

# 四川 大学

## 硕 士 学 位 论 文

题 目 不确定数据聚类关键技术研究

作 者 曾春秋 完 成 日 期 2009 年 4 月 15 日

培 养 单 位 四 川 大 学

指 导 教 师 唐常杰教授

专 业 计算机应用

研 究 方 向 数据库与知识工程

授予学位日期 2009 年 6 月 30 日

# 不确定数据聚类关键技术研究

计算机应用专业

研究生 曾春秋      指导老师 唐常杰教授

**摘 要** 聚类算法是数据挖掘中的重要任务。数据的观测和收集过程存在不可避免的缺失性，非完整性，模糊性，不精确和时效性。这些特性导致了数据与真实世界的偏离，使得这些不确定数据的挖掘结果可信度不高。不确定数据的挖掘是目前研究界和产业界关注的热点和焦点之一。本文建立了不确定数据的聚类数学模型，拓广了现有的聚类算法 K-Means 和 K-Median，提出了新的关于不确定数据的聚类的系列关键技术。主要包括：

- (1) 阐述了现实世界中的不确定数据的普遍性，并对不确定数据进行了形式描述，提出了用不确定对象的概念，用它对不确定数据数学建模。分析论证了不确定对象模型表示现实世界不确定数据的有效性。
- (2) 在不确定对象模型上，分析了不确定数据聚类相对于确定数据的挑战。根据不确定数据聚类的两种方式，将其分为未指定聚类和指定聚类。在未指定聚类中提出了 UA-UK-Median，UA-UK-Means 算法，在指定聚类中提出了 A-UK-Median 和 A-UK-Means 算法。
- (3) 在指定聚类中，对算法 A-UK-Median 进行了效率分析，并提出具有剪枝技术的算法 PA-UK-Median，改进了算法 A-UK-Median。
- (4) 在指定聚类中，分析了算法 A-UK-Means 的效率瓶颈，提出了改进算法 MA-UK-Means。
- (5) 做了详实的实验，验证了本文所提出的基于不确定对象模型上的聚类算法、剪枝算法 PA-UK-Median 和改进的 MA-UK-Means 算法的有效性，实验还表明，新方法提高了不确定数据分析的准确性。

**关键词** 数据挖掘，不确定数据，未指定聚类，指定聚类。

# The Research on the Key Techniques of Clustering over Uncertain Data

Computer Application

**Graduate** Chunqiu Zeng

**Advisor** Professor Changjie Tang

**Abstract** Clustering is an important data mining task. During the course of observing and collecting data, there always involves a lot of data that may be missing, incomplete, vague, imprecise and outdated. These characteristics tend to cause the inconsistency between the data and real world, and hence the mining results may be inconsistent with the real world. Presently, data mining over uncertain data become one of hot spots and focuses concerned in both academic and industrial area. This thesis constructs mathematic model about the uncertain data and proposes a series of key techniques in clustering over uncertain data by extending the traditional clustering algorithm such as K-Median and K-Means. The main contributions of this thesis include:

- (1) Introduces the permeation of uncertain data in the real world. Proposes the uncertain data object model to describe the uncertain data in mathematic form. Analyzes the effectiveness and the efficiency of the uncertain data object model for expressing the uncertain data.
- (2) Describes the two ways of clustering uncertain data based on the uncertain data model, including unassigned and assigned clustering. With the knowledge of K-Median and K-Means algorithms over certain data, proposes UA-UK-Median and UA-UK-Means algorithms in unassigned clustering and A-UK-Median and A-UK-Means algorithms in assigned clustering.
- (3) Proposes PA-UK-Median algorithm with pruning technique by analyzing the A-UK-Median algorithm in assigned clustering, which improves the efficiency of

A-UK-Median algorithm.

- (4) Proposes improved MA-UK-Means algorithm by analyzing the bottleneck in time efficiency of A-UK-Means algorithm in assigned clustering setting.
- (5) Conducts extensive experiments to show that, based on the newly proposed model, the uncertain data clustering algorithms improve the accuracy of the analysis in uncertain data settings greatly. Also, experiments verify the efficiency of both improved PA-UK-Median and MA-UK-Means algorithms prominently.

**Keywords** data mining, uncertain data, unassigned clustering, assigned clustering

# 目 录

|          |                     |    |
|----------|---------------------|----|
| <b>1</b> | 引言.....             | 1  |
| 1.1      | 研究背景.....           | 1  |
| 1.2      | 本文的工作.....          | 2  |
| <b>2</b> | 基本概念和术语.....        | 3  |
| 2.1      | 数据挖掘的相关概念和技术.....   | 3  |
| 2.1.1    | 数据挖掘的产生.....        | 3  |
| 2.1.2    | 数据挖掘的基本步骤.....      | 3  |
| 2.1.3    | 数据挖掘功能.....         | 4  |
| 2.2      | 数据的不确定性.....        | 6  |
| 2.3      | 不确定数据的描述.....       | 9  |
| 2.4      | 不确定数据分析.....        | 12 |
| 2.4.1    | 不确定数据源.....         | 12 |
| 2.4.2    | 不确定数据库.....         | 12 |
| 2.4.3    | 不确定数据查询.....        | 13 |
| 2.4.4    | 不确定数据挖掘.....        | 14 |
| 2.4.5    | 不确定数据聚类.....        | 15 |
| 2.5      | 小结.....             | 16 |
| <b>3</b> | 不确定数据建模.....        | 17 |
| 3.1      | 不确定数据形式化建模.....     | 17 |
| 3.2      | 不确定模型分析.....        | 18 |
| 3.3      | 不确定模型语义.....        | 20 |
| 3.4      | 小结.....             | 22 |
| <b>4</b> | 不确定数据的聚类.....       | 23 |
| 4.1      | 不确定数据对象聚类的特殊难点..... | 23 |
| 4.2      | 不确定数据聚类问题描述.....    | 24 |
| 4.3      | 不确定数据的聚类算法.....     | 26 |
| 4.3.1    | 确定数据的划分算法描述.....    | 26 |
| 4.3.2    | 不确定数据的划分算法.....     | 27 |

|       |                             |    |
|-------|-----------------------------|----|
| 4.4   | PA-UK-Median 剪枝改进算法.....    | 35 |
| 4.5   | MA-UK-Means 改进算法 .....      | 41 |
| 4.6   | 小结.....                     | 46 |
| 5     | 实验和性能分析.....                | 47 |
| 5.1   | 实验环境.....                   | 47 |
| 5.2   | 实验数据.....                   | 47 |
| 5.3   | 不确定数据聚类准确性实验 .....          | 48 |
| 5.3.1 | numUP 对准确性的影响.....          | 49 |
| 5.3.2 | 聚类 K 值对准确度的影响 .....         | 52 |
| 5.4   | 剪枝算法 PA-UK-Median 实验分析..... | 55 |
| 5.5   | 改进算法 MA-UK- Means 实验分析..... | 57 |
| 5.6   | 小结.....                     | 59 |
| 6     | 结论及未来工作.....                | 60 |
|       | 参考文献.....                   | 61 |
|       | 本文作者在攻读硕士学位期间发表的文章 .....    | 64 |
|       | 声明.....                     | 65 |
|       | 致谢.....                     | 66 |

# 1 引言

## 1.1 研究背景

科技和经济的发展极大的提高了人们获取数据的能力,使得数据爆炸式的剧增<sup>[2]</sup>。海量数据的剧增,已经超出了人们对数据的理解能力。数据挖掘作为对数据库研究的一个重要分支,是人们理解和挖掘出海量数据中隐藏的有用知识的有效手段<sup>[3]</sup>。

数据是反映现实世界的一种手段和载体,来源于人们对现实世界的理解。由于人们对现实世界的理解和摄取数据的能力有限,数据当中总存在缺失,模糊,陈旧,不精确,不一致,模棱两可的现象,导致数据固有的不确定性存在。传统的成熟数据库虽然也有对不确定性数据的处理能力,但只停留在对缺失数据的处理和支持少量的模糊查询。对于不确定数据库的研究,在近二十年来局限于理论的研究。近年来,由于对不确定数据处理的需求越来越大,人们开始了对不确定数据库的研究,较著名的开源的研究不确定数据库的项目有 Stanford 大学研究的 Trio, Purdue 大学的 Orion, Washington 大学的 MystiQ 以及 Cornell 大学的 MayBMS 等<sup>[11]</sup>。

另一方面,数据挖掘用于从浩如烟海的数据海洋中挖掘出有用的隐藏知识,数据挖掘为人们理解数据提供了强大的工具。不确定数据的存在造成数据挖掘过程的困难,使数据挖掘的结果的可信度受到了影响。由此提出了数据挖掘的新的研究方向<sup>[11][4][6][7]</sup>,即对不确定数据有效挖掘的研究。

传统的确定数据挖掘技术中,有简单的处理缺失数据的机制。一般是通过一些数据挖掘方法对数据进行填充,把缺失数据集转换为确定数据,然后在转换后的数据集上进行数据挖掘处理。这种方式并没有从根本上对缺失数据进行建模,然后基于建模的不确定数据,把数据挖掘的方法直接运用于处理不确定数据模型上。近年来,对不确定性数据的关注,大多在于对不确定数据的存储和管理上,包括不确定数据的查询<sup>[12][13]</sup>,不确定数据的连接<sup>[19]</sup>,不确定数据的聚集运算<sup>[20]</sup>以及不确定数据的 skylines 的查询上<sup>[14]</sup>。基于不确定数据的挖掘变得越来越必要了。然而把数据挖掘的方法,包括关联挖掘,分类挖掘,聚类挖掘,离群点检测等,运用于不确定数据也越来越受到学术界和工业界的关注。

## 1.2 本文的工作

本文首先介绍数据挖掘的相关概念和技术,然后描述了不确定数据的普遍存在性以及不确定数据的进行数据挖掘的必要性。对于不确定数据的挖掘处理,本文重点对不确定数据上的划分聚类算法进行了研究和实现。主要工作包括:

- (1) 利用不确定对象概念对不确定数据进行了建模。利用形式化的数学语言定义了不确定数据的相关概念。为在不确定数据上进行挖掘算法分析提供了良好的形式化基础。
- (2) 提出了基于不确定数据上的划分聚类算法。并根据不确定聚类算法的分类,在未指定聚类情形提出了 UA-UK-Median 和 UA-UK-Means 算法;在指定聚类算法中提出了 A-UK-Median 和 A-UK-Means 算法。
- (3) 由于在指定聚类算法中相似度计算的复杂性,为 A-UK-Median 提出了基于剪枝技术的改进算法 PA-UK-Median,并分析和比较了 PA-UK-Median 算法和 A-UK-Median 的效率复杂度。
- (4) 分析了算法 A-UK-Means 的效率瓶颈,提出了 MA-UK-Means 算法,有效的避免了大量的不确定数据间相似度计算。提高了 A-UK-Means 算法的效率。
- (5) 做了详实的实验,分析了本文提出的不确定数据聚类算法对不确定数据进行聚类的结果,并比较了 K-Median 和 K-Means 算法在不确定数据上的聚类结果。实验表明不确定聚类算法在准确度方面有极大的改善。利用大量合成实验,分析了改进算法 PA-UK-Median 和 MA-UK-Means 的效率的提高情况。



## 2 基本概念和术语

### 2.1 数据挖掘的相关概念和技术

#### 2.1.1 数据挖掘的产生

科技进步和信息技术的发展促进了数据库技术的产生。数据库功能由简单走向复杂，其容量和管理数据能力得到了有效的提高。由此数据库的内容也变得浩如烟海。

数据容量的几何级数的陡增和内容的极大丰富，人们对数据的理解和充分应用能力，在面对海量数据时，已显得苍白而渺小。大量未被理解的数据造成了“数据丰富但信息贫乏”的局面。再加上数据不可避免的缺失，模糊，陈旧，不精确，不一致，模棱两可等现象构成了数据不确定性的存在，使人们理解数据的困难更是显著加剧。

为了提高对数据的理解能力，缩短数据与信息的鸿沟，数据挖掘便是作为一种从大量的、不完全的、有噪声的、模糊的、随机的数据中，提取隐含在其中的、人们事先不知道的、但又是潜在有用的信息的工具而应运产生。数据挖掘旨在从海量数据中发现感兴趣的数据模式，为商务决策、知识库、科学和医学研究做出了巨大贡献。

#### 2.1.2 数据挖掘的基本步骤

数据挖掘也被视为数据中的知识发现(或 KDD)的同义词。故而数据挖掘过程实际是一个知识发现的过程。如图 2.1，可归结为如下步骤：

- 1) 数据清理：消除噪声和不一致数据；
- 2) 数据集成：多种数据源可以组合在一起；
- 3) 数据选择：从数据库中提取与分析任务相关的数据；
- 4) 数据变换：把数据统一成适合数据挖掘的形式，如可以通过汇总或聚集的操作；
- 5) 数据挖掘：核心步骤，使用各种智能的方法提取感兴趣的数据模式；
- 6) 模式评估：根据某种兴趣度度量，识别表示知识的真正有趣的模式；
- 7) 知识表示：使用可视化和知识表示技术，向用户呈现挖掘出的可理解的有用知识。

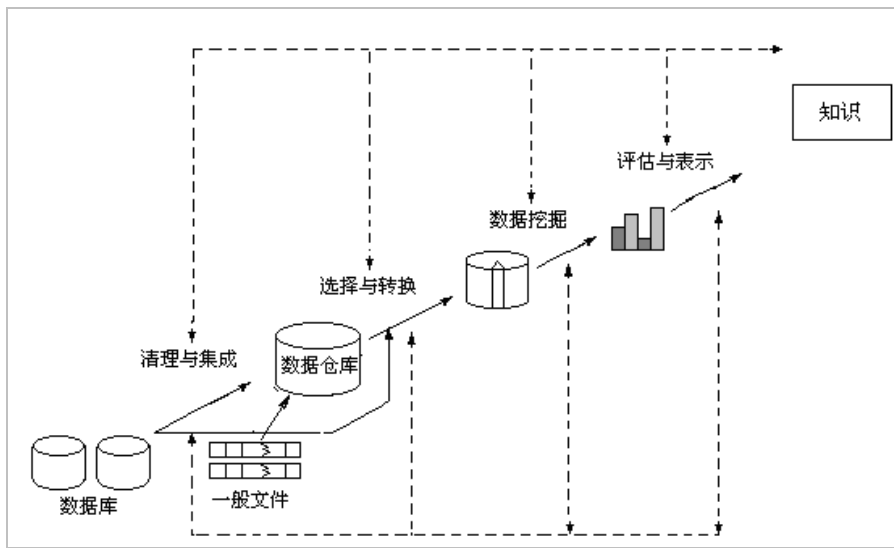


图 2.1 数据挖掘过程

### 2.1.3 数据挖掘功能

数据挖掘功能用于指定数据挖掘任务要找的模式类型。一般而言，数据挖掘任务可以描述为两类：描述和预测。描述性挖掘任务描述了数据库中数据的一般性质。预测性挖掘任务根据已有的知识对当前的数据进行推断，以做出有效的，准确率较高的预测。

- (1) 概念/类描述：特征化和区分。数据可以与类或概念相关联。这种类或概念称为类/概念描述。这种描述可以通过下述方法得到：数据特征化、数据区分、数据特征化和比较。
- (2) 挖掘频繁模式，关联和相关。在数据中提取出频繁出现在事务数据中的数据项集，频繁的序列，频繁的子结构等模式。并利用支持度，置信度来刻画相关的模式和规则。以达到分析数据项间的相关性分析。关联挖掘通常又可根据数据类型不同，可以分为基于分类数据的传统关联，也可基于连续量化数据的量化关联规则<sup>[10]</sup>。
- (3) 分类和预测。分类和预测是这样的过程，它找描述或识别数据类或概念的模型（或函数），以便能够使用模型预测未知对象的类标号或数值。导出模型是基于对训练数据集（即，其类标号或数据值已知的数据对象）

的分析。分类和预测是有监督的学习方法。

- (4) 聚类分析。聚类分析是无监督的学习方法。在分析数据对象时不考虑数据对象已有的类标。对象根据最大化内部的相似性，最小化类之间的相似度的原则进行聚类或分组。也就是说，对象的簇是这样形成的，相比之下，在一个簇下的对象具有很高的相似性，而与其他簇中的对象极大的不同。所形成的每一个簇形成一个对象类，由它可以导出规则，聚类也便于分类法组织形成，将观测组织成类分层结构，把类似的事件组织在一起。
- (5) 离群点分析。数据库中可能包含一些对象，它们与数据的一般行为或模型不一致，这些对象称为离群点。图 2.2 是离群点检测的一个例子，被圈住的点被鉴定为离群点。大部分把离群点视为噪声或异常而丢弃。然而，在一些应用中，如银行的欺骗检测，罕见的事件可能比正常出现的事件更令人感兴趣。故此，离群点的分析被称为离群点的挖掘。

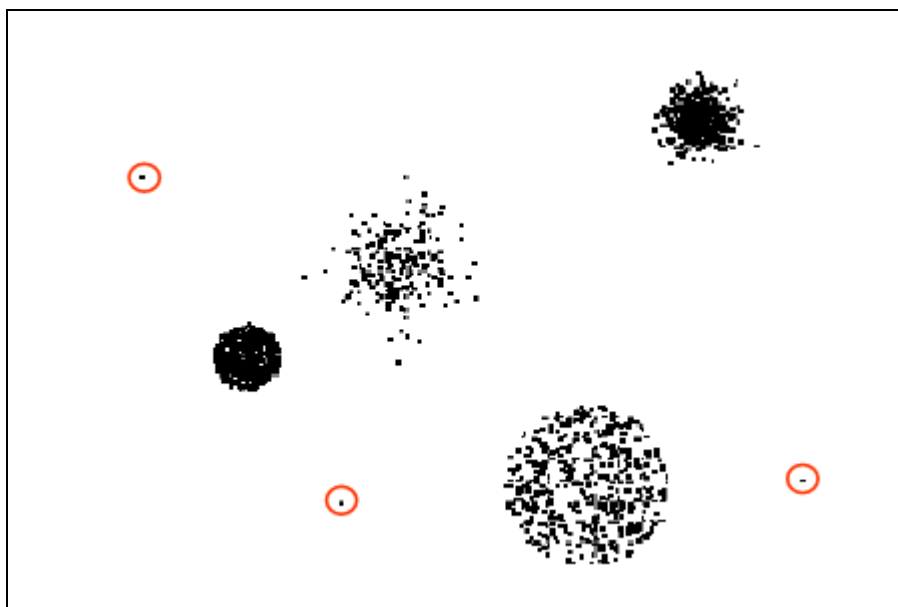


图 2.2 离群点检测

- (6) 演变分析。数据演变分析描述行为随时间变化的对象的规律或趋势，并对其建模。这类分析的不同其他的特点在于需要包括时间序列数据的分

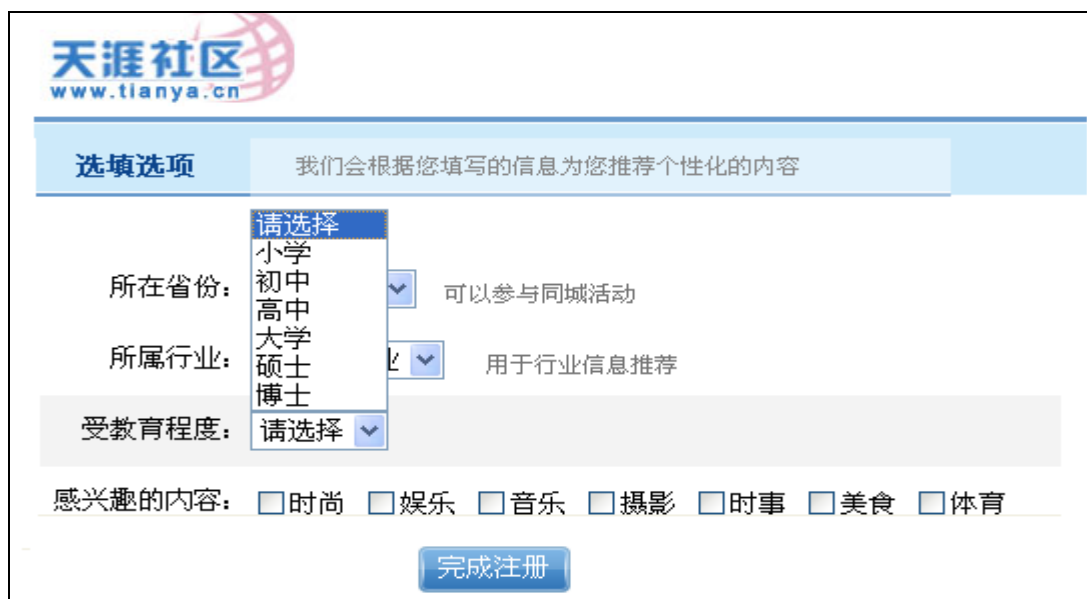
析、序列或周期模式匹配和基于相似性的数据分析。常见应用有数据流的挖掘分析。

## 2.2 数据的不确定性

近四十年来，传统的确定性数据管理技术得到了极大的发展，造就了一个几百亿的数据库产业。数据库技术和系统已经成为建设信息化社会基础设施的重要支撑。在传统数据库的应用中，数据的存在性和精确性均确凿无疑。近年来，随着技术的进步和人们对数据采集和处理技术理解的不断深入，不确定性数据得到广泛的重视。在许多现实的应用中，例如：经济、军事、物流、金融、电信等领域，数据的不确定性普遍存在，不确定性数据扮演着关键角色。传统的数据管理技术和分析方法却无法有效管理和分析不确定性数据，这就引发了学术界和工业界对研发新型的不确定性数据管理技术的兴趣。

数据是用于表达和描述真实的现实世界的一种载体。人们根据自己的知识描述和理解现实世界，并用数据进行表达。这些数据是对现实世界的一个抽象。现实世界是确定的，但由于人们有限的理解和解析现实世界，这些被获取的反映现实世界的数据库的不确定性却是固有存在的。数据库的不确定性主要体现在：数据的缺失性，模糊性，陈旧性，不精确，不一致，模棱两可的现象等。

数据的缺失性主要是因为对当前观察的变量没有数据值与其对应。这种情况通常由于人们获取数据的能力不够或随着人们知识的丰富对同一个个体的认识不同，在整合当前数据和历史数据时通常出现缺失。例如：在医院对出生缺陷病例的特征描述数据中，随着三鹿奶粉事件的发生，当前对病例的调查可能需要多增加一项孕妇是否在怀孕期间食用三鹿奶粉，而过去却没有对这项指标进行调查。所以在集成历史数据时，该项可能表示为缺失的。再如图 2.3 所示，为注册天涯社区时所需填写的注册信息片段。这些信息都是可选填的。对于想立即简单获取一个帐号的社区爱好者而言，有可能根本就不理会这些信息。或者任意处理一下，比如在选择教育程度的下拉列表中，可能直接选择第一个选项，就匆忙结束。这样缺失的数据或虚假的信息随即产生，而不能准确的描述信息。



天涯社区  
www.tianya.cn

**选填选项** 我们会根据您填写的信息为您推荐个性化的内容

所在省份:  请选择

所属行业:  小学  初中  高中  大学  硕士  博士

受教育程度:  请选择

可以参与同城活动

用于行业信息推荐

感兴趣的内容: ☐ 时尚 ☐ 娱乐 ☐ 音乐 ☐ 摄影 ☐ 时事 ☐ 美食 ☐ 体育

**完成注册**

图 2.3 天涯注册信息片段

观测数据可能具有模糊性。从现实世界可获得的数据不够具体，只能得到一个模糊的概念。例如，可能对一个人的描述：“他比较年轻”。这里比较年轻的标准不确定，对于到底是多少岁，无法得出一个具体的值。

观测数据可能具有陈旧性。对于变化比较大的考察对象，由于监控对象的能力有限，往往只能进行定期的采样来初步反映对象的现有状态。例如在考察数据流时，由于数据量比较大，无法存储所有的数据，只能通过采样的方法，然而采样的结果总存在不够新，而只能在一定程度上反映数据流当前的状态。再如对股市的分析过程中，专家一般利用股市过去最近一段时间的股市行情来分析股市未来的发展趋势。这种通过过去的的数据来预测未来的数据都会构成对未来结果预测的不确定性。

观测数据可能不精确。由于设备获取和操作数据总存在误差，也可能由于数据的变化比较大。导致得到的数据是一个范围，而不是一个精确的值。如天气预报当天的温度在 20 到 23 摄氏度。在如 GPS 对运动物体进行定位时，往往给出该物体在以某个位置为中心，再以某个距离为半径的圆形区域范围内。

观测数据可能出现不一致。对某个数据项的考察，存在多个结论，但这些结论却不能同时存在。如描述一个人的年龄在 37 到 45 之间和年龄为 35 岁。这两个值不能同时出现。只能有一个是正确的。

观测数据可能含有歧义。这种情况主要出现在因为没有能够获得足够完整的语义信息，而对同一个数据有多中理解。例如对于姓名拼音“Wei Wang”，中文对应很多种姓名，如：王伟，王薇，王维，王蔚，汪卫，汪玮，汪威，汪巍等。图 2.4，如果需要通过 Google 搜索引擎查找某一个“Wei Wang”，将会出现很多搜索与之匹配的结果，仅仅查看 DBLP 就有 7 个结果。可见如果仅通过姓名的拼音进行查找，将会出现许多干扰项，无法准确定位到查找目标。

The screenshot shows a Google search result for "Wei Wang". On the left, there is a blue box with the text "Wei Wang" and "University of North Carolina at Chapel Hill". Below this, it says "List of publications from the DBLP Bibliography Server - FAQ". To the right of this box, there is a list of "other persons with the same name:" followed by a list of links to various profiles of Wei Wang from different institutions like Fudan University, MIT, Purdue University, etc. On the right side of the search results, there are several sponsored links and organic search results. The sponsored links include "Wei Wang's Home Page", "Wei Wang", and "Wei Wang's group web". The organic search results include "Wei Wang's Home Page", "Wei Wang", "Wei Wang @ CSE, UNSW Australia", "Wei Wang's Homepage", "Wang Wei", and "DBLP: Wei Wang". Each link is followed by a brief description of the page content and a "Cached" link.

图 2.4 “Wei Wang” Google 检索结果片段

以上的各种类型不确定数据往往由于搜集数据的方式和手段的缺陷，同时发生在同一个应用中。如在一次对一个群体进行信息调查中发生如图 2.5 的情况。由于书写潦草，个体张三的年龄有多种理解，有可能是 57 或者 51 岁。而对于婚姻的状况统计中，有可能是填写者错选了已婚改写单身，也有可能填写者本来就是应该选择已婚。对于李四的情况，他的年龄有可能是 25 岁也有可能是 26 岁，同时婚姻状况没有填写，所以二种婚姻状况都有可能。再看王五，由于对年龄没有填写，且已婚。我们可以估计其年龄为一个均匀分布  $\text{Uniform}(18, 118)$ 。赵六的信息是确定的，因此可以按常规方式处理。鉴此和以上各种不同类型的数据不确定性的描述，不确定数据在我们的日常生活中处处存在，无法

避免。对于包含不确定数据的数据集，当然可以直接粗糙的删除不缺定的数据项，含混的实体，只处理确定数据。通过这种方式所有已有的确定数据上的分析方法都能有效的运用在这些处理后的数据上。然而不确定数据往往能带给人们更多有意义的信息。例如在测量温度时，如果知道温度感应器的误差服从某个正态分布，因此我们能通过这个先验知识，对于测定的一个值估计某个概率 $\alpha$ 的置信区间。

|   |  |
|---|--|
| <div>姓名 <u>张三</u></div> <div>年龄 <u>51</u></div> <div>婚姻状况 (1) 单身 <input checked="" type="checkbox"/> (2) 已婚 <input checked="" type="checkbox"/></div> | <div>姓名 <u>李四</u></div> <div>年龄 <u>25</u></div> <div>婚姻状况 (1) 单身 <input type="checkbox"/> (2) 已婚 <input type="checkbox"/></div>            |
| <div>姓名 <u>王五</u></div> <div>年龄 <u>          </u></div> <div>婚姻状况 (1) 单身 <input type="checkbox"/> (2) 已婚 <input checked="" type="checkbox"/></div>    | <div>姓名 <u>赵六</u></div> <div>年龄 <u>30</u></div> <div>婚姻状况 (1) 单身 <input type="checkbox"/> (2) 已婚 <input checked="" type="checkbox"/></div> |

图 2.5 信息调查表格片段

因此，为了充分利用数据的不确定性提供更准确，更有效的分析结果，针对不确定数据模型上的分析研究已经变得很必要了。

2.3 不确定数据的描述

现有对不确定数据模型的研究中，大多利用可能世界（Possible World）的语义来刻画。可能世界的实例是有不确定元组合法的组合构成的。每一个可能世界产生的概率是由这些元组的联合概率确定。如在交通监测系统中，监测器对来往的车辆进行信息监控。由于监控器本身的精确度，当天的天气状况如光线强弱，车辆的驾驶速度等因素的影响，监测的所有信息都不能完全准确。故监测信息记录如表 2.1 所示。

表 2.1 车辆的监测记录

| 监测记录 | 车标识 | 颜色  | 车速 | 置信度 |
|------|-----|-----|----|-----|
| 1    | A   | 白色  | 80 | 0.3 |
| 2    | A   | 银灰色 | 80 | 0.7 |
| 3    | B   | 黑色  | 90 | 0.8 |
| 4    | C   | 红色  | 83 | 1.0 |

在表 2.1 中,对 A 车有两个监测结论。两条记录结果唯一不同是颜色的区分,一个是白色一个是银灰色。对于 B 车的记录只有一条,但只有 80%认为这一条结论是正确的。另外的 20%可能是其他的结论,比如说根本就不是一辆车等。对于 C 车,监控系统以 100%的置信度确认了 C 车的信息。

分析车辆的记录信息,还可以看出在实际中记录 1 和记录 2 不可能同时出现。所以现实中监控记录只可能为表 2.2 所示的情况之一,这里每一种情况便称为一种可能世界。

表 2.2 可能世界

| 可能世界 (PW) | 成员      | 置信度  |
|-----------|---------|------|
| pw1       | {1,3,4} | 0.24 |
| pw2       | {2,3,4} | 0.56 |
| pw3       | {1,4}   | 0.06 |
| pw3       | {2,4}   | 0.14 |

在表 2.2 中,每一个可能世界都有一个置信度,明确的说明了当前的可能世界发生的概率。每一个可能世界的概率可以直接由这个可能世界的成员元组的发生的联合概率来确定。如果用  $\text{prob}(X)$  表示 X 的发生概率,则:

$$\text{prob}(\text{pw1}) = \text{prob}(1) \times \text{prob}(3) \times \text{prob}(4) = 0.3 \times 0.8 \times 1.0 = 0.24$$

$$\text{prob}(\text{pw2}) = \text{prob}(2) \times \text{prob}(3) \times \text{prob}(4) = 0.7 \times 0.8 \times 1.0 = 0.56$$

$$\text{prob}(\text{pw2}) = \text{prob}(1) \times (1 - \text{prob}(3)) \times \text{prob}(4) = 0.3 \times 0.2 \times 1.0 = 0.06$$

$$\text{prob}(\text{pw2}) = \text{prob}(2) \times (1 - \text{prob}(3)) \times \text{prob}(4) = 0.7 \times 0.2 \times 1.0 = 0.14$$

实际上大多数的应用中,在表达数据的不确定性时存在两种级别的不确定



性。一种是基于元组存在性级别的不确定性，另一种是基于属性级别的不确定性。

基于元组存在性级别的表示中，每一个元组都附属一个该元组发生概率的置信度，正如表 2.1 所示。表 2.1 的表示便是基于元组级别的表示，表达比较清晰并且能有效的反映出元组的存在性。但仅仅从表 2.1 不能反映出记录 1 和记录 2 之间的互斥关系，在默认情况下假定每一个元组间的关系是独立的。因此为了表达这条规则，需要额外增加一条附属的规则：记录  $1 \oplus$  记录 2。以此来完善不确定信息的表达。

属性级别的不确定性并不会涉及到整个元组的发生概率。而是对元组的某个属性的可能的属性值发生的概率进行描述，以达到描述该属性的不确定性。这些描述的方法可以利用概率密度函数或统计参数（如方差，期望等）来表示属性值的概率分布。例如利用传感器对周围环境的温度等指标的调查中,由于传感器的精确性有限，完全有可能测定的结果是 70%的 25 度，30%的 26 度。也有可能为在 25 到 28 度上的均匀分布。相比元组级不确定性，属性级的不确定性往往占用的存储空间不多。

表 2.3 温度测量

| 时间     | 温度                           |
|--------|------------------------------|
| time_1 | prob(25)=0.7<br>prob(26)=0.3 |
| time_2 | uniform(25,28)               |

有些时候可以把几个不确定性元组视为一个具有不确定属性的元组。如在表 2.1 中记录 1 和记录 2 可以被表示成有关 A 车的一个元组，只是在车的颜色属性处有 30%的白色和 70%的银灰色。同样对与温度的测量中，可以把 time\_1 的信息转化为有关 time\_1 的两个元组，只是温度一个是 25，一个是 26。元组置信度分别为 0.7 和 0.3，并且这两个元组互斥。

同时，在表 2.1 中 B 车的信息记录，如果用属性级不确定性表示时，并不能表示出 B 车的不存在的概率。在表 2.3 中 time\_2 的记录中，如果用元组级不确定性表示时，由于温度是在 25 到 28 上的均匀分布，在 25 到 28 之间有无穷多个值，有限的元组无法表示。

## 2.4 不确定数据分析

不确定数据的分析框架如图 2.6 所示，包括了数据源，不确定数据的评估，不确定数据上的查询处理，以及不确定数据的挖掘分析功能。

### 2.4.1 不确定数据源

许多现实的应用中，例如：经济、军事、物流、金融、电信等领域，数据的不确定性普遍存在，不确定性数据扮演关键角色。在 2.2 和 2.3 节对不确定数据的产生和大量不确定数据实例的介绍，正是说明了不确定数据的数据源在现实世界中很普遍。

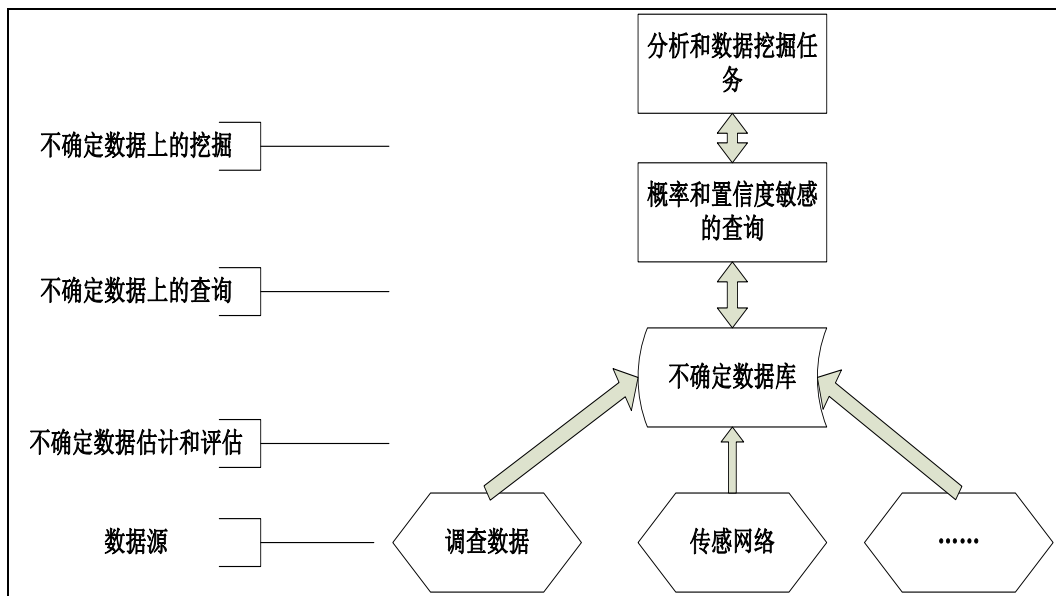


图 2.6 不确定数据的分析框架

### 2.4.2 不确定数据库

不确定数据库用于存储和管理不确定数据。由于不确定数据的普遍存在性，不确定数据库已经变得非常的必要了，并成为了研究热点。表 2.4 列举了一些知名大学以及公司的研究机构正在进行的相关科研项目的基本情况。

表 2.4 不确定性数据管理的相关研究项目<sup>[11]</sup>

| 项目         | 描述  |
|------------|---|
| proTDB     | 主要研究概率半结构化数据的查询处理技术. By University of Puget Sound   |
| Conquer    | 主要研究针对不一致数据库的高效管理技术, 主要应用查询重写技术、实时和动态数据清洗技术. By University of Toronto                                     |
| Orion      | 曾用名: U-DBMS, 是一个通用目的的不确定性数据库系统。它支持离散或连续的概率密度函数; 高效的访问不确定性数据的方法; 优化连接、选择等操作; 图形可视化界面. By Purdue University |
| Trio       | 主要研究不确定数据管理技术, 特别是针对不确定数据的世系分析技术。它基于 ULDB <sup>[29]</sup> 模型, 使用 TriQL 语言. By Stanford University         |
| MayBMS     | 研究内容包括: 查询语言、表示与存储技术、支持数据清洗、高效的查询处理、更新等. BY Cornell University  |
| MystiQ     | 研究内容包括: 数据模型、查询处理技术、关系代数计算等. By University of Washington  |
| HeisenData | 该项目试图将现有的确定性数据管理框架与不确定性查询处理技术相结合, 从而使得新增的功能模块能够嵌入到现有框架中, 增强其性能. By Intel/Berkeley                         |
| ProbDB     | 研究概率聚集查询的计算方法, 开发空间-时间概率数据库. By University of Maryland  |
| Avarar     | 该项目有两大目标: (1) 从非结构化数据中抽取结构化的信息; (2) 基于这类信息构建下一代搜索和商业智能应用. By IBM/Almaden                                  |

### 2.4.3 不确定数据查询

伴随不确定数据库研究一个重要方面, 便是对不确定数据的查询方面。不确定数据的关系代数查询在<sup>[21][31]</sup>中进行了研究, 利用 Trio 系统支持不确定数据库上的查询。常见的聚集函数查询<sup>[20]</sup>也受到了关注。

另外面向确定数据库上的 Top-k 查询的定义非常清晰: 返回 ranking 函数值最大的 k 个元组。但是在不确定性数据库上却存在多种定义方法<sup>[16]</sup>, 例如:

U-Topk<sup>[24]</sup>, U-kRanks<sup>[24]</sup>, PT-k<sup>[5]</sup>, Pk-topk<sup>[23]</sup>查询等。U-Topk 查询返回一个长度为  $k$  的元组矢量, 它在所有可能世界中的发生概率最大; U-kRanks 查询返回在各个级别中出现的总概率最大的元组; PT-k 首先定义一个阈值  $p$ , 返回所有在可能世界实例中成为 top- $k$  的总概率超过阈值的元组; Pk-topk 则返回在所有可能世界实例中称为 top- $k$  的总概率最大的  $k$  个元组。假设一个不确定数据库含有四个元组, 即 $\{t_1=(5, 0.8), t_2=(6, 0.5), t_3=(8, 0.4), t_4=(2, 0.4)\}$ 。当  $k=2$  时, U-topk 返回 $(t_2, t_1)$ ; U-kRanks 返回 $(t_3, t_1)$ ; 当  $p=0.3$ , PT-k 返回 $(t_1, t_2, t_3)$ ; Pk-topk 返回 $(t_1, t_2)$ 。

近年来, 面向不确定性数据的 skyline 查询处理问题也得到了关注。各个元组的值并不确定, 以概率密度函数描述。Pei 等人根据可能世界模型定义了概率 skyline 查询<sup>[14]</sup>。不确定性数据库会衍生出很多可能世界实例, 各元组在各可能世界实例中可能是 skyline 点, 也可能不是 skyline 点。由此,  $p$ -skyline 查询( $0 \leq p \leq 1$ )被定义为返回所有成为 skyline 点的概率超过  $p$  的数据点。文献<sup>[14]</sup>同时提出了两种解决方法: 自下而上方法 (bottom-up method) 和自上而下方法 (top-down method), 分别采用不同的定界、剪枝、精化等启发式规则进行迭代处理。

#### 2.4.4 不确定数据挖掘

基于确定和不确定数据, 数据挖掘功能得到了进一步的分类, 如图 2.7。

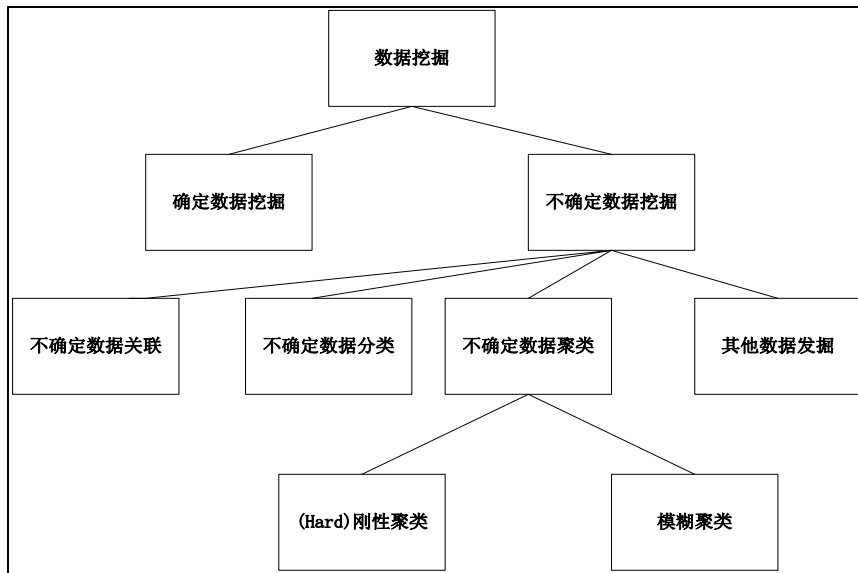


图 2.7 考虑不确定数据, 数据挖掘的分类

对于不确定数据的挖掘任务，可以有两种方式进行。一方面，可以将不确定数据转换为确定数据。把不确定数据从数据集中清理掉，或根据一些先验知识对不确定数据进行评估，使其确定化。然后利用 2.1 节描述的数据挖掘方法对清理后的数据进行分析。有效的避免了挖掘不确定数据的困难。但是正如 2.2 节所述，来自现实世界的的数据固有的不确定性是普遍存在的，如果直接清理掉不确定数据，可能已经剔除了数据的绝大部分。对小部分的数据的挖掘结果，往往不能真实的反映现实世界的隐藏的规律。给数据分析人员的理解构成误导的效果。因此，对待不确定数据，应该进行有效的建模，然后改变确定数据挖掘算法，并应用到不确定数据模型上，并有效对挖掘的结果进行准确度的评估。

在不确定数据数据库模型，查询的研究基础上，不确定数据的挖掘也在逐渐受到关注。在频繁模式挖掘方面，<sup>[15]</sup>提出了基于“可能世界”语义的频繁模式查找，首先提出了离线的不确定数据的频繁模式分析，但由于复杂度比较高，文章提出了基于采样的近似频繁模式的提取。在分类挖掘方面，针对不确定数据，<sup>[17]</sup>提出了相应的改变的决策树分类算法，该文中，每一个元组都带有存在的概率。通过实验证明在精确度方面较传统的决策树算法有较大的提高。为了克服算法的效率问题，提出了剪枝算法，极大的提高了算法的效率。<sup>[18]</sup>对于不确定数据上的基于密度的离群点检测分析进行了研究，并通过实验证明了算法的有效性。<sup>[28][30]</sup>则在数据流的挖掘中有效处理了不确定数据，<sup>[25][26][32]</sup>并提出了相关的近似算法。

#### 2.4.5 不确定数据聚类

在聚类方面，对于数据的聚类语义已经很明确，如在图 2.8 的(1)中，可以较易的分为两个簇 A 和 B。a 点被划归为 A，但当在记录时，a 的位置转移到了 b 时，如图 2.8 的(2)所示，根据不确定算法将会把其划分为 B 簇。

然而不确定数据上的聚类的意义却有多种定义。Sato M.等提出了模糊聚类模型和模糊的 C—Means 算法<sup>[22]</sup>。然而在模糊的聚类模型中，一个簇是数据对象的一个模糊集合。每一个数据对象可能归属于多个簇，并用一个隶属度来衡量一个数据对象归属某个簇的程度。如在图 2.8 的(3)中，数据点 a 是一个不确定点，在 C 区域上均匀分布，在模糊聚类看来，点 a 可能以 60%属于 A 簇，而 40%属于 B 簇。然而本文的不确定聚类模型是把任意一个数据点只归属于其中

的某一个簇，这种聚类称为 **Hard** 聚类(或刚性聚类)。基于 **Hard** 聚类算法模型中，每一个对象都只属于一个簇中的情况。那么针对图 2.8 的(3)中具有不确定点的情形，到底 a 点属于哪个簇呢？本文基于<sup>[1][6][8][9][27]</sup>提出的模型对不确定数据聚类进行了研究。

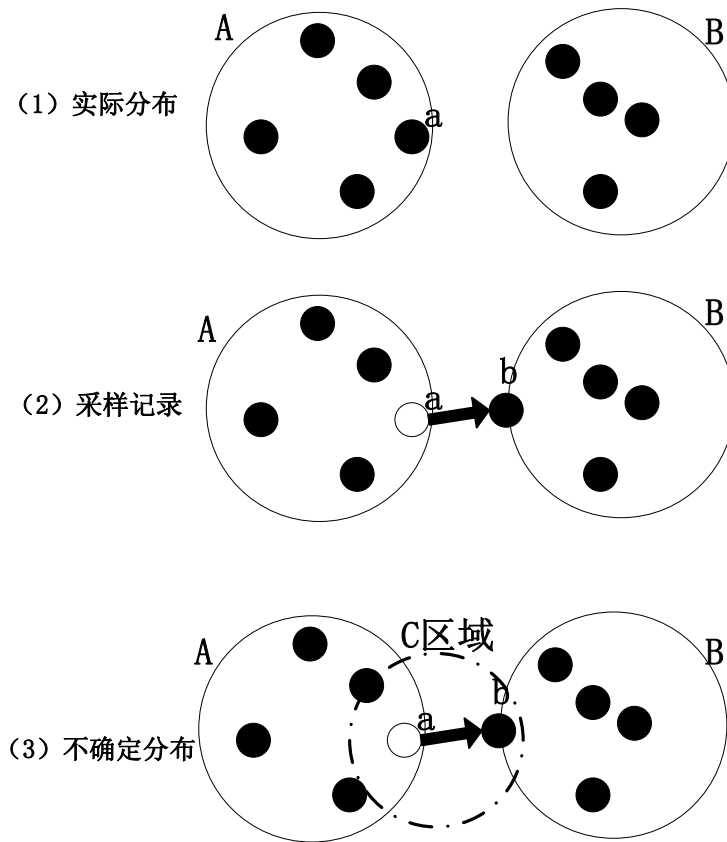


图 2.8 不确定聚类

## 2.5 小结

本章对数据挖掘相关概念进行了综述，同时对不确定数据进行了描述，并通过列举大量应用中的不确定数据来阐明了数据的不确定性的普遍存在性，说明了对不确定数据研究的重要意义。同时也从不确定数据的数据管理、查询、分析挖掘等方面，描述了至今的不确定数据相关研究工作概况。

### 3 不确定数据建模

2.2 节和 2.3 节详细介绍了不确定性数据的基本概念和不确定数据的普遍性。传统的对确定数据上的常规分析挖掘方法不能很好的处理不确定数据。针对 2.4 节提出的不确定数据聚类算法，本文进行了深入的研究。

为了形式化的分析研究不确定聚类算法，首先需要对不确定数据进行建模。

#### 3.1 不确定数据形式化建模

**例 3.1** 在图 2.5 的信息调查片段中，每一个实体可以由姓名、年龄、婚姻状况来描述。如果把三个属性看成空间的三个维度（或分量），则对于这个实例中，一个实体实际上是三维空间上的一个点。如赵六可以看成姓名维的分量值为“赵六”，年龄维的分量值为 30，婚姻状况维的分量值为“已婚”。然而在不确定数据中，每一个实体可能分别以一定概率取空间上的多个点值。如张三实体可能以 42% 被赋予点值（张三，51，单身），18% 为（张三，51，已婚），28% 为（张三，57，单身）和 12% 为（张三，57，已婚）。故此，一个  $n$  维空间上的点以一定概率发生称为  $n$  维空间不确定点。

**定义 3.1（空间不确定点）** 给定一个  $n$  维的向量空间  $R^n$ 。

- (1) 如果点  $p \in R^n$  以概率  $c$  出现某个事件中，则称  $t = \langle p, c \rangle$  为一个  $n$  维空间不确定点。
- (2) 给定两个不确定点  $t_1 = \langle p_1, c_1 \rangle$  和  $t_2 = \langle p_2, c_2 \rangle$ ，如果有  $p_1 = p_2 = p$ ，称  $t_1$  和  $t_2$  是同点项，记为  $t_1 \simeq t_2$ ；反之如果  $t_1$  和  $t_2$  不是同点项，则记为  $t_1 \not\simeq t_2$ 。两个同点项可以合并为另一个同点项，即  $t = t_1 + t_2 = \langle p, c_1 + c_2 \rangle$ ，其中  $t \simeq t_1, t \simeq t_2$ 。

在例 3.1 中  $\langle \text{张三}, 51, \text{单身} \rangle, 42\%$  为一个 3 维空间不确定点。同时如果有  $t_1 = \langle \text{张三}, 51, \text{单身} \rangle, 40\%$  和  $t_2 = \langle \text{张三}, 51, \text{单身} \rangle, 2\%$ ，显然有  $t_1 \simeq t_2$ ，所以合并同点项为  $t = t_1 + t_2 = \langle \text{张三}, 51, \text{单身} \rangle, 42\%$ ，表示为空间点（张三，51，单身）总的发生概率为 42%。

**定义 3.2（空间不确定对象）** 给定一个  $n$  维不确定点的集合  $S$ ，设  $\forall t_1, t_2 \in S, t_1 \not\simeq t_2$ ，满足  $t_1 \not\simeq t_2$  且令  $e = \sum_{t \in S} t.c$ ， $e \leq 1$ 。 $n$  维空间不确定对象  $u$  是满足下列条件的二元

组  $u = \langle S, e \rangle$ :

- (1) 对于  $\forall t \in S$ , 不确定对象  $u$  以概率  $t.c$  被赋予  $t.p$ ;
- (2) 对于  $\forall t' \notin S$ , 不确定对象  $u$  以概率 0 取值  $t'.p$ 。

其中称  $S$  为  $u$  的不确定区域,  $e$  为不确定对象  $u$  的存在率, 也是不确定区域  $S$  的发生概率。

注意, 如果记  $|u|$  为  $u$  可在  $S$  区域上选取的不确定空间点的数量,  $|S|$  为  $S$  区域上的不确定点的数量, 显然有  $|u| = |S|$ 。同时, 令  $UD$  为一个  $n$  维空间不确定对象的集合, 则称  $UD$  为一个  $n$  维的不确定数据集。另本文假定一个不确定数据集的各个不确定对象间相互独立。

**例 3.2** 在图 2.5 的信息中, 考虑 2 个不确定对象, 张三和李四。张三的不确定区域  $S_z = \{ \langle \text{张三}, 51, \text{单身} \rangle, 42\% \rangle, \langle \text{张三}, 51, \text{已婚} \rangle, 18\% \rangle, \langle \text{张三}, 57, \text{单身} \rangle, 28\% \rangle, \langle \text{张三}, 57, \text{已婚} \rangle, 12\% \rangle \}$ , 区域  $S_z$  的发生概率为  $42\% + 18\% + 28\% + 12\% = 100\%$ 。故张三这个不确定对象可以表示为  $u_z = \langle S_z, 100\% \rangle$ 。同理, 对于李四的不确定对象  $u_l = \langle S_l, 100\% \rangle$ , 其中  $S_l = \{ \langle \text{李四}, 25, \text{单身} \rangle, 15\% \rangle, \langle \text{李四}, 25, \text{已婚} \rangle, 35\% \rangle, \langle \text{李四}, 26, \text{单身} \rangle, 15\% \rangle, \langle \text{李四}, 26, \text{已婚} \rangle, 35\% \rangle \}$ 。进而,  $UD = \{u_z, u_l\}$  为一个不确定数据集。

### 3.2 不确定模型分析

在 2.3 节对不确定性数据的描述中, 有两种级别的不确定性。一种是基于元组存在性级别的不确定性表示, 另一种是基于属性级别的不确定性表示。通过 2.3 节的举例分析了两种级别对不确定性数据的表示能力。元组级别的表示中, 能有效的表示元组的存在性, 即出现在数据集的概率, 但由于常常数据库是一个有限的元组集合, 对于具有概率分布的不确定性对象却显得束手无策。另一方面在属性级别的不确定性表示中, 对每一个属性的确定性进行描述, 与元组的不确定性无关, 故对某个属性的所有可能属性值可以用概率密度函数的形式进行表示, 即通过密度函数或相应的参数 (期望, 方差等) 能够表示出连续的概率分布, 同时对于元组的不存在性表示却也显得无能为力。针对 3.1 节描述的不确定性模型, 却能有效的表示不确定性数据的这两个特性。

空间不确定对象可以表示不确定对象的存在性。由于一个空间不确定对象  $u = \langle S, e \rangle$ ,  $e$  表示了不确定对象的存在率, 同理, 可得  $u$  的不存在性概率为  $1-e$ 。



**例 3.3** 在对交通车辆的监控系统中，监控系统记录了一辆 D 车以 100 的速度行驶，但对车的颜色不是很明确，有 20% 的可能是银白色，有 70% 的可能是白色，还有 10% 的可能是系统误把其他物体的运动看成有车辆在行驶，也就是 D 车并没有出现。鉴此，D 车可表示为不确定对象  $u_D = \langle S_D, e \rangle$ ，其中不确定区域  $S_D = \{(\langle D, \text{银白色}, 100 \rangle, 20\%), (\langle D, \text{白色}, 100 \rangle, 70\%)\}$ 。 $u_D$  是一个三维空间不确定对象，根据空间不确定对象的定义， $u_D$  的存在率  $e = 20\% + 70\% = 90\%$ ，此时，易得出  $u_D$  的不存在性概率为  $1 - e = 1 - 90\% = 10\%$ 。

空间不确定对象可以表示取值为连续概率分布的情形。这种表示主要需要概率分布的密度函数和对集合的表示。

**例 3.4** 在利用传感器对环境的温度的进行监测的应用中，如在 T 时刻对 P 地点的温度监测的结果为在 25 到 28 度上的均匀分布。由于涉及到了时间，地点，温度，因此可把这条记录看成 3 维空间不确定对象  $u_T = \langle S_T, e \rangle$ 。其中不确定区域  $S_T = \{(\langle T, P, x \rangle, \text{pdf}(\langle T, P, x \rangle)) | x \in \mathbb{R} \text{ 且 } 25 \leq x \leq 28\}$ ，其中  $\mathbb{R}$  为实数域， $\text{pdf}(\langle T, P, x \rangle)$  用于表示时间、地点、温度三个维度上的概率联合分布密度函数。不妨假设时间、地点、温度三个维独立，且 T 和 P 都已确定，故  $\text{pdf}(\langle T, P, x \rangle) = \text{pdf}(x)$ 。其中  $\text{pdf}(x)$  为 x 的概率分布密度函数，由于是 25 到 28 上的均匀密度函数，所以  $\text{pdf}(x) = 1/(28-25) = 1/3$ 。所以进一步  $S_T = \{(\langle T, P, x \rangle, 1/3) | x \in \mathbb{R} \text{ 且 } 25 \leq x \leq 28\}$ 。同时，对于 e 的计算：
$$e = \int_{S_T} \text{pdf}(x) dx = \int_{25}^{28} \frac{1}{3} dx = 1。$$

当综合考虑以上两种情况，即一个不确定对象的可能取值服从某个概率分布，并且还要考虑该不确定对象的存在性。空间不确定对象同样可以有效表达。

**例 3.5** 在图 2.5 的信息调查中，考察对象王五。王五的年龄是缺失的，但根据其婚姻状况，可估计其年龄为 18 到 118 上的均匀分布。同时假设有 10% 的概率是王五这个对象根本不存在，是错误的处理情况。建立王五的不确定对象  $u_W = \langle S_W, e \rangle$ 。其中不确定区域  $S_W = \{(\langle \text{王五}, x, \text{已婚} \rangle, (1-10\%) \times \text{pdf}(\langle \text{王五}, x, \text{已婚} \rangle)) | x \in \mathbb{R} \text{ 且 } 18 \leq x \leq 118\}$ ，同样  $\text{pdf}(\langle \text{王五}, x, \text{已婚} \rangle)$  为联合分布密度函数，由于各维的独立性和 x 的分布情况， $\text{pdf}(\langle \text{王五}, x, \text{已婚} \rangle) = \text{pdf}(x) = 1/100$ 。即： $S_W = \{(\langle \text{王五}, x, \text{已婚} \rangle, 0.009) | x \in \mathbb{R} \text{ 且 } 18 \leq x \leq 118\}$ 。同时，

$$e = \int_{S_W} \text{pdf}(x) dx = \int_{18}^{118} 0.009 dx = 0.9。$$

另外，对于元组级的表示中，为了反映元组间的互斥情况，还必须定义额外的规则，如表 2.1 中的记录 1 和记录 2 不能同时出现需要规则：记录  $1 \oplus$  记录 2。但在不确定对象的模型中，记录 1 和记录 2 已经作为一个不确定对象的可能被赋予的不确定空间点了，有效的保证了它们之间的互斥性。

特别地，空间不确定对象模型统一了不确定数据和确定数据的建模。可以把一个确定的数据对象  $\mathbf{p}$  看成一个不确定对象  $\mathbf{u}$ ，其中  $\mathbf{u.e} = 1$ ，且  $\mathbf{u.S}$  有且只有一个空间不确定点  $\mathbf{t} = \langle \mathbf{p}, 1.0 \rangle$ 。故本文建立的空间不确定对象模型能统一的表达确定数据。基于空间不确定对象模型，确定数据仅仅是不确定数据的一个特例，能有效的用该模型表示。

### 3.3 不确定模型语义

在 2.2 节讨论了基于属性级别和元组级别下不确定数据的可能世界语义。这一节将针对空间不确定对象，描述对应的可能世界语义。并探讨了不确定数据集上可能世界相关性质。

根据定义 3.2 可知，任何不确定对象可以选择相应不确定区域上的一个空间点对  $\mathbf{u}$  进行实例化。现对可能世界作如下描述：

**定义 3.3 (不确定数据集的可能世界)** 给定  $UD$  为一个  $n$  维空间上的一个不确定数据集， $W$  为一个  $n$  维空间上的一个数据集， $\text{prob}$  为一个概率值，则按下列过程构造出来的集合称为**不确定数据集的可能世界**：

(1) 初始化  $W = \emptyset$ ,  $\text{prob} = 1.0$

(2) 对于每一个不确定对象  $\mathbf{u} \in UD$ ,

a) 如果  $\mathbf{u.e} = 1$ ，则必须选取一个  $n$  维空间不确定点  $\mathbf{t} \in \mathbf{u.S}$ ，使  $W = W \cup \{\mathbf{t.p}\}$ ， $\text{prob} = \text{prob} \times \mathbf{t.c}$

b) 如果  $\mathbf{u.e} < 1$ ，则要么选取一个  $n$  维空间不确定点  $\mathbf{t} \in \mathbf{u.S}$ ，使  $W = W \cup \{\mathbf{t.p}\}$ ， $\text{prob} = \text{prob} \times \mathbf{t.c}$ ；或者要么  $\text{prob} = \text{prob} \times (1 - \mathbf{u.e})$ ， $W$  不变。

注意，以上步骤为不确定数据集  $UD$  的一个实例化过程，称  $W$  为  $UD$  的一个实例， $\text{prob}$  为  $UD$  的实例  $W$  的发生概率。称二元组  $PW = \langle W, \text{prob} \rangle$  为  $n$  维不确定数据集  $UD$  上的一个可能世界。

**例 3.6** 与图 2.5 一样，来自信息调查的另一片段如图 3.1。有两个不确定对象邓文和王灿。邓文的年龄有 40% 为 51 和 60% 为 57。同样王灿的婚姻状况有 80%

为单身而 20%为已婚。

|   |   |
|---|---|
| 姓名 <u>邓文</u>  | 姓名 <u>王灿</u>  |
| 年龄 <u>51</u>  | 年龄 <u>23</u>  |
| 婚姻状况 (1) 单身 <input type="checkbox"/> (2) 已婚 <input checked="" type="checkbox"/> | 婚姻状况 (1) 单身 <input checked="" type="checkbox"/> (2) 已婚 <input type="checkbox"/> |

图 3.1 信息统计片段

同时，王灿实体有 60%为作废的信息，即王灿有 40%的存在率。故两个不确定对象  $u_D$ 、 $u_W$  构成的不确定数据集如表 3.1。同时根据定义 3.3，可以构造出相应的所有可能世界如表 3.2。

表 3.1 图 3.1 的信息统计片段的不确定数据集 (UD)

| 不确定对象 ( $u$ ) | 不确定对象区域 ( $S$ )  | 存在率 ( $e$ ) |
|---------------|--|-------------|
| $u_D$         | $\{ \langle (\text{邓文}, 51, \text{已婚}), 40\% \rangle$<br>$\langle (\text{邓文}, 57, \text{已婚}), 60\% \rangle \}$ | 100%        |
| $u_W$         | $\{ \langle (\text{王灿}, 23, \text{单身}), 32\% \rangle$<br>$\langle (\text{王灿}, 23, \text{已婚}), 8\% \rangle \}$  | 40%         |

表 3.2 信息统计片段的可能世界

| 可能世界 (PW) | 实例 (W)   | 概率 (prob) |
|-----------|--|-----------|
| PW1       | $\{ (\text{邓文}, 51, \text{已婚}) \}$                             | 24%       |
| PW2       | $\{ (\text{邓文}, 51, \text{已婚}), (\text{王灿}, 23, \text{单身}) \}$ | 12.8%     |
| PW3       | $\{ (\text{邓文}, 51, \text{已婚}), (\text{王灿}, 23, \text{已婚}) \}$ | 3.2%      |
| PW4       | $\{ (\text{邓文}, 57, \text{已婚}) \}$                             | 36%       |
| PW5       | $\{ (\text{邓文}, 57, \text{已婚}), (\text{王灿}, 23, \text{单身}) \}$ | 19.2%     |
| PW6       | $\{ (\text{邓文}, 57, \text{已婚}), (\text{王灿}, 23, \text{已婚}) \}$ | 4.8%      |

定理 3.1 记  $PW_s$  为不确定数据集 UD 上所有可能世界的集合， $|PW_s|$ 表示集合

$PW_s$  的元素的数量, 令  $PW$  为一个可能世界且  $PW \in PW_s$  则下列公式成立:

$$(1) |PW_s| = \prod_{u \in UD, u.e=1} |u| \times \prod_{u \in UD, u.e<1} (|u|+1) \quad (3.1)$$

$$(2) PW.prob = \prod_{u \in UD \wedge \exists t \in u.S \wedge t.p \in PW} t.c \times \prod_{u \in UD \wedge \forall t \in u.S \wedge t.p \notin PW} (1-u.e) \quad (3.2)$$

$$(3) PW.prob > 0 \text{ 且 } \sum_{PW \in PW_s} PW.prob = 1.$$

定理 3.1 详细证明较长, 但思路简单, 先给出证明思路: 根据定义 3.3 中  $PW$  的构造过程, 当  $u.e=1$  时,  $u$  可以被赋予  $|u|$  个不确定空间点; 当  $u.e < 1$  时,  $u$  除了能赋予  $|u|$  个不确定空间点外, 还可以有缺失的情况, 故总的有  $(|u|+1)$  种情况。在根据组合的乘法原理即可得到公式(3.1)的证明。同理可根据概率的乘法原理得到公式(3.2)的证明。如在例 3.6 中, 由于  $u_D.e=1$ , 而  $u_W.e < 1$ , 故  $|PW_s| = |u_D| \times (|u_W| + 1) = 2 \times (2 + 1) = 6$ 。  $PW1.prob = 40\% \times (1 - 40\%) = 24\%$ 。

### 3.4 小结

本章从不确定数据的一般性出发, 给出了不确定数据的形式化描述。由于数据的不确定性, 提出了空间不确定点、空间不确定对象、空间不确定区域以及不确定数据集的形式化概念, 从而有效的为不确定数据进行了建模。该模型改进了传统的基于元组级和属性级不确定模型, 有效的表示了不确定数据的元组存在性语义和不确定对象的概率分布。同时, 在建立的不确定数据的数学模型上, 提出了不确定数据模型的可能世界语义, 分析并导出了不确定数据集可能世界的发生概率和规模。为不确定数据的聚类算法研究提供了分析基础。

## 4 不确定数据的聚类

正如 2.4.5 节所述, 常规的聚类算法针对的是确定的数据点, 能确切的把数据归属于与其距离最近, 相似度最大的簇。然而对于不确定数据对象, 却不能确定性的描述对象的相似簇。本章在确定数据上的 K-Means 和 K-Median 聚类算法的基础上, 对不确定数据对象的聚类算法进行了研究。

### 4.1 不确定数据对象聚类的特殊难点

通过对不确定数据的形式化建模, 显而易见不确定数据与确定数据相比, 有着截然不同数学模型。不确定数据基于可能世界的语义变得更加复杂, 故而对不确定数据的聚类分析也变得更具有挑战。

- (1) 可能世界的规模巨大。通过在不确定对象上的可能世界的定义和定理 3.1, 任何合法的不确定对象的不同情况的组合都可以构成一个不确定数据集的实例。假设不确定数据集有  $n$  个不确定对象, 而按照最简单的情况, 每个不确定对象只有一个不确定空间点, 有两种情况出现: 不确定对象要么以这个不确定空间点发生, 要么不发生在可能世界中。这样组合以后将会有  $2^n$  个可能世界(图 4.1)。当每一个不确定对象的候选不确定空间点增多时, 还将会大于  $2^n$  的规模。故可以估计可能世界数量至少是不确定数据集规模的指数级倍数。这样对聚类算法的可伸缩性要求就更高了。

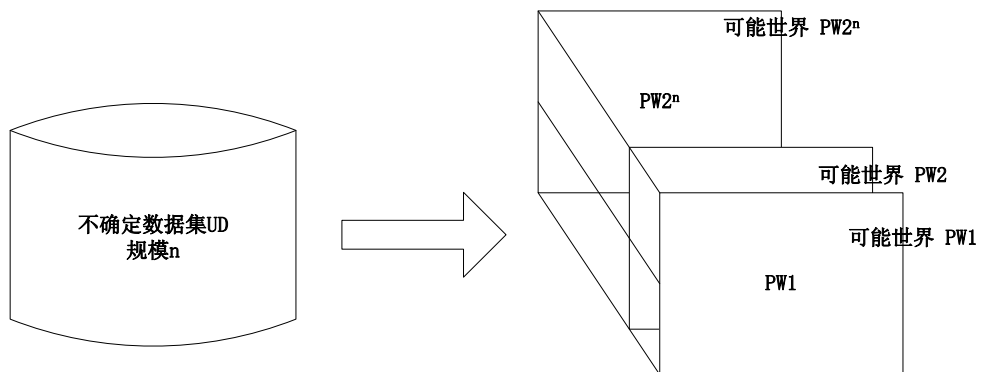


图 4.1 不确定世界指数规模

- (2) 增加了新的维。在  $n$  维确定数据上的传统聚类算法中, 分析的数据仅是确定数据所占空间的维度。然而空间不确定对象的不确定区域中的空间点, 正如定义的形式所示, 不确定数据点  $t = \langle p, c \rangle$ , 除了考虑不确定数据点的空间维度, 即  $p$  的维度, 还要考虑空间点的发生率  $c$ 。这样在维度上来说, 增加了新的维度即空间点的发生率。故而传统的聚类算法需要进行修改以处理这个新增的维度。
- (3) 不确定对象的相似度。在空间不确定对象的模型中, 每一个不确定对象都有一个不确定区域, 而不确定对象的值以一定的概率分布被赋予区域上的任何一个空间不确定点。当在判断一个不确定对象应该归属的簇时, 可能出现如图 2.8 (3) 中的情况。在区域  $C$  中, 如果选定点  $a$  那么这个不确定对象将会属于簇  $A$ , 相反, 如果不确定对象被赋予点值  $b$  时, 则将会归属簇  $B$ 。因此为了判断不确定对象到底与  $A$  更相似还是和  $B$  更相似的问题, 相似性的度量需要重新定义。

## 4.2 不确定数据聚类问题描述

确定数据上的聚类问题可描述为: 将物理或抽象的确定对象集合分成相似的对象簇 (cluster) 的过程称为聚类 (clustering), 其中对象簇是数据对象的集合。划分对象簇的标准是使得同一个类或簇中的数据对象之间具有很高的相似度, 而不同类或簇中的对象却高度相异。类似的可以给出不确定数据集的聚类问题描述。

---

### 问题描述 4.1: 不确定数据聚类问题描述:

---

GIVEN:  $n$  维不确定数据集  $UD$

FIND: 一个包含  $K$  个簇的集合  $\Phi = \{C_1, C_2, \dots, C_k\}$  和一个映射函数  $\varphi$

SATIFY:

(1) 令  $C_i \in \Phi$ ,  $C_i$  的元素为空间不确定点  $t$ , 其中  $t.p$  是  $n$  维空间  $R^n$  上的一个空间点即  $t.p \in R^n$ , 令某个空间点  $p_{ci}$  为  $C_i$  的中心代表。

(2) 对  $\forall u \in UD, \forall t \in u.S, \varphi(u, t) = p_{ci}$ , 即能通过函数  $\varphi$  把  $t$  划分给与自身最为相似的簇  $C_i$  中。

(3) 使得每一个簇  $C_i$  内的对象尽量相似, 簇间的对象尽量相异。

---

然而，对基于空间不确定对象模型的不确定数据，输入的是一个不确定对象集合  $UD$ ，对于  $\forall u \in UD$ ， $u$  按一定的概率分布被赋予空间点值  $t.p$ ，其中  $t \in u.S$ 。在考虑  $u$  和  $C_i$  的相似性时，随着  $u$  被赋予不同值而不断变化。

**例 4.1** 如图 4.2 的(1)中，不确定数据对象  $u$  的不确定区域  $u.S = \{t_1, t_2, t_3, t_4, t_5\}$ ，故对于  $\forall t_i \in u.S$ ， $u$  可以被赋予  $t_i.p$ 。很明显，当  $u$  发生在  $t_1$ 、 $t_2$  时， $u$  应更相似于簇  $C_1$ ，另一方面，当  $u$  发生在  $t_3$ 、 $t_4$ 、 $t_5$  时， $u$  将与  $C_2$  更相似。归因于  $u$  的这种不确定性，常常可以按两种方式来划分  $u$ ：

- 1) 由于  $u$  可以取  $u.S$  上任何空间点  $t.p$ 。并且随着  $u$  不同的取值，归属的簇也将变化。所以按图 4.2(2)所示，直接考虑当  $u$  取  $t_1.p$ 、 $t_2.p$  时，令  $\varphi(u, t_1)=pc_1$ ， $\varphi(u, t_2)=pc_1$ ，即把  $u$  分派给簇  $C_1$ ，当  $u$  取  $t_3.p$ 、 $t_4.p$ 、 $t_5.p$  时，令  $\varphi(u, t_3)=pc_2$ ， $\varphi(u, t_4)=pc_2$ ， $\varphi(u, t_5)=pc_2$ ，把  $u$  分派给簇  $C_2$ 。因而， $\forall t \in u.S$ ，将根据  $t.p$  和簇的相似度来确定  $t$  的簇。如图 4.2 的(2)中， $t_1$ 、 $t_2$  归属于簇  $C_1$ ； $t_3$ 、 $t_4$ 、 $t_5$  归属于  $C_2$ 。这种方式的映射称为不确定对象的**未指定(unassigned)聚类**。
- 2) 把  $u.S$  上的所有空间不确定点  $t$ ，看成一个整体，按照一定标准综合评估  $u$  的归属情况，把  $u.S$  上的不确定点划分给某一个簇。如在图 4.2 的(3)中，可以选定一个规则，比如按照投票的方式来确定簇。因为有 2 个不确定点与  $C_1$  相近，而有 3 个不确定点与  $C_2$  相近。根据投票的多数原则， $u.S$  上的所有不确定点应归属于簇  $C_2$ 。此时， $\forall t \in u.S$ ，有  $\varphi(u, t)=pc_2$ 。这种方式的映射称为不确定对象的**指定(assigned)聚类**。

以上这两种方式在现实生活中都具有重要应用意义。如在银行的管理服务中，对于银行设施、机构的部署问题，可以把每一个客户看成一个不确定对象，每一个客户可能所处的地点集合称为不确定区域。每一个银行客户都要固定到一个银行的支行，以便于从个人的账户经理处得到服务，即使这个客户到了异地，如果要得到相应的服务也必须回到这个指定的支行处。这种方式类似与指定聚类；另一方面，每一个银行客户可以选择离自己最近的 ATM 机取得服务，即使在异地，也可以利用异地的 ATM 机得到有效的处理。这种方式类似于未指定聚类。下一节将对这两种方式的不确定的 K 均值聚类进行研究。

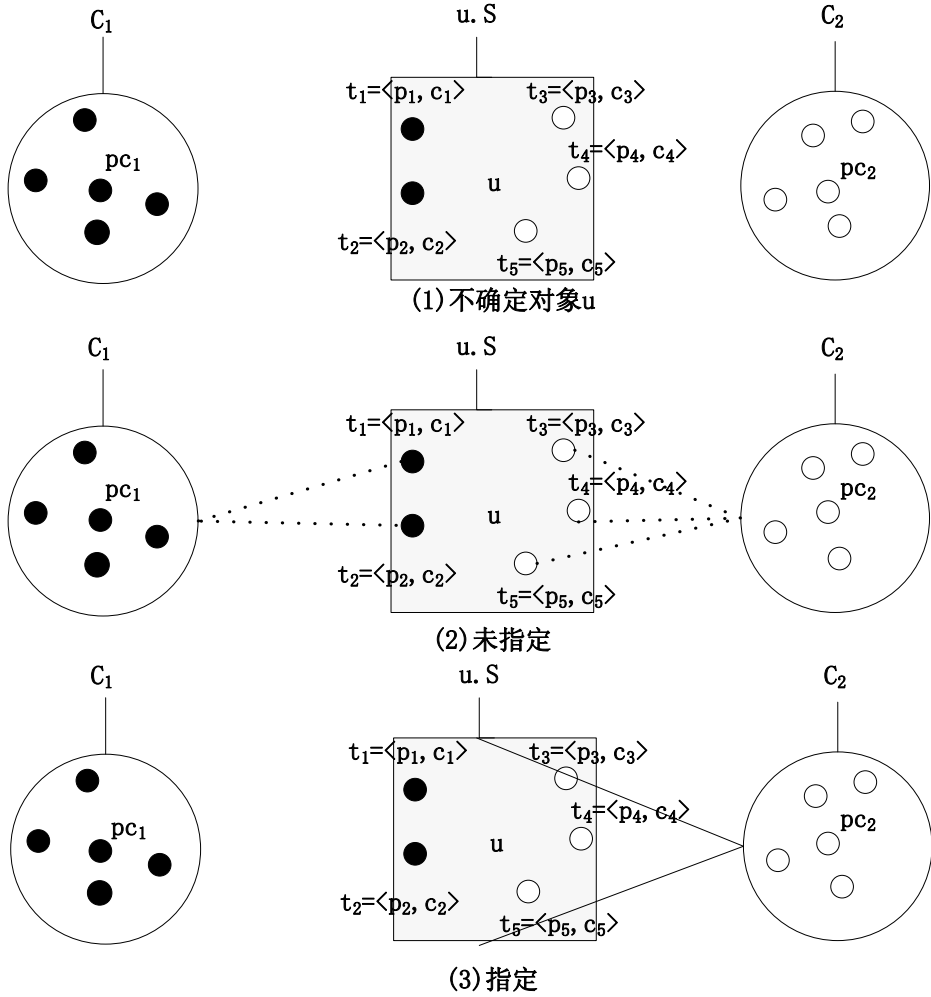


图 4.2 不确定对象的划分

### 4.3 不确定数据的聚类算法

传统的确定数据上的聚类算法很多，在众多著名划分聚类算法中，K-Means 和 K-Median 便是其中之一。本节将基于这两个算法，提出针对不确定对象模型上的对应算法。

#### 4.3.1 确定数据的划分算法描述

已知确定数据集  $D$  是来自  $n$  维空间  $R^n$  的一个子集，K-Means 和 K-Median 聚类算法试图把数据集  $D$  上的数据对象划分成一个包含  $K$  个簇的集合  $\Phi = \{C_1,$



$C_2, \dots, C_k\}$ , 其中每一个簇都是数据对象的集合。须满足:

(1) 如果定义  $\mathbf{pc}_i$  为簇  $C_i$  的中心代表。 $\mathbf{pc}_i$  可如下计算得到:

$$pc_i = \frac{1}{|C_i|} \sum_{p \in C_i} p \quad (4.1)$$

(2) 对  $\forall \mathbf{p} \in D$ , 定义  $\mathbf{p}$  到簇  $C_i$  的相异度为  $\mathbf{p}$  到  $C_i$  的中心点  $\mathbf{pc}_i$  的距离, 表示为  $d(\mathbf{p}, \mathbf{pc}_i)$ 。如果  $\mathbf{p}$  应归属于某个  $C$ , 其中  $\mathbf{pc}$  是  $C$  的中心代表。并定义函数  $\varphi$  为由  $\mathbf{p}$  映射到  $\mathbf{p}$  所在簇  $C$  的中心点  $\mathbf{pc}$ , 即  $\varphi(\mathbf{p}) = \mathbf{pc}$

(3) 所划分的簇集合  $\Phi$ , 需要使得目标代价函数的值最小。 $K$ -Means 的代价函数如(4.2), 而  $K$ -Median 的代价函数为式(4.3)。

$$Cost_{kmeans}(D, \Phi, \varphi) = \sum_{p \in D} d^2(p, \varphi(p)) \quad (4.2)$$

$$Cost_{kmedian}(D, \Phi, \varphi) = \sum_{p \in D} d(p, \varphi(p)) \quad (4.3)$$

在  $K$ -Means 和  $K$ -Median 中, 对于(2)提及的函数  $\varphi(\mathbf{p})$ , 定义为从自变量  $\mathbf{p}$  映射为所属簇  $C$  的中心点  $\mathbf{pc}$ , 其中  $C = \arg \min_{C \in \Phi} d(p, pc)$ 。即  $C$  为与  $\mathbf{p}$  相异度最小的簇。

#### 4.3.2 不确定数据的划分算法

由于不确定数据在维度上增加了数据对象的发生概率, 为了有效的处理概率维, 并充分利用概率维来对不确定数据进行挖掘, 在  $K$ -Means 和  $K$ -Median 划分算法的基础上, 提出了  $UK$ -Means 和  $UK$ -Median 原理和相应的算法。同时如例 4.1, 不确定数据的聚类算法分为未指定和指定两种情形。所以  $UK$ -Means 可细分为: 未指定的  $UK$ -Means 聚类算法, 表示为  $UA$ - $UK$ -Means; 指定的  $UK$ -Means 聚类算法, 表示为  $A$ - $UK$ -Means。类似的,  $UK$ -Median 可以按未指定和指定分为  $UA$ - $UK$ -Median 和  $A$ - $UK$ -Median 两种聚类算法。

已知一个  $n$  维空间  $R^n$  上的一个不确定数据集  $UD$ ,  $UK$ -Means 和  $UK$ -Median 聚类算法试图把不确定数据集  $UD$  划分为一个包含  $K$  个簇的集合  $\Phi = \{C_1, C_2, \dots, C_k\}$ , 每个簇  $C$  的元素都为  $n$  维空间不确定点  $\mathbf{t}$  的集合。须满足:

(1) 如果定义  $\mathbf{pc}_i$  为簇  $C_i$  的中心代表。 $\mathbf{pc}_i$  可如下计算得到:

$$pc_i = \frac{1}{|C_i|} \sum_{t \in C_i} t.p \quad (4.4)$$

- (2) 对给定  $u \in UD$ ,  $\forall t \in u.S$ , 定义  $t$  到簇  $C_i$  的相异度为  $t.p$  到簇  $C_i$  中心点代表  $pc_i$  的距离, 表示为  $d(t.p, pc_i)$ 。如果  $t$  应归属于某个  $C$ , 其中  $pc$  是  $C$  的中心代表, 定义函数  $\varphi$  为由  $u$  和  $t$  映射到  $t$  所在簇  $C$  的中心点  $pc$ , 即  $\varphi(u, t) = pc$
- (3) 所划分的簇集合  $\Phi$ , 需要使得目标代价函数的值最小。但由于数据对象  $u$  的不确定性, 这里的目标函数用代价的期望表示。UK-Means 的目标函数为式(4.5); 而 UK-Median 的目标函数如式(4.6)。

$$Cost_{ukmeans}(UD, \Phi, \varphi) = E\left(\sum_{u \in UD \wedge t \in u.S} d^2(t.p, \varphi(u, t))\right) \quad (4.5)$$

$$Cost_{ukmedian}(UD, \Phi, \varphi) = E\left(\sum_{u \in UD \wedge t \in u.S} d(t.p, \varphi(u, t))\right) \quad (4.6)$$

在 UK-Means 和 UK-Median 中, 对于(2)中提及的  $\varphi(u, t)$ , 相对于 K-Means 和 K-Median 中的  $\varphi(p)$  来说要复杂一些。从 4.2 节所述可知, 不确定数据上的聚类算法通常因为  $\varphi(u, t)$  的映射的方式不同分为未指定聚类 and 指定聚类。故  $\varphi(u, t)$  的定义也将对应两种情况。

- 1) 未指定聚类的情况,  $\varphi(u, t)$  定义为通过  $u$  和  $t$  映射到某个簇  $C$  的中心代表点  $pc$ ,
  - (a) 当 UA-UK-Means 时, 其中  $C = \arg \min_{C \in \Phi} d^2(t.p, pc)$ , 即  $C$  为与  $t$  相异度平方最小的簇。
  - (b) 当 UA-UK-Median 时, 其中  $C = \arg \min_{C \in \Phi} d(t.p, pc)$ , 即  $C$  为与  $t$  相异度最小的簇。
- 2) 指定聚类的情况,  $\varphi(u, t)$  定义为通过  $u$  和  $t$  映射到某个簇  $C$  的中心代表点  $pc$ ,
  - (a) 当 A-UK-Means 时, 其中  $C = \arg \min_{C \in \Phi} E\left(\sum_{t \in u.S} d^2(t.p, pc)\right)$ , 即  $C$  为与  $u.S$  上的所有不确定点  $t$  相异度平方之和的期望值最小的簇。

(b) 当 A-UK-Median 时, 其中  $C = \arg \min_{C \in \Phi} E(\sum_{t \in u.S} d(t.p, pc))$ , 即 C

为与  $u.S$  上的所有不确定点  $t$  相异度之和的期望值最小的簇。

**定理 4.1** (1) 对于未指定聚类情况, UA-UK-Means 的  $\varphi(u, t)$  定义和 UA-UK-Median 的  $\varphi(u, t)$  定义是等价的, 即(a)和(b)等价。(2) 对于指定聚类情况, A-UK-Means 的  $\varphi(u, t)$  定义和 A-UK-Median 的  $\varphi(u, t)$  定义不等价, 即(a)和(b)不等价。

**证明:** (1) 给定不确定点  $t$  和簇集合  $\Phi$ , 一方面, 当利用(a)中的定义, 在  $\Phi$  中获得  $C_i$  为与  $t$  的相异度平方最小, 即  $d^2(t.p, pc_i)$  最小; 另一方面, 当利用(b)中的定义时, 不妨设在  $\Phi$  中获得  $C_j$  为与  $t$  的相异度最小, 即  $d(t.p, pc_j)$  最小。综合这两方面, 将有  $d^2(t.p, pc_i) \leq d^2(t.p, pc_j)$  和  $d(t.p, pc_i) \geq d(t.p, pc_j)$ 。因为  $d^2(t.p, pc_i) \leq d^2(t.p, pc_j) \Leftrightarrow d(t.p, pc_i) \leq d(t.p, pc_j)$ , 而又因为  $d(t.p, pc_i) \geq d(t.p, pc_j)$ , 故  $d(t.p, pc_i) = d(t.p, pc_j)$ , 即  $t$  与  $C_i$  和  $C_j$  的相异度是相同的。(2) 要证明指定聚类情况下, (a)和(b)不等价, 只需列举一个反例即可。如图 4.3 所示, 有一不确定点  $u$ ,  $u.S$  上有两个不确定点  $t_1$  和  $t_2$ , 以及两个簇  $C_1$  和  $C_2$ , 其中  $pc_1$  和  $pc_2$  分别为这两个簇的中心代表。一方面, 当利用(a)的定义时, 因为  $E(\sum_{t \in u.S} d^2(t.p, pc)) = \sum_{t \in u.S} t.c \times d^2(t.p, pc)$ , 故可求得  $u.S$  上的不确定点到

$C_1$  的相离度平方和的期望为  $0.7^2 \times 0.5 + 0.4^2 \times 0.55 = 0.333$ , 而对于  $C_2$  则:  $0.5^2 \times 0.5 + 0.6^2 \times 0.55 = 0.323$ , 故将把  $u$  归为簇  $C_2$ 。另一方面, 当利用(b)的定义时, 因

为  $E(\sum_{t \in u.S} d(t.p, pc)) = \sum_{t \in u.S} t.c \times d(t.p, pc)$ , 故可求得  $u.S$  上的不确定点到  $C_1$  的相

离度和的期望为  $0.7 \times 0.5 + 0.4 \times 0.55 = 0.57$ , 而对于  $C_2$  则:  $0.5 \times 0.5 + 0.6 \times 0.55 = 0.58$ , 故将把  $u$  归为簇  $C_1$ 。综合以上两方面, 可知对同一个情形, (a)和(b)的计算结论是不一样的。所以(a)和(b)不等价。证毕。

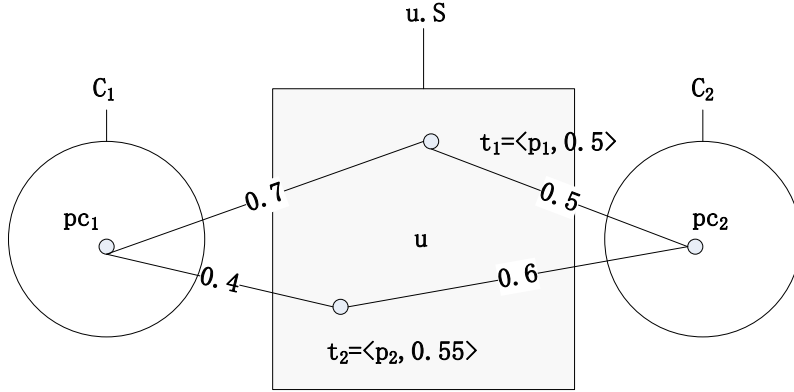


图 4.3 不确定点与簇相离度计算

鉴于以上对给定  $u$  和  $t$  对簇的映射函数  $\varphi(u, t)$  的分析，以下将给出各种情况的算法过程。

在未指定聚类的情形，根据定理 4.1，(a)和(b)是等价的，因此只考虑简单的一种，即选择与  $t$  相异度最小的簇  $C$ 。

---

算法 4.1:  $unassigned\_ \varphi(u, t)$ : 在未指定聚类中，与不确定点相异度最小的簇选择算法

---

输入：不确定对象  $u$

不确定空间点  $t$

簇集合  $\Phi = \{C_1, C_2, \dots, C_k\}$

输出：簇  $C$  的中心点  $pc$ 。相离度  $d$ ，即返回  $\langle pc, d \rangle$  其中  $C = \arg \min_{C \in \Phi} d(t, p, pc)$

过程

- (1) Begin
  - (2) 初始化变量  $d \leftarrow \infty$ ,  $pc \leftarrow \text{NULL}$
  - (3) For each  $C_i \in \Phi$
  - (4)     计算距离  $d(t, p, pc_i)$  并  $temp \leftarrow d(t, p, pc_i)$  // 计算相离度期望
  - (5)     If ( $temp < d$ )
  - (6)          $d \leftarrow temp$
  - (7)          $pc \leftarrow pc_i$
  - (8) EndIf
-

---

```

(9)   EndFor
(10)  Return <pc, d>
(11)  End

```

---

**引理 4.1** 算法  $\text{unassigned\_}\varphi(\mathbf{u}, \mathbf{t})$  的时间复杂度为  $O(K)$ ，其中  $K$  为簇的个数。

**证明：**从算法 4.1 中可以看出，对于每一个  $\mathbf{t}$ ，在寻求其相离度最小的簇时，只需遍历一遍簇集合。在(2)的时间复杂度为  $O(1)$ ；(3)~(9)中，由于  $K$  为簇的个数，故时间复杂度为  $O(K)$ ；(10)的时间复杂度为  $O(1)$ 。所以该过程的时间复杂度为  $O(1)+O(K)+O(1)=O(K)$ 。证毕。

在指定聚类情形，总是把  $\mathbf{u.S}$  上所有不确定点都指派到同一个簇中，所以  $\varphi(\mathbf{u}, \mathbf{t})=\varphi(\mathbf{u})$ ，即指派过程只与  $\mathbf{u}$  相关。在(a)A-UK-Means 和(b)A-UK-Median 的两种定义下，A-UK-Means 考虑的是相异度的平方，A-UK-Median 考虑的是相异度。故可以把二者综合在一个算法  $\text{assigned\_}\varphi(\mathbf{u})$  中。如算法 4.2。

---

**算法 4.2:**  $\text{assigned\_}\varphi(\mathbf{u})$ : 在指定聚类中，与不确定点相异度最小的簇选择算法

---

输入：不确定对象  $\mathbf{u}$

簇集合  $\Phi = \{C_1, C_2, \dots, C_k\}$

输出：与  $\mathbf{t}$  相异度最小的簇  $C$  的中心点  $pc$ 。以及相异度大小  $d$ : 即返回  $\langle pc, d \rangle$

过程

```

(1)   Begin
(2)   初始化变量  $d \leftarrow \infty$ ,  $pc \leftarrow \text{NULL}$ 
(3)   For each  $C_i \in \Phi$ 
(4)        $temp \leftarrow 0$ 
(5)       For each  $\mathbf{t} \in \mathbf{u.S}$ 
(6)           如果是 A-UK-Means,  $temp \leftarrow d^2(\mathbf{t}, p, pc_i) \times \mathbf{t}.c + temp$  // 相离度平方期望
               如果是 A-UK-Median,  $temp \leftarrow d(\mathbf{t}, p, pc_i) \times \mathbf{t}.c + temp$  // 计算期望相离度
(7)       EndFor
(8)       If ( $temp < d$ )
(9)            $d \leftarrow temp$ 

```

---

---

```

(10)      pc ← pci
(11)      EndIf
(12)  EndFor
(13)  Return <pc, d>
(14)  End

```

---

**引理 4.2** 算法  $\text{assigned\_}\varphi(\mathbf{u})$  的时间复杂度为  $O(K \times |\mathbf{u}|)$ 。其中  $K$  为簇的个数。

**证明：**从算法 4.2 中，(2)的时间复杂度为  $O(1)$ ；在(5)~(7)的 For 循环中，将会遍历  $\mathbf{u.S}$  中的每一个  $\mathbf{t}$ ，故时间复杂度为  $O(|\mathbf{u}|)$ ，同时在(3)~(12)的循环中将会把给定的  $\mathbf{u}$  和每一个  $\mathbf{C}$  进行比较，故时间复杂度为  $O(K \times |\mathbf{u}|)$ ；同时(13)的时间复杂度为  $O(1)$ 。所以算法的总时间复杂度为： $O(1) + O(K \times |\mathbf{u}|) + O(1)$ 。证毕。

同时根据传统  $K$  均值的启发式算法思想，获得不确定数据上  $K$  均值算法的思路。首先，随机地从  $K$  个不确定对象  $\mathbf{u}$  的不确定区域  $\mathbf{u.S}$  中，选择不确定点  $\mathbf{t}$ ，然后利用  $\mathbf{t.p}$  代表每一个簇的中心代表，然后对每一个不确定对象中的不确定点通过算法过程  $\text{unassigned\_}\varphi(\mathbf{u}, \mathbf{t})$  或  $\text{assigned\_}\varphi(\mathbf{u})$  指派到对应的簇，再重新计算每个簇的中心代表。这个过程不断重复，直到对应的目标代价函数收敛，即目标代价函数的值和簇的成员不再变化。

由于不确定数据上的聚类分为未指定和指定聚类两种，故以下将分别给出其相应的聚类算法。

在未指定聚类情形，UA-UK-Means 和 UA-UK-Median 只是不确定数据与簇相异的表示方式不一样，所以把 UA-UK-Means 和 UA-UK-Median 可以综合考虑在一个算法中。

---

**算法 4.3:unassigned\_kcluster:**在未指定聚类中，相应的 UA-UK-Means 和 UA-UK-Median 算法

---

输入：簇的个数  $K$

不确定数据  $\mathbf{UD}$

输出：输出  $K$  个簇的集合  $\Phi = \{\mathbf{C}_1, \mathbf{C}_2, \dots, \mathbf{C}_k\}$

过程

---

---

```

(1)   Begin
(2)   从 UD 中随机选择 K 个不同的不确定对象 u, 并在每个 u. S 上选择一个不确定点 t,
      把 t. p 作为每一个簇的中心代表
(3)   Repeat
(4)       Cost ← 0    //目标代价
(5)       For each u ∈ UD
(6)           For each t ∈ u.S
(7)               <pc, d> ← unassigned_φ(u,t)
(8)               把 t 指派到 pc 所对应的簇中
(9)               如果是 UA-UK-Means 时, Cost ← d2 × t.p + Cost //累计期望代价
                  如果是 UA-UK-Median 时, Cost ← d × t.p + Cost //累计期望代价
(10)          EndFor
(11)       EndFor
(12)       For each C ∈ Φ
(13)           根据式 4.4 重新计算 C 的中心代表 pc
(14)       EndFor
(15)   Util 各个簇 C 的成员没有变化, 总目标代价 Cost 没有变化
(16)   End
    
```

---

**引理 4.3** 算法 UA-UK-Means 和 UA-UK-Median 的时间复杂度为  $O(K \times |u| \times |UD| \times r)$ , 其中  $r$  为算法迭代的次数,  $K$  为簇的个数。

**证明:** 如算法 4.3, 在(2)行中, 因为要选择  $K$  个簇的中心, 算法时间复杂度为  $O(K)$ ; (7)行的时间复杂度由引理 4.1 的分析为  $O(K)$ , (6)~(10)的 For 循环中, 因为要遍历  $u.S$  中的每一个  $t$ , 所以时间复杂度为  $O(K \times |u|)$ , 进而(5)~(11)的 For 循环要遍历所有的簇  $C$ , 所以综合的时间复杂度为  $O(K \times |u| \times |UD|)$ 。另外 (12)~(14)的 For 循环将遍历每一个簇, 故代价为  $O(K)$ 。假设(3)~(15)迭代了  $r$  次, 则整个算法的时间复杂度为  $O(K \times |u| \times |UD| \times r)$ 。所以总的时间复杂度为  $O(1) + O(K \times |u| \times |UD| \times r) = O(K \times |u| \times |UD| \times r)$ 。证毕。

在指定聚类中, 同样把 A-UK-Means 和 A-UK-Median 综合考虑在同一个算法

中。故得出算法 4.4 如下所示。

---

算法 4.4: assigned\_kcluster: 在指定聚类中, 相应的 A-UK-Means 和 A-UK-Median 算法

---

输入: 簇的个数  $K$

不确定数据  $UD$

输出: 输出  $K$  个簇的集合  $\Phi = \{C_1, C_2, \dots, C_k\}$

过程

- (1) Begin
  - (2) 从  $UD$  中随机选择  $K$  个不同的不确定对象  $u$ , 并在每个  $u.S$  上选择一个不确定点  $t$ , 把  $t.p$  作为每一个簇的中心代表
  - (3) Repeat
  - (4)      $Cost \leftarrow 0$    //目标代价
  - (5)     For each  $u \in UD$
  - (6)          $\langle pc, d \rangle \leftarrow assigned\_q(u)$
  - (7)          $Cost \leftarrow d + Cost$    //累计期望代价
  - (8)         For each  $t \in u.S$
  - (9)             把  $t$  指派到  $pc$  所对应的簇中
  - (10)         EndFor
  - (11)     EndFor
  - (12)     For each  $C \in \Phi$
  - (13)         根据式 4.4 重新计算  $C$  的中心代表  $pc$
  - (14)     EndFor
  - (15) Util 各个簇  $C$  的成员没有变化, 总目标代价  $Cost$  没有变化
  - (16) End
- 

**引理 4.4** 算法 A-UK-Means 和 A-UK-Median 的时间复杂度为  $O(K \times |u| \times |UD| \times r)$ , 其中  $r$  为算法迭代的次数。

**证明:** 在算法 4.4 中, (2)的时间复杂度为  $O(K)$ ; (6)的时间代价可根据引理 4.2 可知为  $O(K \times |u|)$ ; 同样(8)~(10)For 循环的时间复杂度为  $O(|u|)$ ; 进而可知



(5)~(11)循环逻辑的时间复杂度为  $O(K \times |u| \times |UD|)$ ，而(12)~(14)循环逻辑的时间复杂度为  $O(K)$ 。假设(3)~(15)的迭代了  $r$  次，则总的复杂度为  $O(K \times |u| \times |UD| \times r)$ 。证毕。

显然，算法 4.3 和算法 4.4 的时间复杂度区别在于  $r$  的大小，即迭代的次数，这主要取决于算法的收敛情况。

#### 4.4 PA-UK-Median 剪枝改进算法

算法 `assigned_φ(u)` 旨在为给定的不确定数据对象  $u$  指派一个相异度较小的簇。然而需要遍历每一个簇  $C$ ，以比较出对应相异度最小的簇，故根据引理 4.2，时间复杂度为  $O(K \times |u|)$ ，将会计算  $K \times |u|$  次不确定空间点  $t$  和簇中心代表的距离，代价比较高。然而该过程最终将被算法 4.4 调用  $|UD| \times r$  次。 $r$  为迭代的次数。

实际的算法运算中，如果  $u$  和某个簇  $C$  的相异度最小，就没有必要再计算  $u$  和其他簇的相异度。如果能通过某种方式把这些不必要的相异度计算过程剪枝剪掉  $h$  个，将会避免计算  $h \times K \times |u|$  次距离计算，而把其放入算法 4.4 的调用中分析，因为将要被调用  $|UD| \times r$  次，故总的能避免距离计算  $h \times K \times |u| \times |UD| \times r$  次。如果剪枝算法足够好，即  $h$  足够大，那么算法的效率将得到极大的提高。

令  $pc$  为簇  $C$  的中心代表，则表示不确定对象  $u$  到簇  $C$  的期望相异度为  $u$  到  $pc$  的期望相异度，记为  $Ed(u, pc)$ ，根据在指定聚类中，对 A-UK-Median 描述时，对映射函数  $\varphi(u, t)$  的分析，可得：

$$Ed(u, pc) = E\left(\sum_{t \in u.S} d(t, p, pc)\right) \quad (4.7)$$

则有如下定理：

**定理 4.2** 已知不确定对象  $u$ ，和两个簇的中心点代表  $pc$  和  $pc^x$ ，则有：

$$Ed(u, pc^x) \leq Ed(u, pc) + d(pc, pc^x) \quad (4.8)$$

$$Ed(u, pc^x) \geq \max\{0, Ed(u, pc) - d(pc, pc^x)\} \quad (4.9)$$

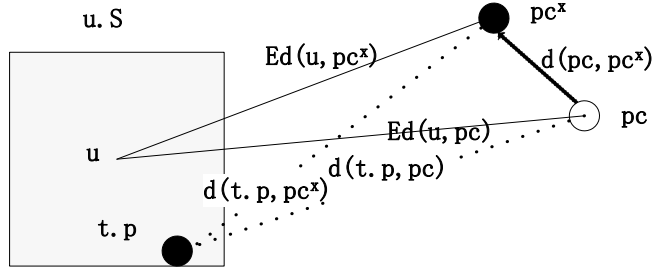


图 4.4 距离的三角不等原理

证明：给定  $\forall t \in u.S$ ，由 (4.7) 式和概率分布的数学期望定义可知

$$Ed(u, pc) = E\left(\sum_{t \in u.S} d(t.p, pc)\right) = \sum_{t \in u.S} (d(t.p, pc) \times t.c), \text{ 所以式(4.8)和(4.9)可等价的}$$

改写为：(a)  $\sum_{t \in u.S} (d(t.p, pc^x) \times t.c) \leq \sum_{t \in u.S} (d(t.p, pc) \times t.c) + d(pc, pc^x)$  以及

(b)  $\sum_{t \in u.S} (d(t.p, pc^x) \times t.c) \geq \max\{0, \sum_{t \in u.S} (d(t.p, pc) \times t.c) - d(pc, pc^x)\}$ ，以下将分别证明(a)和(b)：

1) 将(a)式移项得： $\sum_{t \in u.S} ((d(t.p, pc^x) - d(t.p, pc)) \times t.c) \leq d(pc, pc^x)$ ，又因为  $t$

为  $u.S$  上任意的一个不确定点如图 4.4 所示，根据三角不等式可得  $d(t.p, pc^x) - d(t.p, pc) \leq d(pc, pc^x)$ ，故式的左边项可进一步变换并有如下式：

$$\sum_{t \in u.S} ((d(t.p, pc^x) - d(t.p, pc)) \times t.c) \leq \sum_{t \in u.S} d(pc, pc^x) \times t.c = d(pc, pc^x) \times \sum_{t \in u.S} t.c$$

又因为  $\sum_{t \in u.S} t.c \leq 1$ ，所以左边  $\leq d(pc, pc^x)$ 。所以(a)式成立。

2) 对于(b)，首先左边是大于零的，故当  $\sum_{t \in u.S} (d(t.p, pc) \times t.c) - d(pc, pc^x) \leq 0$

时(b)成立。如果  $\sum_{t \in u.S} (d(t.p, pc) \times t.c) - d(pc, pc^x) \geq 0$ ，则需要证明：

$$\sum_{t \in u.S} ((d(t.p, pc^x) - d(t.p, pc)) \times t.c) \leq d(pc, pc^x), \text{ 这个式已在 1) 中证明成}$$

立。综上, (b)成立。

证毕。

根据定理 4.2, 在已知  $Ed(u, pc)$  和  $d(pc, pc^{pre})$  的前提下可以利用(4.7)对  $Ed(u, pc^x)$  的上限进行估计, 记为  $upper(u, pc^x) = Ed(u, pc) + d(pc, pc^x)$ 。同样可以利用(4.8)式对  $Ed(u, pc^x)$  的下限进行估计, 记为  $lower(u, pc^x) = \max\{0, Ed(u, pc) - d(pc, pc^x)\}$ , 即有  $lower(u, pc^x) \leq Ed(u, pc^x) \leq upper(u, pc^x)$ 。

**例 4.1** 利用定理 4.2, 可以利用 A-UK-Median 算法中上一次迭代的计算结果来估计当前迭代中的一些结果。如图 4.5, 已知不确定对象  $u$ , 在前一次迭代过程中与某个簇  $C$  的中心代表  $pc^{pre}$  的期望相异度  $Ed(u, pc^{pre})$ 。在当前的迭代过程中, 簇  $C$  的中心代表重新计算为  $pc$ , 并可计算两个中心的相异度距离  $d(pc, pc^{pre})$ 。利用定理 4.2 可以估计当前迭代中  $u$  与  $pc$  的期望相离度  $Ed(u, pc)$ , 即:  $upper(u, pc) = Ed(u, pc^{pre}) + d(pc, pc^{pre})$ ,  $lower(u, pc) = \max\{0, Ed(u, pc^{pre}) - d(pc, pc^{pre})\}$ , 所以  $lower(u, pc) \leq Ed(u, pc) \leq upper(u, pc)$ , 这样就能有效利用上一次迭代的结果来评估当前迭代。

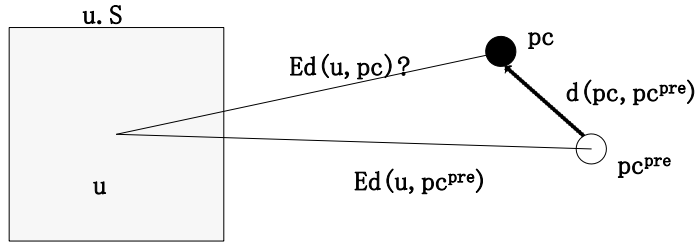


图 4.5 期望相离度计算的剪枝

在 A-UK-Median 算法中, 对任意的  $u$  进行簇指派的时候, 需要把  $u$  指派到与其期望相异度最小的簇, 因此需要对每一个簇进行比较。可以对每一个  $C_i$  的期望相异  $Ed(u, pc_i)$  进行估计得到  $upper(u, pc_i)$  和  $lower(u, pc_i)$ 。并记  $\min\_upper(u) = \min_{C_i \in \Phi} \{upper(u, pc_i)\}$ , 即表示  $u$  与所有簇  $C$  的期望相异度的最小上限值。

**定理 4.3** 给定不确定对象  $u$  和一个簇  $C_i$ , 如果  $lower(u, pc_i) > \min\_upper(u)$ , 则  $u$  一定不会被指派到簇  $C_i$ 。

**证明:** 根据  $\min\_upper(u)$  的定义, 总  $\exists C_x \in \Phi$ , 其对应的  $upper(u, pc_x) =$

$\min\_upper(u)$ , 再根据  $upper(u, pc^x)$  定义, 可知  $Ed(u, pc^x) \leq upper(u, pc^x) = \min\_upper(u) < lower(u, pc_i)$ , 又因为  $lower(u, pc_i) \leq Ed(u, pc_i)$ , 所以有  $Ed(u, pc^x) < Ed(u, pc_i)$ , 故  $pc_i$  对应的簇  $C_i$  不会为  $u$  归属的簇。证毕。

**例 4.2** 如图 4.6, 有一不确定对象  $u$ ,  $u$  到  $pc^x$  的下限估计为  $lower(u, pc^x) = 1$ , 上限估计为  $upper(u, pc^x) = 3$ , 同时  $u$  到  $pc$  的下限估计为  $lower(u, pc) = 4$ , 上限估计为  $upper(u, pc) = 5$ 。这样  $\min\_upper(u) = \min\{upper(u, pc^x), upper(u, pc)\} = upper(u, pc^x) = 3$ 。而  $lower(u, pc) = 4 > \min\_upper(u) = 3$ , 故  $u$  将不会被指派到  $p$  所在簇。所以对  $Ed(u, p)$  的计算将被剪枝剪掉, 以避免不必要的相异度计算。

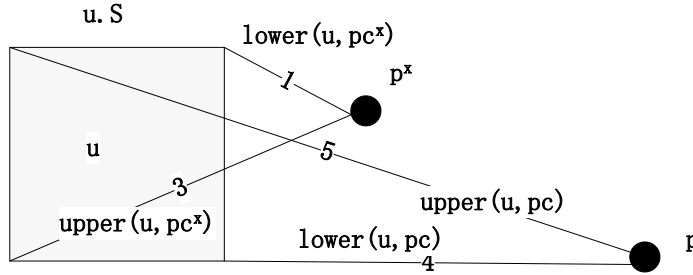


图 4.6 期望相离度计算的剪枝

综合定理 4.2 和定理 4.3, 例 4.1 和例 4.2 的分析, 可以利用上一次迭代计算的结果来估计当前迭代的期望相离度。然后利用估计的结果来对当前迭代的相离度计算进行剪枝。下面将给出相应的算法。

由于选择相离度最小的簇, 故首先需要改进簇的选择算法 4.5。在该算法中, 对于  $K$  簇聚类, 额外增加两个  $|UD| \times K$  个空间  $dist(u, C_i)$  和  $pc^{pre}(u, C_i)$  分别用于保存前一次计算过的  $u$  和  $C_i$  的期望相异度和对应的中心代表。并利用这些值来对算法进行剪枝。

---

**算法 4.5:** PAssigned\_φ(u): 在指定聚类中, 具有剪枝功能的相异度最小簇选择算法

---

输入: 不确定对象  $u$

簇集合  $\Phi = \{C_1, C_2, \dots, C_k\}$

输出: 与  $t$  相异度最小的簇  $C$  的中心点  $pc$ 。以及相异度大小  $d$ : 即返回  $\langle pc, d \rangle$

---

过程

```

(1)   Begin
(2)   初始化变量  $d \leftarrow \infty$ ,  $pc \leftarrow \text{NULL}$ ,  $\text{min\_upper} \leftarrow \text{MAX}$ ,  $C\_List \leftarrow \emptyset$ 
(3)   For each  $C_i \in \Phi$ 
(4)        $\text{min\_upper} \leftarrow \text{Min}\{\text{min\_upper}, \text{dist}(u, C_i) + d(pc^{\text{pre}}(u, C_i), pc_i)\}$  //寻找最小上限
(5)   EndFor
(6)   For each  $C_i \in \Phi$ 
(7)       选择  $C_i$  添加入  $C\_List$ , 其中  $C_i$  需要满足:
            $\text{MAX}\{0, \text{dist}(u, pc_i) - d(pc^{\text{pre}}(u, C_i), pc_i)\} \leq \text{min\_upper}$  //剪枝过程
(8)   EndFor
(9)   For each  $C_i \in C\_List$ 
(10)       $\text{temp} \leftarrow 0$ 
(11)      For each  $t \in u.S$ 
(12)           $\text{temp} \leftarrow d(t, p, pc_i) \times t.c + \text{temp}$  //计算期望相离度
(13)      EndFor
(14)       $\text{dist}(u, C_i) \leftarrow \text{temp}$  //更新计算距离
(15)       $pc^{\text{pre}}(u, C_i) \leftarrow pc_i$  //更新中心点
(16)      If ( $\text{temp} < d$ )
(17)           $d \leftarrow \text{temp}$ 
(18)           $pc \leftarrow pc_i$ 
(19)      EndIf
(20)   EndFor
(21)   Return  $\langle pc, d \rangle$ 
(22)   End

```

**引理 4.5** 具有剪枝技术的簇选择算法  $PAssigned\_ \varphi(u)$ , 假设有剪掉  $h$  个簇, 即  $C\_List$  的大小为  $K-h$  时, 且  $K \ll |u|$ , 时间复杂度为  $O((K-h) \times |u|)$ 。

证明: (2)的时间复杂度为  $O(1)$ ; (3)~(5)遍历每一个  $C_i$ , 查找最小的上限。时

间复杂度为  $O(K)$ ；同样(6)~(8)对簇进行剪枝，时间复杂度为  $O(K)$ ；(14)和(15)利用定理 4.2 对上限和下限进行更新，复杂度都为  $O(1)$ 。所以(9)~(20)的时间复杂度为  $O((K-h) \times |u|)$ ；(21)复杂度为  $O(1)$ 。综上，时间复杂度为  $2 \times O(1) + 2 \times O(K) + O((K-h) \times |u|) = O((K-h) \times |u|)$ 。证毕。

在剪枝的簇选择算法定义后，同时需要 A-UK-Median 改进为 PA-UK-Median 算法。如算法 4.6 所述。

---

**算法 4.6: PA-UK-Median: 在指定聚类中，相应 A-UK-Median 的剪枝版**

---

输入：簇的个数  $K$

不确定数据  $UD$

输出：输出  $K$  个簇的集合  $\Phi = \{C_1, C_2, \dots, C_k\}$

过程

- (1) Begin
  - (2) 从  $UD$  中随机选择  $K$  个不同的不确定对象  $u$ ，并在每个  $u.S$  上选择一个不确定点  $t$ ，把  $t.p$  作为每个簇  $C_i$  的中心代表  $pc_i$
  - (3) For each  $C_i \in \Phi$
  - (4) For each  $u \in UD$
  - (5)  $dist(u, C_i) \leftarrow d(u, pc_i)$ ,  $pc^{pre}(u, C_i) \leftarrow pc_i$  // 距离的计算和对应中心点
  - (6) EndFor
  - (7) EndFor
  - (8) Repeat
  - (9) Cost  $\leftarrow 0$  // 目标代价
  - (10) For each  $u \in UD$
  - (11)  $\langle pc, d \rangle \leftarrow PAssigned\_ \varphi(u)$
  - (12) Cost  $\leftarrow d + Cost$  // 累计期望代价
  - (13) For each  $t \in u.S$
  - (14) 把  $t$  指派到  $pc$  所对应的簇中
  - (15) EndFor
  - (16) EndFor
-

- 
- (17)        For each  $C_i \in \Phi$
- (18)                根据式 4.4 重新计算  $C$  的中心代表  $pc_i$
- (19)        EndFor
- (20)    Util 各个簇  $C$  的成员没有变化, 总目标代价 Cost 没有变化
- (21)    End
- 

算法 4.6 与 4.4 相比, 增加了对  $\text{dist}(\mathbf{u}, C_i)$ 、 $pc^{\text{pre}}(\mathbf{u}, C_i)$  的初始化, 和迭代计算。

**引理 4.6** 带有剪枝技术的 PA-UK-Median 的运行时间复杂度为  $O((K-h) \times |\mathbf{u}| \times |\text{UD}| \times r)$ , 其中  $r$  为迭代的次数; 与算法 4.4 相比, 空间复杂度将额外增加  $O(|\text{UD}| \times K)$ 。

**证明:** 算法 4.6 中, (2)对每一个簇进行初始化, 故时间复杂度为  $O(K)$ ; (3)~(7)对每个  $\text{dist}(\mathbf{u}, C_i)$ 和  $pc^{\text{pre}}(\mathbf{u}, pc)$ 进行初始化, 时间复杂度为  $O(K \times |\text{UD}|)$ ; (11)的时间复杂度在引理 4.5 中已得到证明为  $O((K-h) \times |\mathbf{u}|)$ , 而(13)~(15)的循环时间复杂度为  $O(|\mathbf{u}|)$ , 所以(10)~(16)的时间复杂度为  $O((K-h) \times |\mathbf{u}| \times |\text{UD}|)$ 。当(8)~(20)的迭代为  $r$  次时, 时间复杂度为  $O((K-h) \times |\mathbf{u}| \times |\text{UD}| \times r)$ 。综上, PA-UK-Median 的时间复杂度为  $O((K-h) \times |\mathbf{u}| \times |\text{UD}| \times r)$ 。与算法 4.4 相比, 减少了  $O(h \times |\mathbf{u}| \times |\text{UD}| \times r)$ , 如果  $h$  比较大, 即剪枝算法的效率高, 算法 4.6 将有很大的提高。同时与算法 4.4 相比, 需要额外分配空间来存储  $\text{dist}(\mathbf{u}, C_i)$ 、 $pc^{\text{pre}}(\mathbf{u}, C_i)$ 。而所有的  $\text{dist}(\mathbf{u}, C_i)$  需要  $O(|\text{UD}| \times K)$  的空间,  $pc^{\text{pre}}(\mathbf{u}, C_i)$  同理需要  $O(|\text{UD}| \times K)$ 。总的额外需要  $2 \times O(|\text{UD}| \times K) = O(|\text{UD}| \times K)$  的空间。证毕。

在以上的剪枝算法分析中,  $K$  一般较小,  $K \ll |\text{UD}|$ , 额外空间需要与不确定对象个数成线性增长。同时时间代价当  $h$  足够大, 时间效率将会有显著的提高。

## 4.5 MA-UK-Means 改进算法

对于 A-UK-Means 算法, 其同样调用簇选择算法  $\text{assigned}_\varphi(\mathbf{u})$ , 如同 4.4 节分析, 由于  $\text{assigned}_\varphi(\mathbf{u})$  算法的复杂度, 导致 A-UK-Means 算法的时间代价比较高。本节将对算法 A-UK-Means 进行改进, 并提出算法 MA-UK-Means 算法。

已知  $n$  维空间中的两点  $\mathbf{p}_1$ ,  $\mathbf{p}_2$ , 假定用欧几里德距离平方(即二阶范数)来表示  $\mathbf{p}_1$  和  $\mathbf{p}_2$  之间的相离度, 即:

$$d^2(\mathbf{p}_1, \mathbf{p}_2) = \|\mathbf{p}_1 - \mathbf{p}_2\|^2 \quad (4.10)$$

根据在指定聚类中，对 A-UK-Means 的分析，不确定对象  $\mathbf{u}$  和簇  $C$  的期望相异度为  $\mathbf{u}$  与  $C$  的中心代表  $\mathbf{pc}$  之间的期望相异度。可表示为

$$Ed^2(u, pc) = E\left(\sum_{t \in u..S} d^2(t.p, pc)\right) \quad (4.11)$$

已知  $\mathbf{u}$  为不确定数据集上的一个不确定对象，定义空间点  $\mathbf{um}$  为  $\mathbf{u}$  发生的概率质心点，表示为：

$$um = \frac{1}{u.e} \sum_{t \in u..S} t.p \times t.c \quad (4.12)$$

则有如下定理：

**定理 4.7** 给定不确定对象  $\mathbf{u}$  和一个簇  $C$  的中心代表  $\mathbf{pc}$ ，则  $\mathbf{u}$  和  $\mathbf{pc}$  的期望相离度可计算为：

$$Ed^2(u, pc) = Ed^2(u, um) + d^2(um, pc) \times u.e \quad (4.13)$$

**证明：** 根据式 (4.10) 和 (4.11)，式 (4.13) 可变换为：

$$\begin{aligned} \sum_{t \in u..S} (\|t.p - pc\|^2 \times t.c) &= \sum_{t \in u..S} (\|t.p - um\|^2 \times t.c) + \|um - pc\|^2 \times u.e, \quad \text{又因为 } \|t.p - pc\|^2 \\ &\Leftrightarrow \|(t.p - um) + (um - pc)\|^2 \Leftrightarrow [(t.p - um) + (um - pc)] \cdot [(t.p - um) + (um - pc)] \\ &\Leftrightarrow (t.p - um) \cdot (t.p - um) + (um - pc) \cdot (um - pc) + 2 \times (t.p - um) \cdot (um - pc) \Leftrightarrow \|t.p - um\|^2 \\ &+ \|pc - um\|^2 + 2 \times (t.p - um) \cdot (um - pc). \end{aligned}$$

所以变换式的左边为：

$$\sum_{t \in u..S} (\|t.p - pc\|^2 \times t.c) = \sum_{t \in u..S} \left[ (\|t.p - um\|^2 + \|um - pc\|^2 + 2 \times (t.p - um) \cdot (um - pc)) \times t.c \right]$$

进而：

$$\sum_{t \in u..S} (\|t.p - pc\|^2 \times t.c) = \sum_{t \in u..S} (\|t.p - um\|^2 \times t.c) + \sum_{t \in u..S} (\|um - pc\|^2 \times t.c) + \sum_{t \in u..S} (2 \times (t.p - um) \cdot (um - pc) \times t.c)$$

即：

$$\sum_{t \in u..S} (\|t.p - pc\|^2 \times t.c) = \sum_{t \in u..S} (\|t.p - um\|^2 \times t.c) + \|um - pc\|^2 \times \sum_{t \in u..S} t.c + 2 \times (um - pc) \cdot \left( \sum_{t \in u..S} (t.p \times t.c) - um \times \sum_{t \in u..S} t.c \right)$$

又根据式 (4.12)：  $\left( \sum_{t \in u..S} (t.p \times t.c) - um \times \sum_{t \in u..S} t.c \right) = um \times u.e - um \times u.e = 0$ 。所以：



$$\sum_{t \in u.S} (\|t.p - pc\|^2 \times t.c) = \sum_{t \in u.S} (\|t.p - um\|^2 \times t.c) + \|um - pc\|^2 \times u.e, \text{ 此时左边等于右边。}$$

故式(4.13)成立。证毕。

根据定理 4.7，如图 4.7 所示，在计算给定的不确定对象 **u** 和 **pc** 的期望相异度时，可以转换为计算 **u** 到 **u** 的概率质心 **um** 的期望相异度和计算空间点 **um** 与 **pc** 相异度的问题。

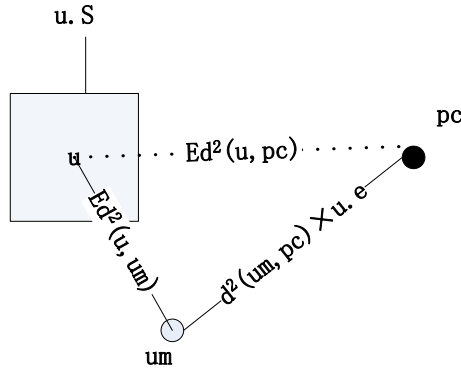


图 4.7 通过 **um** 计算  $Ed^2(u, pc)$

因为只要 **u** 已经确定，**um** 就可以通过式(4.12)进行确定。因此在每一次迭代的过程中，因为 **u** 和 **um** 的确定，所以  $Ed^2(u, um)$  也不会改变。因此只需要计算  $Ed^2(u, um)$  一次。而 **um** 和 **pc** 都是空间点，所以确定 **um** 的最相近的 **pc** 的代价远远没有直接确定 **u** 的最相近簇中心 **pc** 的代价复杂。

利用定理 4.7 及其分析，下面将给出一个算法 MA-UK-Means 对已有的 A-UK-Means 进行改进。首先给出该算法的簇选择算法如算法 4.7 所示。

---

算法 4.7: MAssigned\_φ(u): 在指定聚类中，改进算法 MA-UK-Means 的簇选择算法

---

输入：不确定对象 **u**

簇集合  $\Phi = \{C_1, C_2, \dots, C_k\}$

输出：**u** 的最相近簇 **C** 的中心点 **pc**。相离度 **d**，即返回  $\langle pc, d \rangle$

过程

---

---

```

(1)   Begin
(2)   初始化变量  $d \leftarrow \infty$ ,  $pc \leftarrow \text{NULL}$ 
(3)   For each  $C_i \in \Phi$ 
(4)       计算距离  $d^2(um, pc_i)$  并  $\text{temp} \leftarrow d^2(um, pc_i) \times u.e$  // 计算相离度期望
(5)       If ( $\text{temp} < d$ )
(6)            $d \leftarrow \text{temp}$ 
(7)            $pc \leftarrow pc_i$ 
(8)       EndIf
(9)   EndFor
(10)  Return  $\langle pc, d \rangle$ 
(11)  End

```

---

**引理 4.7** 算法  $M_{Assigned\_q}(u)$  的时间复杂度为  $O(K)$ 。其中  $K$  为簇的个数。

**证明：**该算法的(3)~(9)的循环遍历所有的簇  $C$ ，所以时间复杂度为  $O(K)$ ；而(2)和(10)步骤的复杂度都为  $O(1)$ 。综上，算法复杂度为  $O(K) + O(1) = O(K)$ 。证毕。

进而，改进算法  $MA\text{-}UK\text{-}Means$  可以容易给出如算法 4.8 所示：

---

**算法 4.8:**  $MA\text{-}UK\text{-}Means$ : 在指定聚类中，相应于  $A\text{-}UK\text{-}Means$  改进算法

---

输入：簇的个数  $K$

不确定数据  $UD$

输出：输出  $K$  个簇的集合  $\Phi = \{C_1, C_2, \dots, C_k\}$

过程

```

(1)   Begin
(2)    $\text{baseCost} \leftarrow 0$ 
(3)   For each  $u_i \in UD$ 
(4)        $um_i \leftarrow \vec{0}$ 
(5)       For each  $t \in u_i.S$ 
(6)            $um_i \leftarrow um_i + t.p \times t.c$ 
(7)       EndFor

```

---

---

```

(8)       $um_i \leftarrow um_i \times \frac{1}{u.e}, \text{ baseCost} \leftarrow \text{baseCost} + Ed^2(u_i, um_i)$ 

(9)      EndFor

(10)     从所有  $|UD|$  个  $um_i$  中随机选择  $K$  个作为  $K$  个簇的中心代表

(11)     Repeat

(12)         Cost  $\leftarrow$  baseCost    //目标代价

(13)         For each  $u_i \in UD$ 

(14)              $\langle pc, d \rangle \leftarrow MAssigned\_p(u_i)$ 

(15)             Cost  $\leftarrow d + \text{Cost}$     //累计期望代价

(16)             For each  $t \in u_i.S$ 

(17)                 把  $t$  指派到  $pc$  所对应的簇中

(18)             EndFor

(19)         EndFor

(20)         For each  $C \in \Phi$ 

(21)             根据式 4.4 重新计算  $C$  的中心代表  $pc$ 

(22)         EndFor

(23)     Util 各个簇  $C$  的成员没有变化, 总目标代价 Cost 没有变化

(24)     End
    
```

---

**引理 4.8** 改进算法 MA-UK-Means 的时间复杂度为  $O((|u|+K) \times |UD| \times r)$ , 其中  $r$  为迭代的次数; 与 A-UK-Means 相比, 额外的空间复杂度为  $O(|UD|)$ 。

**证明:** (2)的时间复杂度为  $O(1)$ ; (5)~(7)循环的时间复杂度为  $O(|u|)$ , (8)的时间复杂度在与  $Ed^2(u, um)$  的计算, 为  $O(|u|)$ , 所以(3)~(9)的时间复杂度为  $O(|UD| \times |u|)$ ; (10)的时间代价为  $O(K)$ ; 而(14)根据引理 4.7, 时间复杂度为  $O(K)$ , (16)~(18)的时间代价为  $O(|u|)$ , 所以(13)~(19)的代价为  $O(K \times |UD|) + O(|u| \times |UD|)$ , (20)~(22)的时间复杂度为  $O(K)$ , 假设(11)~(23)重复迭代了  $r$  次, 则总代价为  $O(K \times |UD| \times r) + O(|u| \times |UD| \times r)$ 。综合以上个步的分析, 算法的总时间代价为  $O((|u|+K) \times |UD| \times r)$ 。与算法 A-UK-Means 比较, MA-UK-Means 需要额外的  $|UD|$  个  $um$  来记录各个  $u$  的概率质心, 所以要增加  $O(|UD|)$  个空间。证毕。

由引理 4.4 可知 A-UK-Means 算法的时间复杂度为  $O(K \times |u| \times |UD| \times r)$  的复杂度，A-UK-Means 算法与之相比大大的提高了时间效率。同时额外空间的的增长，与  $|UD|$  呈线性增长，空间复杂度并不高。

## 4.6 小结

本章首先分析了不确定数据上的聚类的困难和挑战，然后基于不确定对象模型，对不确定数据上的聚类问题进行了描述，由于不确定数据的特殊性，把聚类问题分为未指定聚类和指定聚类两种情形。基于确定数据上的 K-Means 和 K-Median 算法提出了在不确定数据上的未指定聚类和指定聚类两个版本的相应算法。并分析了算法的效率，提出了对 A-UK-Median 的剪枝算法 PA-UK-Median 和对 A-UK-Means 的改进算法 MA-UK-Means，都极大的提高了算法的效率。

## 5 实验和性能分析

### 5.1 实验环境

实验运行在 Intel(R) Core(TM)2 Duo T7100 1.80GHZ 的处理器，2.0G 内存的 Window XP Professional 操作系统上。主要的用到的开发工具为 Visual Studio 2008，MATLAB R2008a。编程语言为 C Plus Plus。

### 5.2 实验数据

为了有效的评估不确定数据上的聚类算法的准确性，高效性以及可伸缩性，本文通过数据合成器来模拟真实世界的的数据不确定性。数据合成器主要有以下参数确定。

表 5.1 合成器参数

| 参数名        | 描述  |
|------------|---|
| numOB      | 非负整数，指明数据集中需要产生的不确定数据对象的个数  |
| dim        | 非负整数，指明不确定数据所在不确定空间的维数  |
| exist      | 在 0 到 1 间的实数，指明有 exist 比例的不确定对象的 $u.e < 1$  |
| numUP      | 非负整数，指明每一数据对象的不确定区域的不确定数据点个数范围，即 $ u.S $ 应在 1 到 numUP 之间  |
| Range      | 非负实数，不确定点的每个维的数据产生服从正态分布 $N(\mu, \sigma^2)$ ，且这些数据都必须在 $\mu - \text{Range}$ 到 $\mu + \text{Range}$ 之间 |
| miuRange   | 非负的实数，不确定点的每个维的数据产生服从正态分布 $N(\mu, \sigma^2)$ ，其中 $\mu \in [-\text{minRang}, \text{miuRange}]$ 。       |
| deltaRange | 非负实数，类似与 miuRange 的作用。 $\sigma \in [-\text{deltaRange}, \text{deltaRang}]$                            |

基于对现实世界数据的分布都服从正态分布的假设，对于每一个数据对象  $u$ ，不确定点  $t \in u.S$ ，且  $t = \langle p, c \rangle$ ，其中  $p$  为空间的点。数据合成器通过参数 dim 指定  $p$  的维度。用 numOB 来指定该不确定数据集的不确定对象的个数。而 exist 参数则指定了在整个数据集中，期望有  $\text{exist} \times \text{numOB}$  的数据对象  $u$ ，其  $u.e < 1$ ，即  $u$  的存在率小于 1。同时 numUP 指定了不确定对象的不确定区域  $u.S$  上的不

确定点的个数，即  $1 \leq |\mathbf{u}| = |\mathbf{u}.S| \leq \text{numUP}$ 。假设  $X$  表示不确定点  $\mathbf{t}$  的某个维值的随机变量，则  $X \sim N(\mu, \sigma^2)$  的正态分布，且  $\mu - \text{Range} \leq X \leq \mu + \text{Range}$ 。同时  $\mu$  为  $[-\text{miuRange}, \text{miuRange}]$  上的某个随机实数值； $\sigma$  为  $[-\text{deltaRange}, \text{deltaRange}]$  上的一个随机实数值。另外每一不确定点的发生率即  $\mathbf{t}.c$ ，为  $[0,1]$  上的一个随机数，且满足  $\sum_{\mathbf{t} \in \mathbf{u}.S} \mathbf{t}.c = \mathbf{u}.e \leq 1$ 。

通过对数据合成器的描述，下面将运用本文讨论的不确定数聚类算法，对不确定数据进行实验。

### 5.3 不确定数据聚类准确性实验

不确定数据聚类的准确率表示为：

$$\text{accuracy} = \frac{\text{正确识别的不确定对象个数}}{|\mathbf{UD}|}$$

其中  $|\mathbf{UD}|$  为不确定数据集中不确定对象的个数。

为了有效的衡量不确定数据上聚类的准确率，每一次实验需要合成器产生 3 个数据集， $\mathbf{UDreal}$ ， $\mathbf{UD}$ ， $\mathbf{UDrand}$ 。其产生步骤如下：

首先随机的产生一个包含  $\text{numOB}$  个不确定对象的不确定数据集  $\mathbf{UDreal}$ ，且通过指定  $\text{exist}=0.0$ ， $\text{numUP}=1$ ，使得任意的  $\mathbf{u} \in \mathbf{UDreal}$ ，满足  $\mathbf{u}.e=1$ ， $|\mathbf{u}|=1$ ，即  $\mathbf{u}.S$  只有一个不确定点值。此时，按 3 章的模型所述， $\mathbf{UDreal}$  退化为一个包含  $\text{numOB}$  个确定数据的确定数据集。把这个  $\mathbf{UDreal}$  看成数据的真实状况。

然后产生不确定数据  $\mathbf{UD}$ ， $\mathbf{UD}$  的产生是由数据  $\mathbf{UDreal}$  衍生出来的。需要重新指定合成器的参数，对每一个  $\mathbf{u} \in \mathbf{UDreal}$ ，因为  $|\mathbf{u}|=1$ ，假设  $\mathbf{u}.S$  唯一的不确定点为  $\mathbf{t}$ ，则用  $\mathbf{t}$  的每一维值作为正态分布的期望  $\mu$ ，再配合然后参数  $\text{deltaRange}$  确定正态分布的方差  $\sigma$ 。利用正态分布  $N(\mu, \sigma)$  产生  $[\mu - \text{Range}, \mu + \text{Range}]$  上的实数。把这个实数作为另一个不确定空间点  $\mathbf{t}_x$  的对应维的值。对同一个  $\mathbf{t}$ ，反复使用同样的方式，可以产生多个  $\mathbf{t}_x$ ，把这些  $\mathbf{t}_x$  作为另一个不确定对象  $\mathbf{u}_x$  的不确定区域  $\mathbf{u}_x.S$  上的点值。按这种方式，每个  $\mathbf{u}$  可以产生一个  $\mathbf{u}_x$ ，且  $|\mathbf{u}_x|$  大小由参数  $\text{numUP}$  确定， $\mathbf{u}_x.e$  由参数  $\text{exist}$  确定。这  $\text{numOB}$  个  $\mathbf{u}_x$  构成了不确定数据  $\mathbf{UD}$ 。

最后，数据集  $\mathbf{UDrand}$  的产生。对于  $\forall \mathbf{u}_r \in \mathbf{UDrand}$ ，满足  $|\mathbf{u}_r|=1$ ， $\mathbf{u}_r.e=1$ 。其中，这样  $\mathbf{UDrand}$  也退化为一个确定数据集。 $\mathbf{u}_r.S$  上的唯一不确定点  $\mathbf{t}_r$  随机来源

$u_x.S$  区域, 其中  $u_x \in UD$ 。即 UDrand 是通过 UD 衍生出来的。根据可能世界的语义, UDrand 是不确定数据 UD 的一个可能世界实例。

UDreal, UD, UDrand 之间的关系可以描述为: 首先把数据集 UDreal 看成真实世界的的数据。当对 UDreal 进行采集时, 由于不能准确的采集到真实的值, 对每一对象可能对应一个不确定区域, 这样可以描述为不确定对象 UD。而 UDrand 正是一次采集中, 对真实世界的一个反应。

准确性分析实验的步骤: 分别利用确定的聚类算法于 UDreal 和 UDrand 上, 同时利用本文相应的不确定聚类算法于 UD 上。把 UDreal 上的聚类结果作为聚类结果的标准, 分别用 UD 和 UDrand 上的聚类结果与之比较获得两个准确度。UDrand 的聚类结果的准确度作为 UD 上不确定聚类结果比较的基准(baseline)实验。由于数据的产生的是随机生成的, 故实验中, 在每一个固定参数情况下, 都将重复实验 10 次然后取结果的平均值进行分析评估。

### 5.3.1 numUP 对准确性的影响

$|u.S|$  为不确定对象的不确定区域中不确定点的数量, 即表示了不确定对象可以从  $|u.S|$  大小的集合上按照一定概率选取某个不确定点  $t.p$  作为发生值。正如 5.2 节对数据合成器的描述中, numUP 参数值指明了每一个产生的不确定对象的不确定区域集合  $u.S$  的大小范围。即每个  $|u.S|$  可以随机的选取 1 到 numUP 上的数字作为  $u.S$  上的空间数量。在实验中为了呈现 numUP 与聚类准确性的关系, 固定其他参数为一个特定的值, 变化 numUP 的大小。

在图 5.1 中, 显示的是 UA-UK-Median 算法准确性伴随 numUP 从值 50 变化到 250 的情况。实验中, 数据集 UD 是通过合成器根据某个 UDreal 数据集而合成产生的不确定数据。而 UDrand 是随机的从 UD 中抽取的一个确定数据集, 即 UD 的一个可能世界。本实验的基准实验是利用确定数据集上的 K-Median 算法对 UDrand 进行聚类实验, 并参考 UDreal 上聚类的结果计算准确度。同样利用 UA-UK-Median 算法在不确定数据集 UD 上实验, 参考 UDreal 的结果得出准确度。并对比两者的准确度情况。可以明显看出 UA-UK-Median 算法准确度一直都高于 UDreal 上的聚类结果。其中最好的情况是 UA-UK-Median 准确度的比基准实验高 8%。

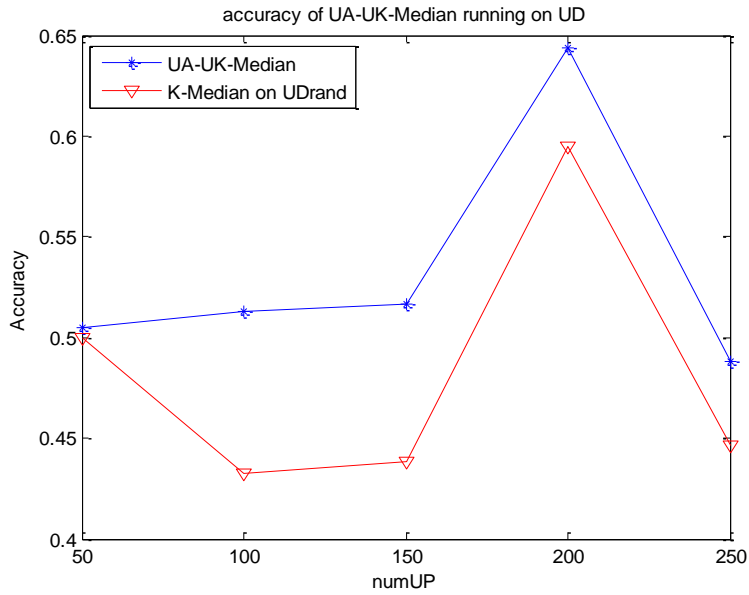


图 5.1 为数据合成器选取参数为 numOB=100, dim=2, exist=0.2, Range=5, miuRange=20, deltaRange=10, K=5 时, UA-UK-Median 算法准确率伴随 numUP 变化的情况

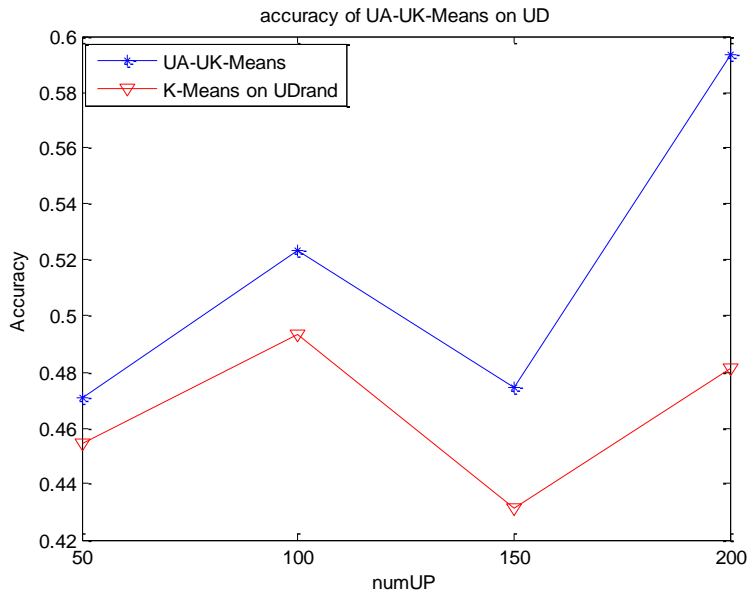


图 5.2 为数据合成器选取参数为 numOB=100, dim=2, exist=0.2, Range=5, miuRange=20, deltaRange=10, K=5 时, UA-UK-Means 算法准确率伴随 numUP 变化的情况

在图 5.2 中, 该实验主要针对 UA-UK-Means 算法的准确度随着 numUP 的变



化情况。基准实验是确定数据的 K-Means 算法在 UDrand 上的聚类情况，参考 UDreal 聚类情况，并计算其准确度与 UA-UK-Means 在 UD 上的聚类情况进行了对比。同样，UA-UK-Means 算法的准确度一致性的高于基准实验。并可以重实验中看出，伴随 numUP 的增长，UA-UK-Means 与基准实验的准确度差距更加明显。就 numUP 从 50 到 200 的变化过程中，准确度最大优于基准实验 11%。

在图 5.3 中，指定聚类中 A-UK-Median 算法相对于 numUP 变化的准确度的情况。基准实验与图 5.1 一样，利用确定数据上的 K-Median 算法对 UDrand 进行聚类实验。并比较两个算法的准确度差异。在 numUP 从值 50 到 200 的变化过程中，A-UK-Median 算法的准确度一致的高于基准实验的准确度。同样从图中可以看出，伴随着 numUP 的增加，两者的差距有逐渐变大的趋势。A-UK-Median 算法的准确度最大优于基准实验 14%。

在图 5.4 中，是对指定聚类中，A-UK-Means 算法伴随 numUP 变化的准确率的变化情况。基准实验与图 5.2 的情况一样，利用确定数据 K-Means 算法在 UDrand 上的实验结果，参考 UDreal 上聚类结果来评估实验的准确度。并与 A-UK-Means 算法在 UD 上的聚类结果。同样明显的有 A-UK-Means 算法的准确度高于基准实验。A-UK-Means 准确度最大优于基准 12%。

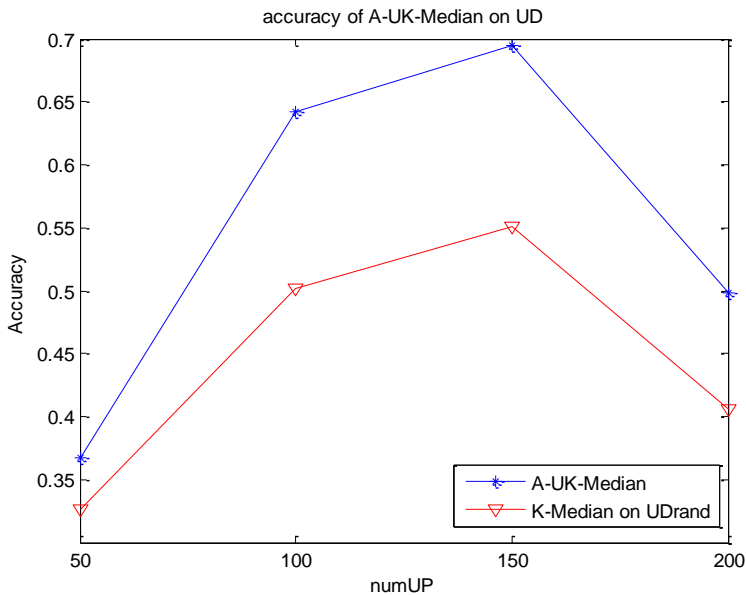


图 5.3 为数据合成器选取参数为 numOB=100, dim=2, exist=0.2, Range=5, miuRange=20,

deltaRange=10, K=5 时, A-UK-Median 算法准确率伴随 numUP 变化的情况

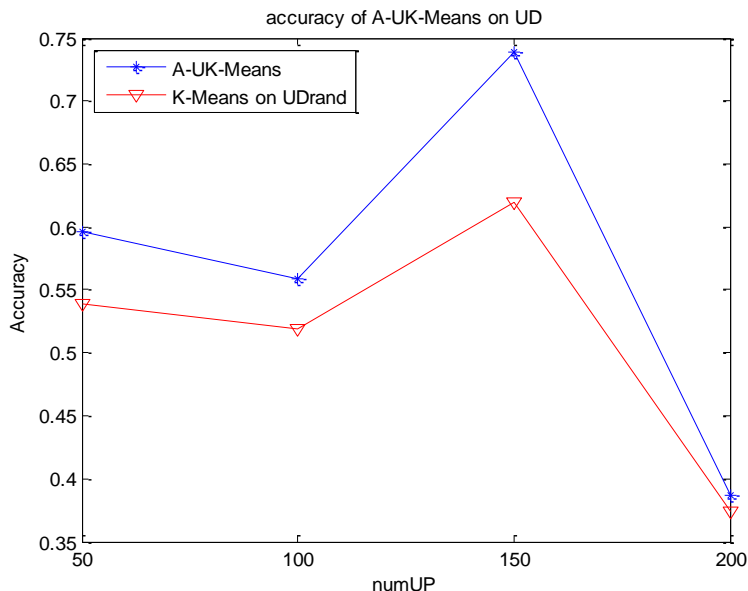


图 5.4 为数据合成器选取参数为 numOB=100, dim=2, exist=0.2, Range=5, miuRange=20, deltaRange=10, K=5 时, A-UK-Means 算法准确率伴随 numUP 变化的情况

### 5.3.2 聚类 K 值对准确度的影响

为了实验证明 K 值对不确定数据上的聚类算法准确度的影响, 所以固定数据合成器的所有其他参数, 以摒弃其对参数的聚类算准确度评估的干扰。本节不确定数据集 UD 上聚类准确度衡量同 5.3.1 节相同。聚类准确度参考确定数据集 UDreal 的真实聚类结果。并利用 UDrand 聚类结果作为相应不确定数据上聚类实验对比的基准。

在图 5.5 中, 主要显示了未指定聚类算法 UA-UK-Median 的聚类准确度伴随聚类簇的数量 K 变化的情况。基准实验是利用确定数据上的聚类算法 K-Median 在数据集 UDrand 上的聚类的准确性, 来与 UA-UK-Median 算法对不确定数据集 UD 的聚类情况进行比较。相比之下, UA-UK-Median 的准确度一致高于基准试验的结果。另外, 很明显在 K=1 时, 所有的数据都被归为一个类中, 这样它们的准确度都为 100%, 随着 K 增大, 簇的数量增多, 准确度也将逐渐降低。但当 K 值的增加, 准确度的下降逐渐平缓。UA-UK-Median 准确度的最大优于基准

实验 12%。

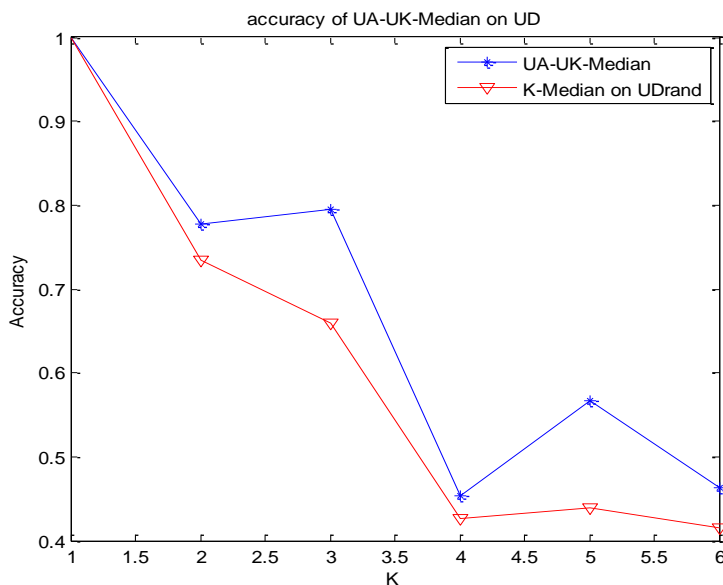


图 5.5 为数据合成器选取参数为 numOB=200, dim=2, exist=0.2, numUP=50, Range=5, miuRange=20, deltaRange=10, UA-UK-Median 算法准确率伴随 K 变化的情况

在图 5.6 中, 实验显示了未指定聚类 UA-UK-Means 算法在不确定数据集 UD 上聚类随着 K 值变化的情况。基准实验是利用 K-Means 算法在确定数据 UDrand 上的聚类的准确率实验。与图 5.5 一样, 在 K=1 时, 两者的准确度都为 100%, 随着 K 值的增加, 准确度逐渐下降, 且下降变得越来越平缓。UA-UK-Means 准确度最大优于基准实验 8%。

图 5.7 中, 实验显示了指定聚类算法 A-UK-Median 算法在不确定数据 UD 上, 随着 K 值增加的准确率变化情况。基准实验为算法 K-Median 在 UDrand 上的聚类准确度实验。实验显示在 K=1 是, 两者的准确率都达到了 100%。但随着 K 的增加, 准确度逐渐减小, 且减小的速度更加平缓。同时 A-UK-Median 算法的准确度一致高于基准实验。A-UK-Median 最大优于 K-Median 为 8%。

在图 5.8 中, 实验显示了指定聚类中 A-UK-Means 算法聚类随着 K 的增加的变化情况。基准实验是 K-Means 在 UDrand 上的聚类结果。相比下, A-UK-Means 准确度一致高与基准实验。在 K=1 时, 两者的准确率都为 100%, 但随着 K 值

逐渐减小，减小趋势趋近与平缓。A-UK-Means 实验中最优高于基准 9%。

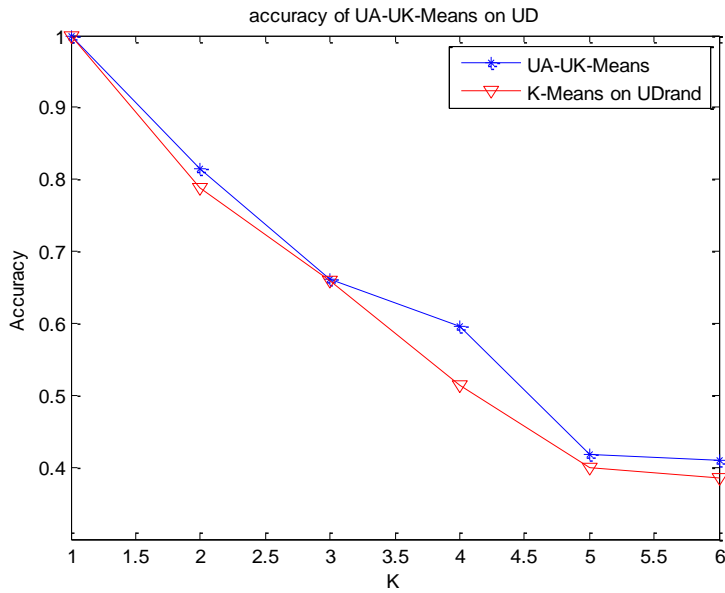


图 5.6 为数据合成器选取参数为 numOB=200, dim=2, exist=0.2, numUP=50, Range=5, miuRange=20, deltaRange=10, UA-UK-Means 算法准确率伴随 K 变化的情况

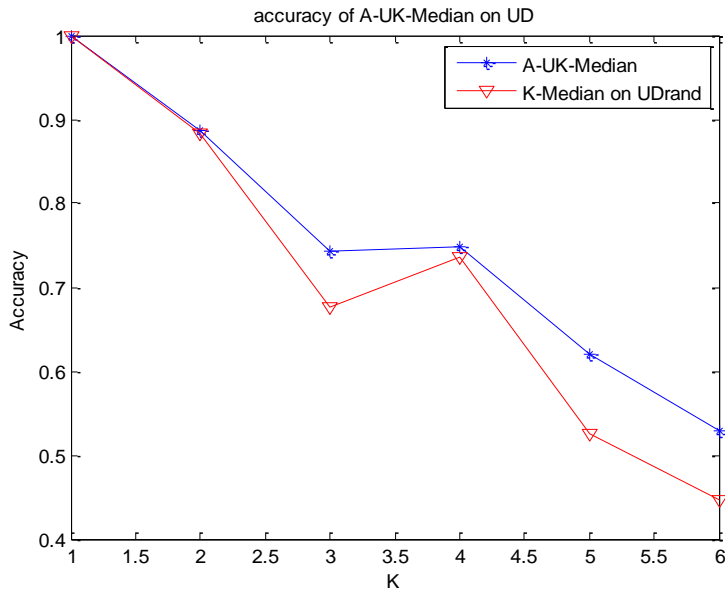


图 5.7 为数据合成器选取参数为 numOB=200, dim=2, exist=0.2, numUP=50, Range=5, miuRange=20, deltaRange=10, A-UK-Median 算法准确率伴随 K 变化的情况

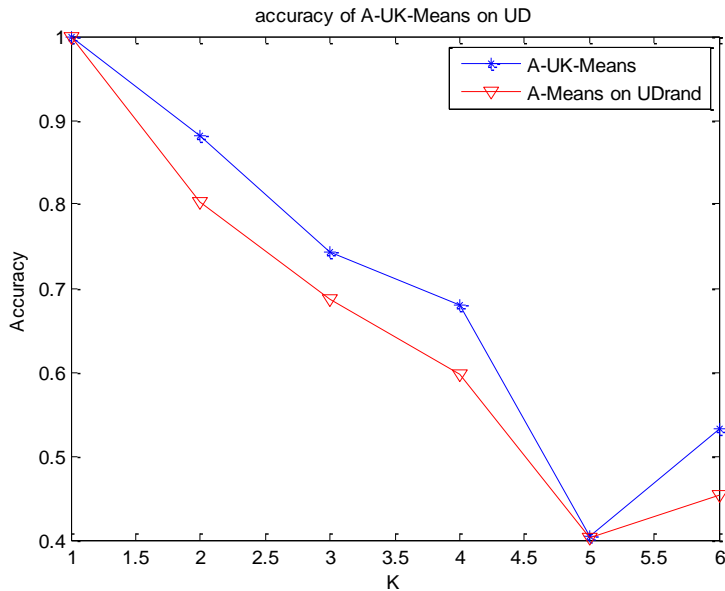


图 5.8 为数据合成器选取参数为 numOB=200, dim=2, exist=0.2, numUP=50, Range=5, miuRange=20, deltaRange=10, A-UK-Means 算法准确率伴随 K 变化的情况

## 5.4 剪枝算法 PA-UK-Median 实验分析

在指定聚类算法中, 剪枝算法 PA-UK-Median 算法是针对 A-UK-Median 提出的效率改进算法。

在图 5.9 中, 比较了算法 A-UK-Median 和 PA-UK-Median 算法在伴随 numUP 变化时的时间效率变化情况。可以看出, PA-UK-Median 算法的效率一致高于 A-UK-Median 的效率, 且随着 numUP 的增加, 两者的时间都成亚线性增长, 两者的时间效率差别有增加的趋势。在图中最好的情况 PA-UK-Median 提高基准实验的 50%。

在图 5.10 中, 比较了算法 A-UK-Median 和 PA-UK-Median 在 numOB 变化时, 两者的时间效率变化情况。可以较易看出 PA-UK-Median 算法效率一直高于 A-UK-Median。两者都随着 numOB 具有线性增长的趋势, 同时 PA-UK-Median 的效率将显得更优。最优的情况提高了相对基准效率的 25%。

在图 5.11 中, 显示了 A-UK-Median 和 PA-UK-Median 在 dim 变化时, 两者的时间效率变化情况。可以看出 PA-UK-Median 算法优于算法 A-UK-Median, 随着

dim 的增加，两者都有显著增加，但 PA-UK-Median 的效率优势将更加明显。

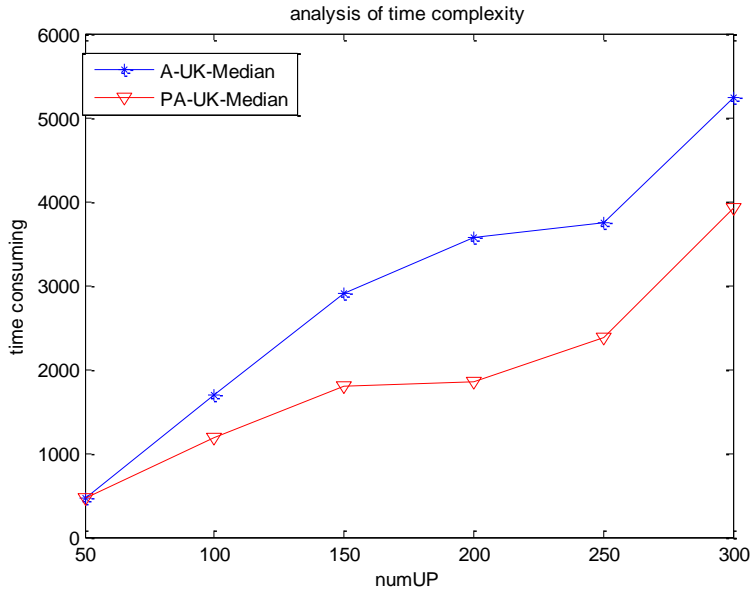


图 5.9 为数据合成器选取参数为 numOB=100, dim=2, exist=0.2, Range=5, miuRange=20, deltaRange=10, k=5 时, PA-UK-Median 算法效率伴随 numUP 的情况

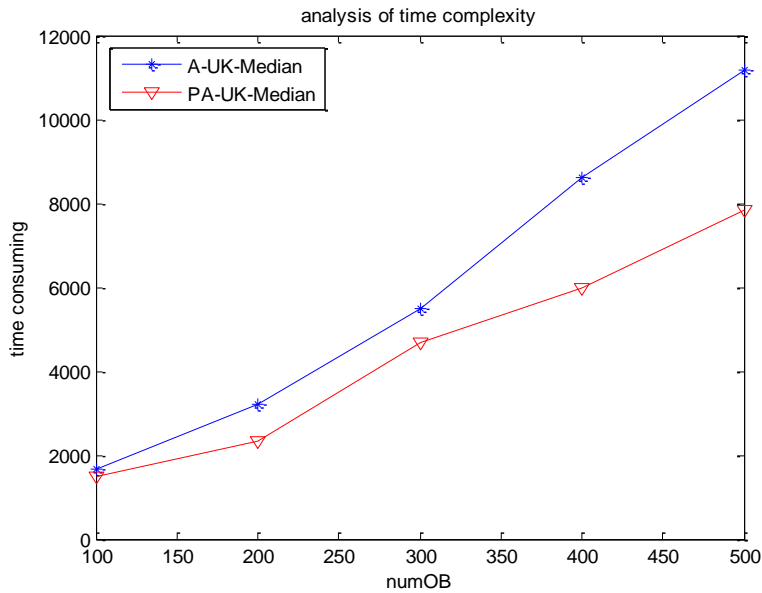


图 5.10 为数据合成器选取参数为 numUP=100, dim=2, exist=0.2, Range=5, miuRange=20, deltaRange=10, K=5, PA-UK-Median 算法效率伴随 numOB 变化的情况

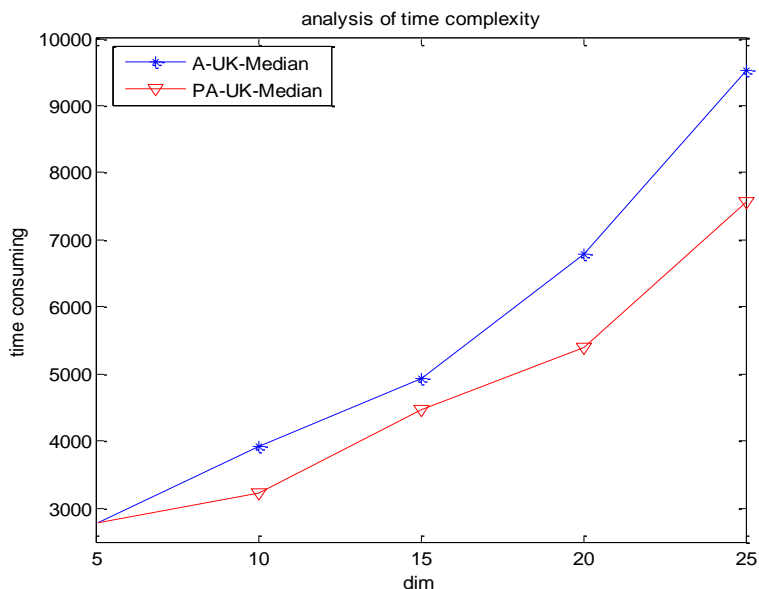


图 5.11 为数据合成器选取参数为 numOB=100, numUP=100, exist=0.2, Range=5, miuRange=20, deltaRange=10, K=5, PA-UK-Median 算法效率伴随 dim 变化的情况

## 5.5 改进算法 MA-UK- Means 实验分析

在指定聚类算法中, MA-UK-Means 是对算法 A-UK-Means 的改进算法。本节将实验分析算法效率的提高。

在图 5.12 中, 实验显示了算法 A-UK-Means 和算法 MA-UK-Means 在随着 numUP 不断变化过程中, 相应的效率变化情况。较易看出, MA-UK-Means 的效率始终高于算法 A-UK-Means。两者都相对 numUP 的增加而成亚线型增长趋势。且同时, MA-UK-Means 的效率优势将显得更加明显。实验结果中最优提高了 相对基准实验 10% 的效率。由于 MA-UK-Means 主要是避免了对不确定对象中的每一个数据点的计算, 所以当 numUP 增加将会更加明显的效率提高。

在图 5.13 中, 显示了 A-UK-Means 和 MA-UK-Means 算法在随着 numOB 变化的效率变化情况。MA-UK-Means 算法的时间效率始终优于 A-UK-Means 算法, 随着 numOB 的增加将体现得更加明显。实验中最优情况 MA-UK-Means 提高了 10% 的效率

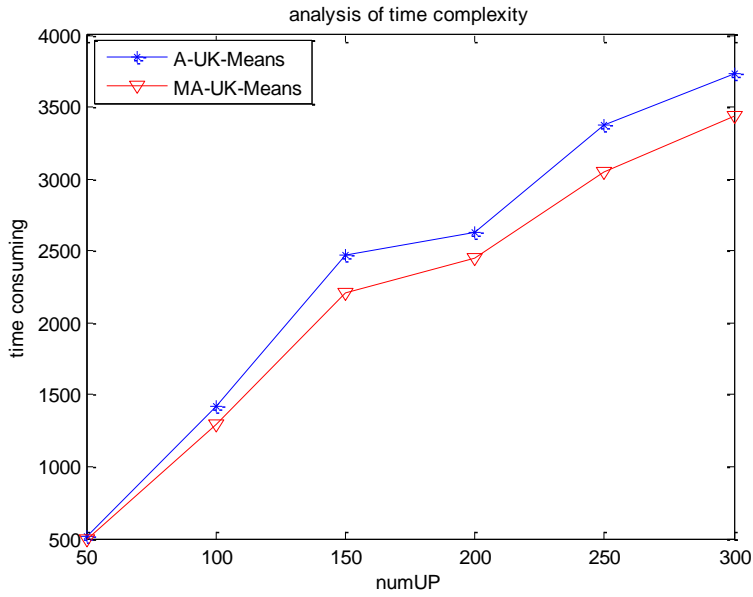


图 5.12 为数据合成器选取参数为 numOB=100, dim=2, exist=0.2, Range=5, miuRange=20, deltaRange=10, K=5, MA-UK-Means 算法效率伴随 numUP 变化的情况

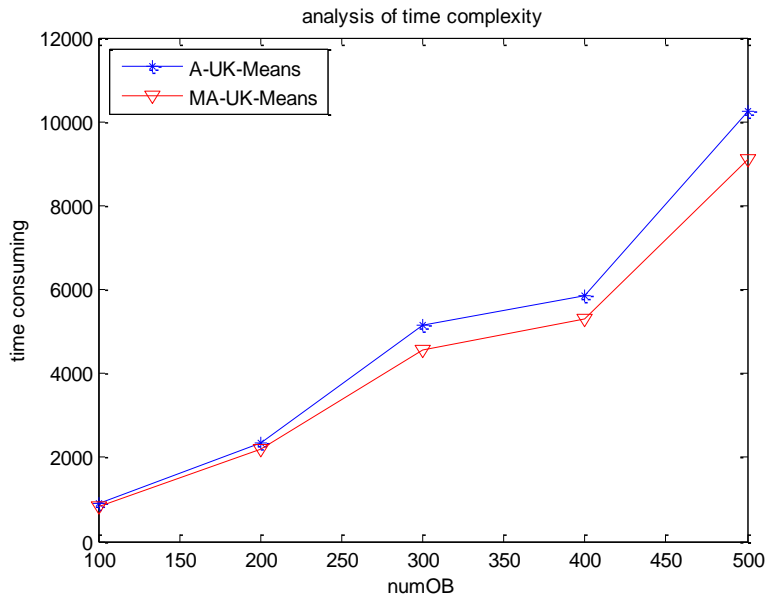


图 5.13 为数据合成器选取参数为 numUP=100, dim=2, exist=0.2, Range=5, miuRange=20, deltaRange=10, K=5, A-UK-Means 算法准确率伴随 numOB 变化的情况



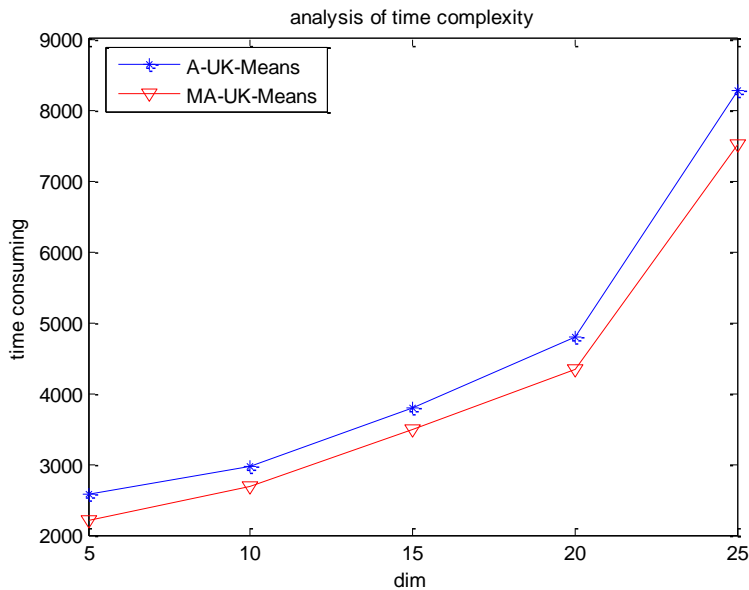


图 5.14 为数据合成器选取参数为 numOB=100, exist=0.2, numUP=100, Range=5, miuRange=20, deltaRange=10, K=5, A-UK-Means 算法准确率伴随 dim 变化的情况

在图 5.14 中, 显示了 A-UK-Means 和 MA-UK-Means 算法在随着 dim 增长的过程中效率的变化情况。MA-UK-Means 算法效率一直优于 A-UK-Means 算法。且随着 dim 的增加逐渐增加。增加的趋势变得更加陡峭。这主要因为维度的增加将大大的影响相异度的计算量。

## 5.6 小结

本章主要对第 4 章提及的算法, 进行了详细的实验评估和分析。主要从对不确定数据上的聚类算法 UA-UK-Median, UA-UK-Means, A-UK-Median, A-UK-Means 算法的准确度上来评估了算法的有效性。同时对算法的时间效率进行了比较实验分析, 剪枝算法 PA-UK-Median 的改进算法与 A-UK-Median 进行了对比实验评估, 显著的提高了 A-UK-Median 的时间效率。MA-UK-Means 对 A-UK-Means 进行了改进, 实验证明了改进算法在时间效率上显著的提高了 A-UK-Means 算法的效率。

## 6 结论及未来工作

本文针对不确定数据上的聚类算法进行了研究。首先描述了数据的不确定现象的普遍存在性。并提出了不确定对象的概念对不确定数据进行了有效的建模。基于不确定对象数学模型，对不确定数据进行了聚类分析。本文的工作包括：

- (1) 提出了不确定数据上的聚类算法 UA-UK-Median、UA-UK-Means、A-UK-Median 和 A-UK-Means 算法。
- (2) 基于合成数据，通过实验分析了以上不确定聚类算法在对不确定数据聚类分析时的准确度。并把对应的确定数据聚类算法 K-Median 和 K-Means 在不确定数据上聚类挖掘的结果作为实验对比的基准。实验表明显著的提高了不确定数据上的分析精度。
- (3) 由于在计算不确定数据间相似度时，要涉及到计算期望的相离度。而这个计算代价复杂，且被频繁计算。为了克服这个瓶颈，针对 A-UK-Median 提出了基于剪枝技术的改进算法 PA-UK-Median。对算法 A-UK-Means 分析提出了改进算法 MA-UK-Means。
- (4) 实验对比了 PA-UK-Median 和 A-UK-Median 算法，MA-UK-Means 和 A-UK-Means 算法的执行效率。实验表明了改进算法效率的明显得到提升。

本文只针对部分不确定数据聚类算法进行了探讨。在未来的工作中可以在不确定数据的背景下，对其他常用算法如分类，关联等挖掘算法进行研究。另外实验只是在合成数据上进行，把这些算法应用于大规模的真实环境数据中将会显得更加有意义。

## 参考文献

- [1] Michael Chau, Reynold Cheng and Ben Kao. Uncertain Data Mining: A New Research Direction[C]. In proceedings of the Workshop on the Science of Artificial, Dec 7-8, 2005
- [2] A. Motro. Management of Uncertainty in Database Systems[C]. In Modern Database Systems: the Object Model, Interoperability and Beyond (W. Kim, Editor), Addison-Wesley/ACM Press,1994, pp.457-476
- [3] Jiawei Han, Micheline Kambr. Data Mining – Concepts and Techniques[M]. Higher Education Press,2001:110-112
- [4] J.Pei, M. Hua, Y. Tao, and X. Lin. Mining Uncertain and Probabilistic Data: Problems, Challenges, Methods and Applications[C]. In proceedings of the 14<sup>th</sup> ACM SIGKDD International Conference on Knowledge Discovery and Data Mining(KDD'08). August 24-27, 2008, Las Vegas, NV, USA.
- [5] M. Hua, J. Pei, W. Zhang and X. Lin. Ranking Queries on Uncertain Data: A Probabilistic Threshold Approach[C]. In proceedings of the 2008 ACM SIGMOD International Conference on Management of Data(SIGMOD'08), June 11-14,2008, Vancouver, Canada.
- [6] Chua M., Cheng R., Kao B. and Ng J. Uncertain Data Mining: An Example in Clustering Location Data[C]. In proceedings of the Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD'06), Singapore, April 9-12, 2006, pp.199-204
- [7] Graham Cormode and Andrew McGregor. Approximation Algorithms for Clustering Uncertain Data[C]. In ACM Principles of Database Systems(PODS'08),2008
- [8] J. Ngai, B. Kao,C. Chui, et al. Efficient Clustering of Uncertain Data[C]. In IEEE international Conference on Data Mining (IEEE ICDM'06), HongKong, Dec, 2006.
- [9] S.Lee, B. Kao and Reynold Cheng. Reducing UK-means to K-means[C]. In the 1<sup>st</sup> Workshop on Data Mining of Uncertain Data(DUNE), co-located with IEEE ICDM, Ohama, US, Oct 2007
- [10] Chunqiu Zeng, Jie Zuo, Chuan Li and et al. MPSQAR: Mining Quantitative Association Rules Preserving Semantics[C]. In proceedings of the 4<sup>th</sup> International Conference on Advanced Data Mining and Applications(ADMA'08) pp.572-580, 2008
- [11] 周傲英, 金澈清, 王国仁, 李建中.不确定性数据管理技术研究综述. 2008

- [12] Reynold Cheng, Sunil Prabhakar and Dmitri V.Kalashnikov. Querying imprecise Data in Moving Object Environments[C]. In Proceedings of the international conference on Data Engineering (ICDE'03), pp.723-725.
- [13] Reynold Cheng,Dmitri V. Kalashnikov,et al. Evaluating Probabilistic Queries over imprecise Data[C]. In proceedings of the ACM Special interest group on Management of Data(SIGMOD'03), pp.551-562, June 2003.
- [14] J. Pei, B. Jiang, X. Lin, and Y. Yuan. Probabilistic Skylines on Uncertain Data[C]. In proceedings of the 33<sup>rd</sup> International conference on Very Large Data Bases(VLDB'07), Vienna, Austria, Sep. 23-28, 2007.
- [15] Q. Zhang, F. Li, K. Yi. Finding Frequent Items in Probabilistic Data[C]. In proceedings of 27<sup>th</sup> ACM SIGMOD International conference on Management of Data . Vancouver, Canada, June, 2008
- [16] K.Yi, F.Li,G. kollios, et al. Efficient Processing of Top-k Queries in Uncertain Databases[C]. In proceedings of 24<sup>th</sup> IEEE international conference on Data Engineering(ICDE'08), Cancun, Mexico, April 2008.
- [17] Smith Tsang, Ben Kao, Kevin Y. Yip, et al. Decision trees for uncertain data[C]. In 25<sup>th</sup> international conference on data Engineering (ICDE'09), Shanghai, China, 29 March-4 April 2009
- [18] Charu C. Aggarwal, Philip S. Yu. Outliner Detection with Uncertain Data[C]. In Proceedings of the 2008 SIAM International Conference On Data Mining(SIAM'08) 24-26 April, 2008
- [19] Hans-Peter Kriegel, Peter Kunath, Martin Pfeifle, et al. Probabilistic Similarity join on Uncertain Data[C]. In proceeding of 11<sup>th</sup> International conference on Database Systems for Advanced Applications (DASFAA'06), pp. 295-309,2006
- [20] T.S. Jayram, Satyen Kale, Erik Vee. Efficient Aggregation Algorithm for Probabilistic Data[C]. In proceedings of the 8<sup>th</sup> Annual Symposium on Discrete Algorithm, p.346-355. 2007
- [21] O. Benjelloun, A. Das Sarma,C. Hayworth, et al. An introduction to ULDBS and the Trio System[C]. IEEE Data Engineering Bllletin, Special Issue on Probabilistic Databases. 29(1):5-16, March 2006

- [22] Sato M., Sato Y., and Jain L. Fuzzy Clustering Models and Applications[C]. Physica-Verlag, Heidelberg 1997
- [23] Jin Cheqing, Yi Ke, Chen Lei, Yu Xu, Lin Xuemin. Sliding-window top-k queries on uncertain Streams[C]. VLDB, 2008
- [24] Soliman M A, Ilyas I F, Chang K C. Top-k query processing in uncertain databases[C]. In proceedings of the 23rd IEEE International Conference on Data Engineering. Istanbul, 2007:896-905
- [25] S.Arora, P.Raghavan, and S.Rao. Approximation schemes for Euclidean k-medians and related problem[C]. In proceedings of thirtieth annual ACM symposium on Theory of computing, pages 106-113. ACM Press 1998.
- [26] V.Arya, N.Garg, R.Khandekar, et al. Local search heuristic for k-median and facility location problems[C]. In proceedings of the thirty-third annual ACM symposium on Theory of computing, pages 21-29, ACM Press 2001.
- [27] D. Pfoser and C.S. Jensen. Capturing the uncertainty of moving-object representations[C]. In proceedings of the 6<sup>th</sup> international Symposium Advances in Spatial Databases(SSD'99), volume 1651 of Lecture Notes in Computer Science, pages 111-132, Hong Kong, China, 20-23 July 1999 Springer.
- [28] C.Aggarwal and P.S. Yu. Framework for clustering uncertain data streams[C]. In IEEE International Conference on Data Engineering, 2008
- [29] O.Benjelloun, A.D. Sama, A.Y.Halevy, et al. ULDBs: Database with uncertainty and lineage[C]. In international conference on Very Large Data Bases (VLDB 2006).
- [30] G.Gromode and M.N.Garofalakis. Sketching probabilistic data streams. In proceedings of ACM SIGMOD International Conference on Management of Data, pages 281-292, 2007.
- [31] N.N Dalvi and D.Suciu. Efficient query evaluation on probabilistic databases[C]. VLDB J.16(4):523-544
- [32] A.Kumer, Y.Sabharwal, and S.Sen. A simple linear time  $(1+\epsilon)$ -approximation algorithm for k-means clustering in any dimensions[C]. In IEEE Symposium on Foundations of Computer Science, 2004

## 本文作者在攻读硕士学位期间发表的文章

1. **Chunqiu Zeng**, Jie Zuo, Chuan Li, et al. MPSQAR: Mining Quantitative Association Rules Preserving Semantics. In proceedings of the 4<sup>th</sup> International Conference on Advanced Data Mining and Applications(ADMA'08) pp.572-580, 2008
2. LI Chaun, TANG Changjie, **ZENG Chunqiu**, et al. Discovering Multi-dimensional Major Medicines from Traditional Chinese Medicine Prescriptions In:BMEI'2008 international conference 2008
3. Liang TANG, Chang-jie TANG, Lei DUAN, Chuan LI, Ye-xi JIANG, **Chun-qiu ZENG** and Jun ZHU. MovStream: An Efficient Algorithm for Monitoring Clusters Evolving in Data Stream In:GrC'2008 IEEE International conference 2008
4. **曾春秋**, 唐常杰, 李川, 段磊. MPSQAR:无损语义的量化关联规则挖掘算法. 计算机科学与探索 2009.4
5. 张悦,唐常杰,李川,朱军,**曾春秋**,唐良,刘显宾. "出生缺陷监测数据中的朴素干预规则挖掘".计算机科学与探索.(2009年第2期,2009.4, Vol.1, 编号 T0808052), ZHANG Yue, TANG Changjie, LI Chuan, ZHU Jun, ZENG Chunqiu, TANG Liang, LIU Xianbin. "Mining Naive Intervention Rules in Birth Defect Data".Journal of Frontiers of Computer Science and Tehnology. 2009 年第 2 期, 2009.4
6. 唐良, 唐常杰, 姜页希, 李川, 段磊, **曾春秋**, 徐开阔. TRAODGrid: 基于 Grid 空间划分的高效离群轨迹检测方法. 第二十五届中国数据库学术会议, 计算机研究与发展, 第 45 卷, 增刊, 2008 年 10 月, pp. 185-190.
7. 倪胜巧, 唐常杰, 王有为, 李川, 张悦, **曾春秋**, 唐良. 基于 GPU 的基因表达式编程性能提升技术. 第二十五届中国数据库学术会议, 2008, 计算机研究与发展, 第 45 卷,增刊, 2008 年 10 月, pp. 227-233.
8. 倪胜巧, 唐常杰, 曾旭晟, 乔少杰, **曾春秋**. E&AMode:一种新的基于引擎粒子系统的图像渲染模式. 四川大学学报(自然科学版) 2007 年 6 期
9. 谢贵霞,唐常杰,李川,**曾春秋**,张悦.基于混合文档频率的 Web 潜在联系查询方法. 四川大学学报(自然科学版). 2009 年

## 声明

本人声明所呈交的学位论文是本人在导师指导下进行的研究工作及取得的研究成果。据我所知，除了文中特别加以标注和致谢的地方外，论文中不包含其他人已经发表或撰写过的研究成果，也不包含为获得四川大学或其他教育机构的学位或证书而使用过的材料。与我一同工作的同志对本研究所做的任何贡献均已在论文中作了明确的说明并表示谢意。

本学位论文成果是本人在四川大学读书期间在导师指导下取得的，论文成果归四川大学所有，特此声明。

## 致谢

时光荏苒，转眼间三年的研究生生活就要结束了。在细细思量自己三年来的收获所得之际，我要由衷地感谢在生活、学习等各方面所有关心过我、帮助过我，指导过我的亲人、老师和同学。

首先，我要特别感谢我的导师唐常杰教授。他以他严谨的治学态度、渊博的学识、独树一帜的教学方法和对学生耐心的谆谆教诲教会了我该如何做人、如何做研究。从他那里，我不仅仅学到了丰富的专业知识，更重要的是学习到了他的治学态度和做人的品质。使我建立了更加完善的人生观和价值观，感受到了如何去做一个踏实的人、勤勉的人、品格高尚的人。在此，谨向唐老师表示衷心的感谢！

其次，我要感谢的是我们实验室的师兄师弟师姐师妹们。我们的实验室是一个温馨和谐的集体，我很幸运能够生活在这个愉快、和睦的大家庭中。在此感谢各位老师和各位博士师兄的指导。感谢众多的师兄师姐师弟师妹，与你们一同度过的时光将成为我的美好回忆，在此，向你们表示诚挚的谢意！

最后，我要感谢我的父母。是他们的辛勤劳动给予了我进入川大学习的机会。没有他们，就没有我的现在。勤勤恳恳在工作岗位上工作的他们，为我的生活树立了最现实的榜样。同时也是他们给了我最无私最永恒不变的关怀、安慰、鼓励和支持！