X]=t₁[X], 则 t;[Y] =

t;[Y] 。 在不失一般性的情况下，假设F:X→Y 已经分解，Y只包含一个属性。由多个 函数依赖组成的集合称为约束集三。为简化讨论，本章使用两种符号形式来表示每个 元组，用t₁ 表示D 中的第1个元组，用t 表示来自源S, 的关于实体e, 的元组。元组t₁ 和亡的属性AEAttr(R) 的值分别表示为t₁[A] 和t[A] 。 有了这些基本定义，下面分 析如何定义代价函数可以最大限度的得到准确的真值信息。

首先，如果没有额外的知识，数据修复的代价函数一般为修复结果 D'和原始数据 D之间的距离。然而，在多源数据模型中，D 是由来自多个数据源的元组组成的。如 上文所述，不同的数据源通常具有不同的可靠程度，如果可以检测出可靠的数据源， 即哪些数据源更有可能提供准确的信息，那么这些可靠的数据源就可以指导模型在检 测到不一致的情况下修复错误。通过这种方式，修改t[A] 到 t′[A] 的代价不仅基于 t[A] 和t′[A] 之间的距离，还要衡量修改t[A] 的权重。其中， t′[A] 是 t[A]

在 D'中的对应修改。综上所述，代价函数定义如下：



大数据治理：理论与方法

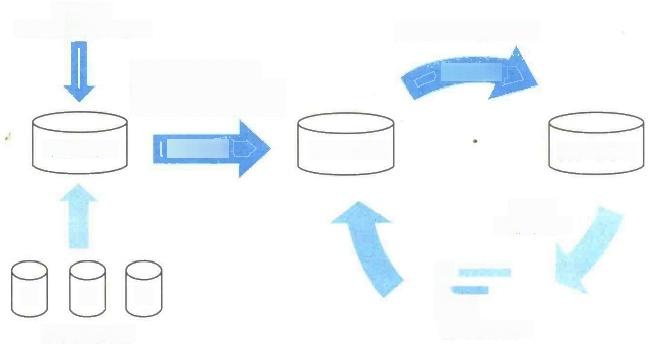
…………………………………





为了简化代价函数，距离的度量定义为0-1损失函数。如果t[A] 由可靠的数据源 S, 提供，意味着t[A]更可能是准确的，则修改t[A] 的代价变得相对较大，因为w 较 高。直观来讲，为降低代价，需通过修改具有较低权重且改动较少的元组来消解根据 函数依赖产生的冲突或不一致。

解决这一问题可以使用AutoRepair 算法。为解决不同实体信息间的不一致和同一 实体数据源间的冲突，需要分别进行数据修复和数据聚合步骤。由于这两个步骤用于 处理数据的不同层面，如何安排顺序至关重要。AutoRepair 选择在数据聚合之前先进行 数据修复，这主要有两个原因。首先，由于数据修复利用函数依赖来检测元组之间的 不一致，拥有的信息越多，错误被发现和修复的可能性越大。相比之下，首先进行数 据聚合可能导致丢失有用的信息(如正确的值),这将降低数据修复的准确性。其次， 由于需要知识来正确地修复错误，使用源可靠性作为证据将大大提高数据修复的准确 性。具体来说，当两个元组之间检测到不一致时，更可靠的数据源提供的元组有更大 的概率是正确的，所以应该修改来自不可靠数据源的元组。相反，首先进行数据聚合 将失去数据源可信度对数据修复的指导。为了进一步提高准确性， AutoRepair 迭代地进 行数据修复和数据聚合。原因在于，数据源可靠性对于得到准确的修复结果很重要。 然而，数据源的可信度通常又是未知的，需要从数据中估计出来。在迭代过程中， Au- toRepair 可以使用最新的修复结果来更准确的估计源可靠性。因此，基于更新后的源可 靠性再次进行数据修复时，可以获得更准确的结果。其框架如图6-7所示。



约束集2

初始化源可靠分数W

算法1 GPS(D,Z,W)

算法2 IndCR(D,Emd,W)

算法3 IntCR(D,Zom,W)

数据修复

更新源

可靠分数W

S;:

S₂:

S;:■

源可靠分数W

算法4 DataAgg(D',W)

数据聚合

*S₁* S₂ S₃

多源同构数据

真值结果D⁴

修复结果D

原始数据D

图6-7 修复框架

伪代码描述如图6-8所示。

4. 基于否定约束的同源多构数据真值发现算法

若数据中存在多种类型的实体关系，基于源可靠性估计的真值发现和基于约束的 数据修复方法都不能很好地获得实体的真实信息。CTD算法可以通过同时利用多种类

108

第6章|数据质量管理

型的实体关系和源可靠性程度，来进行真值发现。 一方面，将多种类型的实体关系作 为约束条件可有助于识别可靠数据源产生的错误。另一方面，来自多个数据源的冲突 信息也为修复错误、发现真值提供了更多的证据。接下来需要解决以下问题：

|  |
| --- |
| Input:原始数据D,约束集 ∑  Output:真值结果D |
| 1初始化源可靠分数W  2将约束集 ∑分为独立约束集Zmd和关联约束集Zon  3 while收敛条件未满足do  4 D\*=IndCR(D,Zimd,W);  5 D'=IntCR (D\*,Zcor,W);  6 D⁴=DataAgg(D',W);  7 for每个w₄ ∈Wdo  8 计算S;  9 更新wg  10 return D |

图6-8 AutoRepair算法伪代码描述

(1)哪种约束可以对多种类型的实体联系进行建模?

(2)在这种约束下，如何从多源数据中准确地得到真值发现结果?

(3)当约束数量较大时，如何保证方法的可扩展性?

考虑有 K 个数据源提供的L 个实体，每个实体由P 个属性组成。第k 个源提供的 第1个实体的第p 个属性的值表示为 。第k 个源提供的信息表表示为X₄, 其中v 是 其第φ个项。第1个实体的第p 个属性的真值表示为vg 。 给定K 个信息表|X,X₂,…,

X}, 所有实体的所有属性真值都存储在真值表X\* 中，该表的第4p个项表示为v。 源可靠分数表示为W=|w₁,w₂,…,w₄}, 其 中w₄ 是第k 个源的可靠性得分。较高的w 说明第k 个源更可靠，且该源提供的值更可能是正确的。给定一组运算符B=|=,…,

≤,≥},DCs 是定义在真值表上的一阶逻辑表达式。每个 DC 的格式为

4:-(C₁A…AC₂)

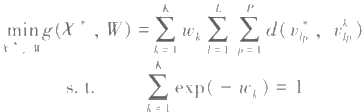
其中，每个子句C₂ 的格式是v ◎r(vm),r(v,) 表示c 或者v,c, 是一个常量，

⊙∈B,m,n 指第m,n 个 属 性 ，i,j=1,…,L。

如果X\* 中的值满足φ中定义的所有约束，则真值表X\* 满足φ,表示为

X\*F4

否定约束将一个实体或多个实体内的各种关系表示成统一的格式，可以方便地作 为约束条件纳入真值发现过程中。基于此，每个实体的真值都可以从其多源信息，以 及相关实体信息中被找到。同时考虑否定约束和数据源可靠性估计的真值发现问题定 义如下。问题定义给定一组实体的数据源信息表{X₁,X₂,…,X}, 以及定义在这些实体 属性值上的一组否定约束三，目标是得到真值表X\*和源可靠分数X, 使得X\*满足三并 且最接近真实值。此真值发现问题形式化如下：



X\*Fφ,Q∈≥

109

大数据治理：理论与方法……………………

其中， d(·) 是用于计算数据源信息表和真值表之间距离的损失函数。对于连续性数 据，损失函数定义为：



对于离散型数据，损失函数定义为：



该模型的基本思想是，真值表中的信息需要来自可靠的数据源，同时满足 DCs的 要求。为了实现这个目标，真值应该接近可靠源 (w₄ 很大)支持的值( ),并且满 足三中每个φ。因此最小化真值表与多源信息表的总加权偏差，其中每个数据源都以 其可靠性程度加权，真值由三中的DCs约束。

对于每个DC 。,首先将其一阶逻辑表达式φ:- (C₁A…AC₂) 转换为析取范式，即

4:-C₁V…V-C₂

以便分别处理每个子句。由于每个子句C, 的格式是),可以得到：



其 中 ，i=1,…,L,◎ 是运算符⊙的反向运算符，  表示第z 个子句。对于具有 不同◎的子句，不同的函数f 都包含符号函数sgn(x)。 通过给不同的子句分配不同 的函数，可以得出结论：

如果f(-C₂)>0, 则 -C₂ 为真；否则1f(-C₂)=-

利用这些函数，可将上面的逻辑表达式转换为一组算术约束：



5.基于模式发现的多源异构数据真值发现算法

在面向多源同构数据的真值发现领域，对于一个实体，它的正确值是通过解决多 个源信息之间的冲突来找到的。对于一组实体，为每个实体提供信息的数据源越多， 就越可能识别可靠的数据源并找到真值。然而，对于多源异构数据，由于其在实体层 面和属性层面上存在的多种冲突，为每个实体提供信息的数据源个数可能是稀少的， 这种信息的不足使得面向多源同构数据的真值发现方法难以识别可靠的数据源和发现 真值。由于实体个数通常远远多于属性个数，模式层面的异构性可以通过领域专家来 匹配解决，实体层面的异构性利用领域专家匹配却是不现实的。因此，提出面向存在 实体层面异构性的多源数据真值发现方法是至关重要的。

假设有K 个数据源，每个数据源包含描述一组实体的多条记录，不同数据源观察 到的实体可能是不同的。即使不同数据源提供的某些信息描述的是同一个实体，由于 实体对应关系未知，本章也将其视为是代表不同实体。假设每条记录都有 M 个属性

{A₁,…,Ay} 。 第 i 条记录表示为o;={v₁,V2,…,vu}, 其 中v 是记录o, 关 于A。的值。

代表第k 个数据源提供的记录集合，N₄ 为第k个源提供的记录总数。 一个存 在实体层面异构性的多源数据 .由K 个数据源的条记录组成。在D 上

第6章|数据质量管理

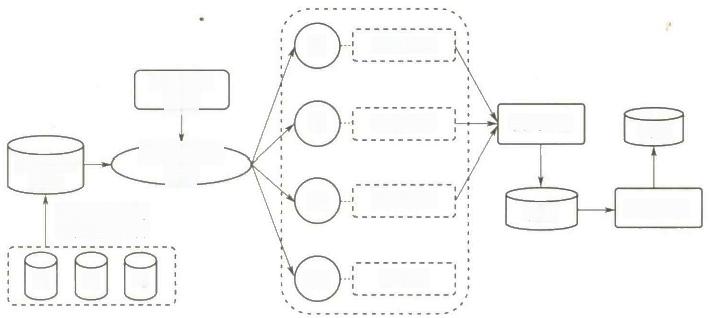
定义的模式为一个三元组 (R,X,t), 其中R, 为应用记录集，它由记录o,∈D 组

成；X 是特征属性集，由{A₁,…,Ay} 中的属性组成；t 是X上的值组合。对于每个 属性A。∈X,t 是A。在D 的域中的常量值。可以断定，为了避免一个记录出现在两个 模式的应用集中，每个模式的应用集必然是不相交的。因此，模式发现问题可以视为 是推断潜在分组的任务。

潜在分组的定义如下：

给定潜在分组个数L,G 是 n×L的矩阵，其元素ga 代表o,∈D 的组指示符。若 gu=1, 代表o;属于组1,否则gu=0 。根据组指示符形成潜在分组{C₁,C₂,…,C₂}, 其 中第1个组 C,由组指示符gμ=1 的记录o, 组成。

PatternFinder算法可以用来在多源同构数据中找到真值，其框架如图6-9所示。



潜在分组

优化分组

策略

组级代表

PatternFinder

算法

属性权重

实体层面异构

属性层面同构

源权重

模式 真值生成

多源异构 数据

S₁ S₂ S₃

模式生成

真值

W

G

C

P

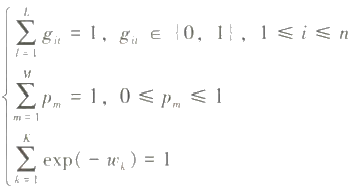
图6-9 PattermFinder 算法

其基本思想是，对于每个潜在的分组，可靠的数据源提供可信的属性值，重要的 属性组成模式的特征属性集。然后，组级代表在重要属性上的值应与可靠数据源提供 的值接近。因此，优化目标应该最小化组级代表到组内记录的总体加权偏差，其中每 个源以其可靠性加权，每个属性以其重要性加权。目标函数定义如下：





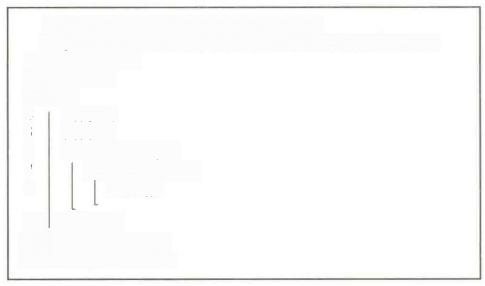
其中



大数据治理：理论与方法………………………

通过最小化目标函数f(G,C,P,W) 可以搜索四组未知变量组指示符G 、组级代表 C 、属性权重P 、 源权重W 的值。直观地说，如果一个数据源更可靠 (w₄ 较高),并 且一个属性更重要 (p 。 较高),在确定组级真值时，该源提供的关于A。的信息更可能 是正确的。 一方面，当组级真值c 偏离该源提供的值v 时，惩罚项较大。另一方面， 当 来自具有较小w₄ 的不可靠源，或者来自具有较小p。的不重要属性时，惩罚项较 小。根据该优化目标框架，需同时求解组指示符G、组级代表C、属性权重P 、 源权重 W。 然而，在优化方法中很难直接计算四组未知变量。因此，应该将目标函数在每个 变量集合上循环地最小化，同时将剩余的三个变量集合固定在它们最后更新的值上， 这种方法称为块坐标方法。为了最小化问题定义中的目标函数，本节提出了一个迭代 算法 PatternFinder 。PatternFinder 通过同时学习组指示符G、组级代表C、属性权重P 、 源权重W 四组变量来最小化目标函数。

算法伪代码描述如图6-10所示。

Input:记录集合 D, 潜在分组 L, 平衡参数μ.

Output:组指示符 G.组级代表 C,属性权重 P, 数据源权重 W.

1 初始化 G和 C;

2 初始化 W;

3 repeat

4 更新 P;

5 更新 G;

6 for 每个分组C;do

7 for 每个属性4,do

8 更新cm;

9 | 更新 W;

10 until 满足停止条件；

11 return G,C,P和 W.

图6-10 PatternFinder 算法伪代码

6.5 **错误检测与修复**

数据中还有可能存在其他错误，本节简述一些错误检测与修复策略。

**6.5.1** **格式内容清洗**

如果数据的来源是系统日志，那么通常在格式和内容方面，会与元数据的描述一 致。而如果数据是由人工收集或用户填写而来的，则有很大可能在格式和内容上存在 一些问题，简单来说，格式内容问题有以下几类。

**1.** **显示格式不一致**

这种问题通常与输入端有关，在整合多来源数据时也有可能遇到，如时间、日期、 数值、全半角等表示不一致等。这种情况下的数据错误检测通常需要根据数据类型预 定义数据显示格式，根据显示格式将不符合标准的数据处理成一致格式的数据。这种 方法表面看起来很直接，其难点在于识别不一致的属性值中对应的部分。有时候数据 格式比较复杂，需要用正则表达式描述，这种情况下，检测与修复则需要更复杂的算

**第6章** |数据质量管理

法，感兴趣的读者可以查阅其他文献，我们不在这里做深入探讨。

**2.** **有非法字符**

某些属性值只允许包括一部分字符，比如，身份证号只允许包括数字+字母X, 中 国人的姓名只允许包括汉字(通常登记时不允许名字中有其他字符)。有非法字符最典 型的情况就是头、尾、中间包括空格，或姓名中存在数字符号、身份证号中出现汉字 等。这种情况下，可以在元数据中规定属性值包含的字符集合，并去除非法字符。

**3.** **内容与该字段应有内容不符**

在一些情况下，用户误将本来属于一个属性的数据填写到了另一个属性中，例如， 将姓名写成了性别，身份证号写成了手机号等，出现这类问题的数据并不能用简单的 删除来处理，因为其仍包含了有用的信息，可以通过识别数据的格式，并基于格式辨 识其应当隶属的属性来将属性值放到正确的位置上。

格式内容问题看起来仅仅是细节问题，但造成了很多分析的失败，比如，非法字 符导致跨表信息关联的识别失败(认为“王宏志”和“王宏志”不是一个人)、统计 值不全(掺杂了字母的数字无法参加求和)、分析输出失败或效果不好(把日期和年龄 弄混导致数据分析结果语义不正确)。因此，这部分清洗工作需要引起重视，特别是在 数据来源不可靠的时候或者数据库对于数据的语法约束缺乏必要约束的情况下。

**6.5.2** **逻辑错误清洗**

逻辑错误清洗的工作是去掉一些通过逻辑推理就可以发现问题的数据，防止分析 结果的偏差。这部分主要包含如下几个步骤。

**1.** **去重**

顾名思义，去重就是去掉数据中的重复信息，由于数据存在的同名(不同的事物 具有相同的名字)和异名(相同的事物具有不同的名字)情况，去重通常要通过上文 介绍的实体识别技术来实现，这类数据中出现的冲突值可以通过上文介绍的真值发现 技术来进行消解。而且为了解决由于格式问题直接导致的异名问题，建议将去重的步 骤放到格式清洗之后进行。

**2.** **去除不合理值**

有时候，用户会填入一些不合理值，如年龄为2000岁，月收入为10亿元。我们需 要有效检测和修复这种不合理值。这类不合理值的检测主要依靠属性值上的约束。例 如，人的年龄取值在[1,150]之间，月收入在[0元，100万元]之间。由于这类不 合理值提供的有用信息非常少，因此其修复需要按照缺失值处理。

**3.** **修正矛盾内容**

有些字段是可以互相验证的，例如，某个用户电话的区号是“010”,但是城市是 “上海”,我们就可以知道区号和城市两个属性中有一个是错误的；又如，两条记录中 邮编相同但是城市不相同，也可以从中发现两条记录中某一个属性是错误的。这种错 误的检测可以通过规则来实现，这就经常用到函数依赖和条件函数依赖。

条件函数依赖是函数依赖在语义上的扩充，其经常用于数据清洗工作，在数据库 一致性修复上应用十分广泛。 一个关联规则可以等价于一个条件函数依赖，发现关联

113

大数据治理：理论与方法

规则就相当于发现条件函数依赖。

我们在6.2节已经给出了函数依赖和条件函数依赖的概述，此处给出二者的形式 化定义

定义6-5(函数依赖) R(U) 是一个属性集 U 上的关系模式，X 和 Y是 U 的子集。

对于R(U) 的任意两个可能的关系r₁ 、r₂, 若 r₁[x]=r₂[x], 则 r₁[y]=r₂[y], 或者

若r₁[y]≠r₂[y], 则r₁[x]≠r₂[x], 称X 决定Y, 或者Y依赖X。

例如，在设计学籍信息表时，我们知道了某位同学的学号，那么一定能知道其姓 名，这就是姓名依赖于学号的例子。

定义6-6(条件函数依赖) 设存在一个关系模式R,attr(R) 表示定义在其上的属

性集，对每个属性A, 有ACattr(R), 定义在R 上的一个条件函数依赖φ可以表示成

φ:(R:X→Y,T₁) 。 其中：

(1)X 和 Y是定义在attr(R) 上的属性集；

(2){R:X→Y| 是一个标准函数依赖；

(3)T, 是与X 和 Y相关的模式元组，定义了相关属性在取值上的约束条件。

比如，李建中老师为研究生开的课只用英语进行教学(其他老师开的课或者李建 中老师为本科生开的课不一定用英语进行教学),可得φ:[教师='李建中',授课对象= '研究生'] → [授课语言='英语!]。

**6.5.3** **非需求数据清洗**

针对分析过程中不需要且有错误的数据，可以考虑直接删除。这种处理方法表面 上看起来简单，但是实际操作起来，需要考虑许多问题。例如，在一些情况下可能删 除了看上去不需要但实际上对业务很重要的字段；有时候用户觉得某个字段有用，但 又没想好怎么用，无法确定是否该删除；在处理时删错了字段等。考虑到这些情况， 可以在出现错误的数据上增加错误标记，这样在数据分析的过程中会忽略这些错误值 或者针对存在的错误值设计劣质容忍的数据分析算法，最小化错误值对分析结果的 影响。

**6.5.4’面向图数据的错误检测技术**

**1.** **知识图谱错误检测方法**

知识图谱的构建有两种手段：手动构建的知识图谱和自动构建的知识图谱，目前 的机器智能大规模的依赖于这两种知识图谱，手动构建知识图谱的例子包括 WordNet 和Cyc, 自动构建知识图谱的例子包括Know-ItAll 和Probase。人工构建的知识图谱精确 度高，但是规模有限；相比而下，自动构建的知识图谱覆盖率高，但是准确度相对较 低。因此，检测并消除自动构建的知识图谱中存在的错误对于提高机器智能至关重要。

当前知识图谱的研究只是基于平面上图的关系，节点为实体或概念，边为相应的 属性。大部分的研究仅仅关注于知识图谱的层次结构和相关关系的出现频率。但是几 乎没有一种有效的算法可以实际的考虑到知识之间相互关联的关系，几乎没有一种方 法可以通过知识库本身进行自我学习，从而通过自身进行错误检测。针对以上的不足 之处和缺陷，本书准备从知识图谱内在关系入手，找寻新的有效方法对于自动构建的

**第6章** |数据质量管理

知识图谱进行错误检测，并对于检测出的错误提出相关处理方法。

目前国内外已经提出了一些基于知识图谱的错误检测方法， 一些方法在构建知识 图谱时就进行了相关错误检测，如判断所加入的关系是否在知识图谱当中等。降低所 构建的知识图谱的错误率，这类方法并不属于本书关心的内容。本书所应用的范围是 针对于已经构建成功的知识图谱进行进一步错误检测。这些错误检测方法目前可以分 为简单方法和高级方法。

1)简单方法

目前较为简单的知识图谱错误检测方法有两种：使用频率，以及引入新的知识对 现有知识进行修正，下面简要介绍这两种方法。

使用频率修正错误：目前较多的针对知识图谱的错误检测算法都着重于利用知识 图谱本身所具备的频率。以微软的 Probase 知识图谱为例， Probase 知识图谱包含了 1600万个IsA 关系，主要使用 Hearst 语法模式使得机器自动从17亿个网页中进行关系 的提取，主要提取语法是 such as语法结构。Probase 当中保存的是知识之间的 IsA 关 系，对于每个关系都附有一个频率，该频率为从网络语料库中观察到的该关系出现的 次数。从而可以通过该频率判断，如果关系所属频率较低，则该条关系在语料库中出 现的次数较少，从而可以判断该关系的正确率较低。这种方法也同样适用于其他包含 关系频率的知识图谱错误检测中。

引入新知识关系：引入新的知识对现有知识进行修正也是很多目前知识图谱使用 的方法。引入的新知识包括：另一个知识图谱中所包含的知识关系、新的网络语料库 关系等。这种方法利用外部知识库来消除冲突并提高分类的质量。将外部知识作为对 内部本身具备知识的一种二次验证，从而进行错误检测。

2)高级方法

目前较为高级的知识图谱检测方法也有两种：使用统计学回归方法进行回归计算， 以及查找知识图谱中存在的圈关系。下面简要介绍这两种方法。

统计学回归方法：统计学回归方法是一种数学方法，该方法的思想使用统计技术 用于数据整合。该方法主要是针对于处理知识图谱当中的线性关系，通过处理数字属 性之间的定量关系，从而得到对于线性关系的归回方法。

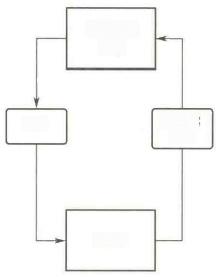
查找圈关系：查找知识图谱中存在的圈关系是该领域最新提出的一种方法，该方 法的着重点将知识图谱中间存在的关系视为一种有向图关系，并使用相关算法在该大 图上查找圈关系，若存在圈关系，则判定该圈中包含的关系至少有一个是错误的。该 方法的出发点是，理想的知识图谱应该是一张有向无环图，如果图谱当中存在环，则 证明至少有两个关系是互为 IsA关系，这在人类的知识认知当中是不正确的，所以可以 通过删除其中的错误边来破坏圈结构，从而尽量使知识图谱呈现有向无环结构。

3)存在的问题

上述提到的四种目前常用的知识图谱错误检测方法均存在一些问题。使用频率的 方法无法处理频率较为集中的知识图谱关系。用Probase 知识图谱为例，Probase 当中 的 IsA 关系的频率信息本身遵循具有长尾的幂律分布，这意味着具有或不具有错误的大 多数关系都具有较低的频率。例如，在 Probase 中，大约七百万个边缘的频率为1。但 是通过查阅资料得知，里面近乎78%的关系都是正确的。如果使用将高频率关系进行

大数据治理： 理论与方法

保留，而删除低频率关系，会使得错误检测的准确率大大降低，删除掉许多有用的关 系，从而妨碍知识图谱的正常使用。第二种引入新知识关系的错误检测方法忽略了不 同知识图谱的独特性， 一些知识库如上述提到的Probase 有许多其他知识库中不存在的 特定概念。因此，外部知识库中缺少一个具体概念的实例的成员关系。例如， Probase 有270万个概念，而Yago只有48万个类型。由于 Probase 的概念覆盖与外部知识库之 间存在巨大的差距，因此无法利用它们来发现 Probase 中的冲突。因此，引入新的知识 库并不能高效地进行错误检测，在很多情况下没有普遍性。第三种统计学回归方法的 问题显而易见，因为回归只对发现线性关系有效。然而这种方法对于非线性关系却无 法得出有效的解决方法。因此，这种方法对于大多数的知识图谱都是不太有效的。最 后一种找圈的关系面临的最大问题就是较低的错误召回率。因为知识图谱的关系是从 大型网络语料库中获得，多数的错误关系并没有存在于圈结构当中，所以这样无法较 多的找到知识图谱当中存在的错误。存在圈结构的错误如图6-11 (a) 所示，不存在圈 结构的错误如图6-11 (b) 所示。

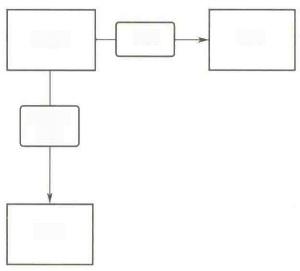
exciting

Eagie

bird

IsA

city

wrong

wrong IsA

IsA

IsA

Paris Fish

(a) 存在圈结构的错误 (b)不存在圈结构的错误

图6-11 两种错误分类

**2.** **基于集合距离的纠错算法**

尽管目前已经提出了一些知识图谱的错误检测方法，主要的知识图谱检测方法可 分为两种。

第一种是基于频率的方法，关系之间的低频率对应着其在语料库中很少被注意到， 因此利用频率可以判断关系的正确性。频率越高正确性越大，频率越低正确性越小。 例6-6表明了这种方法存在的问题。

例6-6 在 Probase 当中，三元组(蛇，IsA, 草 本 ) ; ( 果 蝇 ，IsA, 生物)的频率 都为1。但是非常明显是，第一个三元组是错误的，第二个是正确的。

在表中，本文使用 Probase 知识库作为分析对象，如表6-5所示，遍历整个 Pro- base所有关系对，统计了频率为1和频率大于1的关系占的比例。从图中我们可以 看出，大量的关系都是频率为1。之后随机选取了100个关系来检查正确性，答案由 人为判断给出。从这个实验中看出只用频率来纠错是一种不明智的方式，因为准确 率相对较低。

116

第6章 **|数据质量管理**

**表6-5** **频率关系**

|  |  |  |  |
| --- | --- | --- | --- |
| 频率(个) | 百分比(%) | 频率(个) | 百分比(%) |
| 1 | 65.3 | 1 | 65.0 |
| >1 | 34.7 | >1000 | 100.0 |

第二种方法是基于结构的纠错方法。理想的知识图谱关系应该是有向无环图，不 存在由抽象概念指向具体概念的IsA关系，于是我们可以通过查找知识图谱中的圈关 系，并将其消除来构造有向无环图，从而达到纠错的目的。但是并不是所有的错误关 系都存在在环结构当中，我们用 一 个例子6 - 5来说明这种方法的局限性。

例6-7 在 Probase当中，三元组(鹰， IsA, 鱼)是一条错误关系，但没有(鱼， IsA, 鹰)的关系，于是不存在环结构，通过构造有向无环图无法查找出这个错误。

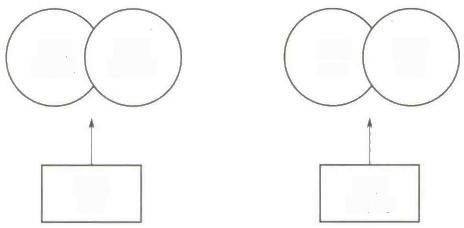
所以我们需要发现更加通用的纠错方法，使得召回率和准确率尽量同时提高。通 过观察知识图谱，我们发现许多错误都是由于实例被错误分类到错误概念引起的，例 如三元组(鹰， IsA, 鱼),鹰应该被分类到鸟类概念集合，而鸟类概念集合在人类认 知中是和鱼类集合冲突的，于是如果 一个物种同时属于鱼类和鸟类，那么这个IsA 关 系 在很大程度上是错误的。而知识图谱当中的错误基本全都是概念错分类，包括之前使 用圈结构找到的错误。由此启发，我们可以使用判断相互冲突集合这一方法来检测 错误。

定义6-7(概念集合) 给定存在IsA 关系的知识图谱，概念集合是基于此知识库构 建的，以概念点作为集合名称的，包含所有此概念的直属IsA 下概念关系的实例，每个 实例与其权重w(p) 关联。

定义6-8(矛盾集合对) 给定一个知识图谱，构建知识图谱的概念集合集，考虑任 意两个概念集合，如果它们在语义上是不兼容的，则这两个集合为矛盾集合对。

例如，我们可以通过定义6-7构建概念集合：鸟和鱼。通过定义6-8,我们将这两 个集合确定为矛盾集合对(鸟，鱼)。因为鸟和鱼在语义上是不兼容的。

两种概念之间的关系有两种：相关，不相关。其中不相关关系包括：无关联，矛 盾。我们寻找的就是知识图谱当中的不相关概念集合。图6-12和图6-13展示了两种概 念集合的关系，图6-12展示了两个相似概念集合和它们的交集所具有的内容；图6-13 展示了两个矛盾概念集合和它们的交集所具有的内容。



物种

species

北京 Beijing

动物 anima

城市 city

区域

area

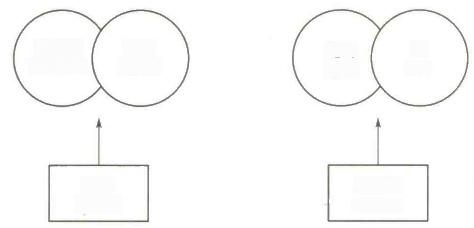
蝙蝠

Bat

(a)相似概念集合对a (b)相似概念集合对b

图6-12 相似概念集合对及交集

大数据治理：理论与方法

音乐家 musician

疾病

illness

运 动

sport

树

tree

恐惧 椰子树

phobia coconut

(a)矛盾概念集合对a (b)矛盾概念集合对b

图6-13 矛盾概念集合对及交集

上图清楚地表明了从知识图谱中建立概念集合的正确性。相似概念集合对表明两 个概念集合在语义上存在相似性，所以它们的交集里面的内容理想情况下都是符合 IsA 属于关系的，并不会出现错误。而矛盾概念集合对表明两个概念集合在语义上存在矛 盾性，所以他们的交集理想情况下应该是空集，因为一个实例不应该属于两个语义上 矛盾的概念。所以，求解矛盾集合对是在大规模知识图谱错误检测的一个重要突破口， 它可以帮助我们检测到一些频率相差非常大的关系，从而进行高准确率的错误检测。 因为理想的知识图谱矛盾集合对的交集应该为空，所以只要两个被判定为矛盾集合对 的交集不为空，则它们的交集之中存在错误的概率是非常大的。人类认知是非常容易 做到对两个概念之间的关系进行判断，可以使用众包的方式判断是否为矛盾集合对。 然而能够使机器自动判断集合的关系是一个重大挑战。因此，我们接下来的重点则是 如何让机器智能的判断两个集合是否为矛盾集合对。

受到大规模网页查重及文档相似度度量和文档查重算法的算法启发，本书首次将 应用于大规模网页查重的文本距离判断算法应用于知识图谱概念集合之间的距离判断， 通过求解概念集合之间的距离来判断概念集合对之间的关系。我们假定目前自动构建 的知识图谱提取的知识可以基本包括自然界所有常用知识和不常用知识，所以构建的 概念集合中包含的实例应当是此概念下所有IsA 关系的总和。。因此，我们可以通过知识 图谱获得所有完整的概念集合。接下来，距离的定义对于概念集合对的判断是至关重 要的。

目前针对网页查重文本相似度度量等问题已经有很多算法被提出，然而为了考虑 知识图谱所能提供的信息，以及便于求解集合距离。本书选用两种距离算法作为概念 集合距离的评判标准：Hamming 距离和Jaccard 距离。

定义6-9(H amming 距 离)给定两个概念集合A,B, 给定哈希函数h(.) 将概念集 合映射为0- 1串，集合A,B 之间的 Hamming距离为：

Hamming(A,B)=h(A)④h(B)

定义6-10(Jaccard距 离)给定两个概念集合A,B, 它们的Jaccard 距离为：



式中 |A|,|B|——A,B 概念集合中实例的个数。

**第6章**|数据质量管理

使用Hamming距离的目的是考虑利用高频率例如大于1000的频率进行集合对的判 断 ，Hamming距离可以快速地计算带有权重的集合并生成哈希签名，通过比较哈希签 名来判定集合之间的关系。Jaccard 距离排除频率因素，着重于两个集合之间交集关系 所占百分比。直观地说，如果两个概念集合为相似概念集合，则它们实例点之间的交 集应该存在大量重复；若两个概念集合为无关联概念集合，则它们实例点之间的交集 应该存在很少量重复；若两个概念集合为矛盾概念集合，则它们实例点之间的交集应 该为空。所以可以通过求解概念集合对之间的距离来判断集合的关系。

基于Hamming距离的算法伪代码描述如图6-14所示。

|  |
| --- |
| 算法Hamming Detection |
| 输入：概念集合A,B;阈值t.  输出：若为矛盾概念集合，则返回TRUE,若为相似概念集合，则返回FALSE 1:h(A)-0,h(B)0  2:for每个instance。E A,instance,EB do  3:h(A)+=Hash(instance。)\*weight(instance。)  4: h(B)+=Hash(instance,)\*weight(instance,)  5:end for  6:ifh(A)@h(B)>tthen  7: retum TRUE  8:else  9: return FALSE  10:end if |

图6 - 14 Hamming Detection 算 法

Jaccard 距离算法的伪代码描述如图6-15所示。

|  |
| --- |
| 算法Jaccard Detection |
| 输入：概念集合A,B;阈值t.  输出：若为矛盾概念集合，则返回TRUE,若为相似概念集合，则返回FALSE  1:cnt-0  2:for每个hash ∈H do  3: h(A)←hash(A)  4: h(B)—hash(B)  5: if Minhash,(A)=Minhash,(B)then  6: cnt++  7: end if  8:end for  9:Jaccard(A,B)←cnt/H  10:if Jaccard(A,B)>1 then  11: retum TRUE  12:clse  13: return FALSE  14:end if |

图6-15 Jaccard Detection 算 法

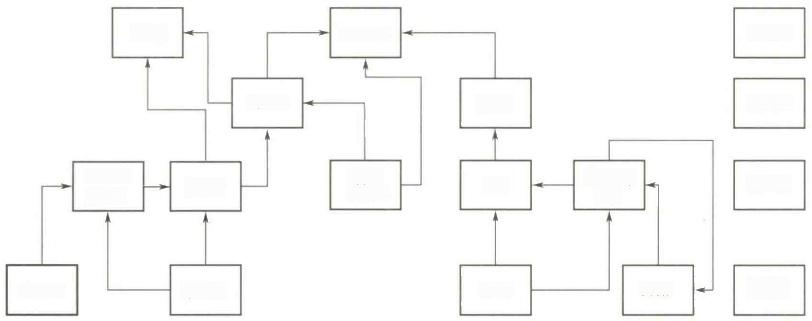
**3.** **基于图层次结合距离的纠错算法**

将层级模型应用于知识图谱的启发是由社交网络的层次模型而来。社会层次的分 级是根据社会个体在社会中的权利、财富、知识和重要性等因素进行划分。分层在社 会的各个群体当中是显而易见的，高层次人群的社会地位高于低层次的人群。随着互 联网的广泛深入，各种社交工具的发展，给定一个社交网络，社交网络中存在社会等 级推荐：定义U→V, 其中U 是V 的跟随者，则该链条关系表明U 对 V 在社交网络当

119

大数据治理：理论与方法………………

中的推荐，在社交网络图中用箭头表示。因为在现实的社交网络中，高层次人群并不 会推荐低层次人群，因为这样会导致他们的社交痛苦。该模型和知识图谱的模型结构 非常相似。类比而言。知识图谱当中的一个个实例相当于该模型当中的一个个个体， 个体和个体之间的推荐关系相当于知识图谱实例之间的 IsA关系。而最重要的一点是， 理想的知识图谱不应该存在从抽象高级关系指向具体低级关系的IsA 关系，这一点和社 交网络的社交痛苦非常相似。于是利用这一点，可以将社交网络的层次结构应用于知 识图谱的层次结构构建中。IsA 关系之间的结构关系如图6-16所示。



level 4

level 3

exciting

city

xi'an Paris level I

mobile

phone

apple

device

concept

iphone

phone

level 2

xiaomi

place

thing

device

city

图6-16 知识图谱的层次结构

在建立完知识图谱的结构后，我们可以从下而上的搜索每两个存在相交点的概念 实例，从而可以高效地找到存在不相关关系的矛盾集合对，在寻找的过程中如果发现 从上而下指向的边，则可删除，并判定为错误关系，从而构建有向无环的知识图谱。 并且如果两个集合被判定为相关集合对，则搜索终止，所以算法是收敛的。

一个理想的知识图谱应该是具有层次结构的有向无环图，其中的每个节点都具有 自己的层级。除此之外，理想的知识图谱应该是由层级低的点指向层级高的点，因为 在现实生活中，具体的概念应该属于抽象的概念。这个模型很像上述提到的 Agony Model, 所以我们可以构造一个分层函数LA, 因此， 一旦我们找到一个知识图谱的层级

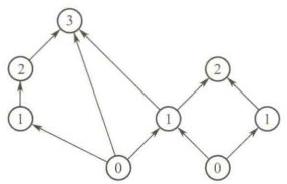


图6-17 IsA 关系有向无环图

分配，我们就找到了这个知识图谱构造有向无环图的 方法。因此，如果有一个好的层次分配函数，就意味 着可以利用层级来发现错误的 IsA 关系。在图6-17中 我们展示了一种有向无环图的分层结构，而对于知识 图谱的分层结果也应完成图6-17所示的任务。

构建如上的 IsA 关系有向无环图面临的最大挑战就 是怎样确定分层函数 LA, 而分层函数当中一个重要的

问题就是，如果出现错误层级的指向边，即从高层次

的实例指向低层次的实例，则这条边需要受到惩罚。

算法模型如下：

如果x,y₁,y₂ ∈V 且至少存在边e₁,e₂ ∈E 使 得e₁=(x,IsA,y₁),e₂=(x,IsA,y₂)

且

**第6章** |数据质量管理

l(y₁)-l(x)=1;l(y₂)=1(y₁) 且 e₃,e₄∈E, 使 得e₃=(y₁,IsA,y₂),e₄=(y₂,IsA,y), 则

对概念集合Y₁,Y₂ 进行矛盾概念集合对判断，使用上节提到的 Hamming距离和 Jaccard 距离进行判断。

上述算法模型的设计依据是知识图谱的层次结构，算法的开始实施是从下而上的 对每个概念集合进行比较，算法进行的终止条件是如果两个概念集合被判定为相似概 念集合，则停止向上搜索，直到遍历完所有的概念集合。设计这个算法的优势是因为 基于集合距离的错误检测算法需要对所有的概念集合进行两两对比，然而其中有一大 部分的比较工作是无用的，所以可以通过图层次结合距离模型进行矛盾概念集合的求 解。该算法的最坏情况是知识图谱当中的所有概念集合都为矛盾概念集合。但是这种 最坏的情况是不存在的，因为假设知识图谱足够大并包含了世界上已有的知识，则知 识图谱所存储的概念不可能是全部矛盾的，因为概念之间是有联系的。所以该算法确 实可以加速矛盾集合的计算。然而该算法存在的问题是求解出的矛盾集合对个数小于 等于基于集合距离的错误检测算法，所以检测出的错误召回率低，但是可以保证较高 的检测速度。

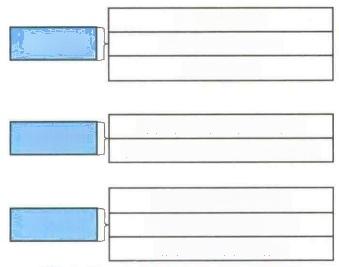
6.6 面 向 大 数 据 的 数 据 清 洗

**6.6.1** **大数据清洗的探索**

在当今的信息化时代，随着互联网的兴起和产业的数字化，各种数据量急剧增长，

也使得数据质量越来越不容忽视。数据质量管

混合错误清洗

理更依赖技术：数据规模大 →难以全面掌握数 **模态多样** 时间序列错误清洗

据特征；数据来源多样 → 依靠人去掌控数据质 元数据清洗

量成本高；数据变化快 → 人的响应速度慢，难

以满足要求；数据获取与加工流程复杂 → 人难 并行数据清洗算法与系统

**计奠困难**

以全面监控流程。 扫描数据一次的错误检测算法

大数据清洗的迫切需求使得面向大数据的 基于众包的数据清洗

数据清洗领域涌现出非常多的探索研究，大数 **缺少知识** 基于知识库的数据清洗

据清洗面临模态多样、计算困难和缺少知识的 基于Web的数据清洗

三大挑战，如图6-18所示。 图6-18 大数据清洗面临的挑战

高质量的数据是人们所追求的目标，下面我们将简要介绍一下大数据清洗的几类 方法。

**6.6.2** **时间序列错误清洗**

数据序列中经常会存在一些脏数据或不精确的数据，如 GPS 轨迹数据或传感器读 入的序列。据调查，股票价格的数据中有时也存在着脏数据。为了清洗这些脏数据， 人们经常使用基于限制的修复方法。现有的清洗序列数据的研究方法考虑了速度的限 制，即序列数据中值变化速度的限制。例如，燃油表的速度限制条件要求起重机的油 量消耗不能为负值，且不能超过每小时40公升。基于限制的清洗能够识别出违反上述



大数据治理：理论与方法

速度限制条件的情况，并且修复脏数据，使得修复后的数据满足速度限制的条件。

当速度限制方法成功地识别到大幅度变化的错误时，它将序列数据中脏的一个数 据修复为最大或最小的速度。这种将错误值调整为最大或最小值的方法源于数据修复 中的最小变化原则。基本原理是，在实际中，人们或系统都尝试将错误最小化。最小 变化原则被广泛应用于修复关系型数据中。

下面我们介绍一种引入速度变化的可能性分布的时间序列错误清洗方法，这一方 法通过计算速度变化的可能性，找到具有最大速度变化可能性的序列作为修复后的序

列，能够弥补基于速度限制的序列数据清洗方法的不足。

这一方法的核心思想如下。

速度是根据值的变化来定义的，例如，从点i-1 到 点 i 的速度为：



其 中x, 代表第i 个数据点的值，t, 代 表x, 的时间戳。于是有：

(1)

我们将 u,作为第i 个点前后速度的变化。

一个序列x 的可能性L(x) 表示为：

(2)

其中，P(u₁) 代表速度变化u,发生的可能性，L(u₁) 代表相应的可能性(用log①计算)。

速度变化的经验概率P 可以根据序列的数据被预测。

下面我们通过一个例子来说明概率分布和可能性计算。

例6-8 给定一个序列x={11,12,15,14,15,15,17}, 对应的时间戳t={1,2,3,

4,5,6,7}。图6-19表示出了序列中所有的数据点，图6-20展示了相应的速度变化概率

分布。

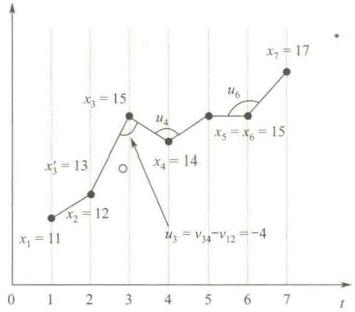


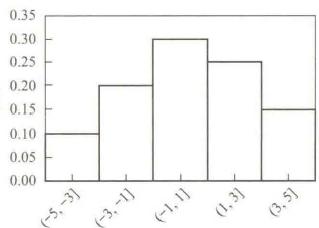
图6-19 序列中所有的数据点

① 本书中，log表示以10为底的对数。

122

第6章|数据质量管理

概率



速度变化

图6-20 速度变化概率分布

第3个点 (x₃) 的速度变化概率为：



可能性L(u₃)=log(0.1)=1.2 。 同样地，我们计算其他点的可能性。最终可以得 到序列x 的可能性：

L(x)=log(0.25)+log(0.1)+log(0.25)+log(0.2)+log(0.25)=-8.1

x₃=15 的值是错误的，相应的正确值应该是x'=13。 根据这个真实的值，可以得到 x' 的可能性L(x')=3log(0.3)+log(0.2)+log(0.25)= -6.6, 比包含脏数据的序列x 的 可能性更高。

根据上述核心思想，可以构造出基于动态规划的伪多项式时间算法。根据公式 (1)和公式(2),我们可以知道在点i 上的速度变化和相应的可能性取决于前一个点 i-1 和后一个点i+1 。 因此，为了获得最优解，我们需要构造出一个关于x/-2,x{, 和 x' 的递归表达式，如下所示：

D(i,c;,x{,x()=max[D(i-1,c;-,x(-2,x'-)]+L(u()

其 中c; 表示修复的代价。

对于 i=2, 有：

D(2,c₂,x{,x²)=0

令D(n,c,x′,x()为所有c,x,x 的最大可能性。通过收缩所有的D(i,c;,x,x) 可以得到这个最大的可能性，从而获得最优解x'。

通过对计算过程进行近似化，我们可以得到线性时间的启发式算法和常数因子近 似算法。通过将离散的可能性分布近似为连续可能性分布，可以获得变形近似算法和 贪心近似算法。

下面介绍基于相关性的高维时间序列清洗技术。

时间序列上的约束可以捕捉和描述数据的性质和数据间的关系，通常具有实际的 含义，而基于约束的时间序列数据清洗方法通常遵循最小修复代价原则，从而避免了 基于平滑的清洗方法过度修改大部分正常的数据的情况，具有较好的清洗效果。但是， 目前对于时间序列约束的讨论较少，基于约束的时间序列异常检测存在以下几个难点。

(1)约束的定义。有两种方式可以制定约束：从数据中挖掘得到，或结合实际生 活与领域知识进行定义。从数据中统计得到的约束只能反映当前部分数据的信息，不

大数据治理： 理论与方法…

能简单地认为后续数据一定满足该约束，这种方式可以较好地把握近一段时间数据的 特征，但需要根据数据增长不断调整。如果某数据反映了物理世界中的某个变量，那 该数据一定遵循物理世界的客观规律，使用这种规律来制定约束，可以保证约束的永 真性，但这种方式定义的约束通常范围较大，会漏检违反程度较小的异常。

(2 )优化目标的选择。基于约束的时间序列数据清洗方法通常遵循最小修复代价 原则，通过最少的数据值修改使得数据满足约束。但在异常检测中，通常使用删除异 常点或替换异常点的操作来消除异常，即通过删除或替换部分数据点使序列满足约束。 由于优化目标函数不同，可能导致检测结果不同，所以不宜直接使用基于约束的清洗 算法进行异常检测。

高维时间序列 (Multiple Time Series)S包含N 条时间序列S,,S₂,…,S, 。 每条时间 序列都可以用一个三元组表示： S=(x₁,t₄,n₄) 。 约束是定义在高维时间序列上的、用 于描述数据或数据关系必须要遵循的限制。根据约束的定义域不同，将约束分为四类， 如图6-21所示。

|  |  |  |
| --- | --- | --- |
|  | 定义在一条序列上 | 定义在多序列上 |
| 定义在一个时刻的数据上 | 第一类：定义在单序列的 单数据点上 | 第三类：定义在多序列的 一个时刻的数据上 |
| 定义在多个时刻的数据上 | 第二类：定义在单序列的 多个数据上 | 第四类：定义在多序列的 多个时刻的数据上 |

图6-21 四类约束

考虑到对于时间序列异常检测而不是直接修改数据的需求，同时在进行异常检测 时，我们通常会将异常点确认后剔除，然后将其视为缺失值填充进行修复，因此优化 目标是最小化异常点数量。首先定义速度约束下的异常点：给出单维时间序列S=(x,t,

n) 及速度约束SC(sm,S ), 如果去掉 E 中下标对应的数据点，剩余任意两个数据点 均满足速度约束，那么 E 为时间序列S 在速度约束 SC 下的异常点下标集合。

给出单维时间序列S=(x,t,n) 及速度约束SC(sm,Sm), 求解最小的异常点下标集 合E, 即求解具有最小|E| 的 E。

接下来介绍一个时间复杂度为 O(n²) 的动态规划算法求解输入序列S 中的最小异 常点集合E,n 是输入序列的长度。令S[1:i] 表示输入序列的子序列，包括序列的第 1-i个数据点。令dp[i] 表示：在不删除第i 个数据点的情况下，使子序列S[1:i] 满足 速度约束，最少需要删除的异常点数量。如果两个数据点下标i,j(i<j) 对应的数据满足

速度约束，即s ≤(x[j]-x[i])/(t[j]-t[i])≤s , 那么可以通过删除第i+1 到 j-1

下标的数据点，保留第；个数据点，使得子序列S[1:j] 满足速度约束。于是得到 dp

[j]=dp[i]+(j-i-1) 。 假设通过删除 dp[j] 个数据点，使得S[1:j] 满足了速度约束。为

了使整条序列满足速度约束，简单地删除第j+1 到 n 下标的数据点。于是， 一共需要删 除dp[j]+(n-j) 个数据点，使得整个序列满足速度约束。在通过dp[i] 更新 dp[j] 时， 记录更新路径 trace[j]=i 。 在计算完成后，沿着 trace 追踪到所有保留下的数据点，那 么剩余的数据点就是被删除的异常点。于是就得到了朴素动态规划算法。该算法的伪 代码描述如图6-22所示。

第6章|数据质量管理

|  |
| --- |
|  |

图6-22 朴素动态规划算法

该算法的主要时间花费在第2~6行中，即对于一个j, 需要遍历所有的i=1,2,…, j-1, 寻找满足速度约束的i, 尝试使用dp[i]更新dp[j]。可以看到，对于一个下标j, 算法第3~6行是为了寻找满足下列条件的下标i:

1. i<j

2. 

3. 

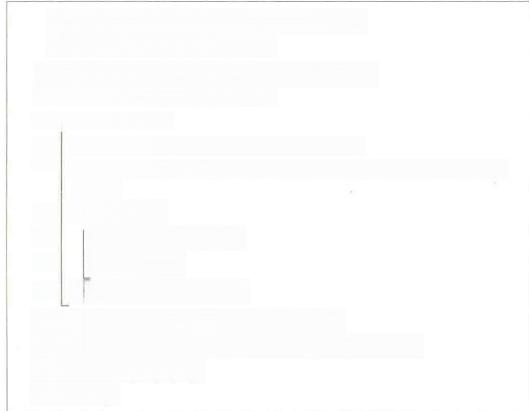
4.dp[i]+(j-i-1) 最小

这相当于在X≤X[j],Y≥Y[j] 的二维平面内，寻找f[i] 值最小的i 。这个问题可 以使用二维区间树在O(log₂n)时间、0(nlogn)空间内完成。

可以使用二位区间树来对算法进行优化：在第二维区间树的非叶节点上额外保存 了以下信息：该子树中最小的f[i] 值和对应的i。对于一个范围在X≤X[j],Y≥Y[j] 的查询，首先在第一维区间树上查询X≤X[j] 的范围，得到了一些非叶节点，它们的 子树包含的所有叶节点对应的数据点均满足X≤X [j] 。 然后，访问这些非叶节点对应 的第二维区间树，在上面查询Y≥Y[j]范围内最小的[i] 和其对应的下标i。由于非叶 节点上保存了其子树中叶节点的统计信息，所以可以直接得到子树中最小的f[i]和对 应的i, 不需要再去花费额外的时间依次遍历每个数据点。这样一次查询的与修改的代 价均为O(log₂n) 。 伪代码描述如图6-23所示。

时间复杂度为O(log,j)≤0(log₂n)。共计执行n 次查询与插入操作。初始化、寻找 未被删除的点、得到异常点下标的操作时间复杂度均为O(n), 于是该算法的时间复杂 度为O(nlog₂n), 空间复杂度为0(nlogn), 其中n 是输入的时间序列的长度。

大数据治理： 理论与方法



Input: 单维时间序列S, 速度约束SC(Smin,5mm)

Output: 最小的异常点下标集合E

1初始化 dp[j]—i-1,trace[]--1 fori-1,…,n;

2初始化一棵空的二维区间搜索树T.

3 forj-1,…,n do

4

X[]-x[j]-[U]\*Smmand YU]-x[]-/[/]\*smx;

5 在T中查询x ∈(-,X[])and y∈[YU],)范围内最小/[i]值对应的下

标 i;

6 ifi 存在 then

7 dp[/]—dp[]+(-i-1);

8 trace[j]—i;

9 向T中新增点(XT),YUJ);

10找到使dp[index]+(n-index) 最小的index;

11使用index -trace[index] 追踪未被删除的数据点集合/; 12E-{1,2,3,…,n}}I;

13 return E;

图6 - 23 基于二位区间树的优化算法

**6.6.3** **基于众包的数据清洗**

众包质量管理是指一定数量的工人对任务集合给出了答案，我们根据所给出的答 案对任务的正确答案和工人的质量进行预测。

例如， 一个数据科学家想要为微博设计一个情感分析算法。为了训练出该算法， 他需要一个对相应微博的情感做出了等级评定的训练数据集。每条微博需要被评定为 1、2或3。其中，1表示消极情感，2表示中立，3表示积极。如果将这项工作在众包 平台上被外包给工人，那么由于工人可能出现错误的回答，因此我们需要多个工人来 评定每一条微博的情感。这里，众包质量管理的目标就是预测出每条微博的正确情感 及每个工人的准确率。

接下来，我们将介绍面向全局最优的众包数据质量管理方法。

一个直观的方法是考虑所有从项到等级的映射，针对每个映射，推理出它的可能 性大小，最后选择具有最大可能性的映射作为全局最优解。

假定我们的数据集中包含500 条微博，且每一条微博都有三个可能的情感等级。 那么,从微博到等级的映射总数就是350个，这是一个相当大的数字。因此，这个直观 的方法并不可行。

为了减少这个指数级的复杂度，我们采用两个简单的修剪技术来减少映射的数量， 把3这个大数变化成一个能够管理的数字。我们首先假设所有工人都是同一的，他们 符合相同的误差分布曲线。然后，我们采用第一个修剪技术，即将收到相同等级评定 的微博哈希到同一个桶中。假定有300条微博，每条微博都收到了三个等级为3的评 价，有100条微博收到了一个等级为1、 一个等级为2和一个等级为3的评价，还有 100条微博收到了三个等级为1的评价。于是，我们把所有的微博划分成三个桶，与工 人的答案集合相对应，即B₁={3,3,3},B₂={1,2,3},B₃={1,1,1}。 我们把每个桶中

 126

**第6章**| **数据质量管理**

的微博同等看待。因此，只需要考虑不同桶中互不相同的映射即可。由于图6-24中有 三个不同的桶，所以我们只需要考虑3³个映射即可。

接下来，我们采用第二个修剪技术来减少 映射数量。我们认为收到工人评价更高的项最 终被分配的等级也会更高。根据这个，我们对 桶进行支配排序。在图6-24中， B₁≥B₂≥B₃。

我们将现有的27个映射中不符合支配排序的映 射剪去，剩余的有(3,3,3),(3,3,2),(3,3, 1),(3,2,2),(3,2,1),(3,1,1),(2,2,2),

(2,2,1),(2,1,1),(1,1,1)10个映射。

在叙述了核心修剪思想之后，我们以过滤 问题为例，说明全局最优的众包质量管理算法 的主要流程。

工人评价 项

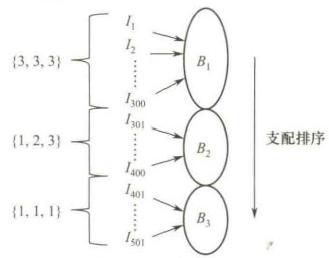


图6-24 映射实例

例6-9 给定一组图片，过滤任务是找出这组图片中所有包含奥巴马的图片。对于 每一张图片，我们询问工人，“这张图片上是否是奥巴马?”。工人可以给出0或者1作 为答案，其中0表示“不是”,1表示“是”。

算法的第一步是对工人的答案进行装桶化。假设每一张图片由m 个工人来判断， 我们有m+1 个桶，每个桶中的答案从0个0到m 个0不等。我们用图6-25简要表示这 些桶。 x 轴代表每一张图片收到的答案为1的个数，y 轴代表每张图片收到的答案为0 的个数。同一个桶中的图片最终会被分配相同的值。

在图6-25中，(3,0)代表该图片收到了三个1和0个0,(2,1)代表该图片收到

了两个1和一个0,以此类推。因为相同桶中的图片具有相同的值，即f(a)=f(b), 且图片I₃=1, 所以f(l₃)=f(1₄)。

算法的第二步是对桶进行支配排序。收到答案1越多的图片的真实值越有可能是 1,因此有 (m,O)→(m-1,1)→ …→(1,m-1)→(0,m) 。 图6-25通过有向边表示了 不同桶间的支配关系。我们将所有为一个桶分配0却为其支配的桶分配1的映射丢弃， 如将为(3,0)分配0却为(2,1)分配1的桶丢弃。

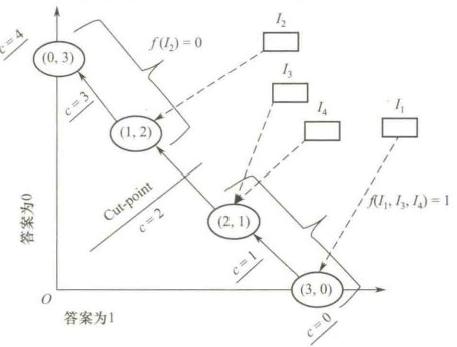


图6-25 分割点划分实例

**大数据治理：** 理论与方法………………

我们将满足算法第一步和第二步限制的映射称为支配性一致的映射。为了构建出 支配性一致的映射，我们选择一个分割点将排序好的答案集合划分为两部分。为前半 部分分配答案1,为后半部分分配答案0。例如，如图6-7所示，如果在(2,1)和 (1,2)之间选择分割点，会使得项I 、I₃ 、I₄ 被映射为1,项I₂ 被映射为0。我们有5 个可能的分割点，即0、1、2、3、4,每个分割点都对应着支配性一致的映射。如分割 点0对应着将所有项映射为0,分割点4对应着将所有项映射为1。

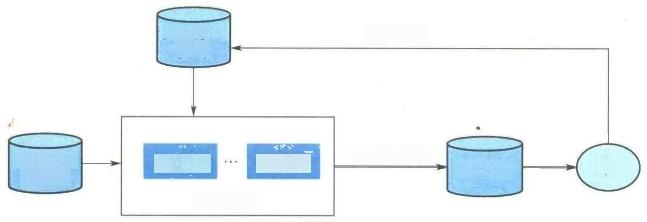
该算法枚举了所有的支配性一致映射，对它们的可能性进行了计算，并返回了最 大可能的映射。如果有m+2 个映射，那么每个映射的可能性能够在O(mlll) 的时 间内计算出来，算法的时间复杂度为0(m²lIl)。

**6.6.4** **基于用户反馈的数据清洗**

**1.** **用户反馈的意义**

主动学习以有限的标记实例和大量没有标记的实例开始。这些标记的集合形成了 初始训练数据，用于分类器的初始训练。主动学习的目的是从无标记的这些实例中寻 找出能最快速度加强分类器精确性的实例。那我们应该使用什么样的标准来挑选这些 实例?初始的分类器对其中的一些无标记实例的分类是很肯定的，但对大部分实例分 类是不确定的。这些不确定的实例落在分类器中的模糊区。这个模糊区在训练集小时 是很大的。分类器可以通过对这些不确定实例的分类来减小它的模糊区。这种直觉形 成了主动学习主要标准的基本依据之一。我们举几个例子来展示基于不确定得分选择 的实例是如何帮助减少分类器模糊区的。

图6-26展示了基于用户反馈主动学习模型的整个框架。整个过程主要有以下四个 阶段。



少量标记

实例

委员会1 委员会”

记实例

学习模型

不确定得分

用户标记

大量无标

N个实例

用户

图6 - 26 基于用户反馈主动学习模型

阶段1:学习模型的初始化。这里的学习模型可以是贝叶斯分类器、决策树或支持 向量机等机器学习模型中的任何一种。这个阶段主要是通过手工标记的少量实例对学 习模型中的各个委员会进行训练，从而得到一个初始的学习模型。

阶段2:选择标记实例。这个阶段主要是根据某种得分机制从大量没有被标记的实 例中选择价值最大的N 个实例出来。这N 个实例经过标记加入训练集后可以最大程度 加强学习模型分类的正确性。

阶段3:用户反馈和学习模型重训练。在这个阶段，用户对学习模型挑选出来的带

128

**第6章**|数据质量管理

标记实例进行标记并将它们加入训练集中。然后学习模型重新训练，由于阶段2中选 择了价值最大的实例进行标记，因此在下次迭代中学习模型的分类正确性得到最大 加强。

阶段4:循环训练。重复阶段1~3,直到用户对学习模型预测正确性满意为止。此 时即使没有用户参与，学习模型对为标记实例的分类结果是非常准确的。下面，我们 先用一个数据实例来说明基于用户反馈主动学习模型运作过程。然后讨论模型的主要 应用场景。

基于用户反馈的主动学习模型通过上述四个阶段的运作，不但减少了用户的参与 程度，还以最小的训练集得到非常精确的学习模型。

**2.** **不一致性修复**

为了将数据库中的不一致性数据找出来。我们需要数据依赖，因此可以通过违反 依赖的准则来找出不一致性数据。但是我们将使用那些依赖呢?本章我们主要考虑三 种依赖：条件函数依赖，条件包含依赖和匹配依赖。

条件函数依赖 如前文所述，关系R 上的条件函数依赖 (CFD) 可以表示为：φ:

R(X→Y,T,) 。 这里X→Y 是一个标准的FD, 而 T, 是在属性集X 和 Y的一个匹配模

式，{XUYAttr(R)} 。 对于每个属性Aε(XUY) 和每个模式元组t ∈T 。,t,[A] 要么

是域 Dom(A) 中的一个常量a 或者是一个变量“-”。4: R(X→Y,T,) 如果是标准形 式，那么T, 仅仅由一个模式元组t, 组成，Y由单个属性A组成。我们将φ简写成(X→ A,t₀), 将X 和A表示成LHS(φ) 和 RHS(q)。

一个CFD规则φ:R(X→Y,T,), 这里Y={A₁,A₂,…},T,={t,₂,…} 能表示成 CFD规则集2。。2。中的每个 CFD都是标准形式，2。=|4:(X→A₁,tn),φ₂:(X→A₁,

tg),…} 。 考虑标准形式的CFD规则φ:R(X→A,t), 如果t,[A]≠“-”, 我们称 为常量 CFD; 如果t,[A]=“-”, 我们称为变量CFD。

条件包含依赖 条件包含依赖我们也在前文中讨论过。定义在关系模式R₁,R₂ 上 的条件包含依赖 (CIND) 可以表示w:(R₁ [X;X,] ≤R₂ [Y;Y,],T,) 。 其中， (X,X₂) 和

(Y,Y₄) 是属性集Attr(R₁) 和 Attr(R₂),R₁[X]CR₂[Y] 是标准IND,T, 是规则w 在属性

集X, 和Y,上的匹配模式。对于每个属性B∈(X,UY), 每个元组模式t,εT, 在属性B

上的值t,[B] 是域Dom(B) 上的一个常量。

一个条件包含依赖规则w:(R₁[X;X,]≤R₂[Y;Y,],T,) 如果是标准形式，那么T, 仅 仅由一个模式元组t, 组成，X 由单个属性A组成，Y由单个属性B 组成。我们将规则w 表示为(R₁ [A;X₁]≤R₂ [B;Y₁],t), 并且将A 和B 表示成LHS(w) 和RHS(w)。 一个条 件包含依赖规则w:(R₁[X;X,]≤R₂[Y;Y,],T₂), 这里X={A₁,A₂,…},Y={Y₁,Y₂,…|,

T,=1,tg,…| 能用一个条件包含依赖规则集2。来表示。Z 。中的每个 CIND 都是标

准形式， E 。=\w₁:R₁[A₁;X,]CR₂[B₁;tμ],…}。

匹配依赖 定义在关系模式R 上的匹配依赖MD 可以表示为：

ψ:(X→Y,λ)

这里XCAttr(R),YCAttr(R),λ 是属性集XUY 匹配相似度的阈值。

一个匹配依赖规则ψ:(X→Y,λ) 如果是标准形式，那么Y由单个属性A组成。我们 将ψ简写成ψ:(X→A,λ), 将 X 和A 表示为LHS(ψ) 和RHS(ψ)。

大数据治理：理论与方法……………

一个匹配依赖规则ψ:(X→Y,λ), 这里Y={A₁,A₂,…} 能用匹配依赖规则集之。来 表示。之。中的每个规则都是标准形式E。={ψ₁ :(X→A₁,λ),…}

下面将介绍一种遗传模型来产生候选修复方案。这个遗传模型尝试在最小化代价 函数的，同时满足约束条件的情况下，选择解决方案。

问题定义：给定由表R₁,…,R, 组成的数据库D, 定义在该数据库中的约束集合W 和一个代价函数 Cost。任务是寻找干净数据库D', 使得它满足约束集 W, 同时代价 Cost(D') 最小

为了解决这个问题，我们引入三种约束条件：

(1)CFDS;

(2)CINDs;

(3)MDs 和一个代价函数。对于这个问题，首先应该决定如何来表示问题的答案。 我们能使用约束条件W寻找数据库D 中的脏元组，同时创建一个遗传模型来寻找元组 的最好修复元组使得 Cost(D) 最小，同时满足约束条件D。

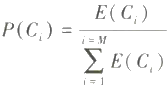
为了产生候选修复方案，传统启发式算法是效率低并且这个问题可以抽象成一个 优化问题。正如上面所述，数据库的一个修复方案能编码成一个序列。因此，我们尝 试使用遗传算法解决这个问题。遗传算法模型定义如下：

GM=(C,E,P₀,M,φ,F,ψ,T)

这里C=[u(t),u(t₂),…,u(t,)] 是个体编码方法，它是个体适应函数， P₀=[C₁,

C₂,…,Cy] 是初始入口，它可以通过对每个入口分配一个随机变量得到，M 是人口的大 小，φ是选择因子，F 是交叉因子， 是变异因子， T 是终止条件，它是适应度函数的一 个阈值。

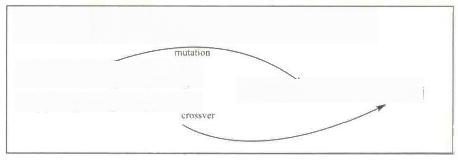
选择因子：它的目的是从当前人口中选择最优秀的个体，以至于使他们有机会繁 衍下一代。这里，我们使用轮盘赌选择因子。这样一个个体被选择的概率



变异因子(ψ):变异是对一个已存在的答案产生随机的改变。在本例中，变异可 以通过挑选其中一个元组修复方案，用改元组的另一个修复方案来代替。

交叉因子 (F): 交叉是挑选最佳的两个答案，由他们组合成新的答案。在本例 中，交叉是通过挑选一个答案中的若干元素，用另一个答案的其他元素来代替。

这两个因子可以用图6-27来阐述。

fug)a(s),u()a(s)-1u)a)ma

*[utm]a(,m(),u(la)a()]*

[u((₀),u(h₂),u(₃),u(t₄),u(t₅)]

*[u(),μ(),a(),n(ka),a(as)]*

图6-27 交叉和变异因子

GenerateRapair 算法伪代码如图6-28所示。

**第6章** **|数据质量管理**

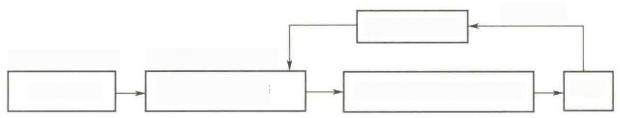
|  |
| --- |
| 算法6-1 GenerateRepair(D,Cost,W) |
| 输入：数据库D,代价函数Cost和约束集W  输出：最优解[u(t₁),u(t₂),…,u(t)]  1:通过数据质量约束规则W识别出数据库D中的脏元组，并将其存放在DirtyTuples列表中 |
| 2:根据DirtyTuples,随机建立一个满足约束规则的修复方案集合P₀=[C₁,C₂,…,Cx],即初代人口 |
| 3:while max{E(C)}<T do |
| 4: 计算代价函数以得到一个排序后的解列表 |
| 5: 通过轮盘赌φ,变异因子ψ和交叉因子厂建立下一代 |
| 6:end while |
| 7:选择最优解[u(),u(t₂),…,u(t,)] |

图6-28 GenerateRepair 算法伪代码

第1步是通过数据质量约束规则识别出数据库中的脏元组。第2步是随机的创建 数据库的修复方案集合，就是我们熟知的人口。

在优化的每一步(步骤3~6),整个人口的代价函数被计算出来，并且对它们进行 排序。在这些方案排序后，新的人口作为下一代被创建。首先，将当前人口中的精英 加入新人口中，其他的新人口完全由上一代的精英通过变异和交叉产生。

在第7步中，遗传算法产生了数据库最好的修复。这个算法的时间复杂度不会超 过O(aN), 其中N 是数据库中元组的数量，a 是常量。创建一个用来嵌入反馈的贝叶斯 机器学习模型的目的是对上文产生的候选修复方案的正确性进行确认。我们的学习模 型是建立在贝叶斯理论的基础之上。在这个模型中，对于每个属性C;e Attr(R), 可 以使用训练集 T 创建用于投票的若干委员会的贝叶斯分类器。为了改善这个学习模型， 依据不确定得分挑选出分类器最不确定的n 个候选更新，用户对这些更新的反馈构成 新的训练集对学习模型进行再训练。如图6-29所示。



FeedBack

Training data T

Learning component

{M₂,M₂,…,M

Suggested repairs

u=<1,A,v>

Select n repairs for labeling User

图6-29 基于贝叶斯理论的学习模型

整个过程的伪代码描述如图6-30所示。

**3.** **真值发现**

一般情况下，我们认为一个真值由大多数数据源提供，而一个错误值只有特定的 几个数据源提供。这样我们可以应用投票 (Vote) 算法，将大多数数据源提供的值作 为真值。

**算法描述：** 给定对象0和N 个数据源在其属性A上的N 个值集W=|v₁,…,vx}, 对 象O 属性A 的真值是集合W中频率出现最高的元素。

BVote 算法的流程如图6-31所示。

大数据治理： 理论与方法

|  |
| --- |
| 算法6-2 UFeedBackRepair(D,E,T) |
| 输入：待修复数据库D,清理规则卫，训练数据T  输出：修复后数据库D'  1:利用T训练一个贝叶斯分类模型的集合{Me,Mc,…Mc}  2:利用 ∑识别出D中的脏元组，并存放在DirtyTuples列表中  3:while存在脏元组do  4: 对所有t∈DirtyTuples且A∈Attr(R)调用GenerateRepair()生成格式为<t,A,v>的更新  5: 贝叶斯机器学习模型预测出推荐的修复  6: while用户不满意所给出的预测do  7: 学习模型选择n个修复等待标记  8: 用户针对这些修复交互式地给出反馈  9: 将这些被标记的修复添加进训练数据集T,重新训练模型  10: 重新生成预测  11: end while  12: 将机器学习模型的决策应用至数据库  13: 检测新的脏元组并生成修复  14:end while |

图6-30 基于贝叶斯理论的学习模型伪代码

|  |
| --- |
| 算法6-3 BVote(T,V,A) |
| 输入：数据源集T,对象V和属性A  输出：对象V在属性V中的真值，用<V,A,v>表示  1:从数据源集T中找到对象V在A上的取值集合(v,,…,vy 2:统计出现频率最大的v  3:返回<V,A,v> |

**图6-31** **BVote算法的流程**

但不幸的是，数据源的复制在实际中经常发生，尤其在Web 领域。由一个数据源 提供的变量，无论其真假，都可能被其他数据源直接复制过去，这种情况下寻找真值 变得非常困难。另一个问题是如何确保产生真值的正确性。

NFTF 即朴素用户反馈真值发现框架。它假设不同数据源之间相互独立，即不存在 一个数据源中的数据时从其他数据源复制而来。这种情况下每个源对每个对象的贡献 值为1。考虑到基本投票算法 BVote() 产生的真值并不能保证正确性，我们提出了融入 用户反馈机制的NFTF框架。NFTF框架其实就是利用基本投票算法 BVote() 产生候选 真值，为了确保真值的正确性，创建了用户反馈机制的主动学习模型对候选真值进行 确认。我们针对每个属性A,∈Attr(S) 通过用户反馈的主动学习方法训练了一系列分类 器{M₄,M₄,…,M₄} 。 如图6-32所示。

朴素用户反馈真值发现伪代码如图6-33 所示。

第1步，对给定的样本集合U, 我们针对每个属性A∈Attr(S₁) 训练一系列机器学

习模型AL=|M₂M₄₂,…,M₄, 。 第2步，我们通过基本投票算法BVote() 产生对象0

在属性A中的后候选真值<0,A,v>, 并将他们存在在列表 TruthList 中。

步骤3~9是整个框架的核心。为了确定通过BVote() 产生候选真值的正确性，我

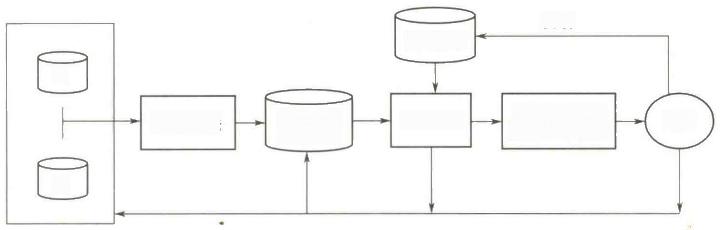
们使用机器学习模型AL对所有候选真值进行确认(步骤4),然后从候选真值集中选择

 132

**第6章** **|数据质量管理**

反馈

n 个学习模型最不确定的更新推送给用户(步骤6),用户对这些更新最初反馈，并将 反馈结果添加到样本集U中对学习期进行重训练(步骤7~8),使得在下一轮迭代中学 习模型 AL的预测结果更加精确。



训练样本

S₁

基于不确定得 分选择n个实例

S

候选真值 <0,A,v>

BVote算法

学习组件

用户

图6-32 朴素用户反馈真值发现框架

|  |
| --- |
| 算法6-4 NFTF\_Process(T,U) |
| 输入：数据源集T={Sj,S₂,…,Sw}和训练样本集U  输出：正确数据源S  1:通过训练样本集U针对每个属性A∈Attr(S)训练一系列机器学习模型 AL={M₄,M₄…,M} |
| 2:针对每个对象O的每个属性A调用BVote()算法产生形如<O,A,v>候选真值 集合TruthList |
| 3:while用户可供利用且存在没有确定真值的对象do |
| 4: 使用机器学习模型对所有候选真值进行确认 |
| 5: while用户对机器学习模型的预测不满意do |
| 6: 从候选真值集中选择n个最不确定的候选真值 |
| 7: 用户交互对这些候选真值给出反馈 |
| 8: 新标记的候选真值加入到样本集U中对机器学习模型进行重训练 |
| 9: done |
| 10: 用户和机器学习模型确认的候选真值添加到真确数据源S中 |
| 11: 更新候选真值集TruthList |
| 12:done |
| 13:return S |

**图6-33** **朴素用户反馈真值发现伪代码**

在第10步，用户和学习模型 AL确认正确的后选择很值添加到正确数据源S 中。 第11步，更新整个候选真值集TruthList, 主要是移除已经被确认的候选真值，并对拒 绝的候选真值用心的值代替 。

最后，输出正确数据源S 。所以对象的真值都存储在S 中。

**面向半结构化数据、图数据的数据清洗**

**1.** **知识图谱数据清洗**

知识图谱是采用结构化的形式，描述客观世界中存在的各种概念、实体和它们

133

大数据治理：理论与方法………………

之间的关系的一种语义网络结构，为互联网时代的海量信息管理提供了更加强大、 便捷的工具。知识图谱的概念在2012年由谷歌正式提出，作为其增强搜索引擎功能 的知识库。除互联网语义搜索外，知识图谱已陆续应用到海量数据分析、智能问答 系统、智能推荐系统、专家系统等研究方向，成为推动人工智能领域发展的关键技 术之一。

在详细介绍基于需求结合语义的知识图谱概念层次剪枝算法前，本节将对知识图 谱剪枝过程中涉及的关键概念进行定义。用 G=(V,E,T) 描述一个知识图谱，其中V是 概念或类型节点的集合，T 是关系集合，E 用于描述三元组集合，三元组(a,b,t)εE,

其中a,beV,t∈T

本章所介绍的知识图谱剪枝任务，适用于具有非自反的、非对称的、可传递的关 系的知识图谱G, 当知识图谱G中存在以下两种情形时，可以定义为存在噪声：

(1)存在不满足前述非自反的、非对称的、可传递的关系的三元组。以维基百科 概念为例，存在大量不满足上述三个条件的关系路径，如人工智能→人工神经网络→ 人工神经网络。

(2)存在非必需的节点。以本文的剪枝任务为例，在针对某一领域进行剪枝的过 程中，可能出现不属于该领域或相关性极弱的节点，此类节点即非必需的节点。除此 之外，还存在语义重复节点，如“神经网络”和“人工神经网络”,两个节点所表达 的意思相同，指向的节点也相同。以上两类节点的存在会造成浪费存储空间、降低查 询效率、降低查询结果的准确程度，甚至在关系推理中导致错误的推理结果，随时间 推移，造成积累性的错误，大大降低整个知识图谱的正确性。

(3)存在冗余路径。仍以维基百科中的概念为例，从机器学习到深度学习存在两 条路径：机器学习→深度学习，以及机器学习→人工神经网络→深度学习。

当以上任意一种情况存在时，即代表该知识图谱G 是有噪声的知识图谱。

对知识图谱问题的研究，可以等价于对图问题的研究，知识图谱的剪枝任务可以 抽象成为一个子图搜索问题。根据定义一中对噪声的定义以及用户的需求，子图搜索 的目标有三条：

(1)‘子图中的所有关系满足非自反、非对称、可传递；

(2)子图中所有的节点符合用户兴趣领域，也不存在语义重复的节点，即子图中 没有非必需的节点；

(3)子图中没有冗余的关系路径，即任意两节点间，有且仅有一条有效路径。

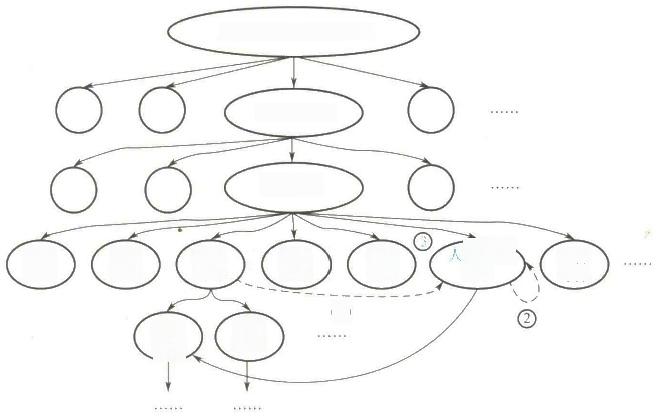
在图结构中，目标(1)可以抽象为图中不存在环路和指向自身的节点，节点间仅 存在单向关系。

目标(2)可以抽象为语义匹配问题，即将知识图谱G 中的节点与用户需求领域进 行语义相关度匹配，去除相关度为0和匹配程度极低的部分节点，同时去除语义重复 的节点。

目标(3)从层次的角度出发，可以将路径冗余问题抽象为：存在节点指向同层节 点或更高级别的概念节点，在剪枝任务中需要对此类关系进行处理。从语义的角度出 发，路径冗余有可能是由于某些概念分类不细致造成的，在剪枝任务中，保留分类更 细致的节点和相应的路径即可。

第6章 |数据质量管理

图6-34中展示了维基百科中工程、技术与应用科学领域下的一些分类节点，图中 清晰地按照层次展示了知识图谱剪枝过程中需要处理的三类节点。类别一代表冗余路 径，类别二代表知识图谱中的环路，类别三代表语义重复的概念节点对。



工程、技术与应用科学

计算机科学

人工智能

自动

驾驶

深度

学习

工神经

网络

知识 表示

机器 学习

聚类 分析

生物 计算

问答 系统

神经

网络

①

图6-34 包含三类需要处理的节点的示例图

在知识图谱剪枝任务中，将所有节点分为三种类型，对于不同的节点也有不同的 处理方式。

根节点 (root): 描述某一领域或兴趣大类的节点，在一个完整的百科类知识图谱 中，可以有多个根节点，代表多个领域，比如食物 (food) 、 设 备 (equipment) 、 动物 (animal) 等。从层次的角度考虑，根节点所处的层次永远是第0层。在不同的剪枝任 务中，根节点的选择可以是不同的。在一个剪枝问题中的根节点，在另一个剪枝问题 中可能是一个类型节点；

类型节点 (category): 描述根节点所代表分类的子类的概念性节点，仍然可以进 行分类，如“动物类”属于“生物类”,可以继续分类为“哺乳动物”和“两栖类”。 从层次的角度考虑，类型节点所处的层次可以是从1开始的任意一个层次；

实例节点 (instance): 描述某一类型中不可再进行分类的实例节点，如“波斯 猫”,从树结构的角度来考虑，实例节点即叶节点。

本节将从目前已有的剪枝算法在方向上的分类角度，即自顶向下和自底向上，介 绍目前较为经典的几类知识图谱剪枝算法。

图结构的深度优先搜索： 通过手动选择感兴趣的类别和术语节点，如果一个节点 出发有多条路径到达另一个节点，则利用公式计算这两个节点间每条路径的权重，在 剪枝环节保留权重最高的路径，每个节点的每条路径都要计算权重，计算复杂性较高。

应用于互联网搜索中的简单剪枝：起始节点和目标节点间可能存在多条路径，通

过 比 较 包 含 重 复 概 念 的 概 念 对 (pair), 保 留 分 类 更 为 细 致 的 概 念 对 ， 该 方 法 的 缺 点 是

每组包含重复概念的概念对都要进行比较，不适用于规模较大的知识图谱。

大数据治理： 理论与方法……………………

结合语义的剪枝：在剪枝过程中，仅保留语义与需求高度相关的类别节点，通过 计算和比较能够到达的最终节点数目，在前进路径上选择节点。这种方式尽管加入了 对语义的考虑，但是剪枝过于暴力，容易造成大量语义信息的丢失。

自底向上的剪枝：此类算法体现了知识图谱的层次结构，加入了对环路的处理， 但处理方式仅限于直接减掉环路中的一条边，同时文中未体现语义信息的重要性，在 剪枝过程中并未结合语义信息进行处理。

然后介绍基于用户需求结合语义的知识图谱概念层次剪枝算法的详细流程。算法 的输入为带有噪声的、具有非自反的、非对称的、可传递的关系的知识图谱G, 输出为 子图G, 。 算法首先将全部节点的层次数赋值为-1,并将根节点加入第0层，层次数赋 值为0,根节点是最高层级的概念节点。接下来，对于第1层中的所有节点，将其层次 值赋值为1,并根据用户兴趣语义相关性或搜索频度对当前层次节点由高到低进行排 序，存入候选列表R, 对当前层次中实体名称进行语义判断，若存在冗余，则删除实体 及相关关系。完成本层处理后，按照列表中的实体顺序，依次将与其相连的下位词加 入新的候选列表。对于第2层及其之后的层次，除进行语义判断和实体排序外，还需 要对层次信息进行判断。可能出现的层次信息情况有以下三种：

(1)赋值前节点层次信息为-1,说明此节点在此前并未被访问过，因此可直接赋 值为当前层次值。

(2)赋值前节点已有层次信息，并且与当前候选列表所在层次数相同，此时需对 下一节点进行判断，如果指向的下一节点是自身，则说明存在自身环路，去掉自身到 自身的关系即可，否则，说明此节点有多个上位词，由于上位词已按语义相关度进行 排序，因此，去掉当前节点与上位词之间的关系，保留与语义相关度较高的节点之间 的关系。

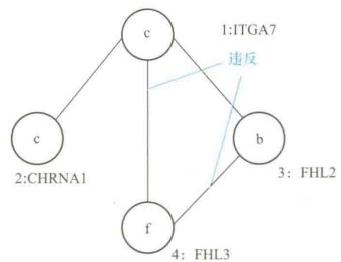
(3)赋值前节点已有层次信息，并且层次数小于当前候选列表所在层次，说明在 此前的某一层次，该节点已经被访问并加入过候选列表，在本算法中，这类情况可能 是由于分类不细致造成的。因此，去除节点与此前上位词之间的关系，并将节点层次 赋值为当前层次数。接下来，重复语义判断、实体排序、层次信息判断等过程，直至 最后一个叶节点完成层次赋值。如果此时节点集合V中还有未遍历过的节点，则证明 有非根节点出发的节点，或该节点不与任何一个领域相关节点相连，去除此节点即可。 最后，如果叶节点包含属性信息，则对属性信息进行领域相关度处理。

**2.** **图数据清洗**

现在，数据清洗已经引起了越来越多数据工作者的重视，针对传统的关系数据清 洗研究已具有一定的规模，但是在图数据领域，相关的研究还非常有限。随着图数据 应用的扩大，针对图数据清洗技术的研究也逐渐显现出它的价值。本文对该领域进行 研究，不仅具有实际意义，且具有重要的学术价值。

由于数据质量问题的存在会给大数据的相关工作带来致命的影响，因此数据清洗 在数据工作领域具有重要的地位。近些年来，关于数据质量和数据清洗的研究不断涌 现，但对于图数据清洗的研究十分有限。本节首先简单阐述传统数据清洗和图数据清 洗的研究现状。

目前，传统的数据清洗研究已较为完善，本节将围绕数据清洗的定义，原理，条

**第6章**|数据质量管理

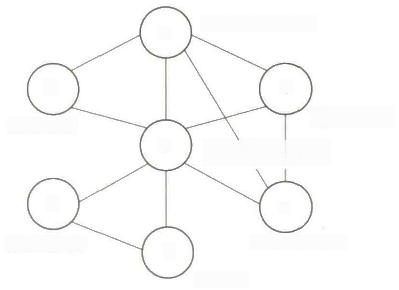
件，方法等进行阐述。当前对于数据清洗尚没有公认的定义，在不同的领域，对于数 据清洗的定义也有所不同。 一般来说，只要是有助于解决上述数据质量问题的处理过 程，都可以被认为是数据清洗。数据清洗的基本原理，是通过分析“脏数据”的产生 原因，利用一些技术手段和方法来检测数据中的“脏数据”,并将其转化为满足数据质 量要求或者后续处理要求的数据。数据清洗过程需要满足的条件：无论是单数据源还 是多数据源，都要检测并去除数据中所有明显的错误和不一致；尽可能地减小人工干 预和用户的编程工作量，而且要容易扩展到其他数据源；应该和数据转化相结合；要 有相应的描述语言来指定数据转化和数据清洗操作，所有这些操作应该在统一的框架 下完成。数据清洗的方法主要有以下几类：统计学方法、基于聚类的方法、基于关联 规则的方法、基于距离的方法和基于分类的方法，这些方法具有不同的特点，能够在 不同领域发挥其作用。

目前，图数据清洗的研究主要建立在属性图模型的基础上，清洗的方法是在邻域 约束条件下修复顶点标签。以下将具体介绍这些内容。

1)基于邻域约束的图数据清洗

许多类型的数据都可以建模为用数据值作为顶点标签的图，如相似性网络、工作 流网络以及蛋白质网络等 。 而作为顶点标签的数据值通常由于各种原因(如录入错字 或错误的实验结果)而变成脏数据。使用邻域约束可以指定允许出现在图中的相邻顶 点上的标签对，以此检测和修复错误的顶点标签。本节将具体阐述实例图、约束图、 成本等概念，明确基于邻域约束的图数据清洗问题及解决方法。

如图6-35所示，实例图表示为G(V,E), 即含有脏数据标签的待清洗的原始图数 据(或经过预处理的图数据)。其中G 表示实例图本身， V 表示实例图的顶点集合， E 表示实例图的边集合。对于 V中的每个顶点，都对应一个标签，不同顶点对应的标签 可能相同或不同。这些顶点的标签中可能存在错误，需要进行检测并修改。



Plasma membrane

c

d

Cytoskeleton

a

Celluter component

e

Mitochondrion

Nucleus

g

Extracellular

b

Cytoplasm

f

(a) 约束图 (b) 实例图

图6-35 数据修复示例

约束图表示为S(L,N), 即表示顶点标签的邻域约束的图。其中S 表示约束图本 身 ，L 表示约束图的顶点集合，也就是实例图顶点所对应的标签集合， N 表示约束图中 的边集合，也就是指定L 中标签的成对的邻域约束，若两个顶点(标签)之间有一条



大数据治理： 理论与方法……………

……

边，则表示这两个标签之间存在约束关系。约束图是根据待清洗图数据的特征，由领 域专家拟定或从原始数据中总结而来，用于对实例图中的脏数据进行检测和重新标记。

标签映射或称为标签函数，表示为λ:V→L, 即从实例图的顶点集V到约束图的标 签集L 的映射关系，它确定了实例图与约束图中顶点标签的对应关系。实例图中不同 的顶点可能映射到相同的标签，从而使得约束图的规模通常远小于实例图的规模。

成本函数即重新标记的成本，表示为δ(λ(v),λ'(v)), 它表示用λ'(v) 代 替

λ(v) 来重新标记顶点v 需要花费的成本。重新标记成本一般由原始数据与重标数据 的差异来进行评估，可以是字符串距离函数或者简单的修复计数。将其作为代价指 标，在清洗过程中作为分支选择的考量因素。另外，针对整个图的重新标记成本， 表示为△(G,G'), 它表示将实例图G 经重新标记变为G'后所需要花费的总成本，由 下式计算：

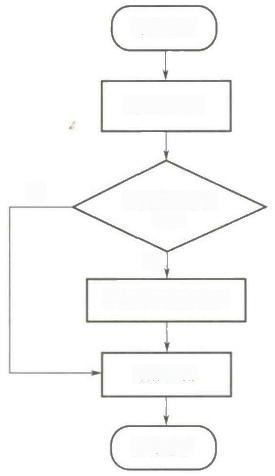


基于上述概念，以下具体阐述了基于邻域约束的图数据清洗规则：若标签1,l₂ 相 同 (l₁=l₂), 或者1₁,1₂在约束图中相邻((1₁,I₂)∈N), 则称标签l₁ 和1₂ 匹配约束图S。

若在实例图G(V,E) 中，对于V(u,v)∈E, 都 有 ((u),λ(v)) 匹配约束图S(L,N),

则称实例图 G 满足约束图S 。违反在实例图G(V,E) 中，若存在(u,u)eE, 但(λ(u),

λ(v)) 不匹配约束图S(L,N), 则称边(u,v) 是约束图S 的一个违反，顶点的违反数指 与它相邻的存在违反的边数。综上，基于邻域约束的图数据清洗问题可以定义为：对 于实例图 G 和约束图S, 通过检测并消除 G 中所有的违反，得到重新标记图C', 使得 G'满足S, 并且使成本△(G,G') 最小化。但是，确定是否存在G'且使成本小于常数的 问题是NP 完成的，所以我们试图寻求该问题的近似解。



原始数据

数据预处理

检测是否存在 违反

是

执行算法进行清洗

数据回流

目标数据

否

图6-36 近似修复流程

2)近似修复

下面将具体阐述解决邻域约束下的图数据清洗问 题的算法思想，包括贪心算法、收缩算法和中心收缩 算法，在此之前，我们要明确解决该问题的整体思想。 重新标记算法的执行主要分为两个部分，检测违反和 修复顶点标签，这两个部分交替重复执行，直到实例 图满足约束图为止。近似修复流程如图6-36所示。

贪心算法是基于贪心策略所提出的简单易行的算 法。由上述程序流程可知，图数据清洗的关键步骤有 两个： 一是对实例数据进行检测，找出其中存在的违 反；二是采取一定的策略修改数据中部分顶点标签。 重复这两个步骤直到消除所有违反为止。而贪心算法 的思想是在每次检测完成后，选择能够消除违反数目 最多的重新标记方案进行处理。由于贪心算法每次处 理只修改一个顶点标签，并且会选择当前情况具有最 高修改价值的顶点进行重新标记，因此具有较高的执 行效率和较好的清洗效果。但不幸的是，贪心算法可 能会出现无法终止的情况。因为贪心算法每一步处理

 138

**第6章**|数据质量管理

只考虑当前情况下最优的解决方案，而不会顾及对全局的影响，实际执行中当修改某 一顶点的标签时，可能会在图中引入新的违反，甚至使得总的违反数目不减反增，所 以在部分情况下会出现消除违反数为零或负值的情况，或者对某几个顶点进行循环往 复地处理，致使程序无法终止(实验证明绝大多数情况都会出现无法终止的情况)。

收缩算法主要为了解决贪心算法中无法终结的问题，其思想如下：将所有顶点划 分为不同的节点 (node), 保证每个节点内部的顶点之间不存在违反的边，之后检测存 在违反最多的两个节点R, 和 R₂, 把其中修改成本较低的节点R, 收缩到节点R₂ 中，即 将R, 中所有顶点重新标记为统一的标签以消除违反，再将R, 和 R₂ 两个节点合并，重 复该步骤直到实例图中不存在违反(满足约束图S) 为止。由于在收缩的过程中节点 数会严格减少，因此可以保证程序能够终止。要执行收缩算法，首先需要定义以下几 个符号：节点表示为R, 它表示一组收缩到R 的顶点；每个节点有一个主顶点，表示 为h(R); 还有一组客节点的集合，表示为 U(R), 其中每个客体节点内所有顶点的标 签都是一致的；另外，用V(R) 表示节点内所有的顶点集合。收缩算法的执行过程主要 分为两个部分，节点初始化和节点收缩。最初，图中的每个顶点v 将单独形成一个节点 R(v), 其主顶点h(R)=v, 客节点集合U(R)=9, 而所有顶点集合V(R)={v} 。 之后 在两个存在违反的节点间进行收缩，将其中一个节点内的所有顶点重新标记，合并入 另一个节点中。在收缩算法的执行过程中，需解决以下几个问题：收缩节点的选择基 于贪心策略考虑，每次处理选择存在违反数目最多的一对节点进行收缩。

收缩算法伪代码如图6-37所示。

输入：实例图G 与约束图S

输出：满足S 的重新标记图G

1.for G中的每一个顶点vdo

2. 创建新节点 R

*3.h(R):=v*

4.U(R):=Ø

5.while G不满足Sdo

6.R₁,R₂:= 具有最多违反项的两个节点

7.1₁,l₂:=R₁,R ₂的最佳候选标签

8.if cost(R₂)>cost(R₁)then

9.swap R,R₂

.for 的每个顶点vdo

12.V(R₁):=V(R₂)UV(R)

13.U(R):=U(R₂)U{R₂}

14.retun G

图6-37 收缩算法伪代码

相比于贪心算法，收缩算法的突出优势是能够保证终止，随着收缩的进行，节点 的数目会逐渐减少，最终必然能够满足约束图。但是，收缩算法也有一定的缺陷，由 于算法较为复杂，其效率也会较低；另一方面，由于贪心策略的局部性，收缩的成本 可能会难以控制。

虽然贪心算法可能不会终止，但却拥有较高的效率和良好的成本控制，另一方面， 收缩算法虽然在成本控制和效率上显现劣势，却能够保证终止。因此，可以将这两种 算法结合起来，优势互补，以达到更高的执行效率和清洗效果。

混 合 算 法 伪 代 码 如 图 6 - 3 8 所 示 。

大数据治理： 理论与方法…………………… ……………

输入：实例图G 与约束图S

输出：满足S 的重新标记图G

1.while G 不满足Sdo

2.v:= 具有最大违反项数目的顶点

3.λ'(v): = 能够消除最多违反项的v的候选标签

4.i f 消除违反项数≤0 then

5. break;

6. else

7. 将顶点v 的标签λ(v)重新标记为λ(v)

8.对 G 执行收缩算法

9.return G

图6-38 混合算法伪代码

通过结合贪心算法和收缩算法的优势，使得混合算法能够兼具二者的特性，在一定 程度上提高了效率，也使得成本被控制的更低，并且能够保证程序的终止，改善了贪心 和收缩算法的缺陷。因此混合算法在执行效率和清洗效果上都优于单一算法的执行。

中心收缩算法基于贪心策略，每次处理选择存在违反数目最多的顶点进行重新标记。 与上文所述的贪心算法不同的是，中心收缩算法每次处理都会对以目标顶点为中心的星 形结构进行处理，统一重新标记该结构内的部分节点，以消除该星形结构上存在的所有 违反，而并非只对单一的目标顶点进行重新标记。该算法在一次处理中能够重新标记多 个顶点，因此能够更高效地清洗数据。此外，由于该算法基于贪心策略，每次处理后还 可能会在所处理的星型结构以外引入违反，因此也不能保证终止，但是该算法能够在无 法继续执行之前处理更多的违反，从而能在混合算法中体现更大的价值，

同贪心算法一样，中心收缩算法也基于贪心策略，因此需要考虑贪心函数的选择。 首先，需要选择存在违反最多的节点，生成贪心函数。选出目标顶点v。之后要对以顶 点v 为中心的星形结构进行重新标记处理，即对该结构中的一个或多个顶点进行重新标 记，以使得该行星形结构内不再存在违反。在阐述策略之前，需要给出以下定义

N(v) 表示与顶点v 相邻的一组顶点，这些顶点不需要重新标记；

C(v) 表示星形结构内的部分顶点，这些顶点需要被重新标记。

以下阐述算法细节：每次处理选出目标顶点v 后，对N(v) 和 C(v) 分别进行初始 化 ，N(v) 包含顶点v 的所有邻接顶点，C(v) 只包含顶点v 自身。之后找到对C(v) 中的 顶点进行重新标记的候选标签组L(v), 其中的标签应使得星形结构中不再存有违反。

中心收缩算法伪代码如图6-39所示。

|  |
| --- |
| 输入：实例图G与约束图S  输出：满足S的重新标记图G  1.while G不满足Sdo  2.v:=具有最大违反项数目的顶点  3.N(v):=v的所有邻接顶点  4.C(v):={v}  5.L(v):=所有使得以v为中心的星形结构满足S的候选标签  6.while L(v)为Odo  7. u:=N(v)中度最大的顶点  8. 将u从N(v)中删除并加入C(v)  9. 更新L(v)  10.L:=L(v)中成本最小的标签  11.for N(v)中每个顶点u  12. 将顶点u的标签λ(u)重新标记为/  13.return G |

图6-39 中心收缩算法伪代码

140

第6章**|数据质量管理**

由于贪心算法每次处理仅重新标记单一顶点，其决策所考虑的范围较小，所以对 全局的把握不够准确，从而容易在局部陷入死循环；而中心收缩算法每次处理一组顶 点，决策时考虑的范围更大，在一定程度上能够缓解贪心策略的缺陷，也容易更高效 地取得结果。

**数据质量标准**

由于数据质量管理涉及多个纵向环节和横向领域，在这些过程中，各环节、各领 域的参与者众多，应当以什么样的标准来要求、评估、保证数据质量是需要慎重考虑 的。如果能够有规范统一的数据质量标准，那么各环节、各领域乃至各组织、各国家 之间就可以遵循标准，进行更高效、全面的数据质量管理。同时也能大幅削减后续的 数据集成、共享、分析等方面遇到的困难和问题，减少因为数据质量问题而造成的决 策失误。为此，国内外的标准化组织进行了大量工作。

我国目前已经发布了一系列与具体应用领域相关的数据质量标准，包括《车载导 航电子地图数据质量规范》 (GB/T 28441—2012)、《CAD/CAM 数据质量保证方法》 (GB/T18784.2—2005)、《机载激光雷达点云数据质量评价指标及计算方法》 (GB/T 36100—2018)等。同时，也有一些相对较为通用的数据质量相关的规范，如《工业数 据质量通用技术规范》 (GB/T39400—2020) 和《信息技术数据质量评价指标》 (GB/T36344—2018) 等。关于数据质量的国际标准有ISO 8000。以下我们将对其中几 个较为重要的标准做简要介绍。

ISO 8000由 ISO工业自动化系统与集成技术委员会 (TC184)SC4 小组委员会制 定，是数据质量和企业主数据的国际标准。该标准描述了数据质量原理，定义了数据质 量特征，并且给出了数据质量的认证过程。TC184/SC4 是负责工业数据的国际标准组 织，是国际标准化组织 (ISO) 的184技术委员会的第4小组委员会，负责制定并维护 ISO 标准，用于描述和管理整个产品生命周期内的工业产品数据。SC4标准广泛用于车 辆、飞机、船舶、制造工厂、建筑物及所有相关设备和零部件的设计、工程、制造和 管理数据，关注数据的可移植性和长期的数据保存。

ISO 8000 是新兴技术标准之一，一些组织正在转向采用这些标准来改进业务流程 和控制运营成本。该标准正在以多个独立文件的形式发布，ISO 称为“部分”。ISO 8000标准于2002年首次提出，第一部分于2009年获得批准，描述“质量标识符”的 “质量标识符前缀”的第115部分于2017年获得批准。目前正在开发几个重要的增强 功能，包括116部分。各部分关注的重点不同，可以分为如下4组。

(1)通用数据质量，第0~99部分。

(2)主数据质量，第100~199部分。

(3)业务数据质量，第200~299部分。

(4)产品数据质量，第300~399部分。

目前 ISO 8000 已经发布的部分如表6-6所示。

**大数据治理：** 理论与方法…………………

**表6-6** **ISO 8000** **已经发布的部分**

|  |  |  |
| --- | --- | --- |
| 序号 | 编 号 | 标 题 |
|  | ISO/TS 8000-1:2011 | Data quality —Part 1:Overview |
| 2 | ISO 8000-2:2017 | Data quality —Part 2:Vocabulary |
| 3 | ISO 8000-8:2015 | Data quality —Part 8:Information and data quality:Concepts and measuring |
| 4 | ISO 8000-61:2016 | Data quality —Part 61:Data quality management:Process reference model |
| 5 | ISO 8000-63:2019 | Data quality —Part 63:Data quality management:Process measure-  ment |
| 6 | ISO 8000-100:2016 | Data quality —Part 100:Master data:Exchange of characteristic  data:Overview |
| 7 | ISO 8000-102:2009 | Data quality —Part 102:Master data:Exchange of characteristic data;Vocabulary(Withdrawn) |
| 8 | ISO 8000-110:2009 | Data quality —Part 110:Master data:Exchange of characteristic data:Syntax,semantic encoding,and conformance to data specification |
| 9 | ISO 8000-115:2017 | Data quality —Part 115:Master data:Exchange of quality identi- fiers:Syntactic,semantic and resolution requirements |
| 10 | ISO 8000-120:2016 | Data quality —Part 120:Master data:Exchange of characteristic data:Provenance |
| 11 | ISO 8000-130:2016 | Data quality —Part 130:Master data:Exchange of characteristic data:Accuracy |
| 12 | ISO 8000-140:2016 | Data quality —Part 140:Master data:Exchange of characteristic data:Completeness |
| 13 | ISO/TS 8000-150:2011 | Data quality —Part 150:Master data:Quality management frame  work |
| 14 | ISO/TS 8000-311:2012 | Data quality —Part 311:Guidance for the application of product da- ta quality for shape(PDQ-S) |

ISO 8000 是一套相当全面的数据质量标准，包括了数据质量术语、数据质量概念、 数据质量活动等。基于ISO 8000数据质量标准，数据可以被独立的交易而不受限于特 定的软件或者服务。目前已经发布的部分中，其中相当大的篇幅是关于主数据的。关 于主数据的概念我们已经在第4章有所介绍，感兴趣的读者可以阅读相关章节获取所 需要的信息。

**6.7.2** **《信息技术数据质量评价指标》**

国家标准《信息技术数据质量评价指标》 (GB/T36344—2018) 是国家标准，由 全国信息技术标准化技术委员会 (TC28) 归口上报及执行，主管部门为国家标准化管 理委员会。

 142

**第6章** | **数** **据质量管理**

该标准于2018年6月发布，2019年1月实施，对数据质量的评价指标进行了规范 和说明，基本覆盖了数据生命周期的全部环节。主要部分如表6-7所示。

**表6** **-** **7** **《信息技术数据质量评价指标》主要部分**

|  |  |  |
| --- | --- | --- |
| 序号 | 标 题 | 主 要 内 容 |
| 1 | 范围 | 内容简述及适用范围 |
| 2 | 术语和定义 | 本标准的术语和定义列表 |
| 3 | 指标框架 | 定义了数据质量评价指标的整体框架及包括的数据质量维度 |
| 4 | 概述 | 标准概述 |
| 5 | 指标说明 | 数据质量评价主要维度说明，包括规范性、完整性、准确性、 一致性、时效 性、可访问性 |
| 6 | 附录 | 数据质量评价过程 |

该标准中详细规范了规范性、完整性、准确性、 一致性、时效性、可访问性这几 个 数 据 质 量 评 价 的 主 要 维 度 ， 通 过 一 级 指 标 和 二 级 指 标 编 码 并 组 织 了 各 维 度 和 维 度 内

的主要要素。各级指标及编码如图6-40所示。

|  |  |  |
| --- | --- | --- |
|  |  | 0101 数据标准  0102数据模型 |
|  | 规范性 | 0103元数据  0104业务规则  0105权威参考数据  0106安全规范 |
|  | 完整性 | 0201 数据元素完整性 0202数据记录完整性  0301 数据内容正确性  0302数据格式合规性 |
|  | 准确性 | 0303 数据重复率 |
| 数据质量评价 |  | 0304数据唯一性  0305脏数据出现率 |
|  | 一致性 | 0401 相同数据一致性 0402关联数据一致性  0501基于时间段的正确性 |
|  | 时效性 | 0502 基于时间点的一致性 0503 时序性 |

可访问性 0601可访问

0602可用性

图6-40 《信息技术数据质量评价指标》规范是数据质量评价维度



大数据治理： 理论与方法

……**…**

………

该规范中的维度可以认为是数据质量评价的最小集合，在具体的应用场景中，可 能需要丰富上述维度来达成更准确的评价。

**6.7.3** **《工业数据质量通用技术规范》**

《工业数据质量通用技术规范》(GB/T39400—2020) 是国家标准，由全国自动化 系统与集成标准化技术委员会 (TC159) 归口上报，全国自动化系统与集成标准化技术 委员会工业数据分会 (TC159SC4) 执行，主管部门为中国机械工业联合会。

该标准关注工业数据质量持续改进模型、质量描述、识别、评价、控制、报告等 方面，适用于工业数据采集、传输、维护和使用过程中的质量管理。该标准涉及的工 业数据涵盖了主数据、事务数据和产品数据。

该标准于2020年11月发布，2021年6月实施，是目前我国较新的数据质量标准 之一。主要部分如表6-8所示。

**表6-8** **《工业数据质量通用技术规范》主要部分**

|  |  |  |
| --- | --- | --- |
| **序号** | **标** **题** | **主** **要** **内** **容** |
| 1 | 范围 | 内容简述及适用范围 |
| 2 | 规范性引用文件 | 必不可少的引用文件 |
| 3 | 术语和定义 | 本标准的术语和定义列表 |
| 4 | 工业数据质量持续改进 | 持续改进模型的定义及主要阶段 |
| 5 | 工业数据质量描述 | 数据质量的描述框架及要素 |
| 6 | 工业数据质量识别 | 定量及非定量的数据质量信息 |
| 7 | 工业数据质量评价 | 数据质量评价方法和步骤 |
| 8 | 工业数据质量控制 | 数据质量控制规则和控制方法 |
| 9 | 报告数据质量信息 | 数据质量报告规范要求 |

144



**第** **7** **章** **数据标准化**

长久以来，大多数系统都是在某些特定业务需求的基础上建立的。这些系统各自 为政，并不考虑与其他系统在功能和数据上有多大的重复性。然而， 一旦存在数据集 成、交换、共享等需求，由于缺乏对数据整体的设计考虑，就会造成多种问题，例如：

(1)数据需求缺乏规范，造成数据对象多份存储，存储结构各异，严重影响数据 共享；

(2)数据定义不一致，造成同样表示的数据语义不同；

(3)数据接口各异，造成数据传递和转换困难。

这些问题往往贯穿了整个数据管理和使用过程，要减少这些问题，需要从数据定 义、功能设计、代码实现、业务管理等多个方面共同入手。

在大数据时代，数据的交换、传递、共享非常重要，可以说是很多应用成功的关 键。数据可以在多个级别上共享，在最低级，多个记录使用相同的数据项。在中级， 多个应用使用相同的文件或数据项组合，多个数据项构成数据库， 一个系统可以包含 许多单独的数据库。在最高级，每个完整的数据库可以用于许多部门，同时也可以供 其他行业乃至全社会使用。应用数据库技术组织数据是实现数据共享的唯一途径。数 据库可以使数据与使用它们的各个应用程序相互独立，互不依赖。无论程序改变还是 数据改变都不引起另一方的改变，所以能够很容易重新组织数据，加入新的数据。同 样，在数据结构、数据内容或数据存储的物理介质发生改变时，都不需要重写程序。

为了能够更好地建立良好的大数据共享与开放环境，数据标准化势在必行。数据 标准化能够使各个应用系统对客观实体的分类和描述手段一致，或者提供相应的转换

大数据治理：理论与方法……………………

接口。在理解一致的前提下，应用标准数据编码系统和统一的逻辑描述方式，使共享 数据库成为可能。

数据标准化是对数据的定义、组织、监督和保护进行标准化的过程。数据标准化 与之前章节介绍过的许多内容相关，包括数据交换、数据质量和数据说明文件等内容， 涉及组织架构、工作流程、数据标准等多个领域的工作。数据标准化是数据共享和系 统集成的重要前提，数据标准化可以节省新系统的开发费用，提高开发效率，拓展应 用领域，有利于系统推广应用，实现数据共享，减少数据采集费用。

本章我们将概述数据标准化并介绍其方法和实例。

7.1 数据标准化概述

**7.1.1** **数据标准**

根据中国信息通信研究院的定义， 数据标准( Data Standard) 是保障数据内外部使 用一致、准确的规范和约束。直观来说，数据标准可以理解为若干描述规范和要求的

文档。更严谨地，数据标准是为了使组织内部和外部使用交换的数据保持准确且一致， 经协商一致制定的并由相关主管机构批准的，可共同使用和重复使用的一种规范性文 件。更一般地，数据标准不仅是一套规范，还是由管理规范、管控流程、技术工具共 同组成的体系，通过这套体系，能够逐步实现信息标准化。

此外，需要说明的是，对于企业或组织而言，数据标准不仅仅是技术或者业务一 个部门的事。企业数据标准化的建立是企业信息化的基础工作，既能提高数据的共享 性，又能为企业提供统一的信息视图、数据规范及相应的编码标准。它是数据层面上 对重要业务主题的统一规范，也是业务规范在数据层面上的实现。数据标准实施依赖 于业务部门之间的共识，以及业务和技术之间的配合。

根据不同的数据分类，数据标准也可以分为不同类别。数据标准的分类和数据的 分类是对应的，下面给出几种典型的定义和划分方法。

**1.** **根据数据类型**

一种典型的方式是将数据划分为基础类和指标类，并进一步定义和规范相应的数 据标准。

所谓基础类数据，是指企业或组织在工作和生产的过程中产生的、可直接使用的、 未经统计分析的基本数据，如商品价格、商品类型、商品编号等。基础类数据标准主 要是为了保证技术类数据使用的一致性，其约束了业务活动产生数据的类型、值域、 标识符等规范。

所谓指标类数据，则是经过统计分析，具备统计意义和指标作用的数据，如均值、 增幅、中位数等。指标类标准又可以进一步分为基础指标标准和计算指标标准，分别 指在基础数据上直接计算得到的指标和由不同指标二次合成得到的指标。

**2.** **根据数据处理和使用的阶段**

根据数据处理和使用的不同阶段，也可以定义不同的数据标准。例如，可以将数 据标准定义和划分为数据处理标准、数据安全标准、数据质量标准、产品和平台标准、

**第7章** |数据标准化

应用和服务标准等不同类别。

数据处理标准包括针对数据整理、数据分析、数据访问的标准。

数据安全标准包括通用标准和隐私保护标准。

数据质量标准包括元数据质量、质量评价、数据溯源等方面的标准。

产品和平台标准包括关系型数据库产品、非结构化数据管理产品、可视化工具、 数据处理平台方面的标准。

应用和服务标准包括开放数据集、数据服务平台等方面的标准。

**3.** **根据数据标准涉及的实体**

根据涉及的实体，数据标准可以定义数据模型和数据规范。其中数据模型包括数 据模型本身和业务规则的定义，数据规范则包括数据建模规范、数据编码规范、数据 交换规范、数据 ETL规范、 ·数据集成规范等。

**7.1.2** **数据标准管理**

数据标准管理是指研究、制定、应用和维护数据标准的过程和活动①。基于这个定 义，可以大致分为4个主要阶段来讨论数据标准管理。

**1.标准研究**

标准研究阶段主要的目的是回答两个问题，即“是否需要标准”和“需要何种标 准”。工作内容主要包括标准规划和标准需求管理。

标准规划包括标准调研、案例收集、业务分析、制订计划等活动。其中标准调研 和案例收集的主要目的是调研现有标准并收集相关领域的最佳实践，业务分析和制订 计划则根据本组织或企业，了解与标准相关的内容，包括现有定义、使用习惯、数据 分布、数据流向、业务规则、服务部门等，确定所需标准的大致范围和框架。

标准需求管理则指定专门的人员组织、收集、精简具体的标准，并根据规范流程 和语言来描述标准需求。

**2.标准制定**

在标准制定阶段，依据前期的调研和需求分析结果，参考国际及国家标准化组织 的相关标准，完成标准的细节定义。具体包括：统一的数据定义标准和编码标准；清 晰定义每个数据元素的名称、内容、类别、关联关系、用途、源系统等属性；保证可 行性、可扩展性、可共享性、稳定性等数据标准应当具备的基本性质。

**3.** **标准应用**

标准应用阶段的主要目的是将制定好的标准实施到各个部门和系统中。对于新开 发的系统，要遵循数据标准中给出的数据编码规范、数据接口规范、数据建模规范、 数据交换规范等；对于已经存在的系统，如果其不符合数据标准，则制订相应的计划， 分阶段、分批次地完成修正，直至满足标准。

**4.标准维护**

数据标准也不是一成不变的，随着业务发展，有些标准需要不断修订和完善。因

① 数据归一化 (Data Normalization) 在一些场景下也称为数据标准化，读者需要注意区分。

大数据治理： 理论与方法……………

此数据标准还有一个关键的管理环节，那就是持续维护改进。在标准维护阶段，需要 有相应的需求收集、需求评审、变更评审、发布等多个步骤，并能对所有的修订做版 本管理，以方便将来进行问题查找。

目前标准管理也得到了国内外标准制定组织的重视，如在国际标准化组织 (ISO) 中，我国大数据标准委员会代表中国在工业和信息化部的指导下也一直参与相关的 工作。

**7.1.3** **企业数据标准化**

企业数据标准化是一个渐进的过程，需要在组织架构、工作流程、数据标准等层 面同时展开。

在组织架构方面，企业需要由高层领导负责，并建立专门的工作小组，负责标 准管理的不同阶段。需要由高层领导负责是因为数据标准化工作是跨部门、跨功能 的，需要有足够的上层支撑才可能顺利实施。同时，工作小组的分工合理性和技能 完备性也非常重要，工作小组可以由小规模、经验丰富、了解业务的多个领域的专 家组成。

在工作流程方面，需要保证贴近业务和循序渐进。目前企业信息化过程中数据标 准化建设有两种：全面标准化和渐进式标准化。

(1)全面标准化首先实施独立的、全面的数据标准化项目，然后各子系统基于已 经稳定的标准来进行建设。

(2)渐进式标准化则从框架开始，结合系统更新、重组、新建的工作阶段，确定 合适的标准化范围，有选择地逐步进行标准化。 一般情况下，在遗留系统较多或较重 要的场景下，数据标准化应采取渐进式标准化，防止速度过快导致重大问题。

在数据标准方面，针对特定企业，数据标准管理应该根据具体业务有所调整。在 通用的标准管理流程框架中，根据业务定制适合的管理活动和管理方式。在具体的标 准定义方面，企业至少应当做到以下几方面的标准化。

(1)数据定义标准化：规范数据的语法和语义，防止数据冗余和歧义，保证数据 质量。

(2)数据模型标准化：数据模型需要规范定义，以保证能跨应用、跨业务地完整 表述企业数据的统一视图。

(3)基础设施管理标准化：统一规划使用数据的存储、通信等基础设施，提高资 源使用效率。

(4)其他标准化：包括生命周期管理、质量管理、服务管理等方面的标准化。

7 **.2** **数据标准化例析**

**7.2.1** **工业大数据**

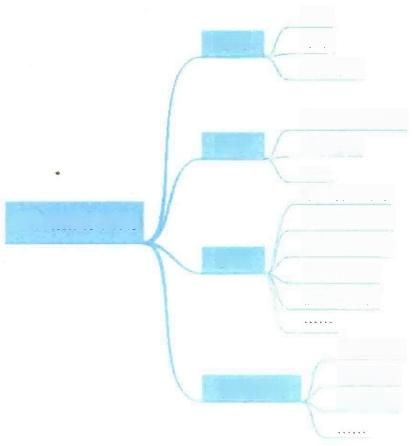
工业大数据是指在工业领域中，围绕典型智能制造模式，从客户需求到销售、订 单、计划、研发、设计、工艺、制造、采购、供应、库存、发货和交付、售后服务、

**第7章** |数据标准化

运维、报废或回收再制造等整个产品全生命周期各个环节所产生的各类数据及相关技

术和应用的总称①。

为了使工业大数据中所蕴含的价值得以挖掘和展现，业界专家也在尝试着搭建工 业大数据标准体系。工业大数据标准体系如图7 - 1所示，由4个类别的标准组成，分别 为基础、技术、管理、产品/应用。



总则

术语

参考架构

全生命周期处理 互联互通

………

数据管理体系 数据资源管理 数据质量管理 主数据管理

能力成熟度

平台/工具 应用服务 应用领域

工业大数据标准体系

产品/应用

基础

技术

管理

.

图7-1 工业大数据标准体系

在上述体系中，各部分标准各司其职，共同构建了工业大数据标准体系。各部分 标准的大致作用如下。

基础标准主要用于统一基础概念，解决基础共性关键问题，包括总则、术语和参 考架构等。

技术标准主要用于对工业大数据的分析处理涉及的关键技术进行规范，包括全生 命周期处理、互联互通等。

管理标准主要对工业大数据在应用过程中的数据管理方法、流程、机制进行规范， 包括数据管理体系、数据资源管理、数据质量管理、主数据管理、能力成熟度等。

产品/应用标准主要针对工业大数据涉及的产品、应用服务及垂直领域应用中涉及 的技术进行标准化规范，包括平台/工具、应用服务、应用领域等。

目前，工业大数据的国际标准化工作主要由ISO/IEC JTC 1展开，仍集中于传统的 工业控制领域所涉及的工业数据的标准化工作。我国工业大数据标准化工作主要依托 全国信息技术标准化技术委员会大数据标准工作组工业大数据专题组开展。目前《信 息技术工业大数据术语》(20180988-T-469)、《 信息技术大数据工业应用参考架构》 (20173819-T-469)、《 信息技术大数据产品要素基本要求》(20173820 -T-469)、《 智 能

① 中国电子技术标准化研究院，全国信息技术标准化技术委员会大数据标准工作组，工业大数据产业应用 联盟.工业大数据白皮书.2019.

**大数据治理：** 理论与方法…………………

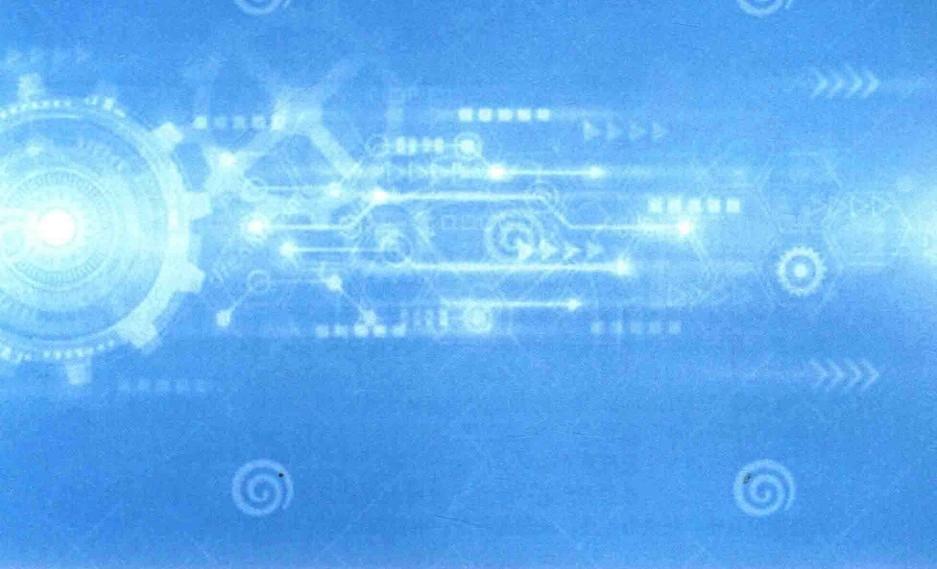
制造工业数据空间模型》(20182054-T-339)、《智能制造多模态数据融合系统技术要 求》(20182040-T-339)、《 智能制造工业大数据平台通用要求》 (20182053-T-339) 、 《智能制造工业大数据时间序列数据采集和存储框架》(20182052-T-339) 等7项国家 标准已完成立项。

**7.2.2** **信息技术元数据注册系统**

元数据注册系统是支持注册功能的元数据数据库，目标是实现标识、来源、质量 监控。合理、规范的元数据注册系统可以帮助各应用系统定位合适的元数据对象。为 了实现元数据注册系统的功能，标准化的框架、定义、标识、语义描述等必不可少。 为此，国家质量监督检验检疫总局、国家标准化管理委员会在2009年发布并实施了一 套信息技术元数据注册系统标准。标准的具体信息如表7-1所示，其类别归属于数据处 理下的数据整理，共分6个部分，分别是框架、分类、注册系统元模型与基本属性、 数据定义的形成、命名和标识原则、注册。

**表7-1** **信息技术元数据注册系统标准列表**

|  |  |  |  |
| --- | --- | --- | --- |
| **一级分类** | **二级分类** | **国家标准编号** | **标** **准** **名** **称** |
| 数据处理 | 数据整理 | GB/T 18391.1—2009 | 信息技术元数据注册系统第1部分：框架 |
| GB/T 18391.2—2009 | 信息技术元数据注册系统第2部分：分类 |
| GB/T 18391.3—2009 | 信息技术元数据注册系统第3部分：注册系统元模型 与基本属性 |
| GB/T 18391.4—2009 | 信息技术元数据注册系统第4部分：数据定义的形成 |
| GB/T 18391.5—2009 | 信息技术元数据注册系统第5部分：命名和标识原则 |
| GB/T 18391.6—2009 | 信息技术元数据注册系统第6部分：注册 |



**第** **8** **章** **数据资产化**

在大数据时代，随着各行各业的全面信息化，各类数据高速积累，各类数字、文字、 图像、声音、日志等类型的数据在不同的场景中发挥着重要作用，为人们的日常生活和 国民经济的发展提供了便利，产生了重要的价值。近些年，我国从国家战略的层面上肯 定了大数据技术发展的必要性，如何更好地管理和使用数据成为重要议题之一。

在当前时代，数据是否算得上是一种重要资产呢?所谓资产，是指“由企业拥有 或控制的、预期会给企业带来经济利益的资源”。当前，数据同样满足资产的定义，其 被企业或组织所拥有和控制，可以交易，并且产生重要的价值。例如，在2013年，美 国有统计指出，在健康护理领域，利用大数据每年产出约3000亿美元；2019年第二季 度，我国银行业金融机构共处理电子支付业务536.71亿笔，非银行支付机构处理网络 支付业务1777.77亿笔。可以看出，数据发挥的作用不可忽视。在经济业务活动中， 会计主体可通过生产、采集、加工、购买等方式拥有并控制数据资源，并且可以通过 出让数据、加工数据或提供数据服务来获取利益。

当前，数据的产生量，早就超过了可用存储量。并非所有的数据都可以被视为资 产，只有可控制、可计量、可变现的数据才可能成为资产。实现数据资产的可变现属 性，体现数据价值的过程，即称为“数据资产化”。如果不能良好管理，数据也可能变 成一种“负债”。如何管理数据资产，“盘活”数据以充分释放其附加价值非常重要。 数据资源是能给会计主体带来预期经济利益或产生服务潜力的资源。

本章将概述数据资产管理的相关概念及方法，讨论数据资产发现与评估、数据交易 与数据定价等议题，探讨怎样识别数据资产，以及怎样利用现有的数据资产创造价值。

**大数据治理：** 理论与方法 …

8**.1** **数据资产管理概述**

**8.1.1** **数据资产**

在《企业财务会计报告条例》①中，资产是这样定义的：“资产，是指过去的交 易、事项形成并由企业拥有或控制的资源，该资源预期会给企业带来经济利益。”参考 资产的定义，可以将数据资产定义如下②:

数据资产 (Data Asset) 是指由企业拥有或者控制的，能够为企业带来未来经济利 益的，以物理或电子的方式记录的数据资源，如文件资料、电子数据等。

上述定义指明了数据资产必须包括三个内涵。

拥有或者控制：数据资产不一定是企业在内部信息系统中拥有的数据资源，也可 以是通过合作、租赁等手段，从企业外部获取使用权的各种数据资源。

带来未来经济利益：这表明数据资产拥有直接或间接导致资金或现金等价物流入 企业的潜力。这种潜力是将数据作为一种经济资源纳入企业经济活动的能力。通过为 企业的管理控制和科学决策提供合理依据，减少和消除企业经济活动中的风险，进而 为企业间接带来预期的经济利益，也可以通过交易或事项，直接为企业带来经营收入。

数据资源：这指出了数据资产的具体形态，表现为以物理或电子方式记录的数据， 如工作记录、表单、配置文件、拓扑图、系统信息表、数据库数据、操作和统计数据， 开发过程中的源代码等。

如前文所述，数据资产的属性，主要包括以下三个方面：可控制、可计量和可

变现。

**1.** **可控制**

目前，数据所有权归属定义并不明确。人们每天使用各种应用，会产生大量数据， 这些数据中包含了大量的个人信息。服务提供商有意识地收集、整理、分析这些数据， 就取得了这些数据的控制权。

对于企业或组织来说，其可控制的数据包括多方面。 ”

用户数据：包括用户的基本信息(如用户ID 、用户类别等)和用户的行为数据 (如用户浏览记录、购买记录、评分记录等)。

生产数据：包括企业或组织在自身的业务活动中产生的数据，如产品参数、运输 数据、营销数据等。

分析数据：包括企业将前两种数据二次加工得到的数据，如统计信息、分析报 告等。

对于通过合法途径收集，并享有存储、使用、加工、共享权利的上述三种数据， 均可视为企业的可控制数据。

① 国务院公布，自2001 年1月1日起施行，以此为依据制定了《企业会计制度》。 ② 中国信息通信研究院，大数据技术标准推进委员会.数据资产管理白皮书.2019.

152

**第8章** |数据资产化

**2.** **可量化**

可量化的意义在于确定资产的体量和价值，这包括两方面。

首先，作为资产的数据需要被量化。并非所有数据都是资产，不同类型的数据的 计量方式也不相同，例如，3GB的音视频数据和3GB文本数据的体量并不总被认为是 相同的。因此，需要根据具体需求，选择合适的计量方式。

其次，作为资产的数据价值需要被量化。数据作为一种新型资产，其价值评估也 是多元化的。同一组数据在不同的应用场景中价值不同，其价值评估需要和业务场景 密切相关。

**3.** **可变现**

资产必须要能够为企业带来未来经济利益。因此，数据必须能够转化为经济价值 才可以被称为资产。不过，转化为经济价值并不总意味着直接租售，数据的变现方式 可能有很多种，我们举几个例子说明。

直接租售：即将原始数据直接出售或出租。

技术研发：许多智能化的技术(如深度学习的模型训练)只有在拥有足够多的数 据时才能实施研发，企业或组织可以通过此类技术研发掌握前沿技术，进而转化经济 效益。

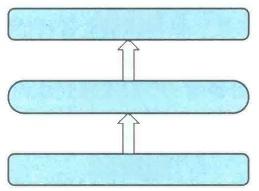
业务支持：大量的数据可以帮助企业或组织开展新的业务，如金融方面可开展基 于大数据的风险定价、股票预测、供应链金融等业务，新的业务可以直接带来经济 效益。

**8.1.2** **数据资产管理**

数据资产管理 (Data Asset Management) 是指规划、控制和提供数据及信息资产的 一组业务职能，包括开发、执行和监督有关数据的计划、政策、方案、项目、流程 方法和程序，从而控制、保护、交付和提高数据资产的价值。

企业及组织通过进行数据资产管理活动来保证数据资产的安全完整、合理配置和 有效利用，从而提高数据资产带来的经济效益，保障和促进各项事业发展。

一般而言，数据资产管理位于大数据平台和数据应用的中间层，如图8-1所示。我 们将从4个领域来介绍数据资产管理：制度建设、资产发现与评估、资产交易与定价、 资产运营和保护。



数据应用

数据资产治理

大数据平台

图8-1 数据资产治理的定位



**大数据治理：** 理论与方法

…………………………

**1.** **制度建设**

为了顺利实施数据资产管理，企业和组织首先需要建立一套完整有效的制度，这 包括组织架构建设、制度体系建设、审计机制建设等方面。

组织架构建设： 典型的组织架构主要由数据资产管理委员会、数据资产管理中心 和各业务部门构成，其中数据资产管理委员会负责制定决策；数据资产管理中心负责 牵头制定数据资产管理的政策、标准、规则、流程及相关的运营、组织、协调工作； 各业务部门则是数据资产的提供、开发和消费者。

制度体系建设：制度体系建设需要制定数据资产的管理规范。从广义上来说，我 们已经讨论过的元数据、主数据、数据质量、数据标准等方面的管理流程和规范都属 于数据资产的制度体系建设范畴。在执行规范和标准的过程需要重视事中检查和事后 监控。事中检查指的是在开发和上线时进行控制，包括检查命名规范、信息完整性、 合理性等；事后监控指的是对存储周期、数据安全敏感信息、加密信息、权限赋权常 态化检查。

审计机制建设：对数据资产的整个管理过程中，需要对敏感、重要数据的使用权 限、使用制度、审批流程、管理机制等方面进行严格审计，还需要保证集中审计的可 行性。

**2.** **资产发现与评估**

资产发现与评估的主要目的是让企业或组织明确其拥有哪些数据资产，以及这些 资产价值如何。理想情况下，资产发现与评估将产出一张或多张资产登记表，用于登 记当前拥有的数据资产，以及在不同业务中的资产价值。我们将在8.2节详细讨论数 据资产发现与评估。

**3.资产交易与定价**

资产交易与定价考虑数据作为资产在交易流通的过程中的重要事项，主要包括数 据的交易模式、定价模型等。我们将在8.3节中详细讨论数据交易与数据定价。

**4.资产运营和保护**

资产运营也包括对数据资产的评估和定价，但我们在此将其单独作为一个领域讨 论，是因为数据资产运营还包括资产的维护和增值。数据作为一种相对易丢失、易过 时的资产，良好的维护非常必要。 一方面，企业或组织需要周期性地对数据进行检查 和整理，以确保数据质量良好，能够提供预期内的价值；另一方面，企业还需要对数 据资产进行有规划、有目的的开发，以具体的业务价值为导向对数据进行调整，扩大 数据规模，提高数据灵活性，并最终提高数据资产的价值。

**8.2 数据资产发现与评估**

**8.2.1** **数据资产发现**

按照数据资产定义，其作为数据资源，包括以物理或电子的方式存放的各种数据， 其中物理存放类数据包括公文、合同、操作单、项目文档、记录、传真、财务报表、

**第8章**|数据资产化

发展计划、应急预案数据，以及各类外来流入文件等；电子存放类数据包括表单、配 置文件、拓扑图、系统信息表、用户手册、数据库数据、操作和统计数据、开发过程 中的源代码等。

数据资产管理的第一步是确保企业或组织知道拥有哪些数据资产及需要什么数据。 这需要进行数据资产盘点，以便创建数据资产登记表，以及用于资产识别和记录。数 据资产登记表为数据资产的管理和评估提供了基础，如果组织要管理和优化其数据资 产，则数据资产登记表必不可少。

通常，数据资产盘点将确定：持有什么数据、谁拥有它、如何持有和管理、它如 何流经组织生态系统。盘点还将评估信息来源的价值：使用了哪些信息、信息对组织 战略目标的重要性、是否应该保留、谁需要现有信息但无法访问、信息和信息资源重 复的地方、组织需要的信息存在哪些差距、用户满意度等。

在对所拥有和控制的数据进行全面盘点之后，企业或组织应最终形成数据地图

以方便后续的业务应用和数据获取。数据地图应指明数据的位置、类型、内容、关联 关系等，并可以使用对用户友好的方式被可视化。首先，从资产化管理和展示数据的 角度出发，数据地图作为数据资产盘点的主要输出物之一，不承载具体数据内容，却 可以帮助业务人员快速精确查找他们想要的数据。其次，数据地图作为企业数据的全 盘映射，能帮助数据开发者和数据使用者了解数据，并成为对数据资产管理进行有效 监控的手段。

无论公共机构是否有数据资产管理和评估的业务需求，随着数据安全相关法律的 实施，监管的需求也迫使机构展开数据资产评估。在公共部门，遵守法律要求的需要 使得以合规为导向的技术解决方案(如数据质量和电子记录文档管理)大幅增加。最 近，对更多面向业务的应用程序(如商业智能和分析)的需求急剧增加，这些应用程 序不仅能清理和归档结构化或非结构化数据以实现合规性，而且能搜索和分析它以提 供可以改变未来的洞察力。

**8.2.2** **数据资产评估**

面对如此纷繁复杂的数据来源，缺少盘点思路，恐怕很容易就陷入了数据的“汪 洋大海”之中。对此，我们建议从业务价值创造生命周期视角来予以推进。因为无论 是企业还是组织，其存在必定有相应的业务目标，也就是创造价值的方向。同时，围 绕这一目标，会设置相应角色、环节，形成一定的流程。不管是固化的流程，还是动 态调整的流程，“凡经过、必有痕迹”,借此，可对数据资产进行探寻。举例来说，在 产品规划环节，规划人员需要通过调查研究，在了解市场、客户需求、竞争对手、外 在机会与风险及市场和技术发展态势的基础上，根据公司自身的情况和发展方向，制 定出可以把握市场机会、满足消费者需要的产品的远景目标及实施远景目标的战略、 战术的过程。

通过对这个环节的业务理解，我们可以采取相应手段，对具体动作设计的数据资 产及对数据资产进行的加工处理进行识别。例如，参考开放组体系结构框架 (The Open Group Architecture Framework,TOGAF), 可以建立从业务架构、应用架构到数据 架构和技术架构的整个企业架构体系，这样就能够以业务为重心，建立其利益相关者

**大数据治理：** 理论与方法 ………………………………………

可理解的、稳定且完整一致的方式，用于定义主要的数据类型和所需数据源。

除了TOGAF框架之外，还可以参考的企业框架包括 Zachman 、EAP 、FEAF等。以 下对这些框架予以简要介绍，以方便读者在盘点数据资产时参考使用。

Zachman 框架：由John A.Zachman 在1987年创立的全球第一个企业架构理论，是 其他企业框架的源泉。Zachman 框架实际上是一种组织架构工具(用来设计文档、需 求说明和模型的工具)的分类学，架构材料往往使用二维表格进行表示。其中，第一 维是角色，主要包括规划者、拥有者、设计者、构造者、转包商、运营企业等；第二 维是描述中心，即和项目相关的什么 (what) 、 怎样做 (how) 、谁做 (who) 、 何时做 (when) 、为什么做 (why) 等问题。通过组合这两个维度，就可以更好地梳理并开发 企业架构。

EAP框架： Steven H.Spewak 在1998年定义了企业架构计划 (Enterprise Architect Planning,EAP), 它用于制定信息架构以支持业务这一过程和实现该过程的计划，它 更偏重于企业架构的动态部分，包括过程、计划、阶段的划分等。EAP 将企业架构的 过程分为4个层级，即开始启动、现状分析、目标分析、实现和整合计划。

FEAF框架：美国联邦政府CEO 委员会于1999年提出了联邦政府组织架构框架 (Federal Enterprise Architecture Framework,FEAF),并用于指导美国政府部门的信息化 建设。FEAF首先是一种组织机制，被用来管理企业架构描述的开发和维护，而在将企 业架构付诸实施方面， FEAF还提供了一种结构，用于组织联邦政府资源及描述和管理 联邦企业架构的相关行为。同时，还对企业架构的开发和维护过程及模型的形式进行 表述，包括架构驱动力、战略方向、当前架构、目标架构、过渡过程、架构片段、架 构模型、标准等。

TOGAF框架：The Open Group Architecture Framework(TOGAF) 框架由国际标准权 威组织The Open Group 制定，于1995年正式发布。它是基于一个迭代的过程模型，支 持最佳实践和可重用的现有架构资产，作为一种协助发展、验收、运行、使用和维护 架构的工具，用于设计、评估，建立组织的正确架构。TOGAF 内容非常丰富，包括架 构开发方法 (ADM) 、架构内容框架、参考模型、指引和技术、能力框架等。

总而言之，在盘点企业数据资产的过程中，需要围绕企业发展战略，只有深入理 解和把握企业业务价值创造周期内的各个流程环节，才可以更好地发现并进一步梳理 数据资产。



**).** **)**

**8.3.1**

**数** **据** **交** **易** **与** **数** **据** **定** **价**

**数据交易**

数据交易将是未来数据重要的发展方向，要想实现数据交易，就必须要有一套交 易模式，本节主要探讨三种交易模式：直接销售模式、Open API模式、多方合作模式。

**1.** **直接销售模式**

直接销售模式是最基础、最直接、最简单的数据交易模式，数据所有者明码标价， 将自身拥有的数据交易给买方。交易过后卖方获得数据的货款，买方获得数据本身。

**第8章**|数据资产化

就像去菜市场买东西，我给你钱，你给我蔬菜，通过简单的交易得到想要的东西。

2013年4月， 一位名叫 Federico Zannier的美国人火了，因为他要将3个月积攒的 大约7GB 的隐私数据卖掉。据 Zannier 自己统计，7GB的数据中共有280万行文本信 息，算下来相当于1500本书，以及软件记录的75.5万次鼠标点击次数，电脑摄像头自 动拍摄的2.1万张图片，总计1.9GB; 还有约2万张屏幕截图，总计1.7GB。最终数据 的价格成功炒到了1100美元。

然而，在数据信息直接买卖交易的过程中，如果数据涉及第三方公司和个人比较 详细的隐私，就属于违法行为，因为这已经侵害了第三方公司或者个人的利益。目前 直接进行数据买卖的数据商人都是通过违法手段获得数据信息的，如通过业务关系获 取银行数据信息，通过个人关系获取公安部门的数据信息，通过售楼中心低价收集买 房者信息。数据商人通过向大量的第三方商业机构进行数据重复售卖，从而获得巨额 利润。

**2.Open API 模式**

Open API(Application Programming Interface, 应用程序接口)并不是一个新概念， 在计算机操作系统出现的早期就已经存在了。在互联网时代，把网站的服务封装成一 系列计算机易识别的数据接口开放出去，供第三方开发者使用，这种行为就叫开放网 站的API, 与之对应的所开放的 API 就称为Open API。

Open API 按照服务类型来看，主要可以分成数据型、应用型、资源型三种。

数据型：也就是将自身的数据开放，让应用开发者根据已有的数据进行二次应用 开发。

应用型：表现为具有交互服务的应用功能，供用户直接使用，也可以支持在应用 之上的定制或功能叠加，例如，Flickr 图片搜索、谷歌日程、百度地图等。

资源型：提供对底层基础架构能力的调用和管理，包括存储服务、运算资源、加 工算法等，如Google 的 Appengine,Amazon 的 S3 等。

Open API的出现最初是因为企业规模日趋庞大、IT 系统日渐复杂，内部的协作也 需要模块化和服务接口化。随着业务的梳理及抽象，服务逐渐地不仅可以满足内部交 互，还对外开放给一些商业合作伙伴，等到 Facebook、Amazon 等公司将之全面商业化， 并且成为一种全新业务模式时，给很多开发者特别是个人和小团队开发者带来了机遇， 因为他们可以用相对较低(大多免费)的成本来实现自己的创意。随之而来的就是数 据资源价值的体现让开放服务的企业得到了回报，也成就了相应的平台价值。

以 Amazon 为代表，其著名的 AWS8已经成为国外许多互联网应用的基础架构，针 对中小型的互联网企业，充分利用这些费用并不高的Open API, 自然比组建一个 IT维 护团队的成本低得多，例如，很多技术人员收看的 InfoQ 视频节目，都是采用 Amazon 的 S3 服务来提供存储服务的。

数据开放一直是 Open API 平台的一个主要方向， Open API 也给破解数据交易难题 提供了一种全新思路，内容如下。

Open API 的使用者需要进行注册认证，这可以初步解决数据交易方的诚信问题。

Open API 使得数据交易的计费过程有了保障，可按次收费、按流量收费、按使用 时间周期收费等，业务模式可以不断组合创新。

大数据治理： 理论与方法………………

Open API 确保了服务过程的质量，由服务提供商来对 API 内容、性能、质量等进 行保障，数据治理也有了具体的对象和手段。

Open API的使用过程可以受到严格的监控，对其服务过程也能够随时进行控制， 这样使得数据交易中的风险防范具备了实施的可能。

Open API 在互联网行业已经得到了全面的认可，各大巨头如雨后春笋般推出了各 种基于Open API 的开放平台，其中，也不乏以数据服务为主要目标的巨头。如国外的 Facebook、Amazon 、Google 等，国内的百度、腾讯、阿里巴巴、奇虎360等。企业之 外，政府方面也同样看重Open API的服务模式。美国白宫制定的数字政府策略对Open API 有着明显的侧重，其引导政府机构采用API 来使政府职员及大众能够灵活访问政 府数据。

2014年， Open API方式在政府各部门间流行起来，无论是美国消费者安全委员会 提供的SaferProducts.gov, 还是消费者金融保护局的抵押贷款数据，又或是之后 FDA (美国食品药品监督管理局)发布的“药物召回及不良反应”数据，都采用了Open API的数据发布方式。其中，旅行者警报 (Travel Alert) 从2013年12月1日上线，只 经过一个季度，就收获了超过8000次的 API 调用。

政府看重 Open API 方式的又一个原因是，基于这种方式的收费模式更行之有效。 虽然 Open API方式默认像任何其他公共服务一样是免费的，但根据不同的服务级别和 场景，还是可以从财政预算角度来适当收费的。例如，对于高频度访问的应用，可以 在设定调用频率的前提下，对超过部分进行收费，也保障了社会资源的公平使用。此 外，还可以根据使用者的身份予以区别，例如，对非营利机构和非政府机构免费服务， 但对于营利机构，则可以收取增值服务费用。Open API 也不是完全没有问题的，其最 大的影响是， 一旦数据应用基于Open API方式来形成服务，则意味着API 本身变成了 一个单点故障的最大风险对象。

此外， Open API也给黑客窃取数据带来可能，例如，黑客利用一定的机器算法， 不停地轮询探测服务者的数据库内容，并且把获取到的数据下载到本地，这样积少成 多，最终就有可能获得完整的数据内容。

**3.** **多方合作模式**

多方合作模式是数据交易的一种特别模式，它立足于数据加工的特性，在多方 之间建立合作关系，共同来提供数据、加工数据、分享信息成果，并在此基础上， 形成利益划分关系。它并不是新出现的事物，在很多企业之间，建立数据合作实验 室、数据沙箱、数据研究分享计划等，就是具体的多方合作的体现。类似于熟人关 系，这种合作方式基于企业间的合作关系或框架协议，能够更好地形成信任，并且 推动深度的数据共享，可以说是一种更为彻底的数据交易模式。例如，移动运营商 与公安部门联合建立数据合作实验室。通过移动用户的通信信息与公安部门己有数据 信息的有机融合，可以利用通信行业通信基站的高覆盖率，实现公安部门监控区域的 高覆盖，利用大数据分析技术分析处理目标区域内用户通信行为和公安部门信息，切 实还原区域内人流的基本属性、行为、轨迹等特征，实现深度洞察公安管控区域的人 流特征。

传统的公安部门通过事后相关人员报警、警察出警来解决安全事件，而基于人流

 158

第8章|数据资产化

监控统计、分析及预测，公安部门可以重点预防和准备，提前采取相应措施，或大幅 提升侦缉能力，更好地打击和威慑潜在犯罪分子，从而最大程度减少公共安全损失。

8.3.2 数据定价

数据定价是进行数据交易的前提，对数据进行估值已迫在眉睫。但目前大数据交 易平台尚在起步阶段，大数据交易规模很小，定价方法过于简单，价格通常设置较低， 不利于数据卖方。

1. 大数据的价值特征

大数据在价值上具有不确定性、稀缺性、多样性等特征①。

不确定性：价值的不确定性来自两方面： 一是大数据本身并不能创造价值，其价 值体现在分析和挖掘后的结果；二是大数据具有大容量和多样性的特征，故其价值取 决于不同主体的需求，大数据在对的人面前才有价值。

稀疏性：价值的稀疏性是指大数据十分庞大，价值密度低，有价值数据所占比例 小，数据价值难以确定。

多样性：由于大数据的价值依赖于使用者，因此不同的使用者可以从大数据中获 取不同的价值。

这三个特征给大数据定价带来了困难。

2. 大数据的定价模式

较为适合大数据定价的理论模型有两种：效用价格论和成本价格论。其中，效用 价格论估计大数据的使用价值，成本价格论则估计大数据的建立和维护成本。二者分 别作为大数据的最高价值和最低价值，给定了一个理论上的价格区间。

具体来说，效用价格 P 为使用大数据前后的预期收益(或损失)的差值，即



其中，Q, 和H,为使用大数据1之前事件发生的概率和预期收益，T; 和J, 为使用大数据 1之后事件发生的概率和预期收益。而成本价格 P… 则表示为

Pmin=C₀+C₀r

其 中 ，C。为生产成本， r 为利润率。生产成本主要包括实施成本和维护成本两方面。就 利润率而言，最低利润率是可以保证基本再生产的利润水平。

于是，大数据的理论价格区间为 [P,P…]。

3. 大数据的定价策略

大数据的定价策略主要有静态定价和动态定价两种形式。其中，静态定价包括多 重定价、歧视定价、捆绑定价和拉姆齐定价。动态定价包括协商定价、拍卖式定价、 反向拍卖式定价。

针对大数据定价中的双向不确定性问题，主要有以下几种应对策略。

预处理策略：在交易之前，通过对大数据进行数据清洗和挖掘，从中提取出易于估

①刘朝阳.大数据定价问题分析 [J]. 图书情报知识，2016(1):57-64.

大数据治理： 理论与方法

值信息。这种策略可以直接消除大数据定价中的双向不确定性，具备保护隐私、保护 元数据等优点；其缺点在于数据挖掘的技术和深度可能会导致对大数据的不充分利用 和错误利用等。

拍卖定价策略：通常，直接给大数据定一个合理的价格是困难的，尤其对于大数据 交易的前期而言。而采用拍卖策略可以保证卖方的利益，也可以兼顾市场原则。对于 大数据拍卖而言，有两个原则： 一是分期拍卖，这可以保证随着拍卖的进行，买卖双 方对大数据价值的认识越来越清晰；二是多种拍卖形式结合，如结合买方竞拍和反向 竞拍，也可结合维克里拍卖。

协商定价策略：协商定价的基础是，买卖双方分别对大数据的成本价格和效用价格 进行预估，从而给出各自的价格区间。协商成功的前提为这两个价格区间存在交集。 在协商过程中，双方逐步试探对方的底线，对买方而言就是卖方的最低价，对卖方而 言就是买方的最高价。这种策略的缺点在于协商的过程可能非常漫长，导致交易的时 间成本较高。

反馈性定价策略：这种策略的核心是通过使用大数据后的反馈对大数据的价格进行 调整。在使用大数据之前，很难对其效用价格进行估计。于是，我们可以利用使用大 数据后得到的实际效用对未交易的大数据进行价格的调整。

**8.4** **拓展：大数据拍卖模型**

**8.4.1** **问题背景**

大数据具有大量、多样、高速、可以复制的特征，因此在价值上也具有不确定性、 稀疏性、多样性。不确定性和多样性是指大数据价值是随不同使用者而定的，不同的 人在消费大数据后将会产生不同的效用。并且大数据通常需要被处理后才能投入使用， 即多次加工，这样对不同的使用人群价值也不同。价值的稀疏性是指大数据十分庞大， 使得其价值密度较低，有价值的数据比例小，数据价值不好确定。大数据的这些特点， 导致了传统定价模式和定价策略并不能有效解决大数据定价不合理的问题。

大数据已经成为经济发展的重要推动力，但大数据定价充满困难。如果某数据商 品无法做到大范围传播，只能将所有权转移到少数人手中，那么我们采用拍卖手段便 可以有效解决大数据定价偏低问题，从而提高卖方的收益。本节以大数据拍卖为背景， 针对大数据具有可复制性，传统拍卖模型无法解决卖方应该拍多少件大数据商品才能 获得最大收益的问题，讨论大数据卖方在不同情境下时应该采取什么样拍卖方式才能 使收益最高，以及将大数据拍卖给多少竞拍者合适。

**8.4.2** **基本拍卖模型**

本节将讨论基本拍卖模型及其基本假设。

假设共有N 个竞拍者，对其中的第i 个竞拍者，他的折扣系数为θ(O=[0j,0₂,

θ₃,…,θ[-]),卖方只拍给一个竞拍者时，他估价Y, 拍给两个竞拍者时，他估价为 θ₁Y, 拍给三个竞拍者时，他的估价为θ₂Y, 以此类推。因为拍品数量增加会使竞拍者

 160

**第8章** |数据资产化

对拍品估价减少，但估价不会低于0,所以θ(1≤h≤N-1) 的范围在0和1之间。例 如，对某个竞拍者来说，卖方只卖一件拍品时，他的估价为100,他的折扣系数为θ= [0.9,0.9²,0.9³,…,0.9\*-'],那么,卖方卖两件拍品时，他的估价为90,卖三件拍品 时，他的估价为81,卖N 件拍品时，他的估价为100×0.9\*-I。

为了解决大数据拍卖面临的问题，后两节提出两种模型，适用于不同的拍卖环境， 一种是卖方不知道竞拍者估价的分布和折扣系数的分布，另一种是卖方知道竞拍者折 扣系数的分布但不知道估价的分布，通过求解得到卖方在具体案例下运用两种模型所 获得的收益的期望，来探讨模型的可行性。

两种模型共有的假设如下。

(1)大数据具有可复制性，双方进行拍卖的成本为0,且不考虑入场费和流拍成 本。双方进行拍卖的入场费和流拍成本与大数据所带来的收益相比很小，可视为0。

(2)每个竞拍者最多只能获得一件拍品，多余的拍品给他带来的价值为0。对竞拍 者来说，他利用大数据是为了获取信息，多件同样的大数据拍品给他带来的信息并不 比一件多，因此多余的拍品给他带来的价值为0。

(3)竞拍者是风险中性的。假定竞拍者不是风险爱好者，也不是风险规避者。

(4)假设所有竞拍者对拍品的估价都会小于某个常数w 。每个竞拍者都有估价上 限，同样地，对所有竞拍者来说，他们对要竞拍的物品都存在一个估价上限，即估价 小于某个常数。

(5)复制拍品对卖方来说成本为0。对于卖方来说，大数据的复制成本和最终拍卖 的收益相比很小，复制成本可视为0。

(6)双方都知道对方拥有哪些知识。卖方可能知道竞拍者折扣系数和对拍品估价 的分布情况，竞拍者了解卖方是否具备这些知识。

(7 ) 假设大数据商品对卖方自用价值为0。卖方一般只对已有大数据进行拍卖，并 不自行利用大数据里面的信息。

(8)假设对所有竞拍者来说，在每个确定的数量下，折扣系数在0和1内的某个 区间内服从均匀分布。 一般来说，对不同竞拍者，当拍品数量增加时，他们估价的变 动大小是不同的，即折扣系数是不同的，可假设其服从某一均匀分布。例如，卖方决 定拍两件时，竞拍者的折扣系数在[0.8,1]内服从均匀分布，卖方决定拍三件时，竞 拍者的折扣系数在[0.7,0.9]内服从均匀分布。

(9)卖方不设立商品的保留价格。

**8.4. 3** **扩展的Vickrey 拍卖模型**

Vickrey 拍卖在数量确定的多物品拍卖下是标准且有效的，标准是指报价高的竞拍 者获得拍品，有效是指估价高的竞拍者获得拍品。但对数量不确定的大数据拍卖该模 型无法适用。本节对这个模型进行修改，使修改后的模型适用于大数据拍卖，帮助卖 方确定最优的拍品数量，获得最大收益。

**1.模型的假设**

本模型有如下假设：

(1)竞拍者的估价服从同一个正态分布，具体哪个正态分布这个知识是竞拍者之

大数据治理：理论与方法

间共有的，卖方不知道；

(2)竞拍者对大数据的估价的折扣系数服从同一个均匀分布，且具体哪个均匀分 布这个知识是竞拍者之间共有的，卖方并不知道；

通常，在拍卖过程中，竞拍者知道他们的估价服从什么类型的分布，也知道他们 的折扣系数服从什么类型的分布，但卖方并不知道具体的分布情况，这适用于卖方对 竞拍者了解较少的情况。

**2.** **定价策略**

这个策略参考Vickrey拍卖模型。Vickrey 拍卖模型适用于数量确定时多件同质拍品 拍卖，将拍品卖给报价最高的那些竞拍者。当拍品数量为k 时，竞拍者i 对这些拍品的 估价由向量X(X=(x{,x₂,xj,…,x)) 给出。x, 表示竞拍者i 对第k 件拍品的估价。 拍卖开始后，竞拍者i 发出一个报价向量b'(b¹=(b,bi,bj,…,bi)),bj 表示竞拍者 对第k 件拍品的报价。在所有的竞拍者报价中，前k 个最高的报价赢得k 件拍品，假如 某个竞拍者赢得了M 件拍品，他将支付除自己外，其余所有竞拍者最大的k 个报价中 最小的M 个报价之和。而在 Vickrey 拍卖模型中，竞拍者按照报价等于估价的策略竞 价，即假如竞拍者认为拍品值x, 他就会报价x。

这个策略要修改 Vickrey拍卖模型，让卖方能够确认使自己收益最大的大数据拍品 数量，且让这个模型具有Vickrey 模型拍卖的良好性质。

本节要讨论的数据拍卖中，每个竞拍者只需要1件拍品，除第1件拍品外，每多1 件拍品对他们来说多带来的边际价值为零。每个竞拍者发出一个报价向量，b=(b₁,

b₂,b₃,…,b₄) 。b₁ 代表当拍品数量为1时，竞拍者的报价，b₂ 代表当拍品数量为2时，竞 拍者的报价，b₄ 代表当拍品数量为k 时，竞拍者的报价。其中b₁,b₂,…,b₄ 是不相等的， 它们之间必须乘上折扣系数0,如果竞拍者的折扣系数是θ=[θ₁,0₂,θ,…,θ、 -],那 么b₄=θ₄-b₁c

卖方需要比较拍卖不同数量的拍品时，具体拍多少件拍品获益最大。假设拍卖1 件拍品时，获得的收益是w,, 拍卖2件拍品时，获得的收益是w₂, 拍卖k 件拍品时， 获得的收益是w 。然后比较w₁,w₂,w₃,…,w₂ 的大小，最后决定拍多少件拍品。最终 卖方收益为：

W=max(w₁,W₂,W₃,…,wx

卖方会确定使得收益达到 W 的件数。假如卖方决定拍卖i 件拍品，那么在N 个竞 拍者中，将每个竞拍者的报价b,从大到小排列。最终报价排在前i 个的竞拍者获得拍 品。而支付规则是，如果某个竞拍者在卖方决定拍i件拍品时获得了拍品，那么他将支 付除自己的报价外，其他所有竞拍者在拍卖数量为i时的报价中排在第i位的报价，也 就是其他竞拍者最高的失败报价。

这种支付规则参考了 Vickrey 拍卖模型。竞拍者在每个数量下的报价策略相当于 Vickrey 拍卖模型在给定拍品数量时每个竞拍者仅需1件拍品时的情况。因此本节设计 的这个模型将传统的Vickrey 拍卖策略作为自己在拍卖某一特定数量拍品的一个策略。

假设共有N 个人参与竞拍，卖方决定拍卖j 件大数据拍品时，对随机抽取的竞拍者 1,除他自身外，其余竞拍者在面对j 件大数据拍品时，将他们的报价从大到小排列， 假如他的报价是x,C, 为其余竞拍者的第；大报价，那么这个竞拍者获胜时，期望支付

第8章|数据资产化

是 E[C;lC<x]。 付为

假设f(x)

他获胜的概率是 P(C;<x), 因此，估价为x 的竞拍者1的期望支

m(x)=P(C;<x)E[C;IC;<x]

是卖方拍卖j 件拍品时竞拍者估价的分布函数。w是所有竞拍者最高的

可能估价，拍j件拍品时，卖方的期望收益为



卖方需要比较不同数量拍品下的期望收益的大小，来决定最终要拍卖多少件拍品。

假如共有4个竞拍者，他们的折扣系数均为θ=[0.9,0.9²],竞拍者1,竞拍者2, 竞拍者3,竞拍者4估价向量分别是[100,90,81],[120,108,97.2],[90,81,79.2],

[108,97.2,87.48],因为报价等于估价，则他们的报价向量也为[100,90,81],[120, 108,97.2],[90,81,79.2],[108,97.2,87.48]。当卖方确定拍三件拍品时，由于竞拍 者1,竞拍者2和竞拍者4在卖方拍三件拍品时的报价最高，分别是81,97.2,87.48, 都大于竞拍者3的79.2。因此竞拍者1,竞拍者2和竞拍者4获得拍品，那么他们将支 付其余竞拍者在拍三件时的最高报价，也就是竞拍者3的报价79.2,卖方收益w₃= 79.2×3=237.6。卖方确定拍两件拍品时，竞拍者2,竞拍者4报价为108和97.2,是 拍两件时报价最大的两个竞拍者，他们获得拍品，他们将支付其余竞拍者在拍两件时 最大的失败报价，因为竞拍者1的报价90比竞拍者3的报价81高。因此获胜者最终都 支付90,卖方收益w₂=90×2=180 。 卖方确定拍一件拍品时，竞拍者2的报价120最 高，获得拍品，其余竞拍者中最大的失败报价为竞拍者4的108,因此他最终支付108, 卖方收益w₁=108。由于w₃>u₂>u₁, 最终卖方会选择卖三件拍品，获益237.6。

3. 模型的性质

在扩展的Vickrey 拍卖模型下，拍卖是标准的且具有有效性，竞拍者都会在不考虑 其他数量拍品下趋向按照自己对产品的估价来报价，使自己的报价和估价相等。

为了证明模型既是标准的又具有有效性的性质，我们给出如下定理。

定理8-1 在扩展的 Vickrey拍卖模型中，竞拍者都会趋向按照自己对产品的估价 来报价。假如估价不等于报价，估价为x, 报价为x', 那么竞拍者要么没有收益，要么 受到损失

证：假设拍卖的是j 件拍品。

(1)如果无论他是否获得拍品，他都按照自己对拍品的估价来报价，则他支付的 价格没有变化，是其他竞拍者第j 大报价，由于他是否获得拍品没有变化，那么他既没 有收益，也没有受到损失。

(2)如果按照自己对拍品的估价来报价，能获得拍品，而不按照此策略不能获得 拍品。则按照对拍品的估价来报价获得拍品时，他需要支付其他竞拍者的第j 大报价， 他的收益是自身报价与其他竞拍者的第j 大报价之差，而由于不按照此策略报价则无法 获得拍品，他便会失去这个收益。

(3)如果按照自己对拍品的估价来报价，不能获得拍品，而不按照估价等于报价 的策略能获得拍品。那么不按照估价等于报价时，估价必然小于其他竞拍者的第j 大报 价，但他获得拍品，那他就需要支付其他竞拍者的第j 大报价，他的损失是其他竞拍者

大数据治理：理论与方法……

的第j 大报价与自身估价之差。

因此竞拍者都会趋向按照自己对产品的估价来报价，这是弱占优均衡策略。

定理8-2 对同一个竞拍者而言，他在不同数量下的报价是不相关的，即他对某个 数量做决策时不考虑其他数量下自己的报价情况。

证：假如拍卖j件拍品。

(1)如果竞拍者考虑了其他数量拍品而导致自己多拍了，即原来按均衡报价，只 能拍0件拍品，报价小于其余竞拍者第j 大报价，估价同样小于其余竞拍者第j 大报价， 但由于多拍了，就必须支付比自己估价大的价格，这就会造成损失。

(2)在竞拍者本来能获得拍品时，却由于考虑其他数量拍品时自身的报价，导致 拿不到拍品，即面对j 件商品时，估价大于其余竞拍者第j 大报价，但由于报价低，使 得竞拍失败，也会造成自身收益减少。

(3)如果考虑其他数量拍品时自身的报价并不会改变竞拍者是否会得到拍品这一 情况，那么收益不变。

定理8-3 扩展的 Vickrey 拍卖模型是有效的且标准的。

证：这种拍卖模型总是将k 件拍品分配给k 个最高的报价且由于竞拍者的报价等于 不考虑其他数量拍品时的估价，因此估价越高报价也就越高，所以这种拍卖模型是有 效的且标准的。

在每个数量情况下，对拍品估价越高的竞拍者，报价也越高，最终总是报价高的 竞拍者获得拍品。

**8.4.4** **扩展的序贯拍卖模型**

由于实际操作方便，第一价格序贯拍卖常用于数量确定的多商品拍卖，但对数量 不确定的大数据拍卖，该模型也无法适用。本节对这个模型进行修改，使修改后的模 型适用于大数据拍卖，获得最大收益。

**1.** **模型的假设**

模型假设如下：

(1)竞拍者的估价服从同一个正态分布，具体哪个正态分布这个知识是竞拍者之 间共有的，卖方不知道；

(2)当人数增多时，竞拍者对大数据的估价的折扣系数的期望θ服从同一个均匀 分布，这个知识是竞拍者之间共有的，且卖方知道。

在拍卖过程中，竞拍者知道他们的估价服从什么样的分布，也知道他们的折扣系 数服从什么分布，而卖方只知道他们的折扣系数服从什么分布，掌握了较多关于竞拍 人的信息。

**2.** **定价策略**

第一价格序贯拍卖是将拍品一件一件拍出的，报价最高的竞拍者获得拍品。竞拍 者获得拍品后退出竞拍，剩余的竞拍者继续参加剩下拍品的竞拍。竞拍者的报价策略 是，拍第一件拍品时，估价为x 的竞拍者报价为E(Y₂lY₁<x), 拍第二件拍品时，估 价为x 的竞拍者报价为E(Y₂IY₂<x<Y), 假设刚开始一共有N 个竞拍者参与竞拍，

**第8章** |数据资产化

那么Y₁ 是其余N-1 个竞拍者最高报价，Y₂ 是其余N-1 个竞拍者第二高报价。第k 阶 段，报价策略为E(Y₄ lY₁<x <Y₄-)。

但直接将这个模型用在大数据拍卖上也不现实。因为它既无法帮助卖方确认拍多 少件合适，也没有考虑到每多拍出一件大数据拍品会给已经获得大数据拍品的竞拍者 造成多大损失。这个策略的目的是要修改第一价格序贯拍卖模型，使它适用于大数据 拍卖的情况。

卖方将所竞拍的拍品一件一件拍出，每轮拍一件拍品，每轮报价最高者获得拍品， 从第二轮开始，每轮拍卖结束后将拍出所得，对由于拍卖件数增多造成已经获得拍品 的竞拍者的损失进行补偿，直到卖方无法用拍卖所得填补已经获得拍品的竞拍者的损 失，则竞拍结束。

实际情况下，由共有的假设，竞拍者知道卖方拥有哪些知识，也知道卖方在补偿 时是如何做出最优决策的，并知道其余竞拍者估价分布和折扣系数分布情况，因此可 以算出自己在卖方弥补损失时可以得到的期望利润(因为卖方不清楚每个竞拍者具体 的折扣系数是多少，因此弥补的数额可能超过竞拍者实际的损失，使竞拍者获得利 润),它就是竞拍者获得补偿的期望与因拍卖件数增加造成的竞拍者的损失的期望 之差。

竞拍者会自然将这些期望利润加在自己的原先报价上，原先报价即不考虑之后卖 方有补偿行为时的报价，因为报价越高的竞拍者越先获得大数据拍品，在卖方每轮拍 卖后弥补损失时获得的期望利润会越大，因此竞拍者获得大数据拍品的顺序是不变的， 且获得大数据拍品的竞拍者最终支付也是自己的报价，因此报价时会自然将期望利润 加在自己的原先报价上，相当于竞拍者提前将这部分利润预支给卖方，在卖方进行补 偿时又将这部分利润返还给竞拍者。因此实际情况是卖方能对竞拍者的损失进行准确 的补偿。

接下来要讨论补偿机制和在此机制下何时会出现补偿失败的情况。由于卖方对竞 拍者的具体损失未知，但共有的假设可知，每多拍一件大数据拍品，竞拍者对大数据 的估价的折扣系数θ服从同一个均匀分布，这个知识是竞拍者之间共有的，且卖方知 道这个知识，θ服从均匀分布，范围是(20-1,1)。

定义补偿系数α=1-0,那么α服从均匀分布，范围是(0,2α)。但卖方不清楚具 体到某一个竞拍者的补偿系数。而那些已经获得拍品的竞拍者看起来没什么区别。因 此卖方只能假设他们的补偿系数都是α'。设某轮拍卖所得为w, 已经获得拍品的竞拍 者人数为N, 竞拍者目前共支付S, 卖方分得的利润为x, 设 F(x) 为补偿系数的分布 函数，f(x) 为相应的概率密度函数，则F(w) 为其中某个竞拍者同意按照该补偿系数进 行补偿的概率，由于每个竞拍者的补偿系数服从同一个均匀分布且相互独立，则补偿 成功的概率为

F\*[(w-x)/S]

卖方在这轮拍卖的期望收益为

W=xF^[(w-x)/S]

为了使期望收益达到最大，需要对x 求导，令导函数为零，即x 为 F[(w-x)/S]+ Nxf[(w-x)/S]=0 的解。可以解出卖方的期望收益。

大数据治理：理论与方法……………………

以上讨论的是卖方在补偿时的决策，而在实际情况下，卖方是按照竞拍者的损失 做出准确补偿的，即不会让竞拍者获得超出损失的补偿。拍第一件拍品时，估价为x 的竞拍者实际支付E(YIY|<x¹), 第二个获得拍品的竞拍者实际支付E(Y2IY2<x²

<Y), 第k 个获得拍品的竞拍者实际支付E(YIY<x⁴<Y), ( 其 中Y 代表第k

轮拍卖的其余竞拍者第N 大报价，x\*表示第k 轮获得拍品的竞拍者的估价)。

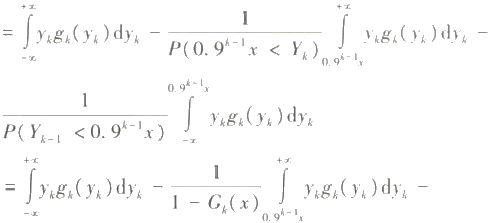
例 假设卖方只拍一件拍品，某竞拍者的估价为x, 他的折扣系数θ=[0.9,0.9², 0.9³, …,0.9\*~1]。即卖方拍两件拍品时，他的估价为0.9x; 拍三件拍品时，他的估价 为0.9²x; ……; 卖方拍k 件拍品时，他的估价为0.9-¹x。假设卖方拍一件拍品时，他的 竞争对手估价分布函数为G₁ (x), 概率密度函数为g₁ (x); 卖方拍两件拍品时，他的竞争 对手估价分布函数为G₂ (x), 概率密度函数为g₂ (x);……; 卖方拍k 件拍品时，他的竞 争对手估价分布函数为G(x), 概率密度函数为g(x)。 那么,他在第一轮获得拍品时， 实际支付E(YIY|<x)。

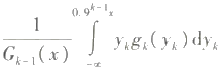


如果他在第k 轮获得拍品，实际支付为：









他获得拍品后，卖方继续拍卖，他会获得和自己损失相同的补偿。直到卖方拍卖 的收益无法补偿他和其他竞拍者为止。

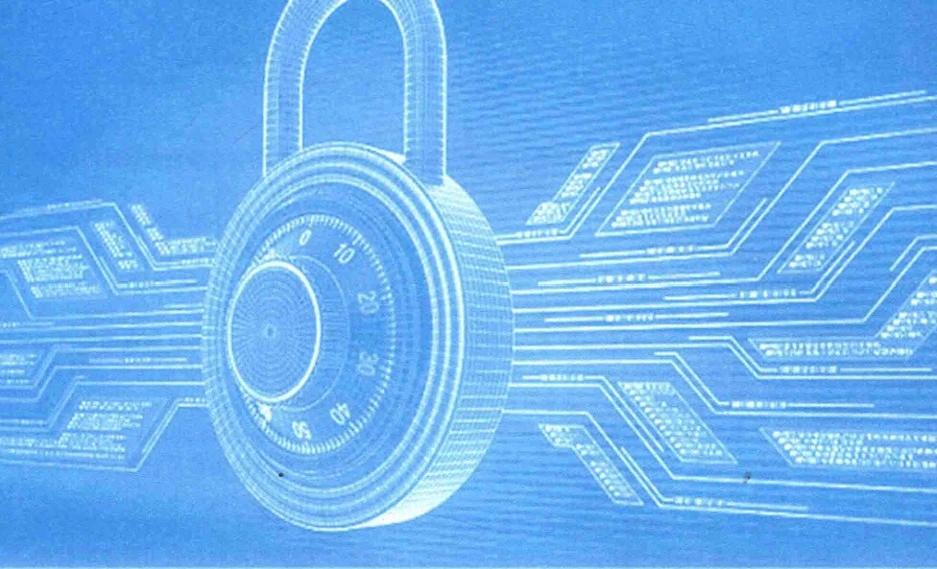
3. 模型的性质

这个模型是有效的。为了证明模型的是有效的，我们给出如下定理。

定理8-4 在这种策略下，每一轮拍卖中，对拍品估价最高的竞拍者获得拍品

证：在扩展的序贯拍卖中，每一轮拍卖，对拍品估价最高的竞拍者报价也最高，获 得拍品。在每一轮下，卖方都选择能使他收益最大，估价最高的竞拍者获得拍品。

因此该模型是有效的。



第 9 章

**数据安全与**

**隐私保护**

在大数据时代，数据可以说是科学研究和技术发展的重要基石。大数据的挖掘、 分析与处理等方面的研究广泛应用到社会、商业、医疗等与我们生活息息相关的各大 领域中，使用这些方法可以实现快速分类和搜索等多种功能。我们在享受个性化推荐、 语音识别、图像识别、无人驾驶等智能技术带来的便利的同时，数据在背后担任着驱 动算法不断优化迭代的角色。

大数据是依托互联网平台而产生的数据集，具有规模大、数据阐述速度快、多样 化的特点。而在网络时代里，人们依托互联网进行各种活动的行为也越来越频繁，在 互联网活动中产生的数据也越来越多。作为以互联网为依托的大数据，它将面临网络 带来的各种安全风险，这些风险威胁到大数据的安全，并可能给用户造成利益损失。 在科学研究、产品开发、数据公开的过程中，算法需要收集、使用用户数据，因此， 一些个人或敏感数据就不可避免地面临安全风险。2019年年初， MongoDB 、Elastic- search 数据库相继被曝出存在严重安全漏洞，可能导致数据泄露，两个数据库在默认情 况下，均无须权限验证即可远程访问并操作数据。CNCERT 抽样监测发现①,我国境内

① CNCERT发布《2019年上半年我国互联网网络安全态势》。

**大数据治理：**理论与方法……………………

互联网上用于MongoDB数据库服务的IP 地址约2.5万个，其中存在数据泄露风险的IP 地址超过3000个，涉及我国一些重要行业。

从信息时代开始，关于数据安全的研究就开始了。随着数据不断地增长，人们对 数据安全越来越重视，因此不同层面的数据安全问题也渐渐变得不可忽视。大致说来， 与数据相关的安全问题可以分为两大类：内生安全问题和衍生安全问题。

所谓内生安全问题，是指数据本身面临的安全问题，如存储安全、访问安全、检 索安全、处理安全、隐私保护等。这些安全问题是数据管理技术本身要解决的安全问 题。只要有数据存储、访问、共享的需求，技术上就存在相应的安全问题。例如， 2014年，“心脏出血漏洞”威胁全球约2/3的网络服务器内存储的用户名、密码及服务 器证书、私钥等敏感数据安全；同年索尼公司遭遇APT(Advance Persistent Threat,高 级持续性威胁)攻击，大量员工信息及影视资源遭泄露。上述攻击都是针对数据本身 的，目的是窃取或破坏数据。

所谓衍生安全问题，是指由数据管理技术本身发展和应用而衍生出的其他安全问 题。衍生安全问题不是技术本身的安全问题，而是由于技术的发展和应用导致的新的 安全问题。举例来说，在智能家居环境中，温度、湿度、光照等传感器采集的数据单 独来说不算隐私数据，但是如果将这些数据放在一起，再进行数据的关联分析，就可 以推导出一些本应受到保护的用户隐私。比如，用户每天晚上几点回家，是否会开启 电灯和空调。传感器采集到的家庭室内温度、湿度、光照会呈现周期性有规律的变化， 这种周期性变化就泄露了用户隐私。

在本章中，我们将重点关注内生安全，从数据安全存储、传输与访问，数据安全 检索与处理，隐私保护三方面进行介绍。

9.1 概述

**9.1.1** **基本概念**

经典的数据安全内容可以简化为保密性 (Confidentiality) 、 完整性 (Integrity) 、 可 用性 (Availability) 等，主要关注如何防止数据在其各个管理阶段遗失、泄露或被破 坏，具体包括数据保密、数据备份、异地容灾等。

保密性是指个人或团体的信息不为其他不应获得者获得。密码、Cookie、交易记录 等敏感或隐私数据需要被安全地存储、访问、传输、使用， 一旦发生泄露，则可能造 成重大损失。

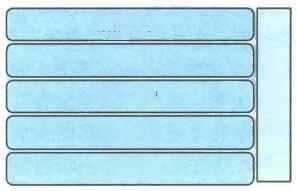
完整性是指数据不被未授权者篡改或在被篡改后能够迅速发现。 一些攻击者会通过破 坏数据来达成攻击目的，数据被破坏可能导致一些分析、挖掘服务失效或被恶意重定位。

可用性是指数据可被合法用户正确、有效地使用。换言之，对于合法用户，数据 可用性保证其在获取其权限内的数据时不被拒绝，且正确有效。

在大数据时代，数据安全的定义已经不仅仅局限于上述三方面，其内涵和外延都 被极大地扩展了。数据安全技术是大数据技术的伴生技术，贯穿了整个数据管理生命 周期，如图9-1所示。

**第9章** |数据安全与隐私保护

数据安全

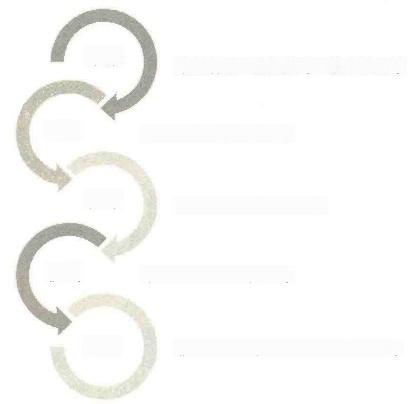
数据共享 数据使用 数据存储 数据传输

数据采集

图9-1 数据安全的定位

**9.1.2** **数据安全与数据管理生命周期**

数据管理生命周期包括数据的采集、传输、存储、使用、共享等多个阶段，每个 阶段各自面临不同的数据安全威胁，如图9 - 2所示。



数据篡改、数据投毒、隐私泄露等

数据窃取、数据失效等

数据丢失、数据毁坏等

隐私泄露、非法使用等

非法访问、数据丢失、数据泄露等

存储

共享

采集

传输

使用

图9-2 数据管理生命周期各个阶段面临的数据安全威胁

对不同类型和用途的数据来说，其生命周期各个阶段采用的技术不同，面临的数

据安全威胁也各不相同。

在采集阶段，常见的数据安全威胁包括数据篡改、数据投毒、隐私泄露等。如果 采集器不能被有效验证，那么其可能会篡改采集到的数据；在众包环境下，如果有恶 意的工人提供虚假数据，那么其可能污染采集到的数据；在物联网中，各种物联网设 备采集到的音视频、传感数据等经常和用户的个人隐私密切相关，因此需要考虑隐私 泄露的风险。

在传输阶段，常见的数据安全威胁包括数据窃取、数据失效等。例如，攻击者可 以通过侧信道攻击窃取数据。在国际信息安全顶级会议“网络与分布式系统安全会议” (NDSS 2020) 上，就发布了 一 项来自浙江大学、加拿大麦吉尔大学、多伦多大学学者 团队的最新研究成果 — — 攻击者可以通过手机内置加速度传感器采集手机扬声器所发

**大数据治理：** 理论与方法

………

出声音的振动信号，实现对用户语音的窃听。另外，由于环境恶劣、信道不稳定的原 因，数据在传输的过程中可能失效，这也是需要考虑的安全风险。

在存储阶段，常见的数据安全威胁包括数据丢失、数据毁坏等。例如，当前很多 应用都基于云服务，数据存储在云端，用户对数据没有直接掌控权，那么云端的备份 恢复、加密存储、数据隔离等问题就是必须要考虑的安全问题。如果没有相应的保障 机制，数据一旦丢失或损毁，则可能直接造成严重的经济损失。

在使用阶段，常见的数据安全威胁包括隐私泄露、非法使用等。许多个性化应用 为了提供更好的服务，必须要采集用户的个人信息，这其中不乏隐私数据。这些隐私 数据应当受到严格的保护，不应被非法用户访问和获取。除此之外，数据分析和挖掘 技术也是一把“双刃剑”,非法使用不仅会影响企业或组织业务的正常进展，还有可能 会导致违法犯罪。例如，如果用户仅购买了云存储服务，那么服务提供商在未经用户 许可的情况下擅自使用用户在云端存储的数据进行分析和挖掘就是非法使用行为。

在共享阶段，常见的数据安全威胁包括非法访问、数据丢失、数据泄露等。由于 共享往往涉及多方，那么如何保证各方在共享的过程中都合理合法地管理数据就尤为 重要。例如，用户的病历数据属于个人隐私，当不同组织在智能医疗方面进行合作时， 就必须保证数据在共享的过程中不会出现丢失、泄露，没有权限的共享参与方也不应 访问敏感数据。

在本章的后续部分，我们将介绍各类数据安全保护技术，以及如何基于这些技术 保护各个阶段的数据安全。

9.2 数据安全存储、传输与访问

**9.2.1** **加密存储和传输**

加密是一个过程，使数据只对正确的接收者可读，其他用户看到的是杂乱无序的 数据。对加密数据只能使用相应的密钥解密之后才能显示出数据本来的内容，以此达 到保护数据不被非法窃取、阅读的目的。其中，待加密的消息被称为明文 (Plain Text), 伪装消息以隐藏它的内容的过程称为加密 (encrypt), 被加密的消息称为密文 (Cipher Text),把密文转变为明文的过程称为解密 (decrypt)。

完整的密码体系要包括5个要素：

(1)明文空间： M是可能明文的有限集，称为明文空间；

(2)密文空间： C 是可能密文的有限集，称为密文空间；

(3)密钥空间： K 是一切可能密钥构成的有限集，称为密钥空间；

(4)加密算法： E 为加密算法，对于任一密钥，都能够有效地计算； (5)解密算法：D 为解密算法，对于任一密钥，都能够有效地计算。

加密算法 (E:M→C) 和解密算法 (D₂:C→M) 必须满足：对于任意x ∈M, 有

*D₄[E(x)]=x。*

密码学是一个被广泛研究的经典领域，目前已经有多种经典加密算法被应用在各 行各业中。对于安全的加密算法，破译者不能在有效的时间内破解出密钥k 或明文x,

170

**第9章**|数据安全与隐私保护

因此，为了保证数据的安全性和隐私性，将数据加密然后存储或传输是非常直观且常 用的技术手段。

经典的加密算法大致可以分对称加密和非对称加密，我们对这两种加密做简单的 介绍。

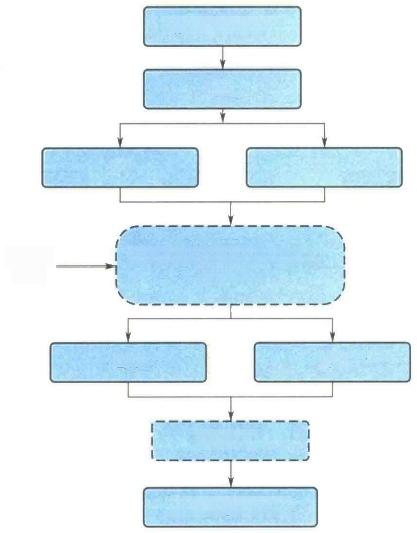
**1.** **对称加密**

对称加密是加密和解密使用相同密钥的加密算法。常见的对称加密算法有 DES、 3DES 、Blowfish 、IDEA 、RC4 、RC5 、RC6 和 AES 等。对称加密的优点在于高效，且密 钥较长时难以破解。然而，对称加密由于加解密使用同样的密钥，其密钥本身的安全 性就显得非常重要， 一旦密钥丢失，那么攻击者就可以很容易地读取全部数据。

我们以 DES为例，简单介绍对称加密①。

DES(Data Encryption Standard, 数据加密标准)是由IBM 公司研发的一种对称加 密算法，美国国家标准局于1977年把它作为非机要部门使用的数据加密标准。它是一 种典型的块密码算法，即将固定长度的明文通过一系列复杂的操作变成同样长度的密 文的算法。对 DES 而言，块长度为64位。同时，DES 使用密钥来自定义变换过程，因 此算法认为只有持有加密所用密钥的用户才能解密密文。

DES有16个相同的处理过程，称为“轮次”(round), 并在首尾各有一次置换 (permutation), 称为IP 与 FP (或称IP-),FP 为 IP 的反函数。DES 基本流程如图9-3 所示，包括三个主要步骤：

明文 (Plain Text)

IP

L0 RO

子密钥

生成

round function, 异或 (XOR)

16轮运算

L16 R16

FP(IP¹)

密文 (CipherText)

图9-3 DES 基本流程

① 加密算法的细节超出了本书的讨论范围，我们在此仅给出算法的基本思想，读者如需进一步阅读，可以 查阅密码学方面的相关书籍。



0F

大 数 据 治 理 ：理论与方法 ……………

(1)将明文做首部置换 (IP);

(2)将输入的数据等分为左右两部分，基于子密钥，使用round function 和异或 (XOR) 计算出中间结果，交给下一轮次，将此过程执行16轮；

(3)将第16轮输出的结果做尾部置换 (FP) 得到最终密文。

DES 加密和解密使用同一个算法，密钥表面上是64位的，但只有其中的56位实 际用于算法，其余8位可以用于奇偶校验，并在算法中被丢弃。因此，其有效密钥长 度为56位。不过，因为DES使用的56位密钥过短，以现代计算能力，24小时内即可 能被破解。也有一些分析报告提出了该算法理论上的弱点，虽然实际情况未必出现。 目前，该标准已经被 AES(Advanced Encryption Standard,高级加密标准)所取代。 AES的区块长度固定为128位，密钥长度则可以是128位、192位或256位。

**2.** **非对称加密**

非对称加密是加密和解密使用不同密钥的加密算法，也称为公私钥加密。当双方 要加密交换数据时，双方公钥是已知的，加密时使用公钥，解密时使用私钥。常见的 非对称加密算法有 RSA 、ECC 、DSA 等 。

非对称加密基本流程如图9 - 4所示，大致包含以下几个步骤：

(1)发送者撰写好明文x;

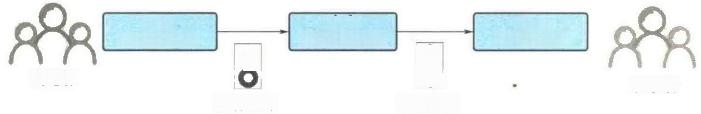
(2)接收者使用伪随机数生成器产生一对密钥，其中一个作为公钥ke, 另一个作 为私钥 k₄

(3)接收者发送公钥k。给发送者(传输渠道不必可信，即使有恶意攻击者在中间 窃听到 k。也没关系);

( 4 ) 发 送 者 用 公 钥k 。 加密明文x, 得 到 密 文 E(x);

(5)发送者可传输密文E(x) 给接收方(传输渠道不必可信);

(6)接收者收到密文，用私钥k₄ 进行解密(D[E(x)]), 得到明文x。



明文

(Plain Text)

接收者

公钥加密

**明文**

**(PlainText)**

密文

(Cipher Text)

私钥解密

**发送者**

日

图9-4 非对称加密基本流程

在上述过程中，如果有攻击者监听窃取了公钥k。和 密 文 E₁(x), 但 由 于 没 有 得 到 私钥ka, 所以依然无法得知明文x。 同理，如果发送者丢失了明文x 并且也没有私钥 k, 那么其同样也无法还原出明文x。

可以看出，由于公钥是公开的，当知道了其中一个密钥时，并不能凭此计算出另 外一个密钥，所以用户只要保管好自己的私钥，密钥的分发和保存将变得比较简单。 非对称加密的缺点是加解密速度要远远慢于对称加密，在某些极端情况下，甚至能比 非对称加密慢上1000倍。因此，当数据量很大时，非对称加密很难应用。

**3.** **组合使用对称和非对称加密**

由于对称加密和非对称加密各有优势，所以 一 种典型的加密方法是组合使用这两 种技术。也就是说，可以利用非对称加密来加密对称加密的密钥，再使用对称加密的

 172

**第9章** **|数据安全与隐私保护**

密钥加密全部数据。由于密钥通常较短，所以不用担心非对称加密的效率问题，将密 钥加密之后，也解决了对称加密的密钥管理困难的问题。

如果担心数据在存储或传输的过程中被篡改或者损毁，还可以通过校验来确保其 完整性。例如，可以使用哈希 (Hash) 算法，对明文生成一段特定长度且唯一 Hash 的 值，如果数据被篡改或损毁，其得到的Hash 值和原明文对应的 Hash 值并不相同①,进 而无法校验通过。这个过程是单向的，即已知 Hash 值时，是无法重新获得明文的。常 见的 Hash 算法有 MD5、HAVAL、SHA 等。

**9.2.2** **访问控制**

直观地说，访问控制 (Access Control,AC) 的目的就是管理什么人、在什么时候， 以什么方式、访问什么数据。访问的行为可以是读取、写入、转移、使用等。确定访 问称为授权 (authorization)。

访问控制在数据安全中扮演的角色至关重要。可以说， 一提起数据泄露，我们首 先想到的就是调查访问控制策略。常见的访问控制策略包括自主访问控制、强制访问 控制、基于角色的访问控制等。

**1.** **自主访问控制**

自主访问控制 (Discretionary Access Control,DAC) 让数据拥有者(用户)来确定 访问权限，系统通常不做强制要求。这种方式相对比较灵活，但给用户造成了很大的 负担，同时也容易出现安全策略管理混乱、不易维护的问题。

**2.** **强制访问控制**

强制访问控制 (Mandatory Access Control,MAC) 也称为非自主访问控制，安全策 略由系统(安全策略管理员)集中控制，用户无权控制。这是一种相对更严格的访问 控制策略，从系统的角度来讲，可控性更强，但也存在着不灵活、实现代价较高等 问题。

**3.** **基于角色的访问控制**

基于角色的访问控制 (Role-based access control,RBAC) 不同于 DAC 和 MAC 之 处在于，其不是直接赋予用户权限，而是将权限赋予角色。RBAC 的核心是围绕角色和 特权定义访问控制，而与策略无关。RBAC的组件，如角色权限 (role-permission) 、 用 户角色 (user-role) 、 角色-角色关系 (role-role relationship) 等，都使得执行权限分配 变得很简单。RBAC可用于具有数百个用户和数千个权限的大型组织中的安全性管理。 尽管RBAC与 MAC 和DAC 访问控制策略不同，但它也可以强制执行这些策略而不会造 成额外复杂性。

基于角色的访问控制由于其灵活性被广泛研究。当数据量非常大时，经典的 RBAC 会存在一些问题，如过度授权或授权不足。当数据复杂且用户构成复杂时，自上向下 为每个角色授权可能会使得有些角色获取了超过所需的权限，而另一些角色却没有获 得足够的授权。为此， 一些角色挖掘的技术被提出。这些技术可以从数据本身出发，

① 请注意，这一点不是绝对的，也可能(小概率)发生碰撞 (collision), 具体取决于算法设计。

大数据治理： 理论与方法

基于数据挖掘技术(如聚类算法)自下向上地分析、挖掘角色，并进一步给出合理的 权限分配。除此之外，还有一些基于风险的访问控制策略，关注于将风险分析引入 RBAC, 通过分析风险和收益来确定访问行为是否能够被允许。

**4.** **其他**

除上述几种策略之外，还有一些其他的访问控制策略，如基于内容的访问控制 (Content-Based Access Control)、基于意图的访问控制 (Purpose-Based Access Control, PBAC)、基于关系的访问控制 (Relationship-Based Access Control,ReBAC) 等。

9.3 数据安全检索与处理

在当前技术环境下，很多检索和处理需求很难由一方独立完成，当有多方共同参 与计算时，如何在互相不完全信任的场景下完成安全的多方计算是一个非常具有挑战 性的任务。我们不妨考虑几个现实场景：

(1)学校需要第三方提供基于学生学习数据的智能化服务，但是又不想泄露学生 的个人数据；

(2)用户想在第三方数据库中检索自己的遗传疾病相关的信息，但又不希望数据 库所有者知道自己具体要查什么疾病；

(3)制造厂商想购买第三方云计算服务，但又不希望泄露已方真实的生产数据；

(4)同一地区的几家医院想要协作开展智慧医疗服务，但又不希望泄露各自的病 历数据。

在上述场景中，如何做到“安全的合作”至关重要。本节将从安全检索和安全处 理两个角度出发介绍一些相关技术，这些技术或许可以为我们提供一些安全“合作” 的思路①。

**9.3.1** **安全检索**

数据被安全存储之后，查询和检索是其最重要的应用之一。如何在数据集上完成 检索但又不破坏其安全性一直以来是数据安全研究人员研究的重点之一。本节我们介 绍三类安全检索的相关技术，分别是隐私信息检索、健忘随机存取存储器和密文检索。

**1.隐私信息检索**

隐私信息检索 (Private Information Retrieval,PIR) 主要是针对公开数据，为了保 障个人隐私在公共平台上的私密性而采用的策略。当用户在检索数据时，PIR 采用一些 方法来阻止数据库服务器端获取用户查询语句和待检索数据的相关信息，从而保护用 户的查询隐私。这类技术也属于隐私保护技术，但由于其和密文检索具有较高的可对 比性，因此，我们在安全检索这里进行介绍。

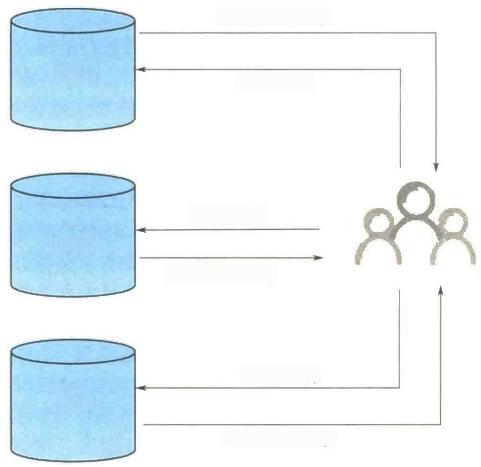
① 除了本节讨论的技术，还有很多研究者关注相关的问题，例如，华人图灵奖得主姚期智先生提出的安全 多方计算 (Secure Multi-Party Computation), 研究在无可信第三方的情况下，如何安全地计算一个约定函 数的问题。其起源于在1982年提出的“百万富翁问题”——两个争强好胜的富翁 Alice 和 Bob 在街头相 遇，如何在不向对方暴露各自具体财富的前提下比较出谁更富有?

**第9章** **|数据安全与隐私保护**

PIR 的目标主要是保护数据访问者的隐私。举例来说，用户可能会在一些公开的医 药数据库中检索自己的症状或病情，以期获得相关的医疗知识。为了获取所需的结果， 用户需要向数据库提供自己病历数据。病历数据是隐私数据，患者只需要获取检索结 果，但通常并不愿意其他人获取自己的病历信息。此时，可以使用PIR 技术来保护用 户隐私，以防私人病历数据被泄露给恶意第三方。

最简单的PIR 实现是将整个数据库全部复制，传送给用户，让用户在本地进行信 息检索。但显然，这种方法执行起来会存在很大的问题。首先，通信带宽未必能保证 数据被迅速传输；其次，数据是资产，数据库拥有者通常也不愿意免费提供自己的所 有数据；最后，普通用户在本地往往也很难具备可用的存储和检索条件。

一个更有效的解决方案是使用多个相互独立的服务器，每个服务器都有相关数据 的完整副本。查询交给所有服务器共同完成，每个服务器处理部分查询和一些迷惑信 息，任意一个单独的服务器都不会处理完整的查询。最终结果由不同服务器的结果共 同组成。这样，每个服务器获得的查询都是残缺的，从而保护了用户隐私。从每个服 务器都查询数据，但是查询的都不是想要的完整的结果。也就是说，每个服务器都查 询一部分数据，最后组合在一起形成最终想要的结果。当然查询的时候会加入一些迷 惑性的信息，查询过程如图9-5所示。

查询结果Ans(Q₁)

部分查询Q₁

DB₁

部分查询Q₂

DB₂

查询结果An s(Q₂)

部分查询Q₃

DB₃

查询结果Ans(Q₃)

**图9-5** **多服务器** **PIR**

用户组合Ans(Q₁)、Ans(Q₂)、

Ans(Q₃)得到最终结果

PIR 已经被研究了多年，其能够适用于一些应用场景，但也存在计算复杂度和执行 效率等方面的问题。PIR 的具体方案包括Percy++ 、Popcorn 、XPIR、SealPIR 等 。

**2.** **健忘随机存取存储器**

健忘随机存取存储器 (Oblivious Random Access Memory,ORAM) 面向私密数据 库，其通过混淆对存储设备的访问方式来防止隐私泄露。在过去的数年中，用户在将

**大数据治理：** 理论与方法…………………

计算外包到云服务器时对数据隐私的担忧已引起人们的密切关注。为了解决这个难题， 人们开始研究基于信任硬件的安全处理器架构。

ORAM 和 PIR 的区别在于， PIR 只保护用户的检索意图，而 ORAM 则认为存储介 质也是不可信的。因此， ORAM希望隐藏用户的访问模式，包括是读还是写、访问的 地址、访问的数据内容等。ORAM 接口将用户对程序地址的访问序列转换为对随机查 找的物理地址的ORAM访问序列，确保物理访问模式独立于逻辑访问模式，因此不会 泄露用户可能依赖于数据的访问模式。

为了实现上述目的，ORAM设计转化协议将1次访问转化为k 次访问，并确保两次 访问在转化后不可区分。这些被ORAM 引入的随机的“冗余”访问可以混淆内存访问 模式，从而以性能损失为代价防止隐私泄漏。

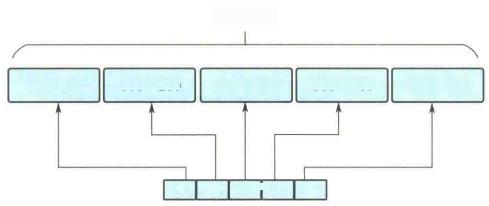
**3.** **密文检索**

顾名思义，密文检索的目标就是在加密的数据上进行检索，但不泄露用户的明 文数据和检索条件。如果不考虑效率，将密文直接传回用户再解密是最安全的做法。 但在实际应用中，受制于传输带宽、存储和计算能力等方面的因素，这种方式很难 实现。因此，需要有技术帮助用户在服务器端完成安全的检索。常见的方式是基于 索引的密文检索，即数据通过传统加密算法加密，但为了检索需要建立专门的安全 索引。

从索引的角度出发，可以将查询分解， 一般需要先对密文数据进行范围查询，定 位查询结果的可能位置，缩小解密范围，然后再解密执行精确查询。因此，索引建立 的重点和难点在于如何尽量提高定位的准确率，缩小返回客户端的密文数据的范围。 我们用一个例子简单说明基于索引的密文检索的基本过程。

例9 -1 考虑目标数据是有序的数值型数据的密文检索，数据大致落在[1,500] 这个区间内，那么就可以建立索引将数据分为5段：1～100,101～200,201～300,

301～400,401～500,如图9-6 (a) 所示。对于类似于“>420”这样的范围查询条 件，就可以先根据索引定位到块，然后解密对应的块，执行精确查询。当然，上述建 立索引的方式还是会引起隐私泄露，因为每一块数据的语义是已知的，为了防止这种 情况，可以对数据值进行 Hash 处理，根据 Hash 处理后的 Hash(x) 的值进行分区，如 图9-6 (b) 所示。



加密数据

1～100 101～200

索引

(a)

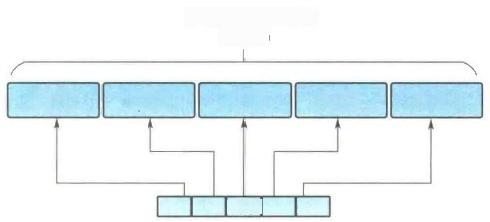
301～400 401～500

201～300

图9-6 基于索引的密文检索

176

**第9章** **|数据安全与隐私保护**



加密数据+Hash操作

Hash(x)

*h₅~h₆*

索引

*h₇~h₈* h₉~h₁₀

*h₃~h₄*

*h₁~h₂*

(b)

图9-6 基于索引的密文检索(续)

从上例容易看出，密文检索虽然可以很大程度上解决服务器端安全检索的问题， 但由于其需要建立索引，所以检索的功能通常不会太复杂。常见的功能包括关键词检 索和区间检索两大类。

关键词检索的主要目的是检索包含用户定义检索条件中的关键词的文本数据，具 体包括单关键词检索、多关键词检索、模糊检索、top-k 检索等。

区间检索主要针对数值型数据，检索返回某属性落在区间范围内的结果，具体包 括单维区间查询和多维区间查询。

**9.3.2** **安全处理**

安全处理关注如何在处理数据的同时不破坏其安全性质， 一类典型的应用是云服 务。用户购买云服务，并将其数据加密上传至云端，当用户需要在云端进行数据处理 时，如果在云端进行解密，那么这一解密操作可能让恶意攻击者有机可乘，窃取数据。 是否能够不解密就进行数据处理呢?同态加密或许是一个有效的解决方案。

之前我们讨论的数据加密都关注存储安全。当两方需要通信时，可以基于将数据 加密，以防被恶意第三方窃取。同态加密则关注数据处理安全。使用同态加密对数据 进行加密后，他人可以不经解密，直接对加密数据进行处理，并且保证数据处理结果 经解密后是正确的。换言之，对于同态加密来说，下面两个流程得到的结果是等价的。

流程1:数据加密——处理加密数据——得到加密后处理结果——解密处理结 果——得到解密后处理结果。

流程2:数据加密——数据解密——处理数据——得到处理结果。

请注意，流程1和流程2的处理操作不一定是完全一样的，最终的处理结果是等价 的。同态加密的思想起源于私密同态 (Privacy Homomorphism), 其允许在不知道解密 函数的前提下对加密数据进行计算，形式化地有如下定义：

定义9-1(同态加密) 设 M 和C 分别是明文空间和密文空间，E 为加密算法， a , b∈M。如果存在算法PLUS, 使得

E(a+b)=PLUS[E(a),E(b)]

则称其满足加法同态；

如果存在算法 MULT, 使得

E(a×b)=MULT[E(a),E(b)]



**大数据治理：** 理论与方法



……

则称其满足乘法同态；

如果E 同时满足加法同态和乘法同态，就称其为完全同态加密函数，如果E 只满 足其中一种同态，则称为部分同态加密函数。

构造完全同态加密方案的问题在1978年就被 Rivset,Adleman 和 Dertouzos①三人首 次提出了，但近些年，由于云计算的发展和人们对数据安全的日渐重视，同态加密渐 渐被越来越多的行业所提及。同态加密可用于保护隐私的外包存储和计算，为云计算 安全提供了理论上的解决方案。在很大程度上解决云计算上的数据安全问题。用户可 以对数据进行加密，然后将其外包给商业云环境进行处理，而所有这些处理都需要加 密。在诸如一些高度管制的行业中，可以通过消除阻碍数据共享的隐私屏障，使用同 态加密在保护隐私的同时开展新的业务。例如，由于医疗数据隐私问题，医疗保健中 的预测分析可能很难应用，但是如果预测分析服务提供商可以对加密数据进行操作， 那么隐私泄露的风险将大大降低。

9.4 隐私保护

本节将重点介绍隐私保护的相关概念和主要技术。

**9.4.1** **基本概念**

第一个重要的概念就是“隐私”。什么是隐私呢?我们可以将隐私定义为“个体的 满足某些特定条件的属性”。这些属性必须满足特定条件，包括如下条件。

(1)可以标识个体身份。换言之，基于这些属性可以识别出个体。典型的例子是 身份证号，如果我们掌握了某个人的身份证号，那么基于这个身份证号，就可以识别 出这个人是谁，进而就有可能知道他从事什么职业、有什么个人特征、有哪些社会关 系等。这些能够唯一标识某个个体的属性，通常都被认为是隐私数据，需要严格保护、 谨慎使用。

(2)泄露后会给个体造成损失。这种损失可以是经济上的，也可以是名誉或者其 他方面的。典型的例子是各种密码。除此之外，生日在一定程度上也属于个人隐私， 因为很多人会将生日作为常用密码之一。

(3)属于个人数据，但个体出于某些原因不愿意公开。如婚姻状况，个体可以主 动公开自己的婚姻状况，但是如果其不愿意公开，那么就应当将婚姻状况作为隐私数 据保护起来。类似的还有病历数据、家庭住址，甚至年龄、性别等。

我们将隐私的讨论范畴限制在个体之内，但并不表示群体的信息就不是隐私。如 果基于群体信息，可以定位到个人，那么群体信息也属于隐私数据。举例来说，如果 有人公开讲“光明小区所有的住户都购买了品牌A 型号B 的电子门锁”,这就属于泄 露隐私，因为对方一旦知道某人住在光明小区，就知道对方家里用的是哪个牌子的电 子门锁。如果此类门锁存在安全漏洞，就可能被不法分子利用，给住户带来生命财产

① Rivest R L,Adleman L,Dertouzos M L.On data banks and privacy homomorphisms [J].Foundations of secure computation,1978,4(11):169-180.

**第9章** **|数据安全与隐私保护**

损失。

用户作为数据的产生者，在将个体信息向外界提供时，需要知道收集和使用数据 的人是否是安全可信的。例如，病人同意医院采集自己的个人信息和身体状况等隐私 数据，是出于对医院的信任。但是医院在采集之后，有义务保证在使用和共享数据时， 不会泄露病人隐私。

为了有效保护隐私， 一类常用的技术手段是向数据中引入干扰，通过在可接受的 范围内牺牲数据质量，混淆不同个体的隐私数据，从而达到保护隐私的目的。例如， 当使用数据匿名化时，数据收集者负责通过隐藏或改变数据中的部分属性值，使得最 后发布的数据无法提供足够的信息来识别个体身份。另外，数据加密也是常用的隐私 保护手段。

如前文所述，隐私是.“个体的满足某些特定条件的属性”。因此，为了讨论隐私保 护，我们首先要对属性做分类。假设数据发布者所持有的数据表为T(A,,A₂,…,A,),

表中每个元组指明一个特定实体的相关信息，如身份证号、姓名、民族、出生日期、 性别、邮编、健康状况等。这些数据根据其作用可以分为4类。

(1)显示标识符：唯一标识个体身份的属性，如身份证号、社保卡号等。 一般在 数据发布前将显示标识符属性屏蔽、删除或加密，达到保护这些私有信息的目的。

(2)准标识符：表中的属性都可以作为准标识符，对于攻击者而言，准标识符是 攻击者可用的属性集合。例如，恶意攻击者拥有“姓名”和“年龄”两个属性的值， 想要基于这两个值窃取疾病隐私，那么“姓名”和“年龄”就可以作为准标识符。

(3)敏感属性：涉及个体隐私的属性，如身体状况、银行卡号、薪资等。敏感属 性与具体应用场景相关，例如，姓名、年龄、性别在任职公示时就不是敏感属性，在 网上购物时就是敏感属性。

(4)非敏感属性：不涉及个体隐私的属性。

数据表中，还可能存在背景知识数据，基于背景知识，可能推导出某些个体的隐 私信息。背景知识可能是外部数据，将外部数据和本地数据链接之后，就可能推导出 隐私数据。这类攻击称为链接攻击。

隐私保护技术根据其应用领域和范围，可大致分为数据失真、数据加密和数据匿 名化三类。

数据失真：在保持某些数据或数据属性不变的情况下使那些敏感的数据失真。例 如，通过添加噪声，阻塞、交换多个数据记录的值，凝聚等技术对初始的数据进行替 换，替换其真值，而替换后的数据必须具有初始数据的某些特性，使得用户能够使用。

数据加密：可以通过数据加密来隐藏敏感属性。在安全多方计算中常用于实现两 方或多方参与的协同计算。当某个计算过程结束时，每个参与者除自身的输入值和输 出结果以外，均不知道任何其他信息。

数据匿名化：发布数据时根据某些限制不发布数据的某些域值，可通过数据泛化 和隐匿扩大数据的值域等方法实现。

上述三类技术中，数据失真会造成一定程度的信息丢失，但该技术具有较高的工 作效能；数据加密则具有较高的安全特征，数据也非常准确，但如前文所述，在密文 上直接处理数据还面临很多技术挑战；数据匿名化能保证所发布的数据真实性，但会

大数据治理：理论与方法

有一定的信息损失。在隐私保护过程中，可以不仅仅局限于使用上述技术方法，在实 际使用过程中也可以融合一些其他的技术，对上述方法进行综合运用。

三种隐私保护技术对比如表9-1所示。

**表9-1** **三种隐私保护技术对比**

|  |  |  |
| --- | --- | --- |
| **技** **术** **类** **别** | **主** **要** **优** **点** | **主** **要** **缺** **点** |
| 数据失真 | 计算开销小；实现简单 | 数据丢失；根据实际数据设计不同算法， 对数据有依赖性 |
| 数据加密 | 数据真实、无缺损；隐私保护效果好 | 计算开销大；部署复杂，实际应用难度 较高 |
| 数据匿名化 | 适用于各类数据、众多应用，通用性高； 数据真实有效；实现简单 | 有数据缺损可能和隐私二次泄露的可能； 匿名算法相对开销较大 |

下面我们介绍一些典型的隐私保护技术。

**9.4.2** **数据脱敏**

我们在前文中介绍过的访问控制可以用来保护用户隐私，可以通过配置访问权限， 阻止未授权的用户访问敏感信息。然而，隐私保护的场景可能比较复杂，仅仅依赖访 问控制并不够。在很多应用场景中，为了获取某些服务，必须提供数据，但又不希望 在提供数据的同时泄露隐私。为了满足这种需求， 一种最直接的方式是数据脱敏。

所谓数据脱敏 (Data Masking), 是指对敏感属性(如个人身份识别信息、商业机

密数据等)进行数据变形处理，使得恶意攻击者无法从经过脱敏处理的数据中直接获 取 敏 感 属 性 的 取 值 ， 从 而 实 现 对 机 密 及 隐 私 的 防 护 。 例 如 ， 将 手 机 号 码 “13812345678”处理成“138 ×××× 5678”就是一种最简单的数据脱敏。在金融、 医疗、电信、电力、科研等诸多行业，数据脱敏都有着非常广泛的应用。

通常，在数据分析或共享时，数据脱敏是基本的操作。例如，医疗组织希望第三 方提供医疗数据分析服务，以追踪某些疾病的演化或传染模型。此时，出于隐私保护 的考虑，不可能将原始的病历数据共享给第三方，必须要先对数据进行脱敏。保证所 有个体标识都被隐去了，才可以使用。隐私数据可能会被遮掩、修改或 Hash 成无语义 的字符串，以防不可信的第三方窃取隐私数据。

数据脱敏的方法有很多，表9-1中的数据失真和数据加密都可以用于数据脱敏。除 此之外，还有数据置换，即使用可逆的置换算法，使数据兼具可逆和保证业务属性的 特征，可以通过位置变换、表映射、算法映射等方式实现。

表9-2给出了一些常见的数据脱敏方法。

根据数据脱敏方式的不同，其可分为静态数据脱敏 (Static Data Masking,SDM) 和动态数据脱敏 (Dynamic Data Masking,DDM)。

静态数据脱敏是指在数据存储时脱敏，存储的是脱敏数据， 一般用在非生产环境， 如开发、测试、外包和数据分析等环境。

动态数据脱敏是指在数据使用时脱敏，存储的是明文数据， 一般用在生产环境， 可以实现让不同用户拥有不同的脱敏策略。

**第9章**| **数据安全与隐私保护**

**表9-2** **常见的数据脱敏方法**

|  |  |  |
| --- | --- | --- |
| **名** **称** | **描** **述** | **示** **例** |
| 掩码 | 利用“·”“×”等符号遮掩部分信息，并且保 证数据长度不变，容易识别出原来的信息格式，常 应用于身份证号、手机号等 | 13812345678→138××××5678 |
| 取整 | 向上取整或向下取整 | 29→30,32567→30000 |
| 置换 | 使用可逆的置换算法，使数据兼具可逆和保证业 务属性的特征，可以通过位置变换、表映射、算法 映射等方式实现 | Alice→userl,Bob→user2,Cindy→user3 |
| 截断 | 舍弃某些必要信息保证数据的模糊性，但对使用 者不够友好 | 13812345678→5678  中国北京市海淀区→ 中国 |
| 加密 | 利用加密算法对数据进行变化，其安全程度取决 于具体加密算法 | 13812345678→Z7EXRTYJewrtyhj& |

从实践经验来看，在能实施数据脱敏的场景中，建议一定要实施数据脱敏。 一种 常见的管理方式是将数据划分为5个等级： L1 (公开)、L2 (保密)、L3 (机密)、L4 ( 绝 密 ) 和L5 (私密)。对于属于为L5 级别的敏感数据，要配以相应的管理措施，例 如，可以要求用户提供材料证明，在有使用明文的必要时才可以使用明文，否则只可 以使用脱敏后的数据。

**9.4.3** **k- 匿名**

k-匿名的概念最早由Latanya Sweeney 和 Pierangela Samarati 在1998 年发表的论文 中①引入，以解决下述问题：

给出特定个人的领域结构化数据，以科学的方式发布数据，使得保证在数据仍然 实用的情况下，无法重新标识作为数据主体的个人。

k-匿名的直观思想是，通过匿名化(如泛化和隐匿)原始数据集中的某些属性值， 形成满足一定匿名要求的匿名数据集并用于数据发布。这种技术一经提出就受到各界 特别是数据库领域的重视，在近些年中，该技术得到迅速发展。我们可以用一个例子 来说明为什么需要提出k-匿名。

例9-2 考虑图9-7 (a) 中的脱敏后的数据表，其数据来源于一次问卷调查，包含 了4条记录，被调查的个体给出了自己的性别、年龄和疾病信息。如果没有外部数据 源或其他背景知识，这种脱敏后的数据似乎已经足够保护用户隐私了，但是如果有人 有背景知识：“Alice 参与了这次调查，Alice 是18岁的女性”。由于这个数据表中只有 一条记录的“性别”属性值是“女”且“年龄”属性值是“18”,所以，拥有背景知 识的人可以知道 Alice 有心脏病。这就是我们前面讨论过的“链接攻击”。然而，如果 调查结果如图9-7 (b) 所示，则不存在上述风险。恶意攻击者即使拥有的背景知识，

① Samarati P,Sweeney generalizatfion and

L.Protecting suppresion

privacy when disclosing information:k-anonymity and its enforcement through

[J].1998.

181

**大数据治理：** 理论与方法………………

也无法得知Alice 是否患有疾病。

|  |  |  |  |
| --- | --- | --- | --- |
| **姓名** | 性别 | 年龄 | 疾病 |
| aaa | 女 | 18 | 心脏病 |
| bbb | 女 | 22 | 无 |
| CCC | 男 | 73 | 无 |
| ddd | 男 | 76 | 心脏病 |

(a)

|  |  |  |  |
| --- | --- | --- | --- |
| 姓名 | 性别 | 年龄 | 疾病 |
| aaa | 女 | 20 | 心脏病 |
| bbb | 女 | 20 | 无 |
| ccc | 男 | 70 | 无 |
| ddd | 男 | 70 | 心脏病 |

(b)

图9-7 两个脱敏后的数据集

在图9-7 (b) 所示的数据集中，如果我们考虑属性“性别”和“年龄”,那么任 意一条记录都至少和其他的一条记录无法区别开来。换言之，具有相同“性别”和 “年龄”的记录，至少有两条。这就满足了k-匿名 (k=2) 的要求。

发布后的数据具有k-匿名性意味着： 发布后的数据中包含的每个个体都无法与至 少k-1 个数据集中的其他个体区分开。我们假设有恶意用户(对手)想要窃取用户隐

私，其可用的属性为“性别”和“年龄”,我们称这两个属性为准标识符。对于具有k- 匿名性的数据集，每个准标识符元组至少出现在k 条记录中。也就是说，如果仅仅从 扰动数据上进行推断，成功地将隐私数据和它对应的个体相链接的概率至多为1/k 。在 k-匿名方法中，参数k 决定了隐私保护的强度， k-匿名方法并不要求原始数据与公开数 据之间有较大的差异。相反，在满足k-匿名的前提下，由于隐私保护强度已经确定， 为了最大化数据可用性，往往还要求原始数据与扰动数据之间的差距尽可能小。 Meyerson 和 Williams证明最优 k-匿名性是一个 NP-难问题①,”但是有很多启发式方法 被提出。

k-匿名要求对准标识符属性做匿名化处理，即修改数据集中的准标识符属性的取 值，以混淆相关记录。在匿名化处理的同时，要尽可能保留相关的统计特征。匿名化 可以通过泛化、微聚集、分解、排列等方法实现。

泛化：泛化通过降低准标识符属性值的精度，使得数据表中准标识符属性值相同 的元组个数增加，从而降低攻击者通过准标识符属性标识个体身份或个体敏感值的概 率。例如，“18”和“22”都可以约等于“20”。准标识符属性分为数值型和分类型两 种，不同类型的属性泛化操作不同。数值型属性一般被泛化成区间，分类型属性则用 一个更一般、更普通的相对应原属性值的值来取代。目前，已经提出了大量的泛化处

① Meyerson A,Williams R.On the complexity of optimal k-anonymity [C]//Proceedings of the twenty-third ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems.2004:223-228.

**第9章** **|数据安全与隐私保护**

理算法来实现匿名数据的发布。

微聚集 (M ircoaggregation): 微聚集通过某种启发式方法将数据集中的记录划分成 若干类。具体地，可以根据数据的相似度将数据分类，保证每类至少包含k 个数据， 用类质心代表所划分类内元组的值，从而实现匿名。划分要求同一类内的数据尽量相 似，不同类间的数据尽量不同。

分解：分解的基本思想是通过对敏感属性值分组，且采取敏感属性与其他属性分 开发布的方式，降低准标识符与敏感属性之间的关联度。具体地，可将待发布的数据 集分割成两部分， 一部分是包含准标识符属性的表，另一部分是包含敏感属性值的表， 两表通过共同的ID 关联。

排列：排列通过将数据记录划分为若干个组，并在各组中打乱敏感属性值的顺序， 降低准标识符与数值型敏感属性的关联度，该方法主要针对敏感属性为数值型的数据。

k-匿名可能会受到同质攻击或背景知识攻击。

同质攻击 (Homogeneity Attack): 这类攻击利用了k 个记录集中敏感值的所有值都 相同的情况。在这种情况下，即使已对数据进行k-匿名处理，也可以准确预测k 个记 录集的敏感值。

背景知识攻击 (Background Knowledge Attack): 此攻击利用一个或多个准标识符属 性与敏感属性之间的关联来减少敏感属性的可能值集。例如，如果攻击者了解日本患者 心脏病发作的发生率较低，就可以用来缩小患者疾病敏感属性值的范围。

**9.4.4** **差分隐私**

差分隐私 (Differential Privacy,DP) 大多关注统计数据库，其目标是在尽量保持 个体记录不被识别的同时最大化数据查询的准确性。所谓统计数据库，是指由在保密 的前提下收集的一组数据组成的数据库，目的是生成统计信息，这些统计信息的产生 不会损害提供数据个人的隐私。

我们用一个例子来说明为什么需要差分隐私。

例9-3 考虑统计数据库中包含个人薪资信息的情况，如图9-8所示，人们可以向 统计数据库发起查询来查询薪资总额(注意：数据库只返回最终的统计结果，“姓名” “性别”“年龄”对查询者来说均不可见)。

|  |  |  |  |
| --- | --- | --- | --- |
| 姓名 | 性别 | 年龄 | 薪水 |
| Alice | 女 | 18 | 10 |
| Bob | 女 | 22 | 20 |
| Cindy | 男 | 33 | 30 |
| Divid | 男 | 46 | 40 |

图9-8 包含个人薪资信息的数据库

考虑这样两个查询。

Q1: 查询所有员工的总薪资。

Q2: 查询35岁以下员工的总薪资。

如果攻击者先后执行这两个查询，则Q1 将会返回结果“100”,Q2 将会返回结果

大数据治理：理论与方法…………………

“60”。如果攻击者拥有背景知识“只有Divid一个人的年龄超过了35”,那么他就可以 知道 Divid 的薪资是100-60=40。

差分隐私就是为了防止攻击者通过上述方式窃取隐私。我们可以基于上例给出一 个比较直观的解释：假设统计数据库D 中包含个体 Divid, 令 Ans(Q,D) 表示D 进行 任意查询操作 Q 所得到的结果，如果Ans(Q,D)=Ans(Q,D-{Divid}) (D- {Divid| 表示从D 中删除 Divid 的记录),那么就可以认为查询不会泄露 Divid 的隐私。

差分隐私保证任意个体在数据集中或者不在统计数据库中时，对最终发布的查询 结果几乎没有影响。2006年，Dwork、McSherry、Nissim 和 Smith 的一篇文章介绍了 ε-差分隐私(ε-DP) 的概念，给出了统计数据库的数据发布相关的隐私损失的数学 定义。

定义9-2(e- 差分隐私) 令ε为正实数， A 表示为将数据集作为输入的随机算 法，Q、表示A 的所有可能的输出构成的集合。对于任意两个只有单条记录不同的数据 集 D 、D'和 Q、的子集 S, 如果有

Pr[A(D₁)∈S]≤exp(e)×Pr[A(D₂)∈S]

则称算法 A 提供ε-差分隐私保护，其中参数ε称为隐私保护预算。

e 通常取很小的值，如0.01,0.1等，其值体现了算法 A 能够提供的隐私保护的 程度。用户可以通过设置不同的隐私保护预算，调整数据的实用性和隐私性。

根据定义9-2,差分隐私保护通过对算法引入随机性，使得对单条记录不同的数据 集D 和 D'进行查询时，获得相同结果的概率非常接近。在例9-3中，如果在查询5个 人的信息和查询4个人的信息时，获得的结果是相同的，那么攻击者就无法通过比较 两次查询结果的差异来窃取隐私。

ε-差分隐私提供了较为严格的隐私约束，在相同的参数条件下，如果有需要，还 可以使用ε-差分隐私的一些扩展版本，如流行的(ε,δ)-差分隐私，其定义将概率约束 放松了δ,即Pr[A(D₁)∈S]≤δ+exp(e)×Pr[A(D₂)εS], 从而允许了一定概率 的错误发生。

差分隐私在很多领域都有使用，如机器学习中常用的主成分分析 (Principal Com- ponents Analysis,PCA)、逻辑回归 (Logistic Regression)、支持向量机 (Support Vector Machine,SVM) 等都有对应的差分隐私化算法。差分隐私机制保证了数据库的每个个 体都不被泄露，但数据库整体的统计学信息，如平均值、总和、计数、方差等，可以 被外界了解。

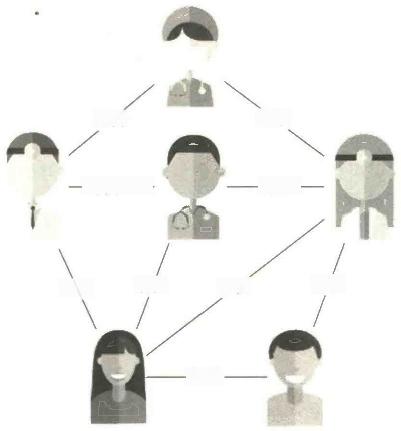
随着大数据、物联网、人工智能等技术的高速发展，各种服务也不断推陈出新， 用户在使用服务时往往不得不提供一些个人隐私数据。随着应用的多样化，隐私保护 面临的题挑战也日益严峻，仅仅从技术上实施隐私保护是不够的，还需要相应的管理 和制度上的保障。

**9.4.5** **图数据隐私保护**

随着大数据技术的深化发展，大量的半结构化和非结构化数据被应用于各类分析 中。图数据是半结构化数据，其能够准确表示事物之间的联系，因此许多场景下的数 据都可以形式化为图数据。例如知识图谱、在线社交网络中的朋友关系、商品交易数

**第9章** |数据安全与隐私保护

据、电话或邮件交流、医患关系、交通数据等。图数据不仅能够表示个体的属性信息 (如用户的姓名、年龄、学历等),还能够表示个体与个体之间的关联关系。基于这些 关联关系，我们很有可能可以挖掘出更深层次的个体属性。例如，如图9-9所示， Alice 与多位肝癌方面的专家保持着密切的社交关系，而她自己又不是医学方面的专业人士， 那么很有可能 Alice 本人或她的亲属患有相关方面的疾病。针对图数据的挖掘可能会很 有用，可以帮助我们获知某些实体的额外的属性、或特定实体在特定群组中的重要性， 但是，如果这类技术被滥用，带来的隐私泄露风险也是非常可怕的。Alice 可能并不希 望医生之外的人获知她或亲属的病情，如果有人通过社交网络中的社交关系挖掘出病 情相关的信息，则可认为是 Alice 的隐私被泄露了。这样的隐私泄露还有可能进一步诱 发诈骗或钓鱼等犯罪行为，给Alice 带来经济损失或人身安全威胁。

夫妻

合作

—同学—

-合作一

认识 认识 认识 医患

-姐弟 ·

图9-9 医疗关系图

虽然我们前文中所讲到的隐私保护技术的举例基本都是使用结构化数据(关系数 据),但这些技术几乎全部都有它们的“图数据版本”。不过，图数据中的隐私保护要 考虑的元素更多，这是因为关系数据隐私保护模型通常仅考虑记录的属性值作为背景 知识，攻击者在获知记录属性值的情况下发起攻击，但是图数据中节点属性、边属性、 节点关系、节点位置、子图结构等信息均可以作为攻击者的背景知识。攻击者可能通 过节点的一跳或几跳邻居来推断节点本身没有展示出来的属性或标签，达到窃取隐私 的目的。

图数据隐私保护中最直接的两类是节点隐私保护和边隐私保护，各类中还可以进 一步细分，如图9-10所示。

节点隐私研究的主体是图中的节点。通常图中节点代表实体，如社交网络中的用 户，或医患关系中的医生和患者。这些实体往往对应着现实世界中的个人或组织，因 此，与节点相关的信息很有可能是需要保护的隐私数据。具体地，节点隐私需要保护 节点相关的一下几类信息。



边存在性

边类别

边重要性

边属性值

边是否存在于图中

边的类别标签

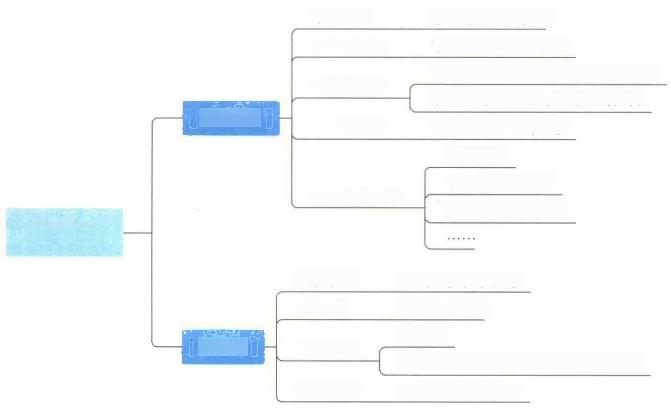
边权重

经过某些算法计算得到的边重要程度

边自身的细节描述

边险私

**大** **数** **据** **治** **理** **：**理 论 与 方 法 … …



节点身份识别信息

节点是否存在于图中

节点在特定节点集合中的重要程度 节点在特定图结构中的重要程度

节点自身的描述细节

节点的度

节点的邻居构成

节点间的可达关系

图数据隐私

节点标识

节点存在性

节点重要性

节点属性值

节点结构信息

节点隐私

图9 - 10 图数据隐私分类

节点标识：节点标识可以作为节点的身份识别信息，例如社交网络中的姓名、身 份证号等信息，获取节点标识通常意味着能够将节点与现实世界中的实体关联起来。 因此，数据发布的时候往往会隐藏节点标识信息。

节点存在性： 一些情形下，节点是否存在也是需要保护的隐私信息。例如，在社 交网络中，节点存在性指明了某个人是否出现在某个社交网络中， 一些人可能并不希 望这样的信息被暴露，此时隐私保护需要建立机制以防攻击者通过背景知识推断出特 定节点的存在性。

节点重要性：通过图数据挖掘，往往可以确定特定节点在特定群体或特定结构中 的重要程度。与存在性类似，某些人可能并不希望被他人得知自己在特定群体中的重 要程度，因此也需要相应的隐私保护机制。

节点属性值：复杂图中的节点不仅有标识信息，可能还带有一系列的标签或属性 值，这些属性值刻画了实体的一些细节信息，如地址、电话、血型、收入等。由于图 数据记录了节点之间的关联关系，因此攻击者可能可以根据邻居信息来推断某些节点 原本没有的属性信息，例如朋友关系网络中的朋友的共同爱好或家族关系网络中的遗 传病等。

节点结构信息：节点的结构信息包括节点的度、节点的邻居构成、节点间的可达 关系等，这些信息可能反映了现实世界中的个体与某些组织或其他个体关系的构成或 联系的紧密程度。结构信息与上述其他几种信息相比，隐藏相对较困难，这是因为修 改了图结构之后，可能会使得后续其他分析不够准确，故需慎重。

边隐私：考虑图中与边相关的隐私信息。图中边通常反映了现实世界中的实体间 存在的关系，复杂图中可能会使用不同类型的边来代表不同类型的关系。例如，在医 疗场景下的图数据中，两个实体间有一条边，可能表示了医生和护士之间的合作关系， 也可能表示了医生和患者之间的治疗关系，如果表示治疗关系，那么患者就可能觉得

这 是 其 隐 私 信 息 而 不 愿 公 开 这 条 边 的 存 在 。 具 体 地 ， 边 隐 私 需 要 保 护 与 边 相 关 的 以 下

 186

第9章|数据安全与隐私保护

几类隐私信息。

边存在性：边存在性反映了两端实体是否存在该边指明的关系，医疗、交易、朋 友等关系都可能是个体想要保护的隐私数据。例如，图9-9中的医患关系边的存在性可 能就是患者不希望别人知道的关系。如何保证不暴露隐私边的存在性也是图数据隐私 保护要研究的问题之一。

边类别：复杂图中，边的类型可能不止一种。例如，在图9-9表示的医疗关系图 中，共有6种类型的边，边的类别可能是夫妻、同学、姐弟、合作、认识或医患。边 的类别包含了个人信息，其表明了某人与其他人具体是什么关系，因此也可能需要被 隐私保护机制所保护。

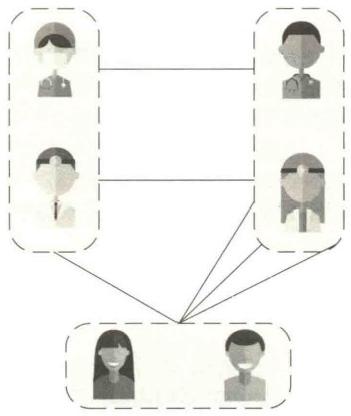
边重要性：边的重要性可以是直接通过边上的权值所反映，也可以是通过数据挖 掘算法计算得到的。边的重要性表明了其对应的关系的重要程度，在金融、医疗、家 庭关系等图数据中，越重要的边约有可能与隐私数据相关。

边属性值：复杂图的边不仅会有类别信息，也可能带有一个或多个属性值。例如， 表示金融交易的边可能带有交易时间、交易类别、交易量等属性值。这些属性值从不 同侧面描绘了边所代表的关系的细节，对于边所关联的个体来说，可能是需要被保 护的。

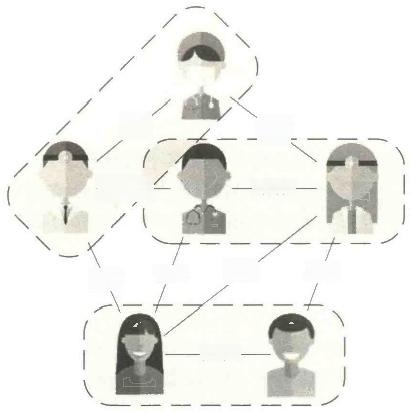
图数据隐私保护最直接的手段就是数据失真和数据匿名化。例如，为了保护节点 隐私，可以在数据发布时对节点的标识做匿名化处理，使用数字或字母构成的无意义 串来标识节点，这样攻击者在没有标识相关的字典的情况下就很难将节点与真实世界 的实体对应起来；为了保护边隐私，可以对边做增删操作，通过增加、删除或交换某 些边来混淆攻击者的认知，使其难以判断特定边的存在性或重要性。我们前文中讲到 的隐私保护技术基本都有其对应的“图数据版本”,以下我们给出一些例子。

图数据中的匿名化： 我们在前面的章节讨论过的 k-匿名针对图数据做一些“定 制”,就演化成了节点k-匿名、k-度匿名、k-同构等。具体的实现包括基于聚类和基于 图修改等方法。基于聚类的匿名可以基于节点或边的相似度对数据进行聚类，通过聚 类算法将节点或边(或节点与边混合)聚类为若干簇，使同一簇中的对象之间相似度 较高，不同簇直接的对象相似度较低，然后进行泛化使得攻击者基于背景知识尝试窃 取隐私时窃取成功的概率降低。不过，基于聚类的方法对于簇内节点的信息详细隐藏 较多，会带来较大的信息缺损，对后续的合法分析不利。基于图修改则采用增加伪节 点、增加伪边、删除节点、删除边等方式来调整图结构，从而满足匿名化要求。 一般 说来，结构修改方法相对聚类方法而言可保持图的原有规模，数据缺损相对较小，可 获得相对较高的数据效用。不过，在增删节点和边时如果不够谨慎，则会对图中节点 或子图的某些重要性质(如连通性、可达性等)带来改变，同样会造成较为严重的信 息缺损。在此我们给出一些典型的定义，并简单讨论相应的实现方法。

节点k-匿名首先对图中节点进行聚类，根据聚类结果将图修改为有若干超点构成 的图，其中每个超点中的节点不可区分，且每个超点至少包含k 个节点，这样当攻击者基 于自己的背景知识尝试识别特定节点时，识别成功的概率小于1/k。例如，图9-11 (a) 表示了图9-9中的社交关系可能的聚类结果，其中虚线圆角矩形框内的节点被聚为一 类，图9-11 (b) 则表示了根据该聚类结果生成超点之后的图结构。在图9-11 (b) 所

大数据治理：理论与方法

示的社交关系中，每个超点中有两个原始图中的节点，考虑边有区别的情况下，每条 边都有4种可能，那么图9-11 (b) 对应的图就有4⁶=4096种可能性，攻击者在没有特 定背景知识的情况下，仅根据图中给出的信息想要识别特定个体并不容易。不过，这 种方法的缺点也显而易见，图9-11 (b) 与图9-9相比，损失了一定的图结构信息，可 能会导致图数据可用性降低，当可用性降低到一定程度之后，数据可能会变得不可用， 这样即使是合法的数据分析人员，也很难从中挖掘到什么有用的信息了。因此，在图 数据中，如何在隐私保护和数据可用性之间取得平衡，同样是一个非常重要的问题。

合作

*夫妻*

一合作一

一同学片

认识 医患

认识

认识

-姐弟一

(a) (b)

图9-11 节点k-匿名示意

k-度匿名的定义与节点 k-匿名定义有些相 似。不过，节点 k-匿名研究的焦点在节点，而 k-度匿名同时需要考虑节点和边。k-度匿名要求 图中任一节点都至少有k-1 个其他节点与其度 数相同。原始图结构中一般很难保证节点的度 都能够满足 k-度匿名的要求，因此为了保护隐 私，可能会对图结构做一些修改，包括增加伪 节点、增加伪边、删除节点、删除边等操作。 图9-12给出了图9-9所示的2-度匿名版本，其 中图的修改操作采用了增加伪边，虚线所示为 增加的伪边(为了简化讨论，我们去掉了边上 的标签)。我们可以看到，增加了两条伪边之 后，图中共有4个度为4的节点和2个度为2的 节点，满足2-度匿名要求。不过，这种修改的

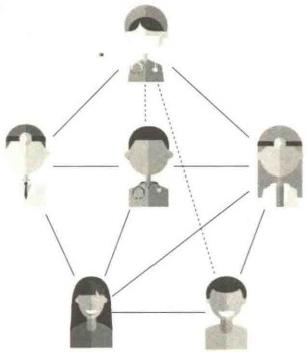
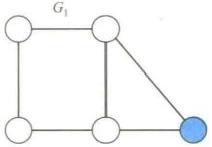
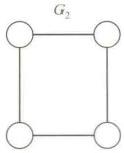


图9-12 k-度匿名示意

缺点也是显而易见的，为了满足2-度匿名要求，我们对图结构做了较大的改变。有一

 188

**第9章** |数据安全与隐私保护

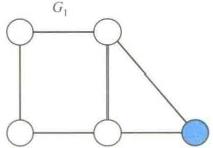
个节点的度甚至从2边成了4,这使得她从一个度较少的节点直接变成了度较大的节 点，改变了她在这个图中的结构性质。此外，对图做增删边的操作还有可能会导致图 的连通性发生改变，这在某些图连通性比较重要的应用场景(如道路交通)中，也会 影响后续数据分析的准确性。

k- 同构将原始图划分为k 个子图，并要求子图之间相互同构(图G, 和 G₂ 是同构的 当且仅当存在一个一一映射f 将 G, 的节点映射到G₂ 的节点，使得： G; 中任意两个节 点 v, 和 v, 邻接当且仅当G₂ 中节点f(v₁) 和f(v,) 邻接)。为了实现 k-同构，可以对 图做增删节点或边的操作，使得图可以被切分为k 个相同大小的同构子图。

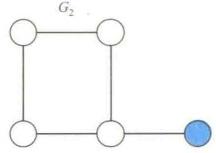
此外，还有k-对 称 图 、k-自同构等图匿名化方式，对于图动态变化的情况，也有许 多研究关注，收篇幅所限，我们在此不做更细致的讨论。

图数据中的差分隐私：差分隐私的通用定义我们在上一节中已经给出，在图数据 中，我们同样可以定义节点差分隐私和边差分隐私。与结构化数据的差分隐私定义类 似，图中的差分隐私也要求当数据集仅有微小改变(通常是个体的出现或缺失)时， 攻击者无法根据算法输出来反推个体隐私信息。在图数据中，数据集的改变对应于图 中节点或者边的改变。节点差分隐私和边差分隐私都要求算法在被视为“邻居” (Neighbors) 的图上有相似的输出。请注意这里的“邻居图”的概念并不是指两个节点 是否邻接，而是用于描述两个图是否足够相似。

在节点差分隐私 (Node Differential Privacy) 中，邻居是指可以通过增删一个节点 及其邻接边而相互转换的两个图。如图9-13 (a) 所示，图G₁ 与 图G₂ 的不同仅在于蓝 色节点及其邻接边，所以它们是邻居。换言之，对于给定的两个图，如果可以通过删 除节点及其邻接边从其中一个图中获得另一个图，则两个图是邻居。节点差分隐私的 目的是保证从原始图中增删单个节点及邻接边时，攻击者无法根据算法输出的不同来 推断个体的隐私信息。



(a) 节点邻居



(b) 边邻居

图9-13 邻居图

在边差分隐私 (Edge Differential Privacy) 中，邻居是指仅在一条边上不同的两个

189

大数据治理：理论与方法…………

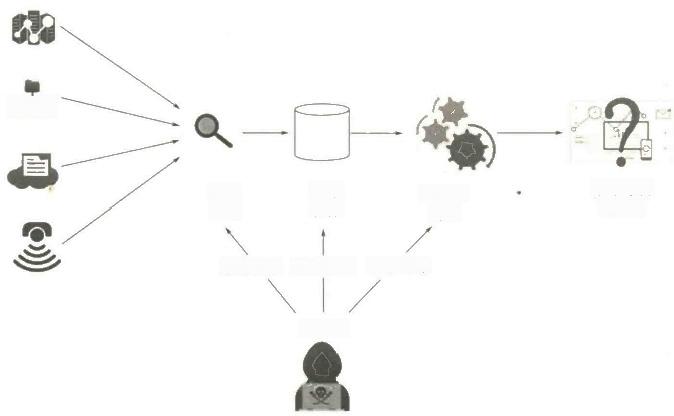
图。如图9-13 (b) 所示，图G, 删除与灰色节点邻接的一条边之后，与图 G₂ 相同，所 以它们是邻居。边差分隐私要求算法在任何一对仅有单条边上不同的图上具有相似的 输出分布，从而保护图中边的变化。边差分隐私对于图的相似程度要求更高，其隐私 保护的力度相对节点差分隐私来说较弱。类似地，还可以定义k 边差分隐私，即边邻 居彼此之间仅有不超过k 条边的不同。

图差分隐私保护的目的是在能较为准确地回答某些查询的同时，不披露个体的隐 私信息。这两个目的在很多场景下是完全矛盾的，因此，如何在“效用”和“隐私” 之间达成平衡，也是图差分隐私研究中需要重点关注的问题。

9.5 **智能化时代的数据安全**

在大数据与人工智能技术井喷式爆发的当下，数据已成为国家战略资源。数据的 重要性不言而喻，高质量数据更是价值不菲。不过，随着数据的价值不断为人们所认 识的同时，针对数据的攻击也逐渐出现。数据投毒就是其中一类很重要的针对数据的 攻击。例如，在推荐系统中，欺诈攻击者往往通过批量注册假用户、添加假评分来误 导推荐系统，从而推举或打压某些商品；在众包服务中，由于数据来自参与众包任务 的普通用户，因此容易通过伪造或控制用户来造成数据投毒攻击；在图片识别任务中， 可以向训练数据中添加某些像素点刻意修改过的图片来使得分类器偏移。这些攻击的 共性在于，都是通过产生并投放恶意数据，从而扰动数据分析模型来达到攻击目的。 我们将这类攻击称为数据投毒 (Data Poisoning) 攻击，攻击过程如图9-14所示。



▲l

数据预

数据 存储

数据 采集

数据分析 与应用

处理

数据投毒

数据投毒 数据投毒

攻击者

图9-14 数据投毒攻击

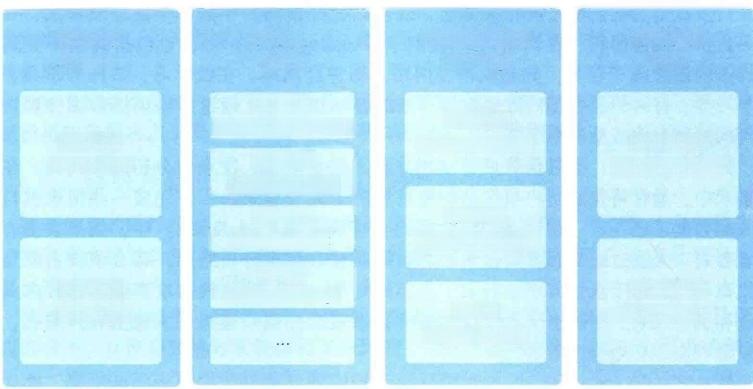
数据投毒会造成许多危害。举例来说，智能汽车的道路识别模型可能是定期训练 而生成的，如果攻击者恶意捏造具有影响力的数据将其注入训练数据中，智能汽车就 可能按照攻击者预想的行为执行一些操作。此外，问答式机器人如微软小冰、小爱同

 190

**第9章** | 数据安全与隐私保护

学等，它们会将与用户之间的对话收纳进自己的语料库并进行学习，如果攻击者出口 成“脏”,那么这些机器人也极有可能被带坏，造成不良影响。有研究表明，在IMDB (Internet Movie Database) 数据集上，只需要添加3%的投毒数据，就可以训练出有偏模 型，将测试的错误率从13%提升到23%。

早期，数据投毒攻击和防御的研究主要面向推荐系统和机器学习算法，近几年， 由于数据来源的进一步丰富，相关研究也逐渐延伸到了其他的数据分析领域。总体说 来，面向推荐系统的欺诈攻击、面向机器学习的训练数据投毒、面向智能电网的假数 据注入等攻击均属于数据投毒攻击。这些攻击的共性特点在于，直接攻击数据集，通 过注入恶意数据来使得数据分析算法或模型输出的准确率下降或按攻击者的意图发生 偏移。数据投毒攻击的攻击手段主要是注入劣质数据或高倾向性数据使得数据驱动的 模型或算法出错或偏斜。。设计具体算法时需要考虑4个要素：攻击场景、攻击目标、 攻击类型和攻击效果，具体如图9-15所示。



|  |  |  |  |
| --- | --- | --- | --- |
| 攻击场景 | 攻击对象 | 攻击所需的知识 | 攻击效果 |
| Scenario | Target  推荐系统 | Knowledge | Goal |
|  | Recommender Systems | 完美知识 |  |
| 封闭世界 |  | Perfect Knowledge | 无目标攻击 |
| Closed World | 机器学习  Machine learning |  | Untargeted Attack |
|  | 众包算法  Crowdsourcing  物联网 | 有限知识  Limited Knowledge |  |
| 开放世界 | Internet of Things |  | 有目标攻击 |
| Open World |  | 零知识  Zero Knowledge | Targeted Attack |

图9- 15 数据投毒攻击的要素

各要素的具体含义如下：

根据攻击场景，可以讲数据投毒攻击分为面向封闭世界和面向开放世界，二者的

区 别 主 要 在 于 数 据 的 来 源 和 语 义 约 束 不 同 。

面向封闭世界：数据的来源和语义都可以被一个已知的封闭集合所限制。例如， 智能电网 (Smart Grid) 中，电力数据采集与监控所涉及的数据主要来源于分布在电网 中的计量装置，其数据语义也可以被反映运行状态的有限集合所涵盖；在一些图片分 类任务中，训练数据的内容和标签都来自已知的封闭集合。

面向开放世界：数据的来源或语义是开放的，无法用封闭集合进行约束。例如， 在群智感知 (Crowdsensing) 中，传感数据来自普通用户的移动设备，因此很难预先约 束数据的来源；在一些自动问答系统中，原始数据可能采集自互联网或普通用户，而 问题和答案的语义也不局限于一个或几个特殊的领域。

我们攻击对象(数据投毒攻击要破坏的应用种类),可以将数据投毒攻击分为针对

**大数据治理：** 理论与方法…………………………

推荐系统、针对机器学习、针对众包算法、针对物联网等。由于不同的应用中采用的 数据分析算法不同，所以投毒攻击要达成的目标和采用的技术手段也各有区别，我们 在本小节的后续部分会举例一些典型的数据投毒攻击。各类别的基本含义如下。

针对推荐系统：通过注入伪造的评分数据来误导推荐系统。推荐系统中的欺诈攻 击 (Shilling Attack) 是一类典型的数据投毒攻击。针对推荐系统的投毒攻击通常是有 目标攻击，其目的是通过批量注入伪造的评分数据来误导推荐系统，使得目标商品在 普通用户的推荐列表中出现或消失。传统的针对推荐系统的数据投毒攻击的类型包括 随机攻击、平均攻击、bandwagon 攻击、平均噪声注入攻击和平均目标位移攻击等。随 着推荐系统技术本身的发展，被攻击的算法也变得多样，具体包括潜在因子模型的推 荐算法、基于近邻的推荐算法，以及最新的跨域推荐 (Cross-domain Recommendation) 算法等。

针对机器学习：通过污染训练数据引发模型倾斜 (Model Skewing)。被攻击的技术 涉及许多重要的机器学习及相关算法，既包括已经使用多年的经典模型或算法，如支 持向量机、线性回归、逻辑回归、自回归 (Autoregressive) 等，也包括近些年受到关 注较多的模型或子领域，如卷积神经网络、图神经网络、在线学习、多任务学习、强 化学习等。针对机器学习的数据投毒攻击的基本思想是，通过污染训练数据使得训练 得到的模型有偏或准确率下降。

针对众包算法：通过伪装成普通用户注入恶意数据，使众包分析结果出错。在众 包技术中，通常将普通用户当作是数据采集的工人 (Worker) 来完成一些很难被自动 完成的数据采集任务。在针对众包服务的女巫攻击 (Sybil Attack) 中，如果攻击者伪 装成普通工人通过提交恶意数据来影响数据质量和数据分析结果，那么该攻击就是数 据投毒攻击。由于众包应用是利用了人类的智能，所以这些投毒攻击通常要考虑人工 管理和成本控制。同时由于众包应用通常会做质量控制来避免工人提交噪声数据，所 以投毒攻击还必须绕过质量控制机制。

针对物联网：通过注入恶意数据误导调控设备或降低数据分析的准确率。物联网 中的假数据注入攻击 (False Data Injection Attack) 是一类数据投毒攻击，目前有不少 研究针对无线传感器网络、智能电网、群智感知 (Crowd Sensimg) 等场景设计了相应 的数据投毒攻击算法。

根据攻击手段，可以将数据投毒攻击分为完美知识攻击、有限知识攻击和零知识 攻击。这三者的区别在于攻击需要的背景知识量不同，各类别基本含义如下。

完美知识( Perfect Knowledge) 攻 击：攻击者具有所有的先验知识，即完全了解全 局信息，包括被攻击的算法和相关参数。在完美知识攻击场景中，攻击者对于被攻击 模型或算法的认知是足够充分的，甚至还能了解环境中其他活动者的相关信息。通常， 在真正的攻防活动中，攻击者很少能有条件发起完美知识攻击，但对于完美知识攻击 的研究仍然很重要，因为这会告诉我们“最坏情况下”攻击者可以做到哪一步。

有限知识 (Limited Knowledge) 攻击：攻击者拥有一定量的局部信息，并且无法基

于已知信息准确推断全局信息。有限知识攻击在实际中相对容易实现，不过攻击者拥 有的知识类别和总量也会影响攻击的效果。同时，知识的获取难度也会影响攻击效果。 通常我们会认为关于被攻击模型的公开信息是相对容易获取到的知识，但是这些知识

**第9章|数据安全与隐私保护**

是否足够支撑发起一次数据投毒攻击要根据具体应用场景来分析。

零知识 ( Zero knowledge) 攻击：攻击者在攻击前不具备任何先验知识。零知识攻 击对于攻击者来说是比较困难的，通常零知识的情况下攻击者只能采用一些较为通用 的攻击手段，如随机攻击、均值攻击等。

根据攻击效果，可以将数据投毒攻击分为无目标攻击和有目标攻击，二者的主要 区别在于对于攻击达成的结果有没有要求，具体定义如下。

无目标攻击 (Untargeted Attack): 攻击目标是使被攻击的模型或算法的输出出错，

但不约束错误的种类和位置。例如，如果被攻击的是投票算法，有 A 、B 、C 三个选 项，正确的投票结果应该是A 。对于无目标攻击来说，通过数据投毒攻击，使得算法 最终输出了投票结果B 或 C 都可算是攻击成功。

有目标攻击 (Targetgd Attack):使被攻击的模型或算法的输出出错，同时对错误 的种类和位置有明确约束。同上，如果被攻击的是投票算法，有A 、B 、C 三个选项， 正确的投票结果应该是A 。对于有目标攻击来说，如果攻击目标是令算法输出结果C, 那么如果攻击之后，算法输出了结果B, 此时攻击虽然导致算法出错，但仍然不能够算 是攻击成功。

在本节的后续部分，我们将以推荐系统和众包服务中的数据投毒攻击为例，说明 数据投毒攻击的攻击手段，并在最后介绍数据投毒攻击的防御方法。

**9.5.1** **推荐系统中的数据投毒攻击**

随着信息的爆炸性增长，人们面对海量信息往往需要工具来帮助自己完成信息过 滤。推荐系统能较为精确的匹配用户的个人兴趣，更有效率地为用户挑选最感兴趣的 商品，进而带来巨大的商业价值。

推荐系统能够精准匹配用户个人兴趣，但整体来说鲁棒性有待增强，容易受到数 据投毒攻击，从而导致推荐系统面临安全风险，误导用户的判断，损害消费者的利益。 具体地，推荐系统的数据都来自普通用户的评分或者行为数据，恶意用户可以伪装成 普通用户在其中为项目(Item) 进行评分，所以，推荐系统很容易被恶意数据所污染 而造成数据投毒攻击，进而影响推荐结果的准确性。数据投毒攻击者基于某些利益因 素会向系统中批量注入恶意数据，目的是扶持或打压某些被推荐的项目，最终扰乱推 荐结果。也就是说，只要注入的数据满足某些条件，攻击者就能够人为控制推荐结果， 让特定的商品有更多(或更少)的机会出现在普通用户的推荐列表中。

欺诈攻击是推荐系统中的一种典型的数据投毒攻击，其采用固定规则生成伪造用 户，向推荐系统中注入伪造用户的伪造的项目评分向量。伪造评分的各类项目按照攻 击者的目的可分为目标项目、选择项目、装填项目，如图9-16所示。

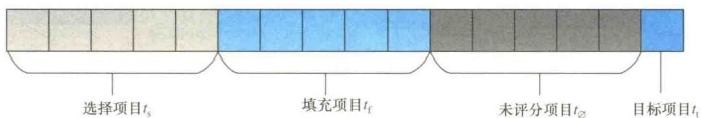


图9-16 伪造评分向量

大数据治理： 理论与方法

其中目标项目t, 是攻击者想要推广或打压的项目，选择项目t, 是攻击者希望让攻 击更有效而筛选出来的项目，装填项目t 是攻击者希望让攻击更难被检测而选择的用 于隐藏自身的项目，未评分项目ta 是伪造评分中评分为空的项目。攻击者首先确定选 择项目和装填项目，接着让批量产生伪造用户，批量伪造用户资料，并对选择项目和 装填项目按照特定的规则进行评分，从而将伪造用户伪装成与被欺骗用户的同好，最 终通过伪造用户的评分来影响推荐系统对被欺骗用户的推荐结果。

最基本的攻击方法是，利用手工规则按照不同的方法选择填充项目为伪造用户设 计评分向量。常见的欺诈攻击如随机攻击、平均攻击、流行性攻击的具体设计模式 如下：

随机攻击 (Random Attack): 选定固定的攻击项目后，选择项目为空，填充项目随

机选择并随机赋值(利用正态分布决定填充项目的评分值)。该方法不需要任何系统知 识，成本较低。

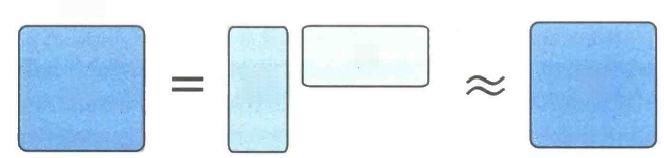
平均攻击 (Average Atack): 选定固定的攻击项目后，选择项目为空，填充项目随

机选择并赋全局评分均值。该方法在攻击前需要的知识是全局评分均值，不过这个均 值在很多推荐系统中会被公布出来，并不难获得。

流行性攻击 (Popular Attack/Bandwagon Attack): 一部分流行的少量的商品会受到

大量的关注，将流行的项目作为选择项目并赋评分最大值，通过这种方式，注入的虚 假用户可以与系统中大量的用户建立相似关系，装填项目随机选择并随机赋值。该方 法在攻击前需要预先知道哪些项目是流行项目，这可能需要一些领域相关的背景知识， 这些知识有时可能需要借助外部知识源才可获得。

上述几种方法是通用方法，即不针对具体的推荐算法，几乎适用于全部推荐场景， 通用方法的优点是易于实现，但在“精准打击”方面力度不够。还有一些更为精准的 攻击手段是针对特定推荐算法的。目前推荐算法可以大致分为基于近邻、基于模型、 基于内容等类型，其中协同过滤 (Collaborative Filtering) 算法是最常见的推荐算法之 一，其通过相似用户和相似商品的历史数据来完成个性化推荐。协同过滤算法通常可 以分为基于近邻的协同过滤和基于模型的协同过滤。基于模型的推荐算法中，基于隐 因子模型 (Latent Factor Model) 的推荐算法在稀疏性和冷启动问题的处理上都有很好 的表现，并且易于扩展和实现，所以较为流行。隐因子模型的基础是对评分矩阵做矩 阵分解，大致原理如图9-17所示。

Items

V

Users M *uT*

M

**图9-17** **矩阵分解的大致原理**

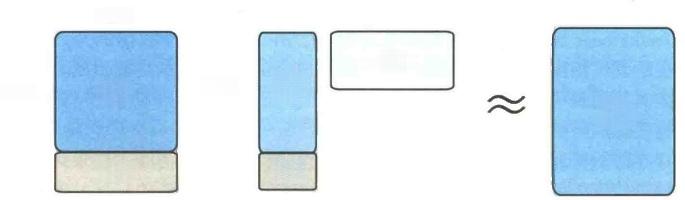
评分矩阵M 的行为用户，列为项目，具体地，每一行表示某个用户对所有项目的 评分，每一列则表示某个项目收到的所有用户的评分。由于用户通常不会评价整个项

 194

**第9章** |数据安全与隐私保护

目集合(如很少有豆瓣用户对豆瓣上列举的每一部电影都做了评分),所以M 中会有 很多空值，第i 行j 列有空值就表示用户i 并没有对项目j 做出评分。我们想要知道用户 i 是否会喜欢项目j, 就是要预测评分矩阵中第i 行j 列的取值。因此，理想情况下，推 荐系统希望能将整个评分矩阵M 中的空值都补全。隐因子模型的大致原理是，对用户 评分矩阵M 分解后，生成用户隐因子矩阵U 和项目隐因子矩阵V, 这两个矩阵可以理 解为用户/项目对具体隐因子的倾向程度，通过截取前k 维矩阵，再合成矩阵后，补全 评分矩阵，完成推荐过程。

要对隐因子模型进行攻击的基本原理如图9-18所示。基本思想是向原本的评分矩 阵M 中注入投毒矩阵M' 、U’, 通过控制最后合成的矩阵M, 使得M。与真实的M 的 均方根误差 (RMSE) 足够大来完成投毒。这个过程可能需要反复训练，具体的训练细 节由于篇幅所限，在本书中不做赘述。



Items

V

二 *uT*

*M。*

*U'*

M

*M'*

Users

**图9-18** **面向矩阵分解数据投毒攻击**

在推荐系统的相关实验中， MovieLens 、Last.fm、亚马逊商品数据集等都是常用的 数据集，这些数据集对于研究推荐系统攻击也同样有用，感兴趣的读者可以自行搜索 这些数据集以及相关的项目。

**9.5.2** **众包环境下的数据投毒攻击**

大数据时代，各行各业都从数据的爆发性增长中受益匪浅，但如何有效增加数据 来源，从而获取大量的数据是一个很重要的问题。各大企业机构都希望在尽可能控制 成本的情况下来高效率的收集信息，其中众包作为能高效收集数据，同时成本低廉的 数据收集模式，受到广泛关注。

通过众包来收集数据，主要是将数据采集、标注、转换等机器较难完成的任务， 通过特定平台外包给普通群众来完成。原则上任何(满足一定条件的)普通用户都可 以接受众包任务，完成任务即可领取酬劳。众包一定程度上扩大了数据来源，使得数 据收集更加容易，但是，通过众包收集到的数据往往不够真实可靠，这是由于发送任 务的发包方在众包环节中，并不了解完成任务的接包方的真实身份，接包方是主动自 愿接包完成任务。这就使得攻击方有机可乘，攻击者可以通过在众包服务中混入恶意 工作者来发起数据投毒攻击，污染数据导致算法模型决策出现偏差，给企业造成不可 估量的损失。

数据投毒针对数据的完整性展开攻击，恶意地向真实数据中混入精心构造的异常 数据，来破坏原有数据的分布，进而导致算法模型决策出现偏差。在实际应用中，无

大数据治理： 理论与方法……………

人驾驶汽车、舆情分析、商业智能应用等需要实时采集数据的学习场景，受到数据质 量的影响极大。尤其是自动驾驶领域，精心训练过的数据投毒攻击，可能导致交通事 故甚至人员伤亡。因此，如何鉴别出众包服务中的恶意工作者，剔除掉中毒数据，成 为防御者研究的关键问题。攻防之间博弈的焦点问题，就在于是否能将中毒数据有效 注入训练数据中。

在很多情况下，由于数据采集设备的误差、用户自身的知识欠缺以及环境动态变 化等原因，普通用户提交的数据并非完全正确，因此当多个用户针对同一个现实世界 实体提交了不同数据时，系统可能会通过一些方法来过滤掉低质量的数据，筛选出正 确的数据。其中真值发现 (Truth Discovery) 作为能从冲突信息中发现真值。具体地， 可以在从工作人员收集数据后运行一个实现真相发现算法的TruthFinder 。TruthFinder 对 每个数据提供者都赋予相应的权重w, 对每条事实也赋予相应的可信度X\*, 通过不断 迭代更新w 和X\*, 来得到一个最贴近真实值 (Ground Truth) 的聚合结果。在这个过 程中， TruthFinder 迭代地计算可能的数据真值，并为每个提交数据的用户分配不同的 权重来表示其可信度。可信度越高的用户会对 TruthFinder 的最终输出的真值产生更大 的影响。真值发现可以在一定程度上缓解不可靠工人的脏数据带来的问题，因此可以 用作数据投毒的防御。

当存在防御机制时，系统可能会首先判断众包工人的可信度，在最终的数据整理 时剔除掉可信度较低的工人的数据。不过，即使存在真值发现这样的机制，恶意投毒 者仍然能够钻机制的漏洞，来完成数据投毒攻击。然而，最近的一些研究工作表明， 类似于TruthFinder 这样的工具也可能被精心策划的数据投毒攻击所危害。如果攻击者 可以获取正常和恶意工人的数据的整体分布、TruthFinder 的基本设置以及其他攻击是 否会成功的先验知识，攻击者就可以利用这些知识来干扰 TruthFinder 的最终结果。

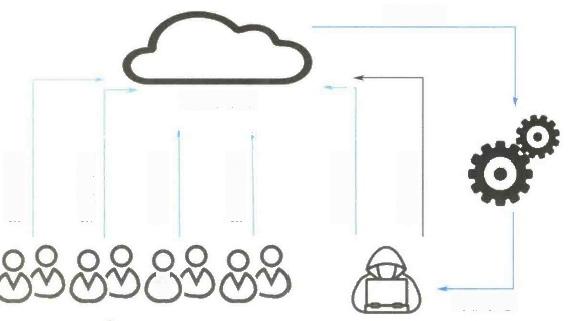
攻击者能够获取的知识量对攻击成功率和攻击收益影响很大，当攻击者具备完 美知识时，攻击者可以绕过系统的防御机制，伪装成正常工人骗取系统信任，甚至 可以找到收益最高的攻击策略，但如果攻击者是零知识状态，则可能因为盲目攻击 而失去系统的信任，提交的数据直接被丢弃，从而毫无收益。针对众包算法的投毒 攻击大部分攻击是有限知识攻击，因为攻击者通常是作为普通数据提供者和参与众 包，进而提交恶意数据，因此很难假设攻击者能够预先获知全局信息。也存在少量 完美知识攻击方面的研究，其主要关注最坏情况下的数据投毒攻击的威胁程度，旨 在设计完美知识攻击，并尝试给出完美知识场景下的最优攻击策略，这类攻击从攻 击场景假设上来讲属于封闭世界假设。在更多情况下，针对众包算法的投毒攻击基

本都采用开放世界假设，攻击场景中除了攻击者和防御者之外，还存在着大量的活 动第三方，即其他数据采集者，这些数据采集者的来源、背景知识、数据质量往往 难以预先控制。

具体地，众包环境下数据投毒攻击思路如图9-19所示。攻击者收集信息，评估自 己当前的状态，判断自己的攻击成功的希望有多大，如果攻击目标成功希望不大，就 让恶意工作者提交正常工人倾向的数据，以此来将自己伪装成高质量的数据提供者从 而骗取系统的信任，以便在后续的数据提交中发起更强的攻击；如果攻击成功的希望 很高，那么就发起猛烈攻击，以期在一轮攻击中获取较高的攻击收益。

196

**第9章** **|数据安全与隐私保护**



攻击数据

正常数据

正常数据

正常数据

正常数据

正常数据

数据采集器

评估状态

收集信息

V

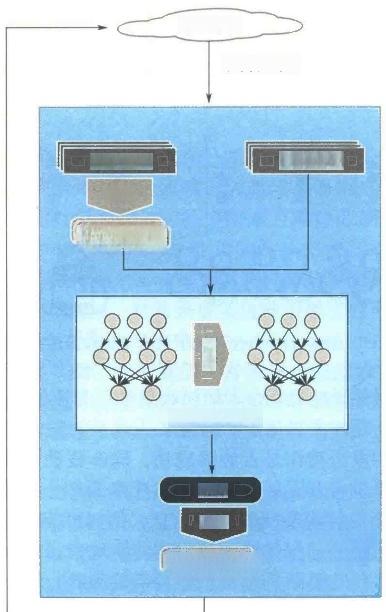
图9-19 众包环境下数据投毒攻击思路

在上述思路中，最重要的是攻击者如何收集信息和评估状态。在许多实际应用中， 攻击者可能难以获得所需的全局信息。恶意工人几乎无法访问其他正常工人提供的数 据，并且不清楚当前的攻击动作是否能够成功，除非该动作被实际执行。攻击者的权 限往往与普通的数据提供者并无区别，只能够观察到部分环境。更具体地说，攻击者 能够获取的信息是本地能够观察到的局部信息，例如可能的真值、TruthFinder 公布的 最终结果以及他们自己产生的假数据。基于这些本地信息，攻击者通常无法预先确定 最佳攻击策略。 一种可行的思路是恶意工作者需要先执行攻击，看看它是否有效，然 后根据执行的攻击的影响完成策略学习，进化他们的攻击策略。例如，图9-20 展示了 一种可能的攻击思路。在图9-20中，攻击者从环境(众包平台)中获得局部信息，例 如对环境的观测结果(众包平台发布的信息、任务的真实数据值等)或攻击奖励(攻 击后结果的偏移量),基于这些观测结果，攻击者可以评估自己当前的状态，推测目前 众包平台是是否信任自己。该攻击的核心是一个深度强化学习模块，攻击者通过持续 不断的发起攻击和收集信息，将这些信息作为训练数据来训练一个动作产生器，该动 作产生器的作用是根据状态估计结果和过往的攻击数据来决策：在当前的估计状态下， 攻击者应当提交偏离真值的虚假数据来发起攻击，还是提交真实数据来隐藏自己骗取 平台信任。根据动作产生器的输出，攻击者产生要提交的数据，完成攻击或隐藏动作。 基于这种思路，攻击者通过发起多轮迭代来优化自己的攻击策略，攻击可能一开始并 不能获得较高收益(收益值未必会高于随机攻击),但随着攻击轮次的增加，攻击者会 在多次动作中学习，逐渐理解在何种状态下应当采取什么样的动作。随着时间的推移， 动作产生器的决策会趋于完善，攻击者收益也会不断提高。

从攻击者的角度来看，学习的目的是优化攻击策略。优化攻击策略的焦点是如何 在平台的防御机制前伪装自己，使防御机制不会对恶意工作者实施惩罚。因此，学习 的主要目标包括估计平台防御机制的算法参数、了解防御机制的惩罚机制以及探测周 围环境参数等。如果能够学习到足够的知识，就可以避免踩到雷区，在攻击成功的可 能性较低时，采用伪装的方式，提交与大多数正常数据提供者一致的数据，来规避防 御机制的惩罚，甚至获得防御机制的奖励(提高可信度权重)。

大数据治理： 理论与方法

发 起 攻 击 或 隐 藏 自 己



众包平台

观测和奖励

观测结果

估计

状态估计结果

深度强化学习

动作

产生

恶意数据 产生者

癌意攻击数据

经验回放

更 新

图9-20 基于深度强化学习的数据投毒攻击

从防御者的角度来看，针对发现真值的研究，焦点是如何对众包中的数据提供者 的可靠性进行评估。可靠性高的数据提供者提供的数据会在后续的数据聚合中被赋予 更高的权重，而可靠性低的数据提供者的数据会被置予较低的权重，在后续的数据聚 合中减少使用甚至不予考虑。这种方式，可以使攻击者尽可能地暴露在防御机制面前， 当数据提供者总是提供与大多数意见不同的数据时，防御机制会降低该工作者的权重， 使其更难发起进攻。而对于经常提供优质数据的数据提供者，则会赋予较高的权重进 行鼓励，使其能更多地参与到聚合过程中。

**9.5.3** **数据投毒攻击的防御**

数据投毒主要是指攻击者在训练数据中加入精心构造的异常数据，破坏原有的训 练数据的概率分布，导致模型在某些条件会产生分类或聚类错误。而数据投毒防御则 是采取技术手段检测和清洗异常数据，或寻找其他缓解数据投毒对模型的影响的方法。

数据投毒攻击的防御既可以在数据采集时进行，也可以在数据采集后进行。数据 采集时的防御主要是通过数据源的行为特征来评估其可信度，从而过滤掉潜在的恶意 数据源；数据采集后防御则是通过分析收集到的全部数据，检查是否存在采集时未被 发现的恶意数据。之所以需要采集后防御，是因为可能会存在清洁标签攻击 (Clean-

 198

**第9章** |数据安全与隐私保护

Label Attack),即数据单独检查完全无害，但放入特定数据集中则会误导模型。

在数据采集期间，较为通用的数据投毒防御机制是数据源的可信度评估。目前这 类方法大致可以分为两类。

基于行为的数据源可信度评估：这类防御机制在推荐系统中使用最多，通过用户 的评分行为来评估其是否是潜在的恶意用户，其中有监督的方法需要基于标记良好的 训练数据集来训练数据源可信度评估模型；无监督的方法一定程度上可以克服有监督 方法数据获取困难的缺点。这些方法不需要标记良好的训练数据，主要基于用户产生 的数据的相似性来完成聚类以发现恶意数据攻击。

基于数据依赖的数据源可信度评估： 一些研究基于社交关系或信任度来完成恶意 群体检测，在社交关系或数据源相关性未知的情况下， 一些方法可以主动地发现数据 源间的潜在依赖，并基于潜在依赖对数据源信度进行评估。

数据采集完成之后，可以从整个数据集出发建立防御机制。在无法完全确定数据 来源的场景下，可以尝试对整个数据集合的质量、特征或语义进行分析以发现潜在的 投毒攻击。简单的数据质量评估手段是校验数据是否完整合法，如果数据不合法，则 无论其是否是恶意攻击数据，都应丢弃。此外，数据质量评估模型的研究目前已经比 较成体系，我们在第6章已经介绍过。这些研究虽然主要针对数据集合中的偶发劣质 数据，但也可以为投毒数据的检测模型构建提供一些思路。例如，有计划的多步攻击 的数据可能存在异常时序依赖，可被时效性分析检测；批量产生的投毒数据可能共同 违背相似的数据一致性约束，可被一致性分析捕获等。

当前，数据投毒的防御还在积极的研究阶段，很多防御方法还停留在思路或者实 验阶段。目前针对机器学习的数据投毒防御研究较多，防御基本归属于采集后防御， 防御的目的是从训练数据集合中剔除掉投毒攻击者投放的恶意数据。其中最直接的防 御手段是数据清洗，在训练数据前通过一些技术手段筛选训练集中潜在的有害数据并 进行修复或者移除。主要有以下两类方法：

数据过滤 (Data Filtering): 这类技术主要考虑对训练数据集的研究，通过检测和 净化的方法防止有害数据对模型的影响。具体方法包括：

(1)依据数据的标签特性制定相应的规则，来寻找潜在的投毒攻击注入的数据点， 在训练模型时将这些点进行过滤。

(2)采用模型对比的过滤方法，即将新训练的模型与前一个模型进行比较来估计 发生的变化，并过滤数据对抗投毒攻击。

回归分析 (Regression Analysis): 这类技术主要利用统计学的方法检测数据集中存 在的噪声和异常值。具体方法包括对模型定义不同的损失函数来检查异常值，以及使 用数据的分布特性来进行检测。

此外，提高模型鲁棒性也是抵御数据投毒攻击的一种可行的方法。如基于集成分 析的防御方法，该类技术强调使用多个子模型的综合结果提升机器学习系统对投毒攻 击的对抗能力，这些模型是相互独立的，它们采用不同的训练集进行训练，这使得整 个系统被投毒攻击的可能性与所造成的影响都大幅降低。

**反侵权盗版声明**

电子工业出版社依法对本作品享有专有出版权。任何未经权利人书面许 可，复制、销售或通过信息网络传播本作品的行为；歪曲、篡改、剽窃本作 品的行为，均违反《中华人民共和国著作权法》,其行为人应承担相应的民 事责任和行政责任，构成犯罪的，将被依法追究刑事责任。

为了维护市场秩序，保护权利人的合法权益，我社将依法查处和打击侵 权盗版的单位和个人。欢迎社会各界人士积极举报侵权盗版行为，本社将奖 励举报有功人员，并保证举报人的信息不被泄露。

举报电话：(010)88254396;(010)88258888

传 真：(010)88254397

E-mail: dbqq@phei.com.cn

通信地址：北京市万寿路173信箱

电子工业出版社总编办公室

邮 编：100036

大数据治理

理论与方法

本书首先对大数据治理的背景和基本概念进行简要介绍，尝试为读者提供

对大数据治理的基础认识；然后从政策、管理和技术等多个方面对大数据治理 相关的概念和方法加以介绍，对数据架构管理、元数据管理、主数据管理、大数

据集成、数据质量管理、数据的标准化、数据资产化、数据安全与隐私保护等进

行深入探讨，以期为读者提供一个比较全面的大数据治理的场景。

◎内容丰富翔实，对大数据治理进行全方位的解构。

@帮助读者理解大数据治理的框架、流程、技术和典型应用。

◎精心设计图表、案例，以实用性为导向，提供了必要的理论和实践指导。

责任编辑：刘璃

封面设计： DflC 期 腮文化

**定价：58.00元**