

*用学校提供的蓝色打印*

**本科毕业设计（论文）说明书**

*一号黑体，居中*

基于不确定性数据的PT-K查询算法的研究与实现

*小二号粗黑体，居中*

**学 院**   **软 件 学 院**

**专 业**  **软 件 工 程**

**学生姓名**  **何宇翔**

**指导教师** **杜 卿**

**提交日期** **2010年5月15日**

*小三号宋体，加粗*

*封面纸推荐用210g/m2的橙色色书*

*论文统一用A4纸打印。边距上 2.54cm，下 2.54cm，左 2.2cm，右 2.2cm。行距为固定值20磅。*

**华 南 理 工 大 学** *四号宋体，加粗，居中*

**毕 业 设 计 （论文） 任 务 书**

*小二号宋体，加粗，居中*

*正文小四号宋体，行距为固定值20磅*

兹发给 06级双专业 班学生 何宇翔 毕业设计（论文）任务书，内容如下：

1.毕业设计（论文）题目：基于不确定性数据的PT-K查询算法的研究与实现

2.应完成的项目：

（1）2010年3月1日前拟定提纲并提交开题报告

（2）2010年5月15日前完成论文初稿

（3）进行与论文题目相关的调研工作并收集相关的一手和二手资料

（4）参考外文文献资料并提交外文翻译译文

3.参考资料以及说明：

4.本毕业设计（论文）任务书于2010年3 月5 日发出，应于2010年5 月15前完成，然后提交毕业考试委员会进行答辩。

专业教研组（系）、研究所负责人 审核 年 月 日

指导教师 签发 年 月 日

毕业设计（论文）评语：

毕业设计（论文）总评成绩： *四号宋体*

毕业设计（论文）答辩负责人签字： *四号宋体*

年 月 日

摘 要

为了适应计算机发展的需要，国外从七十年代开始，建立了软件产业，使软件生产达到了“工程化、产品化、系列化和标准化”。我国软件产业虽然起步较晚，但发展迅速去年国内软件市场的总规模估计已达到800-900亿元，增长幅度高达40%。

金山软件创建于1988年，是中国领先的应用软件产品和服务供应商，从80年代末开始发展起来的民族软件企业，经历了20多年的风雨，足可以代表中国大部分软件企业发展的血泪史的浓缩版，十分具有代表意义。金山软件的海外市场，是从日本开始的，也是作为一个软件国际贸易的转折点。同时，国内很多软件企业都是将日本作为开拓的市场，因此金山软件在日本市场的国际化进程十分具有代表意义。本文依据金山软件公司拓展海外市场的例子研究软件行业今后的发展对策，从而促进我国软件业的软件国际贸易进程。

本文首先介绍了软件贸易的一些概念和理论，构成本文的理论基础；其次，分析我国软件贸易的基本情况；再次，分析软件国际贸易对金山软件公司的影响；最后，分析金山软件公司成功应对国外同类产品的竞争，成功开拓东南亚国家的空白市场。从以上的分析我们可以提炼出金山软件公司软件国际贸易中的一些经验和教训，为我国软件业的发展提供借鉴和指导。

**关键词**：软件贸易，金山软件公司，发展研究，软件业

Abstract

In order to meet the needs of computer development，foreign from the seventies onwards，the establishment of the software industry，making software production reached the "engineering，product oriented，serialization and standardization." Although the software industry in China started late，but rapid development last year，the total size of the domestic software market is estimated to reach 800-900 billion RMB，an increase of up to 40%.

Kingsoft was founded in 1988，is a leading application software products and services provider，from the late 80s developed the national software company，went through 20 years of wind and rain，enough to represent most of the tragic history of development of software enterprises condensed version，is a representative of significance. Kingsoft overseas markets，starting from Japan，are software as a turning point in international trade. Meanwhile，many domestic software enterprises are to Japan，to develop the market，Kingsoft in the Japanese market is a representative of the process of international significance. This article based Kingsoft expand overseas examples of the software industry in the future development of measures to promote China's software industry software process of international trade.

This paper introduces some concepts and software trade theory，the theoretical basis of this thesis; Secondly，the analysis of the basic situation of China's software trade; again，analysis software internationalization of trade on Kingsoft's impact; finally，the successful response Kingsoft similar foreign products competition，opening up gaps in the market in Southeast Asia. From the above analysis we can extract the software company Kingsoft software internationalization of trade in some of the experiences and lessons for the development of China's software industry to provide reference and guidance.

**Key words**: software trade，Kingsoft company，the development of research，software industry

**目 录**

[摘 要 I](#_Toc261862723)

[Abstract II](#_Toc261862724)

[第一章 绪论 1](#_Toc261862725)

[1.1 不确定数据的研究背景及意义 1](#_Toc261862726)

[1.2 研究现状及相关技术 2](#_Toc261862727)

[1.2.1 不确定数据的含义 2](#_Toc261862728)

[1.2.2 不确定数据研究成果分析 3](#_Toc261862729)

[1.2.3 不确定数据查询算法的特点 4](#_Toc261862730)

[1.3 论文研究内容及组织结构 5](#_Toc261862731)

[第二章 不确定数据集与查询处理综述 7](#_Toc261862732)

[2.1 引言 7](#_Toc261862734)

[2.2 U-TOPK算法 7](#_Toc261862735)

[2.2.1 U-TOPK算法基本思想 7](#_Toc261862736)

[2.2.2 U-TOPK算法分析 7](#_Toc261862737)

[2.3 U-KRANKS算法 8](#_Toc261862738)

[2.3.1 U-KRANKS算法基本思想 8](#_Toc261862739)

[2.3.2 U-KRANKS算法分析 8](#_Toc261862740)

[2.4 PK-TOPK算法 9](#_Toc261862741)

[2.4.1 PK-TOPK算法基本思想 9](#_Toc261862742)

[2.4.2 PK-TOPK算法分析 9](#_Toc261862743)

[2.5 SKYLINE查询算法 9](#_Toc261862744)

[2.5.1 SKYLINE算法基本思想 9](#_Toc261862745)

[2.5.2 SKYLINE算法分析 9](#_Toc261862746)

[2.6 本章小结 10](#_Toc261862747)

[第三章 PT-K查询算法理论研究 11](#_Toc261862748)

[3.1 引言 11](#_Toc261862750)

[3.2 术语定义 13](#_Toc261862751)

[3.3 PT-K查询算法定义 13](#_Toc261862752)

[3.4 PT-K查询算法统治集 14](#_Toc261862753)

[3.4.1 统治集属性 14](#_Toc261862754)

[3.4.2 统治集的位置概率 15](#_Toc261862755)

[3.5 处理多元规则 16](#_Toc261862756)

[3.5.1 多规则元组压缩 17](#_Toc261862757)

[3.5.2 前缀共享技术 18](#_Toc261862758)

[3.5.3 减枝技术 19](#_Toc261862759)

[3.6 本章小结 20](#_Toc261862760)

[第四章 PT-K查询算法的设计与实现 23](#_Toc261862761)

[4.1 引言 23](#_Toc261862763)

[4.2 算法流程分析 23](#_Toc261862764)

[4.2.1 运行效率分析 23](#_Toc261862765)

[4.2.2 设计目标 28](#_Toc261862766)

[4.3 不确定数据在PT-K查询算法中的实现 28](#_Toc261862767)

[4.3.1 数据样例分组 28](#_Toc261862768)

[4.3.2 样例实现策略 29](#_Toc261862769)

[4.4 运行平台与开发工具 29](#_Toc261862770)

[4.5 本章小结 29](#_Toc261862771)

[第五章 算法实验及性能分析 30](#_Toc261862772)

[5.1 实验环境和测试数据集 30](#_Toc261862774)

[5.2 实验设计 30](#_Toc261862775)

[5.2.1 数据源 30](#_Toc261862776)

[5.2.2 评估标准 30](#_Toc261862777)

[5.3 PT-K查询算法应用 30](#_Toc261862778)

[5.3.1 数据预处理 30](#_Toc261862779)

[5.3.2 阈值不同时的运行结果 30](#_Toc261862780)

[5.3.3 生成规则发生变化时运行结果 30](#_Toc261862781)

[5.3.4 元组数量发生变化时运行结果 30](#_Toc261862782)

[5.4 实验测试与分析 30](#_Toc261862783)

[5.4.1 TOP-K质量分析 30](#_Toc261862784)

[5.4.2 参数敏感性分析 30](#_Toc261862785)

[5.4.3 数据处理能力分析 30](#_Toc261862786)

[5.5 实验结论 30](#_Toc261862787)

[5.6 本章小结 30](#_Toc261862788)

[第六章 总结与展望 31](#_Toc261862789)

[结 论 32](#_Toc261862790)

[参考文献 33](#_Toc261862791)

[致谢 36](#_Toc261862792)

1. 绪论
   1. 不确定数据的研究背景及意义

近四十年来，传统的确定性数据( deterministic data)管理技术得到了极大的发展，就了一个数百亿的数据库产业。数据库技术和系统已经成为信息化社会基础设施建设的重要支撑。在传统数据库的应用中，数据的存在性和精确性都确定无疑。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( uncertain data)得到了广泛的重视。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却无法有效管理不确定性数据，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。

就计算机和网络发展角度来看，近些年来，随着计算机科学与技术的快速发展，计算机应用也越来越广。然而在这种快速普及和应用的同时，也面临着种种困难。主要的困难包括：知识的表示、信息的组织、软件的复用等。特别是由于因特网的快速发展，面对信息的海洋，如何组织、管理和维护海量信息并为用户提供有效的服务也就成为一项重要而迫切的研究课题。

互联网上的信息资源极为丰富，而且这些信息一直在不断地膨胀，乃至于有人将互联网称为史上最大规模的数据库。根据2009年1月中国互联网信息中心（China Internet Network Information Center，CNNIC）的调查报告，截至2008年底，中国网站总数为287.8万个，全国网页总数约为160.9亿，较2007年增长90%，网页字数为460，217，386，099KB。但是互联网数据的质量却不尽如人意，作为一个典型的分散管理系统，互联网中并不存在一个统一的信息发布机构，各网站均可自由发布和维护信息。因此，当信息维护机构不同、信息更新不及时、工作人员误操作时，极易导致不同数据源（或者同一数据源内部）对同一对象描述的不确定。

面向不确定性数据的查询技术的研究工作并非最近开展起来的，只是在最近几年才在更广的范围内得到更多的关注。早在20世纪80年代末期，就有学者关注这方面的内容，当时关注的焦点是如何对关系数据模型进行扩展。

大部分面向确定性数据的查询任务在不确定性数据环境中仍然具有现实意义，需要进行处理。一个比较有趣的现象是，在不确定性数据环境下，由于引入了概率维度，查询的种类反而会增加。元组的概率维度值从侧面反映了该元组的重要程度，因而影响着查询的定义。以top-k查询为例，在确定性数据处理领域，其意义清晰，返回秩函数的值最大的k个元组。但在不确定性数据管理领域，秩函数值仅是其中一项因素，概率值是表征元组重要性的另一因素。在此基础上，最近出现了多种面向不确定性数据的top-k查询，包括U-Topk、U-kRanks、PT-k和Pk-topk等。

总的来说，不确定数据在一些重要应用领域中是固有存在的，如传感器网络和移动物体追踪。在不确定数据上使用传统的查询方法会使查询结果出现偏差，不能满足用户的需求。因此，基于不确定数据的查询处理受到了越来越多的关注。本文主要针对PT-K算法进行研究和实现，并对实验数据进行分析。

* 1. 研究现状及相关技术

本章主要阐述了不确定数据的概念、相关理论以及不确定数据查询的相关理论，这些都是本文结合统计数据以及各家理论，分析PT-K查询算法的理论基础和依据。不确定数据概念的理解是本文论述的前提，只有深刻地理解不确定数据的定义和特征，才能就不确定数据的查询，乃至PT-K算法做更深层次的探讨。各学者所提出的相关理论为本文的撰写提供了基础素材和阐述思路，而仔细归纳前人的研究成果也是本文创新的基础，有关的理论在本章以下内容中均有详细说明。

* + 1. 不确定数据的含义

申德荣等人认为不确数据就是在可能世界中的不确定信息，而且不确定数据普遍存在，如人们只知道某一属性取值的范围或可能的取值，无法确定该属性的确切值。

裴健认为不确定数据是由于数据产生的过程不精确产生的，而且不确定数据存在于现在已知的大部分领域。

Trio中将不确定数据分为 exact和 inexact两种。有关 inexact数据的描述又有多种，如不确定的数据、 概率数据、 模糊集数据、 近似数据、 不完备数据和不精确数据等。

Motro将不确定数据分为不确定和不精确两类。不确定是指属性值的可信性，如根据疑难病人的各项检测信息得出可能的病症，其可信度不是100%;概率是指属性取某一值的概率，如心脏病人中吸烟者占75%，非吸烟者占 25%，其概率和为 1;肥胖者得心脏病的可能性为 0.7，但没有模糊度和为 1的约束;该病人的年龄在 20～25岁为近似数据;不完备的数据是指有信息丢失，如一部分病人的病例中没有记录病人的血型;不精确数据是指数据的取值可能是集合中的数据之一，等等。

文献中除了将非确定的数据定义为不确定和不精确数据外，还包括不完备、 模糊、 不一致、 不明确。其中除了不明确为语义模糊概念外，其他都涵盖了 Trio中的定义。

归纳已有文章中讨论的数据不确定性，本文把不确定数据分为如下几类：

**表1-1 不确定数据分类**

|  |  |
| --- | --- |
| 范围值 ( range)，如 John的年龄为[37，43] | Or\_set值，如 John的年龄或者为 37或为 43 |
| 否定值 ( negative)，如 John的年龄不是 37 | 概率值 ( probability)，如 John的年龄为 20岁的概率是016，为 22岁的概率是 0.4 |
| 未知值 ( unknown)，如 null值 | 模糊值 ( vague)，如 John为年轻人 |
| 模糊集值 ( fuzzy)，如 John的工资是 high的可能性为0.16，low的可能性为 0.3 |  |

在可能世界中，这些不确定数据普遍存在，并交叠在一起，如 John的年龄为[37，43]的概率为 0.8，而取[40，45]的概率为 0.2，这是将概率值与范围值结合表示属性值;再如 John为年轻人的可能性为 0.6，而为老年人的可能性为 0.3，这是将vague值与 fuzzy值结合表示数据的不确定性。

本文只针对数值型的不确定数据进行定义，并基于相应的定义给出相应的匹配规则。

* + 1. 不确定数据研究成果分析

下面介绍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，并对其进行分析。

**表1-2 知名大学以及公司的研究机构的不确定研究项目**

|  |  |
| --- | --- |
| 多伦多大学 | Conquer项目，研究针对不一致数据库的管理技术。 |
| 普度大学 | Orion项目，研究一个通用目的的不确定性数据库系统。 |
| 斯坦福大学 | Trio项目，研究不确定数据的世系分析。 |
| 康奈尔大学 | MayBMS项目，研究查询语言、表示与存储技术、支持数据清洗、高效的查询处理、更新等技术。 |
| 华盛顿大学 | MystiQ项目，研究内容包括数据模型、关系代数计算、数据库理论等。 |
| 英特尔/加州大学伯克利分校 | HeisenData项目，研究将不确定性查询技术以模块形式加入到确定性数据管理框架中，以增强系统性能。 |
| 马里兰大学 | Pro b DBs项目，研究概率聚集查询的计算方法,开发空间·时间概率数据库. |
| IBM Almaden | Avatar项目，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. |

由上表可以看出，不确定数据管理正成为一个研究热点，国外针对不确定数据研究已经进行得如火如荼，而国内则鲜有动静，因此，对不确定数据进行研究是十分有必要和具有前瞻性意义。

另外，关于国外研究成果方面，还有Ré与Suciu观察到不确定性数据广泛出现在诸多应用之中，并总结了不确定性数据管理所面临的巨大挑战。Dalvi 和 Suciu进一步从理论角度阐述不确定性数据管理的基础与挑战。Aggarwal与 Yu从算法与应用角度综述了不确定数据管理技术。Pei 等人回顾了近期不确定性查询处理方面的进展，特别是他们自己的工作，包括范围查询、skyline 查询与 Ranking查询等。周傲英等人以不确定数据管理框架为主线，综述了不确定性数据管理技术在数据模型、预处理与集成、存储与索引、查询处理等各个方面所取得的重要进展。

近年来，有关不确定性数据管理的文章频频现诸于主流的国际学术会议与国际期刊，例如SIGMOD（Special Interest Group for the Management of Data）、VLDB（Very Large Data Bases）、ICDE（International Conference on Data Engineering）、TODS（Transactions on Database Systems）、VLDBJ（Very Large Data Base Journal）、TKDE（Transactions on Knowledge and Data Engineering）等。此外，一些主流的国际会议也积极召开一些小型的研讨会，讨论热点话题。近期关于不确定性数据管理方面的研讨会有：IEEE ICDM会议的DUNE 2007，IEEE ICDE 会议的MOUND 2009，VLDB会议的MUD 2007和MUD 2008等。国际一流刊物VLDBJ和TKDE也将在2009年和2010年分别刊出一个关于该主题的特刊（Special Issue）。因此，不确定性数据正成为一个研究热点。

* + 1. 不确定数据查询算法的特点

不确定数据和确定数据有着巨大的不同，不确定数据有两方面的内涵，即各元组本身存在性的不确定性和各元组属性值的不确定性。元组本身存在性的不确定性可用概率p描述：即该元组存在的概率是p，不存在的概率是1-p。元组的属性值的不确定性有多种描述方式，最通用的方式是以概率密度函数描述属性值，也可以用一些统计值进行描述，例如方差等。传统数据模型无法准确描述不确定性数据，可能世界（Possible World）模型是描述不确定性数据的通用模型。该模型包含若干个可能世界实例，在各个实例中，一部分元组存在，剩余元组不存在。可能世界实例的发生概率等于实例内元组的概率乘积和实例外元组的不发生概率的乘积之积。所有可能世界实例的发生概率之和等于1。以图1为例。输入数据序列是3个相互独立的元组，存在概率分别是0.7、0.6和0.5，颜色表示各元组存在时的属性值。则共有23=8个可能世界实例，各实例的发生概率依赖于所包含的元组集合。例如，仅包含紫、绿二球的可能世界实例的发生概率等于(1-0.7)×0.6×0.5=0.09。

图1-1 可能世界模型实例

|  |  |  |  |
| --- | --- | --- | --- |
| 数据序列 | | | |
| 时间 | 1 | 2 | 3 |
| 彩球 |  |  |  |
| 概率 | 0.7 | 0.6 | 0.5 |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 1 |  | 0.21 |  | 5 |  | 0.06 |
| 2 |  | 0.09 | 6 |  | 0.09 |
| 3 |  | 0.14 | 7 |  | 0.14 |
| 4 |  | 0.21 | 8 |  | 0.06 |

毫无疑问，管理不确定性数据所面对的最直接的挑战，就是相对于数据库规模呈指数级增长的可能世界实例的数量。在图1中，当输入数据仅含3条记录时，就能够生成8个可能世界实例。那么假设元组独立的不确定数据库含N条记录，若各元组仅有存在级不确定性，可能世界的数目将达到2N个；当各元组还有属性级不确定性时，可能世界的数目会远远超过2N个。可见，简单列举所有可能世界实例的处理开销惊人，更何况还需要进一步处理各项复杂的查询了。部分应用还需要考虑元组相斥的情况，即两个元组无法共存的情况，使得查询处理的复杂度进一步提升。在此情况下，“罗列可能世界实例，计算基于该实例的查询结果，整合各实例的查询结果生成最终的答案”的处理方式显然是不可行的，迫切需要结合各种剪切、排序等技术以快速计算查询结果。

直观来看，不确定性数据与确定性数据的差异并不大，仅多了一个概率维度。这是否意味着可以将概率维度当作一个普通维度，再利用传统技术进行处理？实际情况要更为复杂。概率维度对不确定性数据管理的影响非常深远，体现在查询定义、存储与索引、处理过程、结果呈现等各个环节之中。首先，部分查询定义可能拥有概率参数，例如Pt-k查询（一种top-k查询）需要一个概率参数p，仅返回成为top-k成员的概率超过p的元组集合。其次，传统的索引技术（例如B+树、R树等）无法有效索引不确定性数据，需要开发新的索引技术。再次，处理过程需要充分考虑概率因素，许多算法在执行过程中会优先考虑高概率的元组。最后，查询结果也会包含概率信息。因此，概率维度不是普通的维度，它的出现改变了传统的数据处理模式。

* 1. 论文研究内容及组织结构

本文主要采用定量分析，从理论到实践、宏观与微观相结合的方法。同时，综合应用对比分析和实践研究。通过分析国内外不确定数据研究的现状，指出不确定数据查询亟待解决的问题；又通过针对PT-K算法进行一个深入的研究分析，将本文的理论应用到现实数据中去。最终，评估该算法的效率，力求使本文的研究成果既有理论意义，又有实际意义。

本文的结构和各章研究的主要内容如下：

第一章绪论首先说明了本文的研究背景和意义，接着解释了本文主要讲述的内容。并阐述了不确定数据相关的理论，还有分析不确定数据查询亟待解决的问题。

第二章主要是针对现在比较热门的查询算法，skyline查询和top-k查询进行简单的阐述，并且分析现有的不确定数据查询算法，为后面和PT-K算法进行比较奠定理论基础。

第三章主要针对PT-K查询算法进行深入的理论研究和分析，为实现该算法提供理论依据。

第四章主要是描述PT-K查询算法的设计与实现，分析和设计元组压缩、统治集概率、前缀共享和减枝技术这四个方面的模块，并且针对每一个方面的模块细节进行一个详细的阐述。

第五章主要是对该算法进行测试，接着进行实验，并分析其数据，评估其效率。

第六章是进行总结和展望。

最后是根据前文的实验结果得出结论。

1. 不确定数据集与查询处理综述
2. 1. 引言

不确定数据广泛存在于许多应用领域 ,如传感器网络和移动物体查询。由于不确定数据本身的特性，传统的查询处理方法已经不能高效地解决或无法解决不确定数据上的查询。因此，近些年来不确定数据上的查询处理成为研究的热点 ,并在 SIG MOD、VLDB和ICDE等国际顶级会议上涌现出一批相关的研究。

在数据存在不确定性的情况下，概率不确定性数据模型能够有效反映数据的不确定性信息，为不确定性数据的有效处理奠定基础。基于概率不确定性数据模型的概率性查询能够根据不确定性数据的不确定性区间及其上的概率分布函数得出比传统查询方法含有更多有用信息、具有一定概率保证、更可信的查询结果，但是也付出了更多的计算代价，计算方法更为复杂。本章根据不确定性数据查询方法的分类，介绍各种查询的计算方法，并对某些查询的计算方法提出改进；区别于传统精确数据的查询，不确定性数据查询往往存在一定的模糊性，本章从问题定义和算法等角度对不确定数据上top-k查询和skyline查询分别进行介绍。

* 1. U-TOPK算法
     1. U-TOPK算法基本思想

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。当我们要求Top-k返回的所有元组都来自同一个可能世界的时候，使用U-Topk查询是适合的。

* + 1. U-TOPK算法分析

定义 Uncertain Top–k Query (U-Topk）: 令D为一个不确定数据库，它的可能世界空间为PW={PW1,...,PWn}.令T={T1,...Tm}为一系列长度为k的元组矢量，对于每个Ti∈T：（1）Ti中的所有元组是按照得分函数F排名的，（2）Ti是非空可能世界的Top-k查询结果。基于F的U-Topk查询返回T\*∈T，其中

 （2-1）

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。当我们要求Top-k返回的所有元组都来自同一个可能世界的时候，使用U-Topk查询是适合的。

对于 U-Topk 查询, 使用OPTU- Topk算法，该算法的基本思路是:

1) 设置一个以搜索状态概率优先级排序的队列 Q, 初始化时插入 s0, 0 , 概率P ( s0, 0 ) =1。

2)当 Q不为空时不断执行下面操作:从 Q中弹出sl , i;如果 l = k时返回 sl , i,否则根据 i和 d的比较情况选择下一个要访问的元组 t ;分别对 t可以加入和不加入 sl , i两种情况,将 sl+1, i+1和 sl , i+1插回 Q。

* 1. U-KRANKS算法
     1. U-KRANKS算法基本思想

U-kRanks查询返回在各个级别中出现的总概率最大的元组。这些元组不一定是最可能成为Top-k矢量的，它们也不一定都出现在同一个可能世界，并且同一个元组有可能在结果集里面出现多次。这类查询适合那些对元组来自不同世界没有限制的查询。

* + 1. U-KRANKS算法分析

定义Uncertain k Ranks Query (U-kRanks）: 令D为一个不确定数据库，它的可能世界空间为PW={PW1,...,PWn}.对于i=1...K,令为一系列的元组，每个元组在一个基于得分函数F的可能世界中排名i.一个基于F的U-kRanks查询返回

  （2-2）

U-kRanks查询返回在各个级别中出现的总概率最大的元组。这些元组不一定是最可能成为Top-k矢量的，它们也不一定都出现在同一个可能世界，并且同一个元组有可能在结果集里面出现多次。这类查询适合那些对元组来自不同世界没有限制的查询。

对于 U- kRanks查询使用 OPTU- kRanks算法，该算法的基本思路是：在计算排名 i时,对于一个新来的元组 t ,计算其在所有可能世界在排名 i上出现的概率 Pti；如果 Pt i比目前答案的概率大,并且比将未见元组考虑进来时的概率也大,那么t是结果集中排名为 i的元组。

* 1. PK-TOPK算法
     1. PK-TOPK算法基本思想

Pk-Topk返回所有在可能世界实例中成为 top-k 的总概率最大的k个元组。Pk-Topk与PT-k查询算法很相似，只不过它没有用到阈值p将所有可能在top-k出现的元组输出，而是返回元组的个数是固定的k个。

* + 1. PK-TOPK算法分析

和PT-K算法一样，PK-TOPK算法比较容易理解，在不确定数据的可能世界模型中，PK-TOPK是将处于可能世界的前K个，而且只输出K个值，后面的值就不再输出。可以用以下表达式进行表述，T是可能世界模型中元组的集合，t是元组，



是对t元组进行top-k查询后输出的概率值，p是阈值。

（2-3）



* 1. SKYLINE查询算法
     1. SKYLINE算法基本思想

Skyline 查询能用于解决多准则决策(Muli-Criteria Decision-Making ,MCDM) 问题。给定一个确定性的 n-维数据集合 D ,任一点 d 可被表示为( d.D1 , …, d. Dn)。Skyline 查询返回数据集合 S ,S AD ,则 Pu∈S ,不存在其它点 v ,满足(1) 对于任一维度 i (1in) , u. Di v. Di ; (2) 存在一个维度j (1 j n) 使得 u.Dj <v.Dj 。

Skyline简单来说是定义在多属性对象集(U)上的集合，它由u中所有不被其它对象所“支配”的对象组成。所谓支配是指：如果说对象p支配q()，则p在任意属性上的取值都不“差”于q，且p至少在某一属性上取值“优”于q。

* + 1. SKYLINE算法分析

Skvline查询最早由Brozonyi等人在文献提出，它们还给了一个生动的例子来阐述skyline的应用意义:假设游客Laneelot要到美丽的巴哈马首都Nassau去旅游。他想选一家既能观赏到美丽海景又不是太贵的酒店。然而，鱼与熊掌不可兼得，这两个属性往往是互斥的:靠近海边的酒店费用往往比较昂贵，而费用低廉的酒店往往远离海边。要在旅游业数据库中找到哪一家宾馆对 Laneelot最合适将是个不小的挑战，但旅行社至少可以找出那些Laneelot可能感兴趣的酒店。所谓“可能感兴趣”的宾馆显然就就是那些不被其它宾馆所“支配”的宾馆，这些不被支配的宾馆就构成了skyline(由其外形而得名)。

Skvline只包含那些用户最可能感兴趣的对象。在最优化决策时，用户显然只需要考虑那些属于Skyline的对象，而不必再关心那些被过滤掉的对象。这样用户就可以在小规模的Skyline对象集上按照自己的偏好进行选择。自从Skyline查询在学术界被提出以来，它就一直是一个非常活跃的研究领域，并且被广泛应用于城市导航、多标准决策、数据挖掘可视化及用户偏好查询等领域。

* 1. 本章小结

面向不确定性数据的查询任务丰富。大部分面向确定性数据的查询任务在不确定性数据环境中仍然具有现实意义，需要进行处理。一个比较有趣的现象是，在不确定性数据环境下，由于引入了概率维度，查询的种类反而会增加。元组的概率维度值从侧面反映了该元组的重要程度，因而影响着查询的定义。以top-k查询为例，在确定性数据处理领域，其意义清晰，返回秩函数的值最大的k个元组。但在不确定性数据管理领域，秩函数值仅是其中一项因素，概率值是表征元组重要性的另一因素。在此基础上，出现了多种面向不确定性数据的TOP-K查询，本章的2.2到2.4节主要讲述的就是多种TOPK-K算法。

Skyline查询在数据挖掘领域以及多规则决策应用领域发挥着重要作用。近年来，随着传感器和移动物体数据库的发展，不确定数据日益增多，基于不确定数据的 skyline查询成为研究热点。因此在2.5节也对skyline查询进行了简单的阐述和分析。

本章主要是针对现在比较热门的几个查询算法，top-k查询和skyline查询进行简单的阐述，并且分析现有的不确定数据查询算法，为第六章实验和PT-K算法进行比较奠定理论基础。

1. PT-K查询算法理论研究
2. 1. 引言

文献提出了另一种 top- k查询——概率极限值 top-k查询 ( PT-k查询 )。采用的不确定数据模型也是基于可能世界语义模型。不确定数据在一些重要的应用是固有存在的，例如环境监控和移动对象跟踪。

PT-K算法是TOP-K查询算法在不确定数据上的一种类型，反映的是元组在查询中处于较有利的元组，并且提出阈值概念，增加了查询的可控性，比U-KRANKS和U-TOPK效率更高。因为U-Topk和U-kRanks查询都是对“排名敏感”，这就要求实例化所有的可能状态。而PT-k查询时不用理会各个元组的排列的，只要他们出现在Top-k的概率大于给定的阈值就可以了。因此PT-k查询不需要维护想U-Topk和U-kRanks查询那么多的状态，可以利用更加高效的算法来进行PT-k查询。本章主要研究文献不确定数据集上PT-K查询技术。

**例3-1** 传感器经常被用在侦察濒危动物分布的情况或者某个区域内某种动物移动情况。因此，由于传感器不敏感或者发生故障的概率较高，侦察不能保证到十分精确。但是，侦察的置信度通常是可以精确统计出来。表3-1显示的是一些传感器侦察熊猫的情况。当传感器侦察到动物时，会记录该动物出现的时间和逗留的时间。

在某些地点，目标是运动中的，所以需要设置多个传感器去侦察目标。有两个侦察器在同一个点（如，S206和S231等）同时侦察到目标，记录包括R2和R3，R5和R6。在这种情况下，最多只有一个传感器是正确的。

表3-1 侦察熊猫的记录

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| RID | 地点 | 时间 | 传感器编号 | 逗留时间（分） | 置信度 |
| R1 | A | 6/2/06 2:14 | S101 | 25 | 0.3 |
| R2 | B | 7/3/06 4:07 | S206 | 21 | 0.4 |
| R3 | B | 7/3/06 4:09 | S231 | 13 | 0.5 |
| R4 | A | 3/13/06 22:31 | S101 | 12 | 1.0 |
| R5 | E | 4/12/06 20:32 | S063 | 17 | 0.8 |
| R6 | E | 3/13/06 22:28 | S732 | 11 | 0.2 |

表3-1是带有可能世界语义的含义的。数据可以经过重新处理得出可能世界的集合。可能世界的集合可以通过一些约束规则进行划分。在表3-1中，R2和R3不能同时是真，R5和R6也是，因此我们可以认为他们在生成规则中，R2 ⊕ R3，还有R5 ⊕ R6。以下的表3-2就是展示了所有可能世界的集合。

表3-2 可能世界的集合

|  |  |  |
| --- | --- | --- |
| 可能世界 | 概率 | Top-2 逗留时间 |
| W1= {R1,R2,R4,R5} | 0.096 | R1,R2 |
| W2= {R1,R2,R4,R6} | 0.024 | R1,R2 |
| W3= {R1,R3,R4,R5} | 0.12 | R1,R5 |
| W4= {R1,R3,R4,R6} | 0.03 | R1,R3 |
| W5= {R1,R4,R5} | 0.024 | R1,R5 |
| W6= {R1,R4,R6} | 0.006 | R1,R4 |
| W7= {R2,R4,R5} | 0.224 | R2,R5 |
| W8= {R2,R4,R6} | 0.056 | R2,R4 |
| W9= {R3,R4,R5} | 0.28 | R5,R3 |
| W10= {R3,R4,R6} | 0.07 | R3,R4 |
| W11 = {R4,R5} | 0.056 | R5,R4 |
| W12 = {R4,R6} | 0.014 | R4,R6 |

表3-3 关于表3-1中数据TOP-2的值

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| RID | R1 | R2 | R3 | R4 | R5 | R6 |
| 概率 | 0.300 | 0.400 | 0.380 | 0.202 | 0.704 | 0.014 |

TOP-K算法经常被用来处理不确定数据，例如科学家对TOP-2在逗留时间上有兴趣。在不同的可能世界去回答这个问题将会十分的困难。表3-2就展示了所有可能世界的TOP-2记录。

但是我们对所有可能世界的TOP-2记录的出来的表3-3十分感兴趣，具体的方法会在3.3节进行阐述。这里说明以下表3-3，每一个记录的概率是通过该元组在所有可能世界在TOP-2出现的概率和。通过表3-3，将十分容易找到超过阈值的元组，举个例子，我们假如将p=0.35，则{R2，R3，R5}将会是PT-K查询后的结果。如果是U-TOPK查询的话则返回<R5，R3>，U-KRANKS查询则返回<R5，R5>。

我之所以选择PT-K算法进行研究的一个最大原因就是，U-TOPK和U-KRANKS查询算法都需要将所有可能世界实例化，同时不能避免元组重复出现，另一方面，PT-K算法可以找出概率较高的元组，而不是排名靠前的元组，PT-K算法适合更高效的方法，所以我将研究的重点放在了PT-K算法中。

在接下来的几节中，我们将针对这个算法进行一个详细的分析和叙述，用来奠定我们在实现该算法和实验的理论基础。我们将通过介绍文献不确定数据集上PT-K查询技术，展示一种比较高效的PT-K查询。

* 1. 术语定义

t: 元组，包含一个数据值和一个概率值

T: 包含一系列的元组，这些元组的出现概率Pr(t)>0

R: 生成规则明确了一系列的互斥元组，这些元组的概率加起来小于等于1，并且在一个可能世界实例，每个生成规则最多只能出现一个元组

: 有且只有一个元组的生成规则的集合

W: 一个可能世界，在该世界中，所有生成规则最多只出现一个元组

: 所有可能世界的集合

Sti: 统治集，排在t前面的元组的集合

： 元组偏序，说明经过排序函数f后，元组t’在元组t前面

* 1. PT-K查询算法定义

在一个确定数据中进行TOP-K查询会返回最好的K个结果，但是在不确定数据中显然是不一样的。一个很自然的办法就是去检索每一个可能世界，然后计算哪一个记录的值是超过阈值的，就好像例子3-1一样。但是这样的话效率很低，而且当不确定数据的元组数目较大的时候，一定会需要很大的空间和时间代价的。所以我们需要一种高效的方法去实现PT-K算法。

我们考虑用到可能世界语义模型，利用这个模型我们可以提高查询的效率，就好像文献。

一般地，不确定表 T包含许多元组,每个元组t属于T的概率是 Pr( t)。产生规则R规定了不能同时存在的元组，R：，R的概率 Pr(R) 为 R中所有元组的概率和。T上所有的产生规则构成了产生规则集合 RT。一个可能世界W是 T的子集。W的存在概率计算公式如式 (3-1) 所示。Pr(W)是所有可能世界的集合。



（3-1）



显然，对于一个可能世界W，Pr(W)>0，进一步，

一个生成规则的长度(length)就是这个规则的元组的数目，表示为|R|。

一个可能世界W就是T的一个子集，使得对每一个生成规则 ：如果Pr(R)=1，则R是独立元组，则|R∩W|=1；如果Pr（R)＜1则|R∩W|≤1。如果|R|>1，则R是一个多元规则，里面的元组是依赖关系，否则是互相独立的关系。

对于一个不确定表T，它的生成规则集合为，他的所有可能世界实例的数目就是那些概率为1的生成规则的长度与那些概率不为1的生成规则的长度加1的和的乘积。表示为：

（3-2）



一个Top-k查询表示为Qk（P，f），包含一个谓词P，一个等级排序函数f，和一个整型k>0。Qk（W）表示Top-k在可能世界W上查询Q返回的元组集合，它包含k个元组。一个传统的 top-k查询 Q可以直接应用到可能世界W上，用 Qk(W )表示结果集中的 k个元组。一个元组 t的 top-k概率计算公式如式 (3-3) 所示，在没有歧义的情况下我们简写为Prk( t)。则 PT- k查询结果集中所包含元组的 top- k概

率至少是p， p是给定的极限值。

（3-3）



PT-k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，他的结果集合是：

（3-4）



原始的算法需要实例化所有可能世界实例，这在时间和空间上的开销都是非常大的，所以需要探索一种更加高效的算法来避免搜索所有可能世界实例。

* 1. PT-K查询算法统治集
     1. 统治集属性

P(T)包含了所有满足查询的元组、元组的概率以及生成规则。去除不在P(T)的元组不会影响Top-k查询结果。因此，。我们在Top-k查询的时候只需要考虑P(T)。

在P(T)中的一个元组t是否成为Top-k查询结果只受那些排列在t前面的元组的影响。

统治集（dominant set）：对于P(T)中的一个元组t，它的统治集就是P(T)中所有排在t前面的元组的集合。表示为

**定理 1**统治集属性(The dominant set property)：对于一个元组t∈T，。

证明：假设每个元组都是互相独立的，则可以通过式（3-2）和（3-3）可以推出上面的定理。

统治集属性是经常用在PT-K查询的。PT-K算法通过P(T)的排序和元组t前面的元组预测出元组t的PT-K概率。多元生成规则不能用定理1，，不过通过下一节的元组压缩的技术可以进行元组处理，使其符合定理1的前提。

* + 1. 统治集的位置概率

我们首先假设所有元组都是互相独立的，L=t1……tn是P(T)中的所有元组并且是按照得分函数进行排序的。一个元组ti∈W(1≤i≤n)出现在一个可能世界的第j个位置当且仅当它的统治集有j-1个元组出现，Sti={t1,t2…ti-1}.

在一个可能世界中一个元组出现在第j个位置的概率，表示为Pr(ti, j)。在一个可能世界中Sti出现j个元组的概率表示为Pr(Sti, j)。

一般地，我们可以知道，且（）。

一个元组出现在一个可能世界的第j个位置的概率等于该元组出现的概率乘以该元组的统治集有j-1个元组出现在该可能世界的概率，公式为：

 （3-5）

显然，一个元组ti的TOP-K概率为这个元组出现在第1,2,3...k个位置的概率和，表示公式为：



（3-6）

显然，当的时候，一个元组ti成为Top-k的概率等于它本身的概率，也就是：

 （3-7）

**定理2**：对于，有：

1. 元组ti的统治集Sti出现0个元组的概率为ti-1不出现的概率与ti-1的统治集出现0个元组的概率的乘积，公式为：



（3-8）

1. 元组ti的统治集Sti出现j个元组的概率为ti-1出现的概率与ti-1的统治集出现j-1个元组的概率的乘积与ti-1不出现的概率与ti-1的统治集出现j个元组的概率的乘积的和，公式为：

（3-9）

证明：由于是基于相互独立的元组，所以定理2可以很容易从概率的基本定理推导得出。

定理2可以很高效的计算TOP-K概率，我们可以给一个简单的例子。

**例3-2** 我们可以给出一个按照得分函数进行排序的不确定数据表3-1求出该表每个元组的TOP-3概率

表3-4 排序后的不确定数据表

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| TID | t1 | t2 | t3 | t4 | t5 | t6 | t7 | t8 | t9 |
| Prob. | 0.7 | 0.2 | 1 | 0.3 | 0.5 | 0.8 | 0.1 | 0.8 | 0.1 |

我们首先初始化，知道，且，，同时可以很容易得出Pr3(t1)=0.7, Pr3(t2)=0.2, Pr3(t3)=1（因为

为了计算Pr3(t4)，我们首先利用定理2计算Pr(St3,0)=0, Pr(St3,1)=0.24, Pr(St3,2)=0.62，则由式子（3-9），我们可以计算Pr3(t4)=Pr(t4)( Pr(St3,0)+ Pr(St3,1)+ Pr(St3,2))=0.258

* 1. 处理多元规则

根据上述一节，我们知道我们如果要用定理2，就必须是所有元组都是互相独立的情况，由于元组之间还存在多元规则的关系，我们不能马上就是用定理2，我们必须处理多元规则，让所有元素都符合定理2的前提条件，这样就需要多规则元组压缩策略、前缀共享技术和减枝技术。

假设t1,t2…tn ∈P(T)是已经排序好的不确定数据表，对于ti，有2种情况让多元规则复杂化，我们可以从图3-1了解到。第一，规则R中有些元组可能会出现在ti前面，他们当然不能出现在同一个可能世界中，为了计算统治集的概率，我们需要关注R。第二，ti有可能是多元规则R中的其中一个元组。这两种情况，都会使定理2无法使用。我们需要将非独立的元组组成的统治集，转化成为独立元组组成的统治集。

图3-1 计算每一个元组的

情况1：ti比规则R的高

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | ti |  |  | |  |  | |  | |
|  | | |  | |

规则R

情况2：ti比规则R的所有多元规则要高

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | |  |  |  |  |  | ti |  |
|  | |  | |

规则R

元组压缩之后

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | tR |  |  | ti |  |

Pr(tR)=Pr(R)

情况3：ti被规则R的元组夹着

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  | ti |  |  | |  | |  |
|  | | | | |  | |

规则R

元组压缩之后

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | tR\_left |  | ti |  | |  | |  |  |
|  | | | |  | |

规则R

* + 1. 多规则元组压缩

考虑一下P(T)=t1,t2…tn ，P(T)是已经排序好的序列，就像如果i<j，则

我们如果计算，就是每一个元组在TOP-K的概率，同时已经知道多元规则R：1r1<···<rmn，则我们可以知道会出现一下几种情况（见图3-1）：

情况1：  ti的位置比R中任何一个元组的位置要高，根据定理1，R可以被忽略

情况2： ti的位置比R中任何一个元组的位置都要低，则R中最多有一个元组出现在可能世界。根据定理1，我们可以将规则R记为tR，且他的联合概率为Pr(R)。

情况3：  ti的位置是在R元组之间，令trm0∈R是排名比较高的元组，多元规则R，则被分为2个部分，左边是Rleft={tr1···trm0}，和右边部分Rright={trm0+1···trm}，元组ti的概率，只受左边部分的影响，右边部分可以忽略。

两种子情况也可能会出现，第一种如果ti R，就好像情况2一样，我们可以将ti左边的元组进行压缩，使其联合概率为Pr(tr1…rm0)= ；

第二种情况，如果ti R，ti=trm0+1，在可能世界之中，ti出现了，所以R中的其他元组都不会出现。根据定理1，这时在计算的时候只需要考虑排在t前面的并且不在R中的元组，也就是

**推论1：**多元规则元组压缩：

对于一个元组,如果，那么,，tR就是R中统治t的元组，，其他T中的生成规则依然留在T(R)。

**推论2：** 规则中的元组：

对于一个元组t∈R，P(t)=true并且|R|>1，，T'为不确定表，。

因此，对于一个元组，所有在的元组都是独立的，于是，这样就可以根据定理2来计算Prk(t)，并且只需要扫描一次T(t）。

我们可以根据得分函数将P(T)里面的所有元组进行排序，得到一个有序链表L。对于每个元组ti,扫描一次它前面的元组，得到一个压缩的统治集T(ti），在这个统治集里面所有的元组都是独立的，这样我们就可以使用定理2来计算Prk(ti)了，此时只需考虑集合里面的元组。这样，计算所有元组的Top-k概率的实际复杂度为O(n2），n是不确定表的元组的个数。

* + 1. 前缀共享技术

**例3-3** 我们使用在例子3-2中的表3-1，t6和t7是相邻的两个元组，但是，因为t5在T(t6)，不在T(t7)，所以计算它们的统治集的概率时候，t5是不被用到的。

同时的，T(t6)和T(t7)，共同有相同的子集{t1,t2,4,t3}，这个子集的统治集概率将会重复计算，那就是说，如果他们的前缀共享的话，我们将节省很多的时间成本，因为计算统治集的概率是PT-K查询算法中消耗最多的部分。

公式3-5表明，使用子集概率Prk(Sti,j)来计算Prk(ti)的时候，在Sti-1中的元组的顺序如何并没有关系。这就使得我们可以通过排序不同元组的统治集从而使得对应的子集的统治集概率值能够尽可能多的被共享。

首先，如果元组t的情况正如情况2的话，t是在T'里面，因此，t应该在出现情况3前进行了排序。

其次，如果有多个多元规则包含元组t，每一个规则Rj可以记为tRjleft，我们可以通过更新规则中的包含元组，尽量少的改变共享前缀，使其共享前缀尽量多，。

这时候有两种办法，一种是积极的方法，把所有独立的和完整的规则元组放在多元规则元组之前，把规则中多元规则按它们出现的先后来排序。

一种是消极的方法，尽可能多的利用T(ti-1)中的共有前缀，尽量不重新排序，只有当新来的元组会影响之前的元组的时候才要重排。

例3-4 图3-2展示的是一个不确定数据表T在排序后的情况，有2个多元规则R1 : t1t2t8t11和 R2 : t4t5t10，通过下面的图3-2，我们看到这两种前缀共享的方法不同的地方，特别要关注他们所消耗的资源。

可以看到，在开始的时候，他们的前3个是一致的；

到了t4的时候，积极的方法要重新排t3，T(t3)在前缀共享中无法使用，但是在消极的方法那里没有重排，所以T(t3)共享前缀，因此减少了消耗。

假设L为P(T)中所有元组的有序链表，L(ti)表示T(ti）中所有元组的有序链表，Prefix(L(ti),L(ti+1))为L(ti)和L(ti+1)的最长的共同前缀，那么，总共需要计算的子集概率的数目为：



（3-10）

图3-2 两种重新排序的方法

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | |  | | | | | | | | |  | | | | |
| t1 | | t2 | | t3 | t4 | | t5 | | t6 | t7 | t8 | | t9 | t10 | | t11 | |  |
|  | |  | | | | | | |

P(T)排好序的列表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 元组 | 积极方法 | | 消极方法 | |
| 前缀 | Cost | 前缀 | Cost |
| t1 | ∅ | 0 | ∅ | 0 |
| t2 | ∅ | 0 | ∅ | 0 |
| t3 | t1,2 | 1 | t1,2 | 1 |
| t4 | t3t1,2 | 2 | t1,2t3 | 1 |
| t5 | t3t1,2 | 0 | t1,2t3 | 0 |
| t6 | t3t4,5t1,2 | 2 | t1,2t3t4,5 | 1 |
| t7 | t3t6t4,5t1,2 | 3 | t1,2t3t4,5t6 | 1 |
| t8 | t3t6t7t4,5 | 2 | t3t6t7t4,5 | 4 |
| t9 | t3t6t7t1,2,8t4,5 | 2 | t3t6t7t4,5t1,2,8 | 1 |
| t10 | t3t6t7t9t1,2,8 | 2 | t3t6t7t9t1,2,8 | 2 |
| t11 | t3t6t7t9t4,5,10 | 1 | t3t6t7t9t4,5,10 | 1 |
|  | 总消耗：15 | | 总消耗：12 | |

* + 1. 减枝技术

一个PT-k查询只对那些Top-k概率超过阈值的元组感兴趣，因此可以通过剪枝的方法来避免检索所有的元组是否符合查询谓词。

一些已有的方法如TA算法批处理排序。如果这种办法，我们可以对P(T)进行有效的检索。现在，我们关注如何进行减枝，以达到减少排序排名比较后的元组。

因此，给出3种减枝方法，可以不用计算它的TOP-K概率值就能将其去掉。一次检索就能将P(T)中不能通过阈值P的元组去掉。

但是要注意的是我们仍然要检索排名较低元组t，因为它还是有可能超过阈值P的。

**定理3：** Prk(t)  Pr(t)，对于一个独立元组t，如果Prk(t)<p，那么对于任意

1. t’有，且Pr(t’) Pr(t)则Prk(t’)<p；
2. 对于，有，且Pr(R) Pr(t)则Prk(t’’)<p

要用定理3，则需要将比阈值小的最大的独立元组概率p保存，每次进行减枝时候，去检查元组是否小于这个值。

**定理4**： 对于一个多元规则R，如果有，且Pr(t’) Pr(t)，Prk(t)<p，则Prk(t’)<p

在上面减枝技术的基础上，每一个生成规则R，我们将他其中不能通过阈值P的元组的概率值记录，以后如果检索到同一个R中的元组，如果他的概率没有大过保存的概率，则直接跳过该元组，因为根据定理4，该元组TOP-K概率肯定就是小于阈值P。

**定理5：** 令A为一系列成为TOP-K概率超过阈值p的元组的集合。

如果 ，那么对于每个没有在A中的元组 有：，有Prk(t)<p



我们可以通过观察所有TOP-K概率的超过阈值的值和为k，



* 1. 本章小结

通过上面针对PT-K算法的改进和效率的优化，我们可以将整个算法的流程用伪代码写出来（图3-3）。对于多元规则R：R：1r1<···<rmn，令span（R）=rm-r1。当元组trl(1<lm)进入队列时候，我们需要移除tr…rl-1，还有计算他的统治集的概率。因此，最坏的情况就是对于每一个通过阈值p的元组，多要对每一个属于P(T)的元组进行一次计算，时间复杂度是

图3-3 伪代码

**Input:** 不确定数据表T，元组生成规则的集合，一个查询 Qk(P，f)阈值p；

**Output:** Answer(Q， p，T);

**Method:**

1:对P(T)进行检索排序

for each ti  P(T) do

2: 利用元组压缩计算集合 T(ti) ；

3:计算元组t的统治集概率和TOP-K概率 Prk(ti);

4: if Prk(ti)>p then 输出 ti;

5: 检查ti能否用于减枝技术；

6: if 所有在P(T)中的元组都没有通过阈值p then exit;

end for

我将会在第四章阐述如何实现该算法，并在第五章阐述进行了相应的实验，来证明这些定理的有效性。通常的结果是，我们只需要对整个P(T)的很小一部分元组进行检索就能得到PT-K查询的结果。

1. PT-K查询算法的设计与实现
2. 1. 引言

根据前几章的理论基础和PT-K查询算法的研究，我们接下来的工作就是对该算法进行设计与实现，以提高TOP-K算法的效率。首先我们先对上一章的结论进行分析，图4-1是由上一章得到的伪代码，接着会对该算法进行一个分析，看有哪些地方实现可以更加高效，还有如何避免数据的冗余，提高查询的效率这些将是本章的主要内容。

* 1. 算法流程分析

根据上一章的结论，我们可以从该算法的伪代码进行分析，设计高效的PT-K查询。首先我们先对上一章的结论进行分析，图4-1是由上一章得到的伪代码，本节将会对以下这个算法进行一个分析。

图4-1 伪代码

**Input:** 不确定数据表T，元组生成规则的集合，一个查询 Qk(P，f)阈值p；

**Output:** Answer(Q， p，T);

**Method:**

1:对P(T)进行检索排序

for each ti  P(T) do

2: 利用元组压缩计算集合 T(ti) ；

3:计算元组t的统治集概率和TOP-K概率 Prk(ti)；

4: if Prk(ti)>p then 输出 ti；

5: 检查ti能否用于减枝技术；

6: if 所有在P(T)中的元组都没有通过阈值p then exit；

end for

* + 1. 运行效率分析

我们首先分析在该算法中，每一个环节所需要的开销，然后计算哪一个开销可以减少或者可以避免。表4-1罗列算法中将会有开销的部分：

表4-1 PT-K算法中有开销的部分

|  |
| --- |
| 元组压缩 |
| 统治集概率计算 |
| Prk(ti)计算 |
| 减枝技术 |

接下来我们就分析该算法的效率。

在元组压缩开销方面，应该首先有一个前提，那就是一个元组最多存在于一个规则R之中，就是说生成规则R之间的元组不重合、不重复。

在元组压缩阶段，根据前面3.5.1节的阐述，我们将有两个元组压缩的推论，在该节中，我们用一个比较简单的例子去定性研究了他们。

但是，在实现过程中，我们无法预知后面的原则规则，所以在实现的时候需要有对象去记录元组所在的规则。现在我将各种情况如何处理罗列出来：

图4-2 如何处理多元组生成规则

情况1 元组t不在生成规则之中

情况1.1 生成规则在后面

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| RID | R1 | R2 | R5 | R3 | R4 | R6 |
| Confidence | 0.3 | 0.3 | 0.8 | 0.5 | 1.0 | 0.2 |

R=R3 ⊕ R6

情况1.2 生成规则全部在前面

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| RID | R1 | R2 | R5 | R3 | R4 | R6 |
| Confidence | 0.3 | 0.3 | 0.8 | 0.5 | 1.0 | 0.2 |

R=R2 ⊕ R3

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| RID | R1 | R2,3 | R5 | R4 | R6 |
| Confidence | 0.3 | 0.3+0.5=0.8 | 0.8 | 1.0 | 0.2 |

情况1.3 元组t在生成规则之间

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| RID | R1 | R2 | R5 | R3 | R4 | R6 |
| Confidence | 0.3 | 0.3 | 0.8 | 0.5 | 1.0 | 0.2 |

R2 ⊕ R3 ⊕ R6

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| RID | R1 | R2,3 | R5 | R4 | R6 |
| Confidence | 0.3 | 0.3+0.5=0.8 | 0.8 | 1.0 | 0.2 |

R2,3 ⊕ R6

情况2 元组t是规则R中的元组

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| RID | R1 | R2 | R5 | R3 | R4 | R6 |
| Confidence | 0.3 | 0.3 | 0.8 | 0.5 | 0.5 | 0.2 |

R=R2 ⊕ R4

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| RID | R1 | R5 | R3 | R4 | R6 |
| Confidence | 0.3 | 0.8 | 0.5 | 0.5 | 0.2 |

由上面的图可以很简单分析出元组压缩的伪代码（图4-3），并且可以通过观察发现他的复杂度为O(n2)，如图4-4

图4-3 元组压缩的伪代码

For 在序列化后的每一个元组t

For在统治集St中每一个元组t’

计算t’的统治集

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| RID | R1 | R2 | R5 | R3 | R4 | R6 |
| Confidence | 0.3 | 0.4 | 0.8 | 0.5 | 1.0 | 0.2 |

图4-4 元组压缩检索顺序

从上面的分析来看，元组压缩需要一定的空间和需要检索时间，每一次元组压缩都会让序列发生变化，我所采取的策略就是将整个序列保存在字符串中。这样检索的时候有几个好处：以流的方式保存数据，无论是存储还是查找，效率和速度都会有多提高，每次更新序列的时候，只需要将需要修改部分替换就行，每一个元组t关于更新元组压缩不确定数据表T时候的复杂度为O(n)。

关于前缀共享的方法，在第三章阐述了理论基础，现在将其模型化，用图4-5表示。

图4-5 前缀共享模型

SR4=R1, R2, R5, R3 SR4=R1, R5, R3, R2

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| RID | R1 | R2 | R5 | R3 | R4 | R6 |
| Conf. | 0.3 | 0.3  Reorder  R2 ⊕ R6 | 0.8 | 0.5 | 1.0 | 0.2 |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| RID | R1 | R5 | R3 | R2 | R4 | R6 |
| Conf. | 0.3 | 0.8 | 0.5 | 0.3  R2 ⊕ R6 | 1.0 | 0.2 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| RID | R1 | R5 | R3 | R4 | R6 |
| Conf. | 0.3 | 0.8 | 0.5 | 1.0 | 0.2 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| RID | R1 | R5 | R3 | R4 | R6 |
| Conf. | 0.3 | 0.8 | 0.5 | 1.0 | 0.2 |

SR6=R1, R5, R3,R4 SR6=R1, R5, R3,R4

共享前缀：R1 共享前缀：R1,R5,R3

假设L为P(T)中所有元组的有序链表，L(ti)表示T(ti）中所有元组的有序链表，Prefix(L(ti),L(ti+1))为L(ti)和L(ti+1)的最长的共同前缀，那么，总共需要计算的子集概率的数目为：



（4-1）

在这里可以看到，关于前缀共享的开销是一个代数级增长的关系，所以设计的时候针对其如何更快更新重排后的序列和更新共享前缀，使共享前缀更加多。

关于减枝技术的模型，我在这里简单阐述一下设计模型，见图4-6

图4-6 减枝技术模型

情况1 Ri 和Rj 都是互相独立，不在同一个R

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| RID |  |  |  |  | Ri |  |  |  | Rj |  |  |  |
| Conf. |  |  |  |  | 0.8 |  |  |  | 0.6 |  |  |  |

情况2 Ri 和Rj 不是互相独立，都在同一个R

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| RID |  |  |  |  | Ri |  |  |  | Rj |  |  |  |
| Conf. |  |  |  |  | 0.8 |  |  |  | 0.6 |  |  |  |

… Ri ⊕ Rj…

如果Pr(Ri)≥Pr(Rj)，那么，Prk(Ri) ≥ Prk(Rj)，如果Ri不在结果集中，则Rj也不在结果集中

通过上述模型阐述可以知道，减枝技术也是代数级的增长，我们每次都只需要将不符合结果集的最大的概率记录，每次和新加入的元组进行一个比较，就可以知道是否应该计算它的TOP-K概率或者统治集的概率。在这一步中，可以有效减少无关，或者减少计算概率的元组的数量。

以上的阐述就是关于压缩规则、前缀共享和减枝技术的开销计算，在实验中发现，这几个技术模型，都不是开销主要的方面，开销最大的是进行统治集的计算，特别是当统治集十分巨大的时候，如果前面元组数量超过100的话，时间复杂度是几何级的增长，所以如果针对统治集概率计算模型才是关键因素。

对于整个算法来说，它的复杂度有如下的分析。对于多元规则R：R：1r1<···<rmn，令span(R)=rm-r1。当元组trl(1<lm)进入队列时候，我们需要移除tr…rl-1，和trl进入队列。因此，最坏的情况就是对于每一个生成规则都要计算它的统治集概率值，复杂度为。而且最坏的情况是，对于计算每一个P(T)是否通过阈值，所有在P(T)的元组都要至少读取一次，时间复杂度是

* + 1. 设计目标

实现PT-K算法，并且根据该算法分析其可以改进的地方，结合上面的理论分析，我认为实现后应该达到的目标为：

1. 运行可靠
2. 操作简单
3. 容易获取精确的计算开销
4. 不实现界面，因为点击按钮等响应事件会影响时间开销
   1. 不确定数据在PT-K查询算法中的实现

为了精确计算该算法的效率，这一节中，将阐述一种简单的有效方法生成不确定数据。

* + 1. 数据样例分组

对于一个元组t，令Xt是一个指向在随机数构成的经过排序的不确定数据表的指针。如果t在TOP-K列表中，则Xt=1，否则Xt=0。显然，t的TOP-K的概率是Xt的期望，Prk(t)=E[Xt]。我们的对象就是生成一些不可能世界样例的集合S，计算在S中的期望Xt，E[Xt]。

我们用一些格式化的数据去替代。对于一个不确定数据表T和生成规则的集合，一个样例单元s就是一个可能世界。基于不确定数据表T的分布生成样例单元：每次生成一个样例单元s，就扫描T一次。一个独立元组ti在s中就有概率Pr(ti)，一个生成规则在s中的概率就是Pr(R)。如果s拿到生成规则R中的一个元组，那么元组trl(1<lm)就会有概率值 。一个样例单元s最多就只能拿到生成规则R的一个元组。



只要生成样例单元s，我们就可以计算它的TOP-K的元组。对于每一个在TOP-K序列的元组，它的Xt=1，其他元组则为0。

上面样例的生成过程在一个样例集合S中可以重复获得，那么样例集合S的期望ES[Xt]就可以近似E[Xt]。当样例样本的容量足够大的时候，近似的质量就会越好。

关于样本容量，可以通过以下的定理获取。



**定理6：** 如果对于任意和一个可能世界的样本S，如果

对于任意元组t有

证明：这个定理可以通过切尔诺夫界推导得出。

* + 1. 样例实现策略

上面所说的样本方法有以下两种实现策略：

第一种方法是：我们可以对P(T)进行排序然后放在列表L中，在样例单元s中的前k个元组就是该单元的TOP-K元组。因此，当生成一个样例单元，而不是扫描整个不确定数据表，只需要扫描列表L。只要样例单元中有k个元组的话，那么扫描可以结束。这种方法可以减少生成样例单元时候的开销，但是不影响样例的质量。为了看到效果，考虑一下不确定数据表中所有数据都是互相独立的。如果平均概率是，那么需要生成样例单元而扫描的元组有 。这样的话，在k很小的情况下，扫描数量远远少于|P(T)|。最坏的情况是，对整个列表L进行一次扫描。



第二种办法是： 一般的，在还没有达到基于定理6所得到的样本容量的时候，近似值已经非常精确。因此，可以对样例抽取的办法进行改进：我们可以在得出每一个样例单元之后才计算它的TOP-K概率。对于给定参数d>0和e>0，如果在最后的d个样例单元中，任意元组t的Xt期望小于e时候可以停止扫描。

在第五章的实验中，我们将证明PT-K算法和上面数据的处理办法的效率。

* 1. 运行平台与开发工具

操作系统： Windows7 64位旗舰版

CPU: 英特尔i5-430

内存： DDR3 1333 2G

开发工具： Microsoft visual studio team suite 2008

开发语言： C++

* 1. 本章小结

本章首先介绍了在PT-K查询算法的基本流程，以及根据该算法流程，分析和设计元组压缩、统治集概率、前缀共享和减枝技术这四个方面的模块，并且针对每一个方面的模型细节进行一个详细的阐述。然后分析不确定数据中的数据处理方法，提出了两种样本数据生成的方法模型。在样本数据处理阶段，尽量将P(T)的所有元组进行扫描，避免计算统治集概率，然后采取一种近似值的策略，尽量让数据精确的情况下进行计算，最后得出最终结果。本章是第五章实验的设计基础，第五章通过一些数据会对本章的设计模型进行一个效率分析和验证。

1. 算法实验及性能分析
2. 1. 实验环境和测试数据集
   2. 实验设计
      1. 数据源
      2. 评估标准
   3. PT-K查询算法应用
      1. 数据预处理
      2. 阈值不同时的运行结果
      3. 生成规则发生变化时运行结果
      4. 元组数量发生变化时运行结果
   4. 实验测试与分析
      1. TOP-K质量分析
      2. 参数敏感性分析
      3. 数据处理能力分析
   5. 实验结论
   6. 本章小结
3. 总结与展望

结 论

参考文献

**学术刊物**

1. 崔斌，卢阳. 基于不确定数据的查询处理综述[ J] .计算机应用.2008，28(11): 2729-2731
2. 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[J].计算机学报，2009，(01) :1-16
3. 李建中，于戈，周傲英. 不确定性数据管理的要求与挑战.中国计算机学会通讯.2009，5(4):6-15
4. 申德荣，于戈，寇月，聂铁铮.可能世界内数值型不确定数据匹配模型.计算机应用研究.2008-7：2607-2611

**外国文献**

1. SARMA A D，BENJELLOUN O，HALEVY A，et al.Working models for uncertain data[EB /OL]. (2006203202).http: / / t wiki2 edlab.cs .umass.edu /pub /.
2. MOTRO A. Management of uncertainty in database systems[EB /OL]. (20052032 15).http: / / ise.gmu.edu /～ami / research /publica2tions /pdf /modern94. pdf .
3. M. Hua, J. Pei, W. Zhang, and X. Lin. Eﬃciently answering probabilistic threshold top-k queries on uncertain data (extended abstract). In Proc.International Conference on Data Engineering(ICDE’08), Cancun, Mexico, April 2008.
4. RéC，Suciu D. Management of data wit h uncertainties/ /Proceedings of t he 16t h ACM Conference on Information and Knowledge Management . Lisbon，2007 : 3-8
5. Dalvi N，Suciu D. Management of probabilistic data foundations and challenges/ / Proceedings of t he 26t h ACM SIG2 MOD2SIGACT2SIGART Symposium on Principles of Data-base Systems. Beijing，2007 : 1-12
6. Aggarwal C C，Yu P S. A survey of uncertain data algorithms and applications. IBM Research Report，October 31，2007
7. Pei J，Hua M，Tao Y F，Lin X M. Query answering techniques on uncertain and probabilistic data : Tutorial summary/ / Proceedings of t he 2008 ACM SIGMOD International Conference on Management of Data. Vancouver，2008 :1357-1364
8. Todd J. Green, Vat Tannen. Models for incomplete and probabilistic information. IEEE Date Engineering Bullefin, 2006, 29(1)：17-24
9. HUA M,PEI J,ZHANG W,et al. Ranking queries on uncertaindata:A probabilistic threshold approach[C] .Proceedings of the2008 ACMSIGMOD International Conference on Management of Da-ta. New York: ACMPress, 2008, :673-686 .
10. B RZS NYI S,KOSSMANN D,STOCKER K. The skyline oper-ator[C] .Proceedings of the 17th International Conference on DataEngineering. Washington,DC: IEEE Computer Society, 2001, :421-230 .
11. SILBERSTEIN A,BRAYNARD R,ELLIS C,et al. A sampling-based approach to optimizing top-k queries in sensor networks[C] .Proceedings of the 22nd International Conference on Data Engineer-ing. Washington,DC: IEEE Computer Society, 2006, :68
12. CHENG R,PRABHAKAR S,KALASHNIKOV D V. Querying im-precise data in moving object environments[C] .Proceedings of the19th International Conference on Data Engineering. Washington,DC: IEEE Computer Society, 2003, :723-725 .
13. CHENG R,XIA Y,PRABHAKAR S,et al. Efficient indexingmethods for probabilistic threshold queries over uncertain data[C] .Proceedings of the 30th International Conference on Very Large DataBases. San Francisco: Morgan Kaufmann, 2004, :876-887 .
14. PEI J,JIANG B,LIN X,et al. Probabilistic skylines on uncertaindata[C] .Proceedings of the 33rd International Conference on VeryLarge Data Bases. New York: ACMPress, 2007, :15-26 .
15. BorzsonyiS,KossmannD.andStoekerK.TheSkylineOPerator[A].In:Proe.ofICDE[C].2001,Pages421-430.
16. Chan C. Y. ,Eng P. K. ,TanK.L. Stratified Computation of Skylines with Partially-ordered Domains [A].In:Proc.of the ACMSIGMOD[C].2005，203-214.
17. JinW.，HanJ.，EsterM.MiningThickSkylinesoverLargeDatabase[A].In:Proc.ofKDD[C].2004，Pages255-266.
18. HristidisV，KoudasN.，PaPakonstantinou Y. PREFER:A System for the Efficient Execution of Multi-Parametric Ranked Queries[A].In:Proc.of the ACMSIGMOD[CI，2001，Pages259-270.
19. ABITEBOUL S,KANELLAKIS P,GRAHNE G. On the represen-tation and querying of sets of possible worlds[C] .Proceedings ofthe 1987 ACM SIGMOD International Conference on Managementof Data. New York: ACMPress, 1987, :34-48 .
20. S. Abiteboul, P. Kanellakis, and G. Grahne. On the representation and querying of sets of possible worlds.In Proceedings of the 1987 ACM SIGMOD international conference on Management of data (SIGMOD’87), pages 34–48, New York, NY, USA,1987. ACM Press.
21. Tomasz Imielinski and Jr. Witold Lipski. Incomplete information in relational databases. Journal of ACM, 31(4):761{791, 1984
22. A. D. Sarma, O. Benjelloun, A. Halevy, and J. Widom. Working models for uncertain data. InICDE '06: Proceedings of the 22nd International Conference on Data Engineering (ICDE'06),page 7, Washington, DC, USA, 2006. IEEE Computer Society
23. M. A. Soliman, I. F. Ilyas, and K. C.-C. Chang. Top-k query processing in uncertain databases.In Proceedings of the 23nd International Conference on Data Engineering (ICDE'07), Istanbul,Turkey, April 2007. IEEE.
24. L. Antova, C. Koch, and D. Olteanu. 10106worlds and beyond: Eficient representation and processing of incomplete information. In Proceedings of the 2007 IEEE 23rd International Conference on Data Engineering (ICDE'07), April 2007
25. Ronald Fagin, Amnon Lotem, and Moni Naor. Optimal aggregation algorithms for middleware.In PODS '01: Proceedings of the twentieth ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems, pages 102{113. ACM Press, 2001.
26. D. Angluin and L. G. Valiant. Fast probabilistic algorithms for hamiltonian circuits and matchings. In Proceedings of the ninth annual ACM symposium on Theory of computing (STOC'77), pages 30-41, New York, NY, USA, 1977. ACM Press.

**统计资料**

1. 2009年中国互联网信息中心（China Internet Network Information Center，CNNIC）的调查报告：中国互联网信息中心，2009-1

**参考网址**

1. http://queens.db.toronto.edu/project/conquer/
2. http://orion.cs.purdue.edu/
3. http://infolab.stanford.edu/trio/
4. http://www.cs.cornell.edu/database/maybms/
5. http://www.cs.washington.edu/homes/suciu/project-mystiq.html
6. http://www.eecs.berkeley.edu/Research/Projects/Data/102060.htm
7. ht tp :// www. cs. umd. edu/～vs/research. htm # pdb
8. ht tp :// www. almaden. ibm. com/cs/ project s/ avatar/

致谢

通过三个多月的努力，我的专业毕业论文终于要如期脱稿付梓。本论文得以顺利完成，首先要感谢我的指导老师杜卿老师。杜卿老师在本文撰写过程中，至始至终都给予了悉心的指导，尤其是在论文的选题和拟定大纲时，更是得到了精心的点拨，通过阅览往届毕业生的论文，使我少走许多弯路，也使得资料搜寻和论文撰写工作能够迅速展开。对于论文的每一点细微错漏，她都进行了耐心的指点和纠正。杜卿老师严谨的治学精神和渊博的专业知识使我受益匪浅，在此向我的导师表示最真挚的感谢。

感谢华南理工大学软件学院为我提供了攻读软件工程专业的机会和良好的学习环境，感谢学院各位老师的辛勤培养和教育。老师们渊博的学识和无私的奉献精神，不仅让我从中学到了许多专业知识，还让我学到了许多为人处世的道理。

感谢共同学习和生活的同学和室友，跟他们一起度过了美好的学生生涯，他们也为本文的写作提供了不少好的建议。

还要感谢我的父母，是他们在物质和精神上给予我的支持和关怀，伴随着我走过了十六年的求学岁月，使我得以潜心学习和研究而顺利完成学业。

最后，感谢在百忙之中评阅本文和参加答辩的各位专家和老师！谢谢大家！