编号

**南京航空航天大学**

**毕业论文**

出租车营运大数据分析

|  |  |
| --- | --- |
| 学生姓名 | 陶超权 |
| 学 号 | 161420330 |
| 学 院 | 计算机科学与技术学院 |
| 专 业 | 信息安全 |
| 班 级 | 1614204 |
| 指导教师 | 陈兵 教授 |

二〇一八年六月

**南京航空航天大学**

**本科毕业设计（论文）诚信承诺书**

本人郑重声明：所呈交的毕业设计（论文）（题目： ）是本人在导师的指导下独立进行研究所取得的成果。尽本人所知，除了毕业设计（论文）中特别加以标注引用的内容外，本毕业设计（论文）不包含任何其他个人或集体已经发表或撰写的成果作品。

作者签名： 年 月 日

（学号）：

出租车营运大数据分析

摘 要

随着国民经济的发展以及人们生活水平的提高，出租车越来越成为人们出行不可少的工具，无论是周内的上班工作还是周末的出行娱乐，出租车扮演着越来越重要的角色。在万物联网的今天，每辆出租车都安装了GPS终端，同时每次营运的数据都会被提交至数据库保存，这些终端装置会定时向数据库发送出粗车此时的行驶状态，营运状态等。而这些数据将会成为我们用来挖掘有用信息的重要资源。

通过对出租车营运数据进行处理和分析，可以挖掘出租车热点载客区域，可以为出租车司机进行智能推荐，优化出租车公司的资源调度，以达到更高的效益。

本文借助MapReduce框架，首先对营运数据进行预处理，剔除空数据和重复数据，然后利用MongoDB对MapReduce的支持，通过DBscan聚类算法，按照是否是工作日，以及每天不同的时间段，对营运数据进行聚类分析，最后得到每天不同时段内的乘车热点区域。

关键词**：**出租车，大数据，聚类，MapReduce,MongoDB,DBscan

Abstract

A new kind sandwich structure.

Key Words**：**NC machining; NC verification; Machining environment; Help files

目录

[摘 要 i](#_Toc512019828)

[Abstract ii](#_Toc512019829)

[第一章 引 言 - 1 -](#_Toc512019830)

[1.1 研究背景及意义 - 1 -](#_Toc512019831)

[1.2 研究现状 - 1 -](#_Toc512019832)

[1.2.1 大数据及其隐私保护研究现状 - 1 -](#_Toc512019833)

[1.2.2 出租车数据应用研究现状 - 1 -](#_Toc512019834)

[1.3 论文内容 - 1 -](#_Toc512019835)

[1.4 论文结构 - 1 -](#_Toc512019836)

[第二章 大数据隐私保护 - 1 -](#_Toc512019837)

[2.1 隐私保护概述 - 1 -](#_Toc512019838)

[2.2 隐私保护方法 - 1 -](#_Toc512019839)

[2.3 本章小结 - 1 -](#_Toc512019840)

[第三章 数据挖掘与聚类算法 - 1 -](#_Toc512019841)

[3.1 数据挖掘 - 1 -](#_Toc512019842)

[3.1.1 数据挖掘的概念 - 1 -](#_Toc512019843)

[3.1.2 数据挖掘的过程 - 2 -](#_Toc512019844)

[3.2 聚类算法 - 2 -](#_Toc512019845)

[3.2.1 聚类算法概述 - 2 -](#_Toc512019846)

[3.2.2 聚类算法分类 - 2 -](#_Toc512019847)

[3.2.3 DBSCAN算法 - 2 -](#_Toc512019848)

[3.3 本章小结 - 2 -](#_Toc512019849)

[第四章 数据获取及预处理 - 2 -](#_Toc512019850)

[4.1 数据来源及数据格式 - 2 -](#_Toc512019851)

[4.2 数据预处理 - 2 -](#_Toc512019852)

[4.2.1 数据清洗 - 2 -](#_Toc512019853)

[4.2.2 数据导出导入 - 2 -](#_Toc512019854)

[4.2.3 车牌号隐私保护 - 2 -](#_Toc512019855)

[4.3 本章小结 - 2 -](#_Toc512019856)

[第五章 基于MapReduce框架的热点区域挖掘及可视化 - 2 -](#_Toc512019857)

[5.1 基于MapReduce框架的聚类算法实现 - 2 -](#_Toc512019858)

[5.1.1 DBSCAN算法实现 - 2 -](#_Toc512019859)

[5.1.2 阈值取值分析 - 2 -](#_Toc512019860)

[5.2 热点区域可视化 - 3 -](#_Toc512019861)

[5.3 本章小结 - 3 -](#_Toc512019862)

[第六章 总结与展望 - 3 -](#_Toc512019863)

[6.1 总结 - 3 -](#_Toc512019864)

[6.2 展望 - 3 -](#_Toc512019865)

[参 考 文 献 - 3 -](#_Toc512019866)

[致 谢 - 3 -](#_Toc512019867)

[附 录 - 4 -](#_Toc512019868)

# 第一章 引 言

## 研究背景及意义

随着城市化进程的推进，城市中的交通路网越来越完善，而出租车所占比例也在逐年上升。近几年由于“滴滴打车”等平台的出现，除了传统的路边叫停出租车外，“网约车”也应用而生，极大的方便了市民的出行。各种类型的出租车上的GPS设备，提供了大范围的，海量的可食用数据。但同时随着“网约车”的大量普及，由于缺乏对客源分布的全面认识，极有可能造成出租车过度空驶，同一区域出租车数量过多，无法预测未来一段时间内热点乘车区域而不能获得更大收益等问题。

在此背景下，如果能够充分利用出租车GPS终端所提供的轨迹数据和营运数据，对其进行分析和处理，宏观上得到某一时段内乘客的热点乘车区域，微观上结合数据库、以及特定的算法对车辆信息进行搜索，然后集成为合多功能为一体的、数据以及信息可视化的出租车信息管理查询平台 同时利用大数据挖掘的方法，通过对历史数据的分析，挖掘出某段时间内的热点乘车区域，向出租车司机个性化推荐，整合出租车资源，提高司机收益，方便乘客出行。谷歌在2004年提出的MapReduce[1]框架通过映射和化简两个步骤将大作业分解为若干并行的小作业，借助此框架，通过具体的聚类算法，可以更方便的挖掘出热点区域。

在大数据如火如荼的今天，如何在数据挖掘的同时保护好用户的隐私成为了备受关注的问题。隐私保护又具体分为位置隐私保护、标识符匿名保护、连接关系匿名保护等[2]。本文将针对对于出租车营运数据挖掘过程中，对车牌号进行隐私保护，以防止司机和乘客信息泄露。

## 研究现状

* + 1. 大数据及其隐私保护研究现状

//可以再加个数据挖掘的研究现状

关于大数据尚未有一个公认的定义，其中比较有代表性的是3V[3]定义，即数据量大小(Data Volume)、数据输入输出速度(Data Velocity)、多变性(Data Variety)。其处理过程包含数据采集与预处理，数据分析，数据解释三部分。目前国内外对于大数据隐私保护主要有以下几个方面的研究：

基于位置的服务LBS给人们提供了许多便利，但同时也威胁着用户的位置隐私，典型的位置隐私保护技术有基于k-匿名的位置隐私保护技术。通过对用户C在某一时段内的位置信息进行匿名和泛化处理，使得在该时段内至少有k个用户都在C所处的位置范围内。侯士江基于用户—匿名器—LBS架构，分别针对欧氏空间和路网环境的位置隐私保护、常见查询等问题进行了研究[4];车延辙围绕着改进算法效率和提高抵御攻击能力的目标,研究移动点对点体系结构下的用户位置隐私保护关键技术[5];Gupta针对k-匿名中存在可信第三方的主要限制问题，提出了通过若干个同等级用户相互共享一些位置信息的方法来扩展可信第三方[6].

* + 1. 出租车数据应用研究现状

周洋通过分析出租车的起始点和终点，提取乘客的出行热度和及其空间规律，将居民区进行聚类，得出不同居民区的不同背景信息[7]；王郑委基于Hadoo平台通过K-Means聚类算法，挖掘出租车载客热点区域[8]；Putri F K 基于MongoDB和Spark提出了分布式的热点乘车区域搜索系统[9]；Deri J A利用纽约市的出租车轨迹信息，基于迪杰斯特拉算法在时间和空间上优化了其对轨迹的计算的复杂度[10].

## 论文内容

本文应用南京市出租车的营运信息，对乘车不同天不同时间段内热点乘车区域进行挖掘，主要包括数据获取，数据预处理及隐私保护，聚类算法实现和热点区域可视化四个步骤。首先需要将出租车公司的营运数据从关系型数据库导入到非关系型数据库MongoDB中；由于原始数据包含许多空值字段和重复数据，需要对数据进行清洗，并在清洗过程中对用户敏感信息进行保护；为挖掘出南京市一天内不同时段乘车热点区域，本文采用基于MapReduce框架的DBSCAN聚类算法，将一周时间划分为工作日和非工作日，同时将一天划分为不同时段，通过使用不同的阈值，来达到最佳乘车热点区域；最后通过高德地图API将乘车热点区域进行可视化。

## 论文结构

论文一共分为五个章节。

第一章 引言——主要介绍了论文的研究背景和意义，以及国内外研究现状，包括大数据及其隐私保护的研究现状和出租车数据利用的研究现状，最后介绍了本文的内容和结构。

第二章 大数据隐私保护及数据挖掘相关技术——主要介绍了大数据隐私保护的常见方法，并对数据挖掘做了概述并介绍了其挖掘过程。

第三章 数据获取及预处理——首先介绍了数据来源和数据格式，然后介绍了数据预处里过程，包括脏数据的剔除、数据得到导入导出以及用户敏感信息保护。

第四章 基于MapReduce框架的热点区域挖掘及可视化——首先实现了DBSCAN聚类算法，然后通过不同阈值的取值分析，得到最佳热点区域，最后通过高德地图API将热点区域呈现出来。

第五章 总结与展望——对本文所做工作、不足之处以及将来可改进之处进行了总结。

# 第二章 大数据隐私保护及数据挖掘相关技术



## 大数据隐私保护

* + 1. 大数据隐私保护概述

隐私[11]指主体没有公开的知识、信息等，按照主题的不同，隐私又可分为公共隐私和个人隐私。公共隐私指群体的共同信息或者模式信息，主要针对企业和政府部门。个人隐私指个人独立的基本资料，如医疗信息，电子档案信息等。大数据在分析过程中如果不对用户敏感信息进行保护，极有可能造成用户信息泄露。如何在不泄露用户隐私的前提下，提高大数据的利用率，是目前大数据研究领域的关键问题之一。

* + 1. 大数据隐私保护方法

1. 数据发布匿名保护技术

为防止攻击者在数据发布时使用链接攻击获取用户敏感信息，常常需要在数据发布对敏感信息进行匿名处理。其中最常见的是k匿名技术[12];通过对用户敏感信息使用隐匿和泛化两种方法，使得对用户的同一属性字段至少有k-1个用户与之相同，从而迷惑攻击者。由于k匿名的敏感属性在分类时可能会缺乏多样性，从而导致匿名表产生信息泄露，又提出了l多样化[13]，使得匿名组里敏感属性的多样性大于或等于l。

1. 数据水印技术

数据水印技术[14]指将特定的信息嵌入数字信号中，数字信号可能是音频、图片或是视频等。若要拷贝有数字水印的信号，所嵌入的信息也会一并被拷贝。数字水印可分为浮现式和隐藏式两种，前者是可被看见的水印，其所包含的信息可在观看图片或视频时同时被看见。一般来说，浮现式的水印通常包含版权拥有者的名称或标志。隐藏式的水印是以数字数据的方式加入音频、图片或视频中，但在一般的状况下无法被看见。隐藏式水印的重要应用之一是保护版权，期望能借此避免或阻止数字媒体未经授权的复制和拷贝。

## 数据挖掘

* + 1. 数据挖掘的概念

数据挖掘[15]是一个多学科交叉领域,它融合了数据库技术、人工智能、机器学习、统计学、知识工程、面向对象方法、信息检索等最新技术的研究成果，从大量数据中提取知识和模式。数据挖掘的实际工作是对大规模数据进行自动或半自动的分析，以提取过去未知的有价值的潜在信息，例如数据的分组（通过聚类分析）、数据的异常记录（通过异常检测）和数据之间的关系（通过关联式规则挖掘）。

1. * 1. 数据挖掘的过程
2. 数据预处理

在使用挖掘算法计算之前，必须收集数据集。由于数据挖掘只能发现存在于数据中的模式，所以数据集应该足够大以至于能包含这些模式。对于分析多元数据集来说，数据预处理是必不可少的一个过程，通过预处理过程来剔除噪声数据和缺失数据，以提高挖掘结果的准确性。

1. 数据挖掘

数据挖掘过程通常涉及六种常见的任务：异常检测，用来是被异常数据；关联学习规则，用来搜索变量之间的关系；聚类，在数据集中发现具有某些相同特征的数据子集；分类，将已知的分类或者结构应用于未知的数据，对其进行分类；回归：寻找用最小错误对数据建模的函数；汇总，提供了一个更紧凑的数据表示方法，包括生成可视化和报表。

1. 结果验证

如果数据挖掘方法在不知情的情况下被误用，可能会产生一个看似很重要实则在其他新的数据集上不能复现的结果，这种结果通常是由于使用了太多假设或者不符合假设统计规律所造成，在机器学习中一个例子就是过度拟合。所以在进行数据挖掘之后，往往需要在更大的数据集中验证所得模式是否正确。数据挖掘所得到的模式并不是一直都是有效的，很多挖掘算法所得到的结果并不能够在更宽泛的数据集中复现。

## 聚类算法

* + 1. 聚类算法概述

聚类分析又称群集分析，指把相似的对象通过静态分类的方法分成不同的组别或者更多的簇，将本没有类别参考的数据进行分析和划分，根据数据来发掘数据对象及其关系信息，并将这些数据分成不同簇。聚类分析本身不是特定的算法，但它可以根据不同的需求被各种不同的算法所实现。簇这个概念不能够被精确定义，不同的实现算法会得到不同的簇，这也是为什么会有很多不同的聚类算法的原因之一。

* + 1. 聚类算法分类

按照聚类算法所采用的思想不同，大概可分为五类[16]。

1. 层次聚类算法

通过计算不同类别数据点之间的相似度来创建一棵有层次的嵌套聚类树。在聚类树中，不同类别的原始数据点是树的叶子节点，树的顶层是一个聚类的根节点。创建聚类树分为自上而下的合并聚类和自下而上的分解聚类两种方法。自上而下聚类先将所有原始数据点看作一个聚类，然后根据每个原始节点之间的差异性将其不断分解。自下而上聚类先将每个原始数据点看作一个原子聚类，然后根据彼此之间的相似性不断进行聚合。

1. 分割聚类算法

首先将数据集分割为k个划分，然后通过迭代某特定算法，使得某个指标最优以达到最终结果。这种聚类算法又可详细划分为基于图论的聚类、基于网格的聚类、基于平方差的聚类和基于密度的聚类。

基于图论的聚类将聚类转换为组合优化问题，然后利用图论相关知识，并结合启发式算法来求解原问题。其一般做法是，先构造数据集的最小生成树，然后逐步删除最小生成树中最大长度的边，形成更多聚类。

基于网格的聚类算法[17]，基于网格的聚类方法采用空间驱动的方法，把嵌入空间划分成独立于输入对象分布的单元。基于网格的聚类方法使用一种多分辨率的网络数据结构。它将对象空间量化成有限数目的单元，这些网格形成了网格结构，所有的聚类结构都在该结构上进行。这种方法的主要优点是处理速度快，其处理时间独立于数据对象数，而仅依赖于量化空间中的每一维的单元数。基本思想就是将每个属性的可能值分割成许多相邻的区间，创建网格单元的集合（我们假设属性值是连续的，序数的，区间的）。每个对象落入一个网格单元，网格单元对应的属性空间包含该对象的值。

基于平方差的聚类算法的主要思想是逐步优化聚类结果，通过迭代想目标数据集重复向聚类中心进行聚类以得到最优解。该聚类方法又可详细分为最邻近聚类算法概率聚类算法、K-medoids[18]算法和K-means算法。其中K-means是目前应用最多的一种聚类算法，给定划分数量 k。创建一个初始划分，从数据集中随机地选择 k 个对象，每个对象初始地代表了一个簇中心（Cluster Centroid）。对于其他对象，计算其与各个簇中心的距离，将它们划入距离最近的簇。采用迭代的重定位技术，尝试通过对象在划分间移动来改进划分。所谓重定位技术，就是当有新的对象加入簇或者已有对象离开簇的时候，重新计算簇的平均值，然后对对象进行重新分配。这个过程不断重复，直到各簇中对象不再变化为止。

基于密度的聚类算法根据数据的分布密度，合并密度超过某个阈值的相邻区域为一个区域，可以在有噪音的数据中发现各种形状和各种大小的簇，常用于对空间数据进行聚类。DBSCAN算法[19]就是该类方法中最典型的算法之一。

* + 1. DBSCAN算法描述

先介绍一些关键概念的定义：

假设样本集D=(x1,x2,…,xm)

1. ϵ邻域：对于xj∈D，其ϵ邻域包含样本集D中与xj的距离不大于ϵ的子样本集，即,这个样本集的个数记为
2. 核心对象：对于任一样本xj∈D，如果其ϵ邻域对应的至少包含MinPts个样本，即如果，则xj是核心对象。
3. 密度直达：如果且,则称p由q密度直达。当p和q都是核心对象时，密度直达满足对称性。
4. 密度可达：对于xi和xj,如果存在样本序列p1,p2,…,pT满足p1=x1,pT=xj,且pt+1由pt密度直达，则称xj由xi密度可达。即密度可达满足传递性，但不满足对称性。
5. 密度相连：对于xi和xj,如果存在核心对象样本xk,使得xi和xj均由xk密度可达，则称xi和xj密度相连。密度相连满足对称性。
6. 簇：簇是样本集的一个子集，当样本点满足下列条件时称样本点包含于簇C。
7. :如果p∈C且q由p密度可达，则q∈C
8. :p,q密度相连
9. 噪声点：设C1,C2,…,Ck是样本集D中的簇，则定义噪声点为不属于任何簇的数据集，即

算法流程：

输入：样本集D=(x1,x2,…,xm),邻域参数(ϵ,MinPts)

输出：簇划分C

1. 判断输入点是否是核心对象
2. 如果是核心对象，则该核心对象邻域构成一个簇
3. 遍历核心对象邻域内的所有点
4. 如果该点也是核心对象，则将该点邻域合并到当前簇

伪码描述：

DBSCAN(D, eps, MinPts) {

C = 0

for each point P in dataset D {

if P is visited

continue next point

mark P as visited

NeighborPts = regionQuery(P, eps)

if sizeof(NeighborPts) < MinPts

mark P as NOISE

else {

C = next cluster

expandCluster(P, NeighborPts, C, eps, MinPts)

}

}

}

expandCluster(P, NeighborPts, C, eps, MinPts) {

add P to cluster C

for each point P' in NeighborPts {

if P' is not visited {

mark P' as visited

NeighborPts' = regionQuery(P', eps)

if sizeof(NeighborPts') >= MinPts

NeighborPts = NeighborPts joined with NeighborPts'

}

if P' is not yet member of any cluster

add P' to cluster C

}

}

regionQuery(P, eps)

return all points within P's eps-neighborhood (including P)

## 本章小结

本章首先介绍了数据隐私保护的相关知识及方法。然后对数据挖掘技术进行了阐述，包括其概念和挖掘过程。最后介绍了数据挖掘中的常用方法——聚类算法进行了介绍，包括算法的分类等，并对其中的经典算法DBSCAN进行了详细介绍，为后面的乘车热点区域挖掘奠定了理论基础。

# 第三章 数据获取及预处理



## 数据来源及数据格式

本文采用南京市中北股份有限公司的出租车营运数据作为研究对象，数据采集起止时间为。。。，共有出租车。。。辆，数据总量xxGB,约包含xx条记录。数据源来自数据库名为T\_BUSI\_RUN的数据表，每条记录包含车牌号，上车原始经维度，上车高德经纬度，上车时间等字段。数据字段及其含义对应关系如表3.1所示。

表3.1 营运数据与字段含义表

|  |  |  |
| --- | --- | --- |
| 字段 | 名称 | 类型 |
| FACT\_ID | 厂商标识 |  |
| DEV\_ID | 设备ID |  |
| CPHM | 车牌号 |  |
| SJDM | 司机代码 |  |
| DWDM | 单位代码 |  |
| SCSJ | 上车时间 |  |
| SCJD | 上车高德经度 |  |
| SCWD | 上车高德纬度 |  |
| XCSJ | 下车时间 |  |
| SCJD | 下车经度 |  |
| XCWD | 下车纬度 |  |
| DHSJ | 等候时间，单位：秒 |  |
| YYSJ | 营运时间，单位：秒 |  |
| YYLC | 营运里程，单位：公里 |  |
| KSLC | 空驶里程，单位:公里 |  |
| YYJE | 营运金额，单位：元 |  |
| JYLX | 交易类型，0：现金，1，2，3，5：市民卡，6：支付宝，7：微信，4，144，145，146：银联卡 |  |
| YYID | 营运ID |  |
| PJID | 评价ID |  |
| PJXX | 评价选项，0：未评价；1：满意；2：一般；3：不满意；4：投诉 |  |
| FJF | 附加费，单位：元 |  |
| SCSJGPS | GPS的上车时间 |  |
| XCSJGPS | GPS的下车时间 |  |

## 数据预处理

* + 1. 数据清洗

通常由于设备故障，司机误操作或者天气原因，GPS所采集到的原始数据并不是百分之百准确，会存在一些异常数据。为了使数据挖掘的结果更加准确，需要在进行数据挖掘之前进行数据预处理。本文针对数据预处理主要包括两方面，剔除重复数据和有空字段的数据剔除经纬度越界数据。由于设备异常，同一条记录有时会被多次提交，这种重复记录应该在分析之前被剔除。乘车热点区域挖掘主要针对南京市，由于出租车没有市级范围的限制，有些出租车也会出现在南京市以外的地区，为了提高挖掘的准确性和后期可视化的美观性，需要将南京市以外的记录剔除掉。南京市经纬度大概范围为118.35-119.23，31.236-32.611，剔除上车经纬度在这个范围以外的数据。

* + 1. 数据导出导入

本文所述的数据分析基于非关系型数据库MongoDB进行，由于原始数据存储于关系型数据库Oracle数据库中，需要进行数据的移植工作。考虑到数据挖掘不会用到营运数据表中的所有字段，为节省存储空间，提高挖掘效率，本文在数据移植时只移植了对数据挖掘有用的相关字段：车牌号码，上车高德经度，上车高德维度，上车时间。MongoDB数据库中是以文档的形式来存储数据，MongoDB中文档相当于关系型数据库中的一条记录，集合相当于关系型数据库中的表。

为实现数据的导出与导入，首先在sqlplus中通sql语句剔除重复数据和有空值字段的数据，然后将查询结果保存在csv文件中，最后通过mongoimport命令将csv文件中的数据导入到MongoDB中事先创建好的数据库中，移植后所得文档结构如下。

{

"\_id" : ObjectId("5add799dd9c9d696a38ed0a9"),

"CPHM" : "苏A70007",

"SCWD" : 31.967239,

"SCJD" : 118.795548,

"SCSJ" : "2018-04-22 12:05:00"

}

\_id是一个12字节长的十六进制书，它保证了每个文档的唯一性，其余字段含义如本章第一节所示。

* + 1. 用户隐私保护

通过上节所述可以发现，如果直接将数据存入MongoDB中，车牌号将以明文方式显示对出租车司机或者乘客来说，车牌号码具有隐私性，如果被数据分析过程被不可信的第三方执行，它将会获得车牌号码以及营运数据，结合链接攻击等方式，极有可能造成司机或者乘客的个人隐私泄露，因此，需要对车牌号码进行隐私保护处理。

美国密码学家罗纳德.李维斯特于1992年公开了MD5[20]摘要加密算法,输入任意长度的信息，经过处理，输出为128位的信息。经过程序流程，生成四个32位数据，最后联合起来成为一个128-bits散列。基本方式为，求余，取余，调整长度，与链接变量进行循环运算，得出结果。图3.1展示了其运算过程，一个MD5运算— 由类似的64次循环构成，分成4组16次。F 一个非线性函数；一个函数运算一次。Mi 表示一个 32-bits 的输入数据，Ki 表示一个 32-bits 常数，用来完成每次不同的计算。

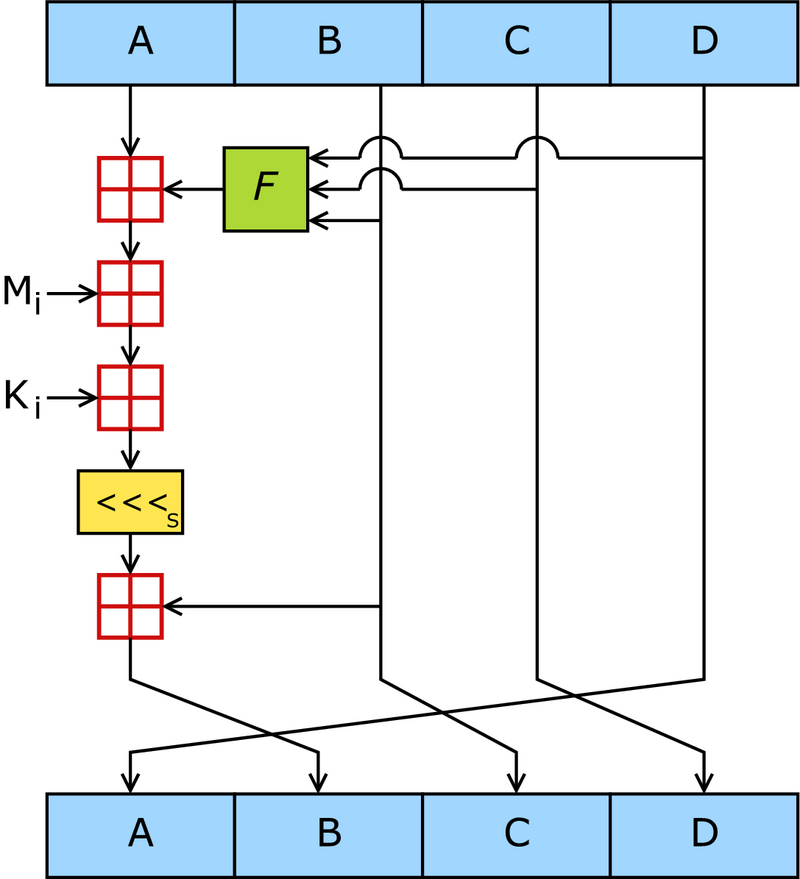


图3.1 MD5运算流程

摘要算法流程如下：

1. 填充：如果信息长度（以bit为单位）对512求余的结果不等于448，则对信息进行填充，在信息尾部添加一个1和若干个0，使得填充后信息长度为N\*512+448位.
2. 记录信息长度：用64位来存储填充前的信息长度。这64位追加在第一步所得结果后面，这样信息长度就变成N\*512+448+64=(N+1)\*512位。
3. 初始化MD数组：A=0X67452301L，B=0XEFCDAB89L，C=0X98BADCFEL，D=0X10325476L
4. 进行循环运算

伪码描述：

//Note: All variables are unsigned 32 bits and wrap modulo 2^32 when calculating

var int[64] r, k

//r specifies the per-round shift amounts

r[ 0..15]：= {7, 12, 17, 22, 7, 12, 17, 22, 7, 12, 17, 22, 7, 12, 17, 22}

r[16..31]：= {5, 9, 14, 20, 5, 9, 14, 20, 5, 9, 14, 20, 5, 9, 14, 20}

r[32..47]：= {4, 11, 16, 23, 4, 11, 16, 23, 4, 11, 16, 23, 4, 11, 16, 23}

r[48..63]：= {6, 10, 15, 21, 6, 10, 15, 21, 6, 10, 15, 21, 6, 10, 15, 21}

//Use binary integer part of the sines of integers as constants:

for i from 0 to 63

k[i] := floor(abs(sin(i + 1)) × 2^32)

//Initialize variables:

var int h0 := 0x67452301

var int h1 := 0xEFCDAB89

var int h2 := 0x98BADCFE

var int h3 := 0x10325476

//Pre-processing:

append "1" bit to message

append "0" bits until message length in bits ≡ 448 (mod 512)

append bit length of message as 64-bit little-endian integer to message

//Process the message in successive 512-bit chunks:

for each 512-bit chunk of message

break chunk into sixteen 32-bit little-endian words w[i], 0 ≤ i ≤ 15

//Initialize hash value for this chunk:

var int a := h0

var int b := h1

var int c := h2

var int d := h3

//Main loop:

for i from 0 to 63

if 0 ≤ i ≤ 15 then

f := (b and c) or ((not b) and d)

g := i

else if 16 ≤ i ≤ 31

f := (d and b) or ((not d) and c)

g := (5×i + 1) mod 16

else if 32 ≤ i ≤ 47

f := b xor c xor d

g := (3×i + 5) mod 16

else if 48 ≤ i ≤ 63

f := c xor (b or (not d))

g := (7×i) mod 16

temp := d

d := c

c := b

b := leftrotate((a + f + k[i] + w[g]),r[i]) + b

a := temp

Next i

//Add this chunk's hash to result so far:

h0 := h0 + a

h1 := h1 + b

h2 := h2 + c

h3 := h3 + d

End ForEach

var int digest := h0 append h1 append h2 append h3 //(expressed as little-endian)

为充分保护用户隐私，本文在进行数据导出时对车牌号码这一关键字段采用MD5算法进行映射，Oracle提供了DBMS\_OBFUSCATION\_TOOLKIT.MD5方法来对字段进行加密，但是此时加密得到的是乱码，并非十六进制数字，需要使用utl\_raw.cast\_to\_raw将字符串转换为RAW类型。

## 本章小结

本章首先介绍了本文进行数据挖掘所需数据的来源和格式，以及各个字段的含义。然后介绍了数据挖掘的第一步——数据预处理，主要包括重复数据和空数据的剔除，紧接着介绍了数据从数据关系型数据库到非关系型数据库MongoDB的移植，最后阐述了通过MD5对用户隐私字段车牌号进行了映射保护。本章为下一章的数据挖掘做了数据准备。

# 第四章 基于MapReduce框架的热点区域挖掘及可视化



## 基于MapReduce框架的聚类算法实现

* + 1. 实验环境搭建

框架：Node.js 8.9.4,

语言：JavaScript,

数据库：MongoDB 3.4.10.

* + 1. 基于MongoDB的MapReduce框架实现

作为数据分析的常用数据库之一，MongoDB自身集成了MapReduce框架的接口。下面对其进行介绍。

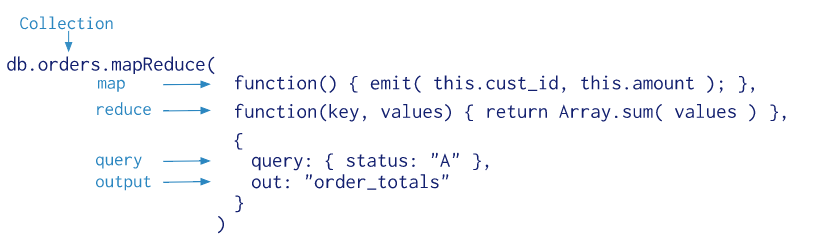


图4.1 MapReduce命令格式

在图4.1中，map函数用于将键和值映射，将同一个键对应的值组合成数组，交由reduce函数处理。Reduce函数对同一键所对应的值构成的数组进行化简处理，最后返回与输入结构相同的结果。

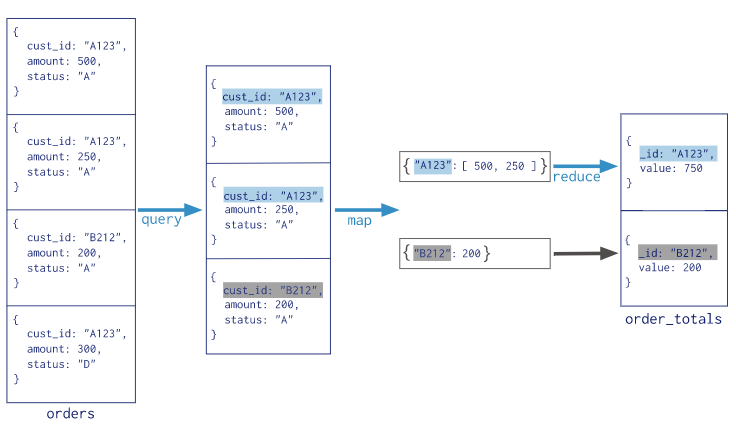


图4.2 MapReduce流程

图4.2描述了MapReduce的工作流程，query阶段通过筛选得到status为A的文档，map阶段以cust\_id为键，以amount为值进行映射，每个键都会对应一个由值组成的数组，reduce阶段对每个键对应的值组成的数组进行化简，最后返回一个键值对。

* + 1. DBSCAN算法实现

算法实现分为五个步骤：读取数据，map,reduce,finalize,结果输出。

读取数据：node.js为mongodb数据库操作提供了mongodb模块，通过require语句引入后即可连接数据库并进行操作。

map:为了突显出周内和周末不同时间段乘车热点区域的不同，本文按照每条数据是否是工作日isWorkDay（即周一到周五），以及所在的不同时间段timeID进行划分。将数据分为工作日和非工作日。在工作日中，将当天24点到七点记为时间段0，将八点到十点记为时间段1，将11点到13点记为时间段2，将14点到16点记为时间段3，将17点到20点记为时间段4，将21点到23点记为时间段5。在非工作日中，将24点到8点记为时间段0，将9点到23点记为时间段1。然后以isWorkDay和timeID为键，以车牌号，上车时间，上车经度，上车维度为值进行映射，得到一个由该四个变量组成的数组。

reduce:这一阶段接收由map阶段传过来的key和values变量，如果map阶段一个key对应的values条数很多，可能会多次调用reduce方法，即前一次reduce的结果可能被包含在values中再次传递给reduce方法，这也要求，reduce的返回结果需要和value的结构保持一致。因此本文不在reduce中进行聚类分析，只是将values原样返回。

finalize:该方法在reduce函数结束后接收其key和values做最后的处理并输出，本文聚类算法将在这个阶段进行。在上述的map阶段，将所有的营运数据按照isWordDay和timeID进行了分类，在finalize中将对每一类别进行再次聚类。具体流程如下：

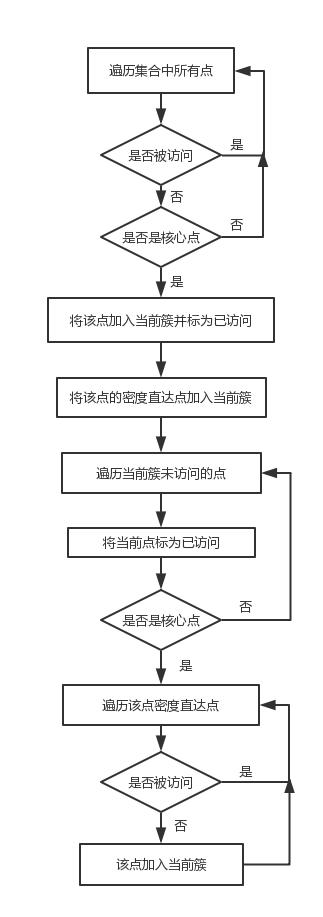


图4.3 DBSCAN算法流程

* + 1. 阈值取值分析

## 热点区域可视化

## 本章小结

# 第五章 总结与展望



## 总结

## 展望

参 考 文 献

[1] Jeffrey Dean and Sanjay Ghemawat. 2008. MapReduce: simplified data processing on large clusters. Commun. ACM 51, 1 (January 2008), 107-113. DOI:

[2]冯登国,张敏,李昊.大数据安全与隐私保护[J].计算机学报,2014,37(01):246-258.

[3]Douglas, Laney. 3D Data Management: Controlling Data Volume, Velocity and Variety(PDF). Gartner. [2001-02-06].

[4]侯士江. 基于空间k-匿名的位置隐私保护技术研究[D].燕山大学,2014.

[5]车延辙. 基于位置服务中用户位置隐私保护关键技术研究[D].浙江大学,2013.

[6]Gupta R, Rao U P. Achieving location privacy through CAST in location based services[J]. Journal of Communications and Networks, 2017, 19(3): 239-249.

[7]周洋. 基于出租车数据的城市居民活动空间与网络时空特性研究[D].武汉大学,2016.

[8]王郑委. 基于大数据Hadoop平台的出租车载客热点区域挖掘研究[D].北京交通大学,2016.

[9]Putri F K, Kwon J. A Distributed System for Finding High Profit Areas over Big Taxi Trip Data with MognoDB and Spark[C]//Big Data (BigData Congress), 2017 IEEE International Congress on. IEEE, 2017: 533-536.

[10] Deri J A, Franchetti F, Moura J M F. Big data computation of taxi movement in New York City[C]//Big Data (Big Data), 2016 IEEE International Conference on. IEEE, 2016: 2616-2625.

[11]刘坚. K-匿名隐私保护问题的研究[D].东华大学,2010.

[12]Wei R, Shen H, Tian H. An Improved (k, p, l)-Anonymity Method for Privacy Preserving Collaborative Filtering[C]//GLOBECOM 2017-2017 IEEE Global Communications Conference. IEEE, 2017: 1-6.

[13] Li N, Li T, Venkatasubramanian S. t-closeness: Privacy beyond k-anonymity and l-diversity[C]//Data Engineering, 2007. ICDE 2007. IEEE 23rd International Conference on. IEEE, 2007: 106-115.

[14] 尹浩,林闯,邱锋,等.数字水印技术综述[J].计算机研究与发展,2005,42(7): 1093-1099.

[15] 邵峰晶, 于忠清. 数据挖掘原理与算法[M]. DynoMedia Inc., 2003.

[16]贺玲,吴玲达,蔡益朝.数据挖掘中的聚类算法综述[J].计算机应用研究,2007(01):10-13.

[17]伍育红.聚类算法综述[J].计算机科学,2015,42(S1):491-499+524.

[18]Hae-Sang Park,Chi-Hyuck Jun. A simple and fast algorithm for K-medoids clustering[J]. Expert Systems With Applications,2008,36(2).

[19]Ester M, Kriegel H P, Sander J, et al. A density-based algorithm for discovering clusters in large spatial databases with noise[C]//Kdd. 1996, 96(34): 226-231.

[20] Rivest R. The MD5 message-digest algorithm[J]. 1992.

致 谢

这次毕业设计可以圆满地完成，首先感谢陈兵老师的谆谆教导，在生活上和学习上都给予我强烈的鼓励和指导，尤其是

附 录

1．XXXXXXX

2．XXXXXXXXXXX

3．XXXXXXX