中文图书分类号：TP391
密　　级：公开
UDC：　004
学　校　代　码：10005

硕士专业学位论文

PROFESSIONAL MASTER DISSERTATION

论　文　题　目：睡眠呼吸疾病分析决策系统的关键技术研究和实现

论　文　作　者：陈羿霖

专业类别/领　域：计算机技术

指　导　教　师：贾熹滨 副教授

论文提交日期：2017 年 5 月

UDC：004
中文图书分类号：TP391

学校代码：10005
学　　号：S201407091
密　　级：公开

北京工业大学硕士专业学位论文
（全日制）

题　　目 睡眠呼吸疾病分析决策系统的关键技术研究和实现

英文题目 RESEARCH AND REALIZATION OF KEY TECHNOLOGY OF SLEEP RESPIRATORY DISEASE ANALYSIS AND DECISION SYSTEM

论 文 作 者： 陈羿霖

领　　域： 计算机技术

研 究 方 向： 计算机应用技术

申 请 学 位： 工学硕士专业学位

指 导 教 师： 贾熹滨副教授

所 在 单 位： 信息学部

答 辩 日 期： 2017 年 5 月

授予学位单位： 北京工业大学

独 创 性 声 明

本人声明所呈交的论文是我个人在导师指导下进行的研究工作及取得的研究成果。尽我所知，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，也不包含为获得北京工业大学或其它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了谢意。

签　　名：陈羿霖

日　　期：2017 年 5 月 12 日

关于论文使用授权的说明

本人完全了解北京工业大学有关保留、使用学位论文的规定，即：学校有权保留送交论文的复印件，允许论文被查阅和借阅；学校可以公布论文的全部或部分内容，可以采用影印、缩印或其他复制手段保存论文。

（保密的论文在解密后应遵守此规定）

签　　名：陈羿霖　　日　　期：2017 年 5 月 12 日

导师签名：贾熹滨　　日　　期：2017 年 5 月 12 日

摘 要

随着互联网的日益发展，各行各业的用户量和数据量都呈现爆炸式的增长。面对越来越多的数据量，传统的数据库和分析系统已经不能满足行业的业务需求，大数据框架成为近年来热点，不仅提供了对结构和非结构的存储查询模式，而且支持智能决策的大数据分析和智能挖掘计算技术。在医疗行业中，不但存在大量传统的病人、医生信息等结构化管理数据，同时存在大量诊断记录、监测、影像等非结构化数据，如何有效利用大数据技术，实现数据的多维统计分析，为医生提供多视角、直观的病人信息，为诊疗决策提供依据；如何利用数据挖掘技术，从大量历史病例中挖掘潜在规律，实现病因追溯、疾病建模、自动诊疗方案建议等智能医疗。因而研究大数据架构下，研究医疗数据分析、智能计算相关技术，为智能医疗平台构建提供经验，具有良好的应用意义。

为实现医疗大数据分析平台，本文以睡眠呼吸疾病为案例，采用大数据实时处理框架 Druid，作为系统的数据仓库，实现联机分析处理（OLAP）的数据统计分析模块。采用开源大数据框架 Spark 作为数据挖掘模块的分析引擎，使用改进的加权 FP-growth 等算法对数据进行聚类分析、关联规则挖掘。在此基础上，完成了核心功能模块设计开发。主要工作和创新点如下：

1、面向睡眠呼吸分析的数据建模及 OLAP 时空分析模块设计实现。基于 OLAP 的技术思想，根据病人、医生、诊断记录等不同维度信息表，构建适合本系统业务的“事实星座”多维数据模型，完成了结合时间和空间的 OLAP 分析。实现了基于时间维度的 OLAP 上卷、下钻查询算子以及基于地区和经纬度的不同空间维度信息的统计分析。

2、研究并实现了基于 K-means 的病情画像和诊断推荐算法。根据睡眠呼吸障碍病人诊断治疗数据，研究生理和疾病指标进行聚类分析方法，实现对病情的画像，在此基础上，提取同类症状治疗方案，针对无创治疗中呼吸机设置，提取设置,实现个性化方案推荐。本文对几种典型的聚类算法进行性能对比实验，选择了折中准确率和效率性能的 k-means 算法，并选用类内距离作为评判依据，确定了 k-means 算法中类别数 k 值为 5。

3、提出了基于加权 FP-growth 的重要指标挖掘算法。根据改进算法实现了对病人指标的关联分析，挖掘不同指标之间的相关性，辅助医生进行诊断决策。FP-growth 相对传统的 Aprior 关联规则学习方法，通过树型结构对项集进行存储，提高了频繁集挖掘效率，本文进一步提出引入医生经验，根据不同指标在实际中的重要程度，对其赋予相应的权值，提升了对指标间的关联规则刻画能力。

4、设计并实现了呼吸睡眠分析系统的整体架构。完成统计分析层和数据挖掘层基础上，设计实现了系统的用户管理模块、数据导入模块、数据存储模块和前端模块。其中用户管理模块主要是面向不同用户的登录注册；数据导入模块主要是包括离线数据和感知数据的导入；数据存储模块主要是基于 Druid 数据仓库对实时数据和离线数据的存储；前端模块主要是使用 Echart 可视化工具和地图接口进行数据的展示，包括折线图、柱状图、热力图等形式。

关键词：大数据；聚类分析；关联规则挖掘；联机分析处理（OLAP）

Abstract

With the increasing development of the Internet, the amount of users and data in all walks of life are showing explosive growth. In the face of more and more data, the traditional database and analysis system can not meet the business needs of the industry Big data framework has become a hot spot in recent years, which not only provides the storage and query mode for both structure and non-structural, but also supports intelligent decision-making large data analysis and intelligent mining technology. In the medical industry, there are not only a large number of traditional structured management data of patients, doctors and the others, but also a large number unstructured data such as diagnostic records, monitoring data, medical images. Therefore, it is essential to find the relative effective big data technologies to achieve multi-dimension statistical analysis to provide the multi-view and direct display of patient information to doctors and to support diagnosis decision. It is also important doctor to use data mining technology to discover potential rules from huge historical cases to achieve intelligent medical care such as the disease cause retrieval, disease modeling and automatic diagnosis. Therefore, under the large data structure, the study of medical data analysis, intelligent computing related technologies provide experience to construct intelligent medical platform experience, which has a good application significance. In order to realize the medical big data analysis platform, this paper uses sleep respiratory disease as the case to achieve the statistical analysis module of OLAP by using the data real-time processing framework Druid as the data warehouse. Spark is used as the analysis engine of data mining module, and the data is clustered and analyzed by using weighted FP-growth algorithm. The main work and innovation are described as follows:

1. Design and realization of data modeling and OLAP temporal and spatial analysis for sleep respiratory analysis. Based on the idea of OLAP, the "fact constellation" multidimensional data model suitable for the system transaction is constructed according to the information tables at different dimensions of patients, doctors and diagnostic records to achieve the OLAP temporal and spatial analysis. OLAP drill-up, drill-down query operators based on time dimension and the statistical analysis based on different spatial dimension based on region and latitude and longitude. to study the physiological and disease indicators for cluster analysis.

2. The K-means-based patient profile and diagnostic recommendation algorithm are studied and implemented. According to patients with respiratory sleep disorders diagnosis and treatment data, we study the clustering analysis algorithm to physical and disease factors to achieve the disease profile. On this basis, to extract the similar treatment and machine settings during non-invasion therapy with CPAP In this paper, the performance comparison experiment is carried out on several typical clustering algorithms, and the k-means algorithm with balanced performance of accuracy and

efficiency is selected. The distance between classes is used as the basis and the K value is determined as 5.

3. An important factor mining algorithm based on weighted FP-growth is proposed. According to the improved algorithm, the association analysis of the patient factors is realized, the correlation between the different factors is excavated to assist doctors making the diagnosis decision. Comparing to the traditional Aprior association rules learning method, FP-growth improve the frequent set of mining efficiency based on the tree structure of the itemset storage. Moreover, this paper makes use of doctors' experience as aprior, which provides the corresponding weight according to the importance of different factors in practice. This enhances the ability to describe the association rules.

4. Design and implementation of the overall structure of the sleep respiratory analysis system. In addition to the statistical analysis layer and data mining layer, we also realize the system user management module, data import module, data storage module and front-end module. The data management module is mainly based on the Druid data warehouse for real-time data and offline data storage; Front-end module is mainly used in the data processing module is mainly used for the import and export of data; Echart visualization tools and map interfaces is for data display, in the form of line charts, histograms, thermograms etc.

Keywords: Big data, Clustering analysis, Association rule mining, Online analytical processing(OLAP)

目 录

第 1 章　绪论

1.1 研究背景与意义

随着互联网在人类社会中的广泛应用以及各种管理应用平台不断涌现，各行各业中的数据量呈爆炸式的增长。根据 2012 年的数据统计，互联网每天所产生的各种内容可以刻满上亿张光碟；每天发出的邮件有近 3 亿封之多；各大论坛每天发表的帖子达 200 万个；每天卖出的手机量将近 40 万部等等，这些都会产生大量的用户数据。据统计，截止至 2015 年，互联网每年产生的数据量接近之前所有数据量的综合，已经远远超过 TB 级别，达到了 PB、EB 级别，不久将会达到 ZB 级别。“大数据”时代已经来临[1]。

爆炸式增长的数据量对传统的数据存储形式、数据存储工具和数据分析工具带来了极大的冲击，原有的工具和模式已经无法适应如此巨大的数据量，新兴的工具和技术层出不穷。如数据仓库[2-4]、大数据分布式框架 Hadoop[5]等。而这些新兴的技术和工具在很多行业中都有着广泛的应用，被人熟知的就是电商、O2O 等行业的数据分析、挖掘。对于海量的用户行为数据，传统的依赖关系型数据库如 MySQL、SQLserver 等进行数据的统计已经无法满足分析业务的要求，大数据分析和挖掘已经深入各行各业，使用数据仓库结合分布式框架的分析方法已经占据了主导。

在医疗领域，大数据分析也越来越被人们所重视。伴随着大数据理念和技术的广泛应用，机器学习和人工智能等领域也得到了迅猛发展，人类社会已经踏入了智能时代。随着智能设备、智能医疗的普及，感知数据在医疗领域中也占据了一席之地。随着医疗领域数据越来越多样化，原有的数据库对于许多半结构化、非结构化数据的存储和处理都是十分费时费力的，所以医疗领域同样需要大数据的理念和技术。随着数据挖掘[6-8]领域的发展，各行各业在分析业务中或多或少的都加入了一些数据挖掘相关的算法，对用户的行为数据进行挖掘，发现其中隐藏的规律和关联。

本系统以某企业睡眠呼吸医疗数据为切入点，构造具有一定功能通用性的医疗大数据管理分析系统。突破传统医疗管理系统基于关键字段匹配为基础的数据查询和结果展示等功能，所开发系统旨在以大数据领域相关研究成果为技术手段，结合大数据分析理念及数据挖掘算法，对医疗档案数据、医疗设备在线感知数据为代表的医疗大数据进行个体及群体数据管理、结合时空特征的疾病分析、疾病挖掘，对病人、医生及相关行业决策者提供全面、深入的分析和建议。其成果可

用于医疗行业大数据平台建设，具有很强的应用价值。

本系统的设计及实现，不仅使用数据仓库及联机分析处理[9-11]技术，打破了传统医疗管理系统的基于关系型数据的查询分析模式；还探讨使用机器学习相关算法实现病情画像、相关疾病要素关联关系挖掘等智能化分析计算，为机器学习应用于医疗健康大数据领域提供了应用经验，具有良好的理论意义。

1.2 国内外研究现状

数据挖掘在各个行业的分析中已经占据了很大的比重，尤其在电商分析中，对用户的行为数据分析起到了重要的作用。其中比较热门的分析方法就是用户画像。

用户画像[12-13]的概念最早是由 Alan Cooper（交互设计之父）提出的。用户画像是建立在一系列真实数据之上，为目标用户模型的过程。通过用户调研去了解用户,根据他们的目标、行为和观点的差异,将他们区分为不同的类型,然后每种类型中抽取出典型特征,如基本信息、行为习惯、人口统计学要素、场景等,形成每一种人物原型 。

用户画像的应满足的七个条件，即 PERSONA（虚拟用户）：

P（Primary）—— 基本性，指该用户角色是否基于对真实用户的情景访谈

E（Empathy）—— 移情性，指用户角色中包含姓名、照片和产品相关的描述，该用户角色是否引同理心。

R（Realistic）—— 真实性，指对那些每天与顾客打交道的人来说，用户角色是否看起来像真实人物。

S（Singular）—— 独特性，每个用户是否是独特的，彼此很少有相似性。

O（Objectives）—— 目标性，该用户角色是否包含与产品相关的高层次目标，是否包含关键词来描述该目标。

N（Number）—— 数量，用户角色的数量是否足够少，以便设计团队能记住每个用户角色的姓名，以及其中的一个主要用户角色。

A（Applicable）—— 应用性，设计团队是否能使用用户角色作为一种实用工具进行设计决策

用户画像在电商行业的作用主要包括以下几种：

（1）精准营销[14] —— 分析产品潜在用户，针对特定群体利用短信邮件等方式进行营销。

（2）用户统计[15] —— 比如中国大学购买书籍人数 top10。

（3）数据挖掘[16] —— 构建推荐系统。例如利用关联规则计算，喜欢红酒

的人通常喜欢什么运动品牌，利用聚类算法分析，喜欢红酒的人年龄段分布情况。

用户画像以及进一步的数据挖掘在很多行业已经具有很成熟的方法和案例，如电商行业的推荐系统、社交平台的用户分析、可穿戴设备的数据分析等。但在医疗行业，目前相关病人画像研究还未成熟。事实上，在医疗大数据条件下，从海量病人数据，依据患者病情、诊断目标、诊疗方案等，抽取如典型病人特征，形成对病人个体画像，有助于实现精准医疗、个性化诊疗方案制定等，具有很强的应用前景。

传统的用户画像的分析目标往往是用户，而现在很多行业都开始将画像的目标从用户转向了自身的产品。在医疗行业中，除了对病人进行画像之外，还可以对病情进行画像，通过对病情数据进行分析学习，得到病情的典型特征，根据病人的适应性提供自动诊断、诊疗推荐等业务。

事实上，医疗管理系统的发展已有比较悠久的历史了，其中最重大的里程碑就是医疗信息系统[17]（Hospital Information System），简称 HIS 系统。早在 60 年代初，美国已经开始研究 HIS 系统，随着信息技术的飞速发展，70 年代 HIS 系统的研究已经成为了医疗行业的热点。美国、日本和欧洲国家都纷纷开始对 HIS 系统进行研究和开发，为现在成型的 HIS 系统奠定了强大的基础。70 到 80 年代的 HIS 系统功能比较简单，主要是实现了一些简单的功能，如医护人员可以录入医嘱和诊疗方案，查询病人的检验结果等。只有较少的医院使用较为完整的 HIS 系统作为医院的信息管理系统。

国内对于 HIS 系统的研究和开发相对较晚，虽然计算机在 70 年代末就已经进入了我国的医疗行业，但是直到 90 年代，随着数据库应用逐渐广泛，医疗管理系统的实现才成为了可能。在初期，都是各大计算机公司联合不同的医院进行各自医院的小型管理系统，这些系统最大的问题就是开发重复率高等，很多系统都有一些共同的模块诸如工资模块、医疗模块、诊断系统等。

完整的 HIS 系统体系结构包括 4 大部分，即临床诊疗部分、药品管理部分、经济管理部分、综合管理与统计分析部分。其中，临床诊疗部分主要包括医生和护士对病人的诊疗方案和监测结果的各种管理系统；药品管理部分主要包括和药品相关的记录和管理系统；经济管理部分主要包括病人在医院的经济支出和医院内部经济支出的记录和管理系统；综合管理与统计分析部分主要包括医疗病案管理系统、病人咨询服务系统以及报表系统等。

随着 HIS 系统的普及，现在几乎所有的医院都在使用 HIS 系统对医疗信息进行管理。虽然 HIS 系统对于医院各科室的管理十分健全，但是不同医院，甚至医院科室间的管理系统，开发平台间缺乏兼容性，数据共享仍然是瓶颈，加之缺乏对非结构化数据的处理能力，研究能够支持异构数据处理的大数据平台，具有现

实意义。

目前国内外医疗器械企业，相继开发了云端监护诊疗系统，利用智能设备等及时获取用户远程监测数据[18-19]，协助医生进行病情管理、分析。澳大利亚瑞思迈呼吸机公司针对自己呼吸疾病病人开发了管理平台，其对呼吸病人实现了睡眠呼吸的智能管理，可以提供医生和病人进行病情指标的监控等。由此可见，医疗领域的智能化管理已经成为了热点。但是传统的管理平台仍然存在一些问题。第一、大部分的分析管理系统都是基于联机事务处理，即对传统关系型数据库进行基本的统计查询功能[20]，对于现在越来越大的数据量支持度不高。第二、对于病人来说，虽然可以观察到病情的变化，但是由于医护人员数量较少，病人无法及时的获取相应的诊疗方案及信息反馈等。

基于联机分析处理技术和数据挖掘算法可以很好的解决上述两个问题，联机分析处理是针对数据仓库的应用而非传统的数据库，不仅可以针对大数据的存储，还可以应用多维数据模型，实现联机分析处理的各种操作。而数据挖掘算法可以通过不同的数据挖掘算法对病人数据的挖掘和建模，对病人进行画像和关联，进而针对不同病人的病情进行诊疗推荐，节省了医生重复性工作的时间和病人寻医的时间。

1.3 主要研究内容

本项目的特色在于为医疗在线感知数据及离线历史档案数据提供统一的数据存储、结合时空逻辑的查询和分析决策平台，从信息化基础设施的层面打破各类医疗数据的存储及管理藩篱，为医疗行业提升临床诊疗以及宏观决策水平，以及远程医疗、主动医疗等新兴医疗服务模式的构建提供信息化支撑技术。

本文在深入的研究总结现有的大数据相关技术和框架的基础上，结合相关技术和算法，针对传统医疗管理系统的模式进行了改进，实现了智能化的睡眠呼吸数据分析系统的。本系统所涉及的主要研究和工作内容如下：

（1）OLAP 技术在睡眠呼吸分析中的应用

由于本文的存储结构采用实时分析框架 Druid[21]作为数据仓库，所以结合数据仓库相关技术对数据进行分析是统计分析层模块重点[22-23]。由于病人、医生、诊断记录等维度信息较多，需要结合数据仓库技术构建合适的多维数据模型，并且根据此模型进行多角度（如结合时间、空间维度等）的 OLAP 分析。

（2）聚类算法在睡眠呼吸分析中的应用

由于聚类算法在数据挖掘中应用最为广泛[24-25]，所以本文基于开源大数据框架 Spark[26-27]对聚类算法在睡眠呼吸分析中的应用做出研究。由于病人的病情不

同，传统的统计分析无法提供针对性的治疗方案，需对不同的病情进行学习，通过聚类的方式进行画像得到典型的类别，在此基础上根据不同的病情进行诊疗方案和相关业务的推荐。

（3）关联规则算法的应用及改进

医生不仅需要得到病人的病情结果，对于病情指标之间的关联和诱因也是十分关注的[28]。因此本文对于关联规则对于病情指标之间的关联挖掘提出了研究。由于本系统的业务涉及到关联规则算法，因此关联规则算法的优劣将很大程度上决定了结果的可靠性。一般的关联规则算法只是针对项集出现的次数进行挖掘，很少关注项目的实际意义，而如何把医生的经验与算法很好的结合在一起是本文一个关键的研究问题。

（4）底层数据管理模块的构建

本系统采用分布式系统 Druid 作为底层的数据仓库和数据管理的框架，对病人的感知数据和离线数据进行管理，包括采集、导入和存储等功能，主要通过对 Druid 的配置和操作实现。

（5）前端可视化界面的实现

本系统的前端可视化界面主要使用 Echart 可视化工具和地图接口进行实现，主要包括热力图、折线图、柱状图等，用于用户进行交互操作。

1.4 论文的组织结构与安排

本文一共分为五章，每章具体内容如下：

第 1 章总述本文研究的动机、难点与解决方案，具体包括研究背景与意义、国内外研究现状与本文研究内容，重点介绍了本系统的意义和特色以及实现的挑战性，阐述了原有医疗管理系统对于大数据时代的滞后性，提出本系统的设计思想与实现方法。

第 2 章主要从需求分析、总体架构等方面，整体阐述了系统的整体结构，包括不同应用不同模块的功能图。主要介绍了系统的采集层、存储层和展示层等部分的功能描述及实现方法。

第 3 章从功能和结构详细介绍了系统的统计分析层。先介绍了本系统统计分析层所涉及到的大数据框架和相关的技术思想。深入讲解了使用数据仓库创建多维数据模型的思路与过程，本系统根据现有的数据模型，结合两种常用的多维数据模型，建立了适合本系统的实事星座多维模型。接着对统计分析层的业务模块进行了功能分析，使用联机分析处理的方法对已经建立的多维数据模型进行查询分析，包括个体分析、医生群体分析、决策者群体分析等业务的流程和操作细节。

第 4 章主要介绍了本系统的数据挖掘层，其主要分为两个模块，一个是使用聚类算法进行挖掘的聚类分析模块，另一个是使用关联规则算法进行数据挖掘的关联分析模块。在聚类分析模块中，首先本文介绍了现在应用广泛的 4 种聚类算法，并使用公共数据集对这 4 种算法进行了比较实验，选出了性能最优的 k-means 算法作为本模块的核心算法。该模块通过对原始数据的聚类，对病人进行画像，衍生出诊疗方案推荐和呼吸机设置推荐两种业务。关联分析模块首先介绍了关联规则算法 Apriori 及其改进算法 FP-growth 的原理，经过对实际问题的分析，提出了对算法进行加权处理，使分析的结果更加切合实际情况。

第 5 章为本系统各个业务层和模块的接口定义及实现和结果展示，如用户注册/登录、数据上传的操作界面，个体分析、群体分析、数据挖掘等模块的交互操作介绍等。

最后，总结了本文的研究工作与贡献，并对本系统未来的应用及改进方向进行了展望。

第 2 章　睡眠呼吸病情分析决策系统概述

2.1 引言

本系统使用实时分析框架 Druid 作为 OLAP 层的数据仓库，使用开源大数据框架 Spark 作为数据挖掘层的分析引擎，既实现了传统的统计查询，又完成了对于睡眠呼吸大数据的数据挖掘分析。在本章中，将从需求分析和系统的总体架构以及各模块的功能图等方面介绍本系统。

2.2 系统需求分析

医疗大数据是指在医院临床、科研、管理等业务行为过程中形成的一切动态和静态数据。医疗大数据分为两大部分，分别为医疗设备感知数据和医疗档案历史数据。本系统以睡眠呼吸大数据为切入点，以呼吸疾病病人、医生以及相关疾病监管决策人员为服务对象，设计并实现医疗大数据管理及分析系统，具体功能需求如下：

1） 可支持对医疗档案等历史数据和医疗设备感知数据的接入、ETL 以及存储；

2） 可支持对于数据的联机分析处理（OLAP）；

3） 可支持基于底层原始数据的聚类挖掘和关联挖掘；

4） 可支持基于 B/S 模式[29]的数据可视化展示和交互操作；

5） 作为具有一定功能通用性和可扩展性的平台基础框架，可被其他医疗大数据管理分析场景使用。

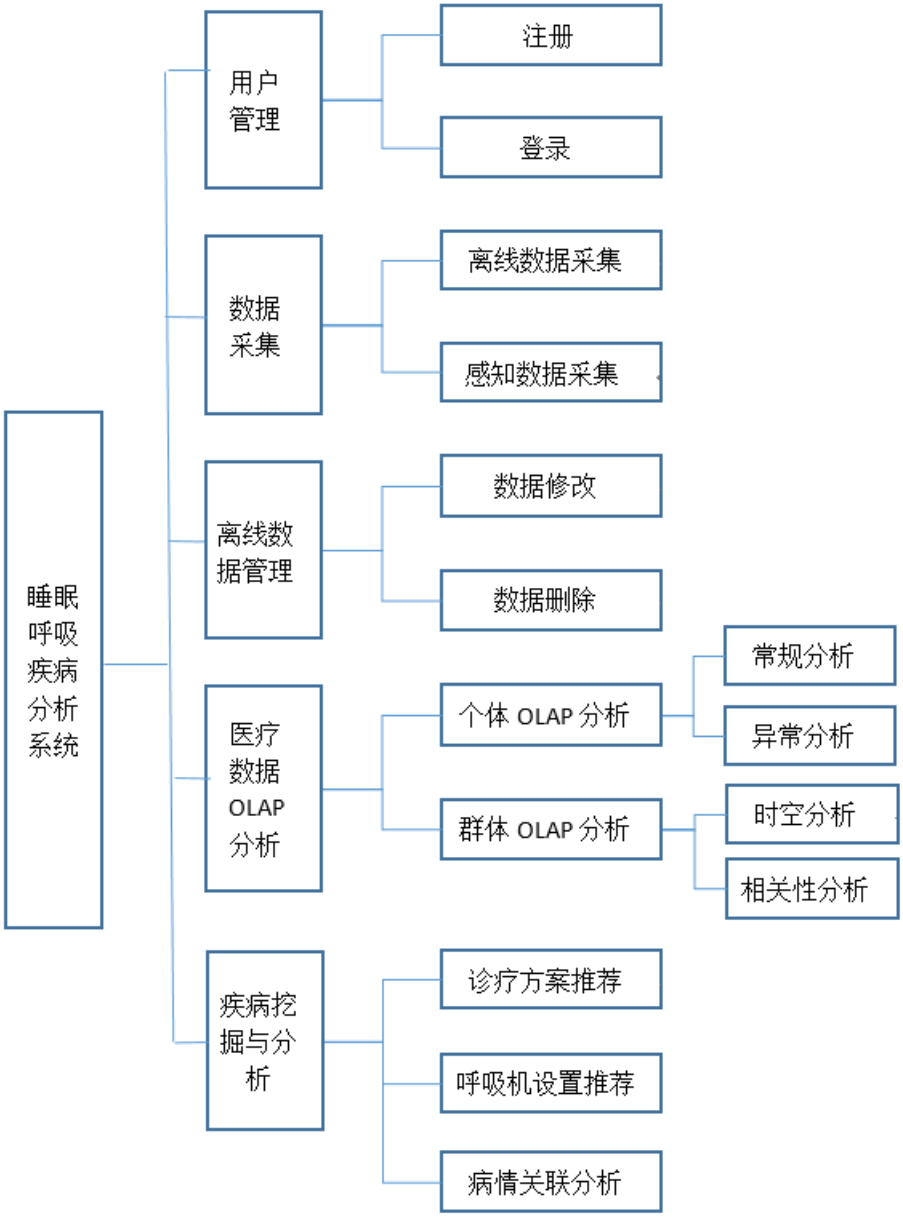


图 2-1 系统功能结构图

Fig2-1 functional structure diagram of system

如图 2-1 所示，本系统的功能结构分为五个部分，每个功能可以进一步细化为若干子功能，其具体描述如下：

（1）用户管理主要负责患者、医生、医院、决策者等多角色的管理，包括注册、登录、管理等功能。

（2）感知数据采集主要包括感知数据采集与离线数据采集两个功能，分别实现对海量采样数据和离线医疗数据的接收与管理。

在感知数据采集中，海量传感器不断地产生新的采样数据，当接收到采样数据时，根据需求对数据按照规定的格式进行存储和管理功能。

在离线数据采集中，由于数据量庞大并且结构复杂，需要对其进行预处理及数据融合，实现离线数据的统一表示形式。

（3）离线数据管理主要负责档案、历史诊疗等数据的管理，包括数据修改、数据删除等功能，其中数据导入功能支持离线数据的批量导入，例如个人的诊疗数据导入等。

（4）医疗数据 OLAP 分析实现统计分析功能，可以细分为个体分析和群体分析两部分功能，其中个体分析侧重患者个体数据的统计，群体分析侧重患者群体的时空分析及与其他因素的关联。

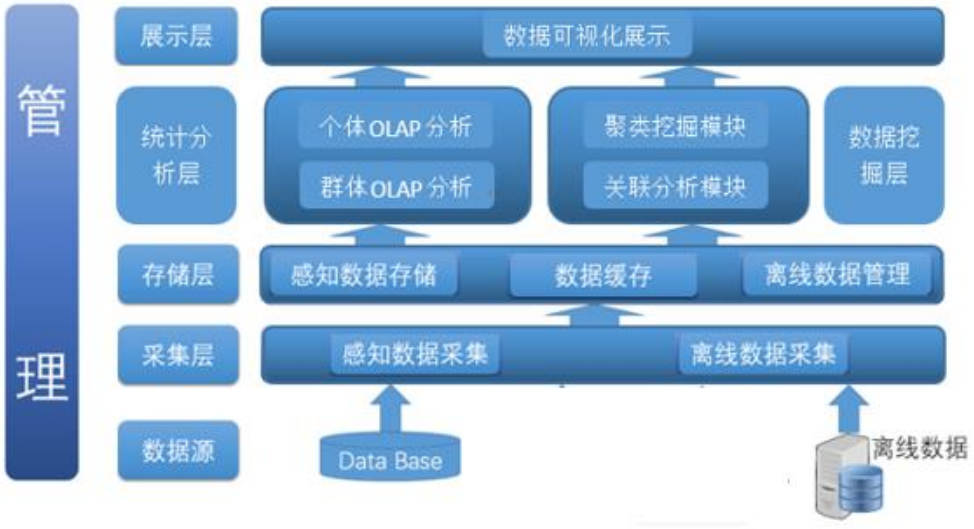
在个体分析中，患者除了可以查看个人的基本健康状况，还可以进行潜在疾病追踪。因此，个体分析中包括常规分析和异常分析等功能。

在群体分析中，患者的群体特征很大程度上反映出时空特性，通过分析不同时空尺度的患者群体，并结合其他因素可以得到更准确的分析和预测。

（5）数据挖掘分析通过对原始数据及统计数据的分析，实现疾病的模型学习，从而挖掘出疾病的规律及关联关系。通过数据预处理和聚类算法对病人进行画像，从而找到典型病人的相关指标情况，还可以通过关联规则算法对指标进行关联分析，挖掘指标的潜在规律。因此，数据挖掘模块包括疾病关联分析、诊疗方案推荐和呼吸机设置推荐等功能。

2.3 系统总体架构

本节给出系统的总体层次结构以及各层次的功能定位，图 2-2 给出本系统的总体层次结构图，共分为五个层次。



2-2 系统层次结构图

Fig 2-2 level structure diagram of system

1）采集层负责在线感知大数据和离线医疗数据的接入和 ETL(Extract-Transform-Load)管理，包括感知数据采集和离线数据采集两个功能。感知数据采集实现多类型感知数据采集框架的部署配置、数据缓存和 ETL 处理。离线数据

采集实现离线数据的导入和 ETL 处理，即离线档案的导入和异构数据处理。

由于数据采集层的感知数据采集和离线数据采集功能相对比较独立，故分别给出逻辑结构设计。

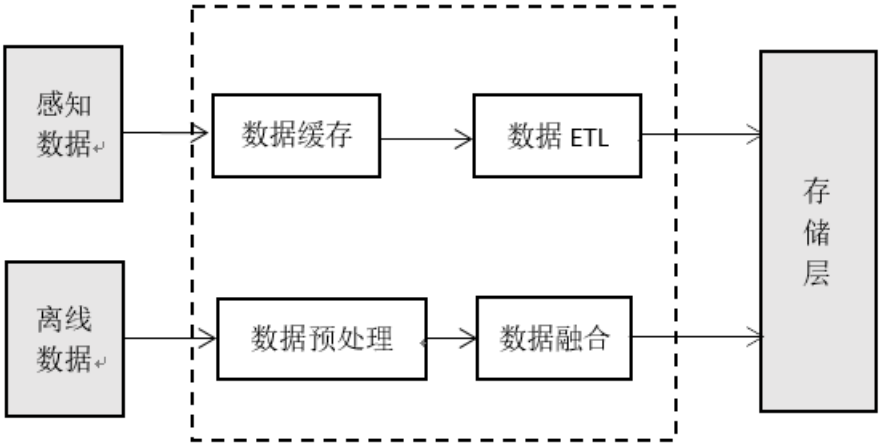


图 2-3 数据采集模块逻辑结构图

Fid 2-3 Logical structure diagram of data collection module

数据采集模块逻辑结构图如图 2-3 所示，包括感知数据采集和离线数据采集两个部分。

(1) 感知数据采集包括三个功能模块：数据缓存、数据 ETL 通道和数据采集控制。

数据缓存 —— 用于对多源在线流式数据进行分区缓存，确保接入数据的完整性。该模块使用 Kafka 软件工具通过配置缓存通道实现。

数据 ETL 通道 —— 用于对接入的流式数据进行 ETL 处理，其中一个数据 ETL 通道可以对应一个或多个在线流式数据源。该模块拟使用 Spark Streaming 软件工具作为基础运行框架，通过开发 Spark Streaming 应用实现各数据 ETL 通道实例。数据 ETL 通道的部署和运行管理可由 Spark 框架的内嵌机制完成。

数据采集控制器 —— 用于根据数据提供者的需求，在线自动配置相应的数据缓存区域，并与 Spark Streaming 平台交互，触发数据 ETL 通道的部署和启停功能。

(2) 离线数据采集包括数据预处理和数据融合两个功能模块。

数据预处理模块 —— 用于对异常或缺失数据进行转换或补全，由于离线数据庞大且杂乱，在对离线数据进行存储之前，需要对异常或缺失数据进行额外处理，使其不被系统进行数据分析。

数据融合模块 —— 用于对离线数据按照合理的数据标准进行融合处理，在对离线数据进行存储之前，需要对各种类型的离线进行融合，融合为统一的数据表达形式，既有利于提高系统效率，又有利于后期的统计方法设计。

2）存储层负责对于感知数据和离线医疗数据的存储管理，包括感知数据存

储、离线数据存储两个方面。由于读写模式的不同，在线和离线两类数据将采用不同的存储管理机制。其中在线数据（感知数据）存储实现对感知数据的缓存和持久化转存；离线数据（历史数据）存储实现对数据的基本原子操作，如增、删、改、查等。

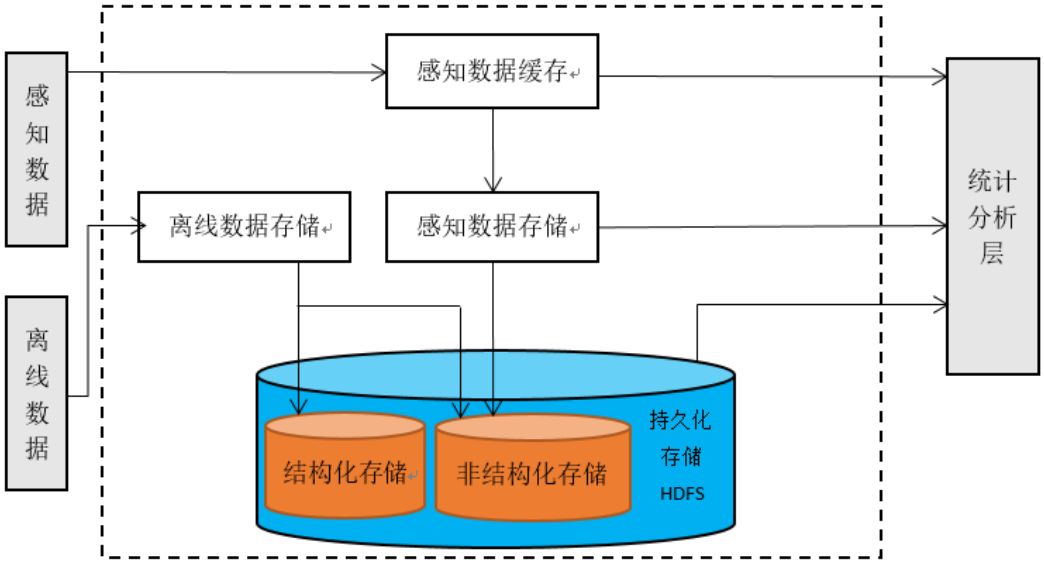


图 2-4 数据存储模块逻辑结构图

Fig 2-4 Logical structure diagram of data storage module

数据存储模块逻辑结构图如图 2-4 所示，包括感知数据缓存、感知数据存储、离线数据存储和持久化数据缓存四个功能模块，以及一个数据持久化存储库。其中，数据持久化存储库包含两类，一类是基于 Hadoop 的 HDFS 文件系统的非结构化存储库，另一类是基于 MySQL 传统数据库的结构化存储库。

感知数据缓存 —— 用于缓存新近接收的流式感知数据，作为实时数据查询的数据来源。该模块拟通过使用 Druid 软件工具并通过配置数据缓存参数实现。

感知数据存储 —— 用于周期性将感知数据缓存中的数据转存至持久化存储系统中。通过调研，针对感知数据普遍具有的只读特征，本系统选择将感知数据存储于 HDFS 文件系统。对于原始的结构化感知数据，通过数据 schema 描述文件，对其进行从结构化到非结构化转换。该模块通过使用 Druid 软件工具并通过配置数据转存参数实现。

离线数据存储 —— 用于对历史档案等数据进行存储管理。根据调研，离线数据可分为具有只读特征（如电子病历档案数据）和读写特征（如医院、病人管理数据）两类数据，其中，读写类数据往往具有结构化特征。因此，本系统将只读类离线数据存储于 HDFS 文件系统，读写类离线数据存储于 MySQL 数据库系统。离线数据存储模块区分两类存储需求，分别进行数据存储操作。该模块需要自行开发，通过调用文件系统及数据库接口实现。

持久化数据缓存 —— 用于定期读取并缓存持久化存储数据，以提升数据查询效率。目前，本系统仅提供对于持久化存储的感知数据缓存。该模块拟通过使用 Druid 软件工具并配置历史数据缓存功能实现。

3）统计分析层负责对在线感知数据和离线医疗数据进行统一的 OLAP 统计分析。既定的 OLAP 统计分析主要包括针对医生及决策人员提供呼吸疾病群体的疾病时空分布和影响因素相关性的统计分析；针对医生和疾病个体提供个体呼吸状态与时、空、生理特征和治疗方案等因素的相关性统计分析。

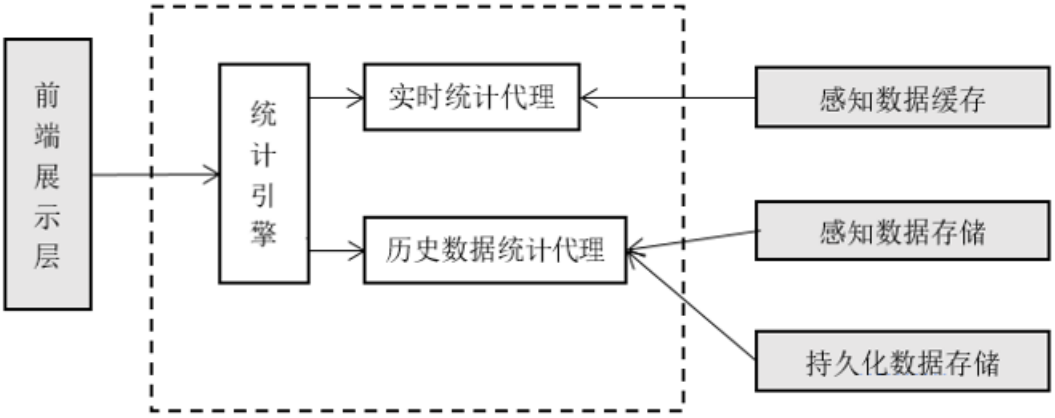


图 2-5 统计分析模块逻辑结构图

Fig 2-5 Logical structure diagram of statistical Analysis module

统计分析模块逻辑结构图如图 2-5 所示，包括统计引擎、实时数据统计代理和历史数据统计代理三个功能模块。

统计引擎 —— 用于接收和解析展示层的用户 OLAP 请求，根据统计数据存储来源的不同（区分缓存数据和持久化存储数据）进行 OLAP 请求分解并分发给相应的统计代理进行处理，最后再将统计代理的结果进行聚合返回。

实时数据统计代理 —— 用于处理统计引擎分发的 OLAP 查询，通过访问存储层的感知数据缓存，从中提取 OLAP 查询所需感知数据，将其整合后发送给统计引擎。该模块使用 Druid 软件工具并通过配置实时数据查询功能实现。

历史数据统计代理 —— 用于处理统计引擎发送的 OLAP 查询，通过访问存储层的持久化感知数据缓存和持久化数据存储，从中提取 OLAP 查询所需持久化存储数据，将其整合后发送给统计引擎。该模块使用 Druid 软件工具并通过配置历史数据查询功能实现。

统计分析层的外部输入来源包括展示层的用户 OLAP 请求、存储层的感知缓存数据和持久化感知缓存数据，外部输出是展示层的统计界面展示。

4）数据挖掘层负责对原始医疗数据或统计分析的抽象数据通过使用机器学习算法进行挖掘分析，为医生优化诊疗方案以及医疗决策者进行宏观决策提供依据。既定的业务包括呼吸机设置值推荐，诊疗推荐和病情关联分析。

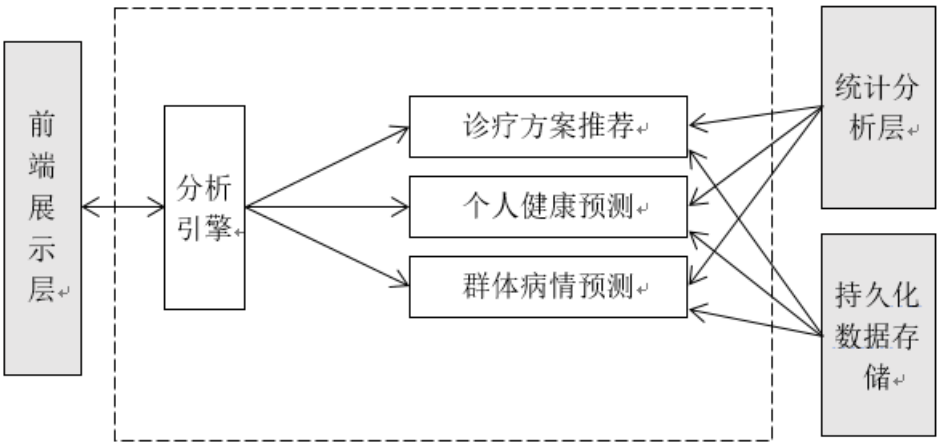


图 2-6 数据挖掘模块逻辑结构图

Fig 2-6 Logical structure diagram of data mining module

数据挖掘层包含两个功能模块：请求分析引擎和数据挖掘分析业务代理。

请求分析引擎 —— 用于解析输入的智能决策请求，激活相应的业务代理，并将分析计算结果反馈到决策界面层。

数据挖掘分析业务代理 —— 用于响应智能决策请求，激活算法目标业务模块，根据计算需求，从存储层的持久化数据存储获取原始医疗数据或从统计引擎中获取统计结果数据，调用 MLlib 库算法或自建 Java 包算法，根据历史数据回朔进行模型学习和知识挖掘，输出包括诊疗建议、健康/疾病早期预测以及疫情风险分析的智能决策分析计算结果，反馈给请求界面层，为管理决策提供依据。

5）展示层负责为最终用户提供图形化的系统访问入口，并为医疗数据查询和分析的结果提供可视化展示。输出内容为可交互的展示图形,主要采用热力图、柱状图、折线图等形式进行可视化展示。

2.4 本章小结

本章介绍了医疗大数据分析系统的整体业务需求和总体架构以及各层次的功能模块。本系统主要分为五个层次，其中主要业务和工作聚集在两个层次。第一个层次是统计分析层，使用的技术为联机分析处理（OLAP），是基于实时分析框架 Druid 作为数据仓库的查询统计分析，主要业务是针对个人及群体对象，结合时空的 OLAP 分析。第二个层次是数据挖掘层，是基于开源大数据框架 Spark 的数据挖掘分析，利用原始数据及统计分析数据，应用相关机器学习算法进行数据挖掘分析，主要业务是对病人进行画像和呼吸机设置推荐及病情关联分析。

第 3 章　基于 OLAP 的数据建模和多维统计分析

3.1 引言

对于医疗行业数据分析来说，统计查询是十分重要的分析手段，在业务中也占据了主要的地位，可以说全面、完善的统计查询，可以提供病人和医生大多数的数据分析以及病情趋势的分析。所以本系统在查询统计模块，应用了数据仓库的多维数据模型以及联机分析处理的技术和思想，基于联机分析处理的技术对呼吸睡眠障碍和治疗大数据进行分析，不仅包括对多维数据模型的的统计查询功能，还在时间维度上实现了联机分析处理特有的上卷、下钻、切片、切块功能，使分析结果更加精准[30-33]，更好满足了用户对复杂数据的多样化统计分析需求。

3.2 相关技术概述

3.2.1 联机分析处理（OLAP）

数据的分析处理主要分成两大类：联机事务处理（OLTP）和联机分析处理（OLAP）。联机事务处理面向的存储系统是关系型数据库（DataBase），业务内容针对的是简单的事务处理[34]。联机分析处理面向的存储系统是大型的数据仓库（Data Warehouse），业务内容更侧重复杂、多维的分析。表 3-1 列出了 OLTP 与 OLAP 之间的比较和区别。

表 3-1 OLAP 与 OLTP 区别

Table3-1 Difference of OLAP and OLTP

	OLAP	OLTP
用户	决策人员，高级管理人员	操作人员，低层管理人员
功能	分析决策	日常操作处理
DB 设计	面向主题	面向应用
数据	历史的，聚集的，多维的集成的， 统一的	当前的，最新的细节的，二维的分立的
存取	读上百万条记录	读写数十条记录
工作	复杂的查询	简单的事物
DB 大小	TB 及以上级别	GB 及以下级别

联机分析处理的特点是可以从多个角度对业务进行深层次的分析，传统的统计分析会根据每一个维度或每两个维度进行查询，将结果形成报表。但是如果维度很多的情况下，每种查询都需要建立相应的数据模型进行查询，这会消耗很大的人力物力，而且很难形成综合、多角度的数据模型，使结果不够全面和深入。

而 OLAP 可以很好的解决上述传统数据分析的问题，直接综合数据仓库中的不同数据维度，预先建立好多维数据模型。该模型包含了业务所需的所有维度，分析人员可以直接对多维数据模型进行查询统计，既可以从某些单一角度进行分析，也可以综合多个维度进行分析，大大增强了分析的灵活性和多样性，也不必像传统分析一样建立多个数据模型，减少了分析人员的工作量。

OLAP 建立的多维数据模型，展现在用户面前的是多维视图。图 3-1 为联机分析处理构建的多维数据模型示意图，多维数据模型包含三种描述，维的层次、维的成员和度量。

维的层次（Level）—— 即人们对数据进行分析的不同角度，如时间、地区等。根据维度层次的不同变化，用维的粒度（Granularity）加以刻画，如时间维度的粒度包括年、月、日等。

维的成员（Member）—— 即某一个维度的一个具体取值，如时间维度的一个成员为（2017 年 3 月 1 日）。

度量（Measure）—— 即多维数据模型的一个具体取值，也可以为综合多个维度的一个具体取值，如综合时间、地区、销量三个维度的一个度量为（2017 年 3 月，北京，100）。

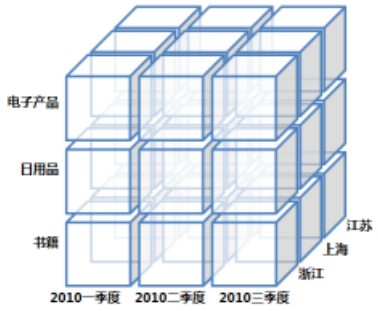


图 3-1 数据模型示意图

Fig 3-1 Schematic diagram of data model

OLAP 的基本多维分析操作有钻取、切片和切块等。

钻取（Drill）—— 即对维的粒度进行不同的变换。钻取包括向上钻取（Drill-up）和乡下钻取（Drill-down）。向上钻取指的是将某一维度较小粒度的数据向较大粒度的层次进行汇总，如在时间维度上由以日为粒度改为以月为粒度。向下钻取与向上钻取相反，是将某一维度较大粒度的数据进行深层探索，得到较小粒度层次的数据。如在时间维度上由以年为粒度改为以月为粒度。图 3-2 和图 3-3 分别为根据图 3-1 的数据模型进行的向下钻取和向上钻取操作的示意图：

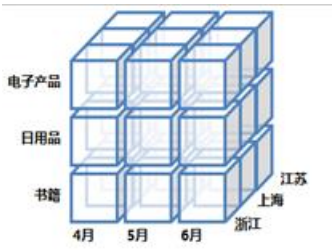


图 3-2 向下钻取示意图

Fig 3-2 Schematic diagram of drill-down

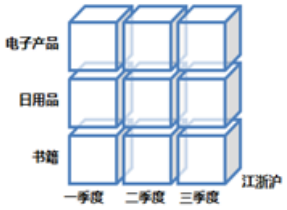


图 3-3 向上钻取示意图

Fig 3-3 Schematic diagram of drill-up

切片/切块 —— 即在数据模型中，一部分维度的成员已经确定的情况下，观察度量数据在其他维度上的分析情况。如果确定的维度成员为一个值，则是切片；如果确定的维度成员为多个值，则是切块。图 3-4 和图 3-5 分别为根据图 3-1 的数据模型进行的切片和切块操作的示意图：

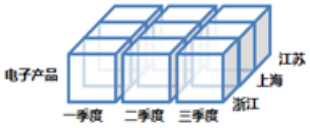


图 3-4 切片示意图

Fig 3-4 Schematic diagram of slice

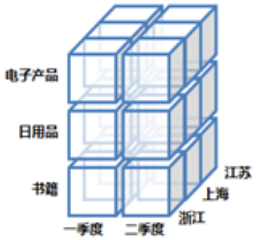


图 3-5 切块示意图

Fig 3-5 Schematic diagram of cut

3.2.2 实时处理框架 Druid

Druid 是一个开源的分析数据存储，用于对事件数据进行商业智能（OLAP）查询。 Druid 提供低延迟（实时）数据摄取，灵活的数据探索和快速数据聚合。现有的 Druid 部署已经扩展到数万亿的事件和 PB 级数据。 Druid 最常用于为面

向用户的分析应用程序提供支持。Druid 系统的可靠性很高，对于意外出现的问题如机器故障、宕机等情况，Druid 通过 zookeeper 协调服务仍然可以保证正常运行。除此之外，Druid 综合了查询的速度和灵活性采取了 Json 的存储格式和查询方式。

Druid 系统综合了 Google 公司的 PowerDrill 和 Dremel 两种分析系统，实现了 Dremel 的全部功能的同时，也借鉴了 PowerDrill 中对一些特殊数据的支持。Druid 具有以下主要特征：

亚秒级 OLAP 查询 —— Druid 的列方向和反向索引可以进行复杂的多维过滤和扫描查询所需的内容，以毫秒为单位汇总和过滤数据。

实时流式摄入 —— 典型通过批量获取数据的数据库。一次捕获事件通常伴随着事务锁和其他开销，这会降低吞吐速率。Druid 采用无锁摄取附加重量数据集，允许以每秒 10000+个事件同时采集和查询每个节点。简单来说，事件发生时间和可见时间之间的延迟只受到事件可以多快地传递给 Druid 的限制。

功率分析应用 —— Druid 有许多内置的多租户功能。强大的面向用户的分析应用程序，旨在供数千并发用户使用。

成本效益 —— Druid 具有众多用于降低成本的功能，如通过简单的配置旋钮降低成本，在设计上极具成本效益。

高可靠性 —— Druid面对系统升级、机器故障等问题，仍然可以正常使用，无论集群数量增加还是减少都不会造成数据丢失。

高效率 —— Druid 系统面对 PB 级的数据量仍然可以保持高效，每天可以轻松处理数以亿记的数据记录，Druid 本身是被设计来解决 PB 级别数据的。

图 3-6 为 Druid 的体系结构，展示了查询和数据是如何运转的，图 3-7 为 Druid 集群的管理层架构，主要展示了外部依赖节点和集群节点之间的关系。Druid 系统主要包括 5 种分离的节点：

HISTORICAL —— 历史节点主要是对历史数据（离线数据）进行处理，包括存储和查询等工作。历史节点从深度存储下载段，响应来自代理节点关于这些段的查询，并将结果返回到代理节点。

REALTIME —— 实时节点将实时数据加载到系统中。它们比索引服务更容易设置，但是以生产使用的一些限制为代价。

BROKER —— 是客户端和数据仓库的代理节点，主要作用把客户端查询分发到对应的历史节点和实时节点，并将查询的结果返回给客户端。

INDEXING —— 索引服务节点中包含一个加载节点，可以将批量数据加载到集群中，索引服务节点可以帮助对系统中的数据进行增、删、改、查等操作。

COORDINATOR —— 协调器节点监视历史节点的分组，使历史节点中的数据

处于可修改、可复制的状态，并且保证“最佳”的配置。

除了这些节点之外，系统还有 3 个外部依赖，共同构成了 Druid 系统的管理架构。

ZOOKEEPER —— 运行的 ZooKeeper 集群，用于集群服务发现和当前数据拓扑的维护。

METADATA STORAGE —— 元数据存储实例，通常为关系型数据库。用于维护由系统提供的数据段的相关数据。

DEEP STORAGE —— “深度存储”主要是由 LOB（large object）存储或文件系统承担，用于存储系统的所有数据。

这种分离允许每个节点只关心节点本身是否是正常的。通过分离历史和实时处理，我们分离了对实时数据流进行监听并对其进入系统进行处理的内存关注。通过分离协调器和代理，我们将查询的需求与在整个集群中维护“良好”数据分布的需求分开。

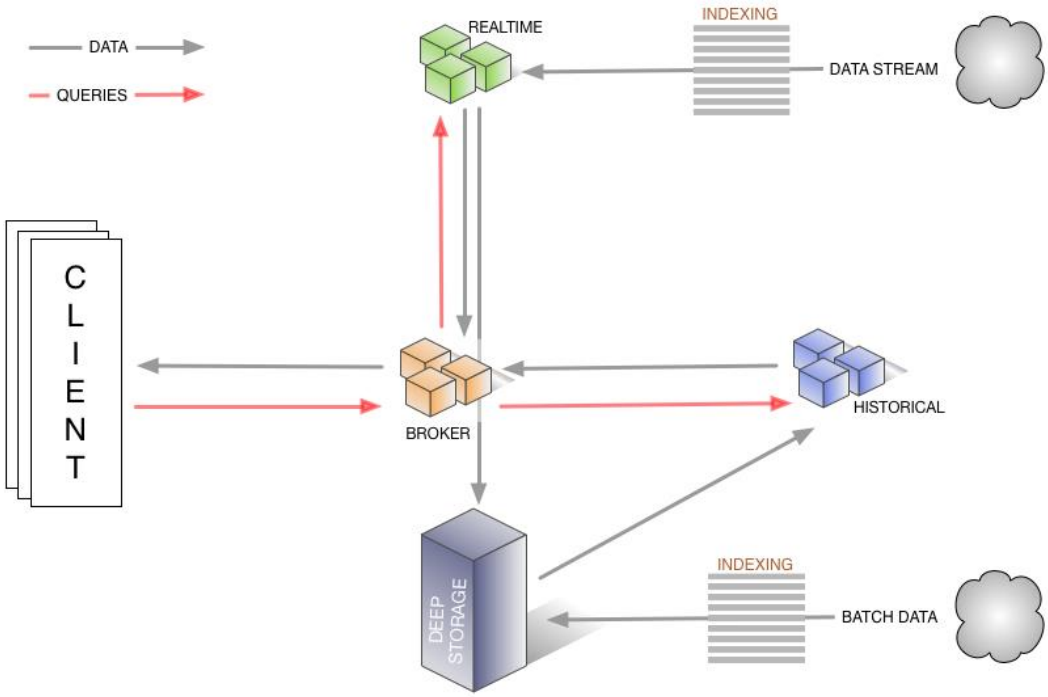


图 3-6 Druid 体系结构图

Fig 3-6 Architecture diagram of Druid

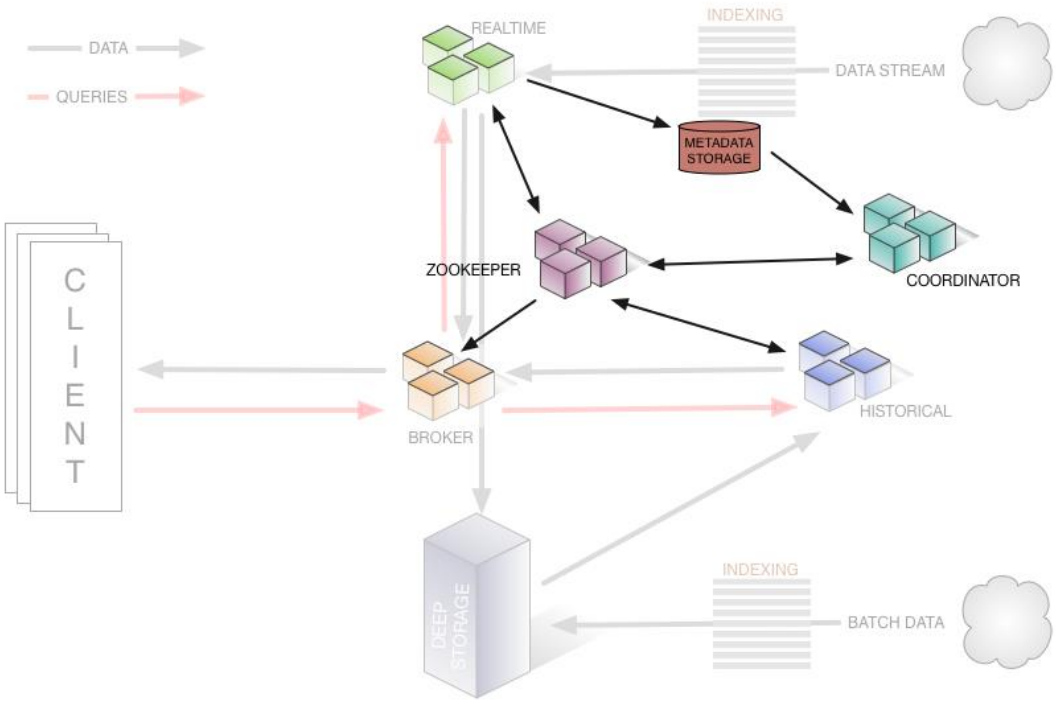


图 3-7 Druid 管理架构图

Fig 3-7 Management diagram of Druid

3.3 联机分析处理模块业务层

3.3.1 OLAP 多维数据模型创建

在 3.2 节中，介绍 OLAP 分析是预先根据数据仓库的数据模型创建一个多维数据模型。模型包括业务人员需要用的事实表和维度表，并将它们按照逻辑关系关联在一起，这样形成的多维数据模型可以使分析人员快速查询到相应的数据，并且可以灵活的切换不同维度来实现不同业务的逻辑。多维数据模型是数据仓库的主要应用方式和实现形式，旨在提供更完整，更灵活的 OLAP 分析决策支持。

在介绍多维数据模型的分类之前，首先介绍一下事实表和维度表的概念。事实表指的是记录具体发生的事件的集合，它包含了具体发生的事件，以及每个事件的具体要素，如（ID，发生事件日期，发生时间地点，事件 ID）为事实表中的一条记录。维度表指的是事实表中发生事件的要素的描述信息的集合，如（事件 ID，事件类型，事件名称）为事件维度表的一条记录。

下面介绍一下多维数据的分类。

（1） 星型模型：一种非正规化的结构，多个维度表直接与事实表相连，所有维度表都服务于同一个事实表，分别记录事实表中的不同要素的描述信息，

不存在渐变维度[35]。

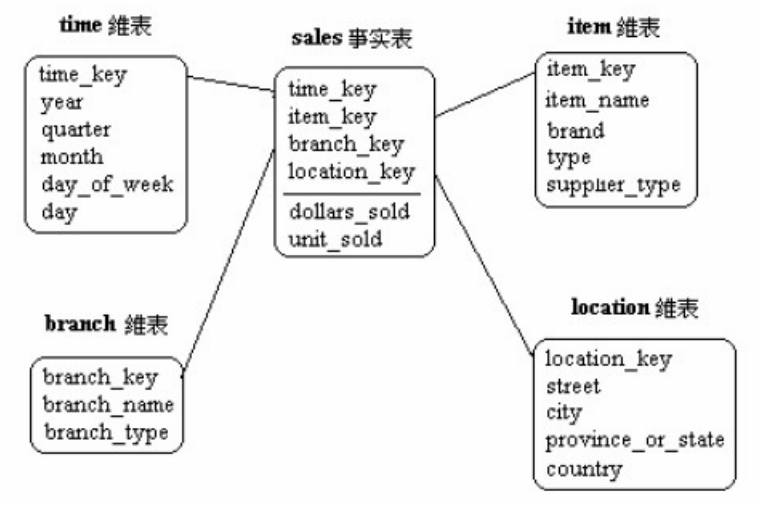


图 3-8 星型模型示意图

Fig 3-8 Schematic diagram of star model

（2） 雪花模型：雪花模型是星型模型一种扩展，即一个事实表仍然连接多个维度表，但是某些维度表可能被扩展，仍有其他维度表与此维度表相连，形成渐变维度。这样可以较大限度的减少数据的存储量，增强查询性能[36]。

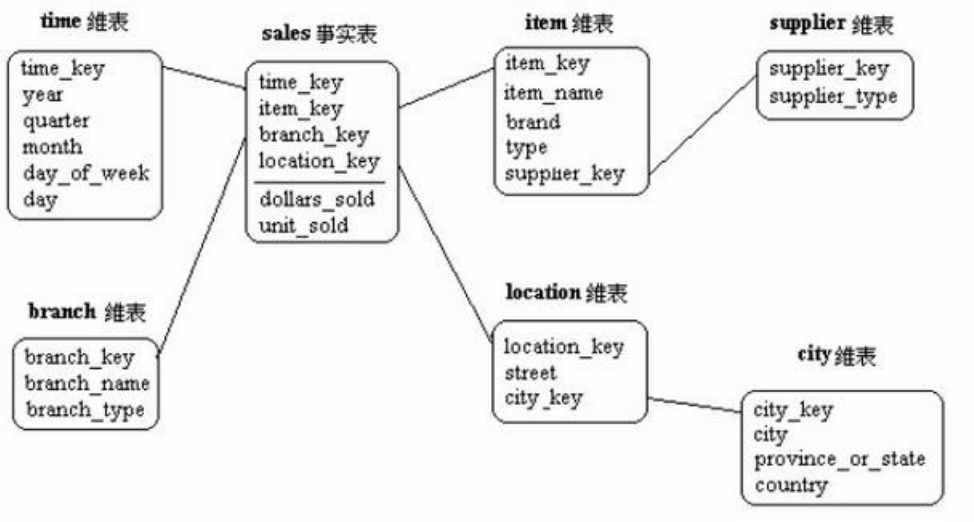


图 3-9 雪花模型示意图

Fig 3-9 Schematic diagram of snowflake model

本系统采用呼吸机数据作为开发数据，数据表结构如下：

事实表包括基本信息采集表和异常事件表。主要记录了病人使用呼吸机的常规采集数据记录和当异常事件发生时的时间记录。表结构如表 3-2 和表 3-3 所示：

表 3-2 基本信息采集表

Table 3-2 Basic information collection table

主键	字段	类型	默认值	备注
√	ID	整型		非空，自增，ID
	COLLECTION_START_TIME	timestamp	0000-00-00 00:00:00	非空，信息开始采集的时间
	LEAK_VALUE	4 位整型	NULL	泄露_值
	TIDALVOLUMEVALUE	3 位整型	NULL	潮气量值
	BPMVALUE	4 位整型	NULL	每分钟节拍数
	CREATETIME	timestamp	当前时间	非空，信息建立时间
	SPO_VALUE	4 位整型	NULL	血氧_值
	HEARRATE_VALUE	4 位整型	NULL	心率_值
	USER_ID	50 位字符型		用户 ID
	DEVICE_ID	30 位字符型		设备 ID

表 3-3 异常表

Table 3-3 Exception table

主键	字段	类型	默认值	备注
√	ID	整型		非空，自增，ID
	EVENT_TIME	timestamp	0000-00-00 00:00:00	非空，事件发生时间
	EVENT_ID	3 位整型	NULL	事件 ID
	EVENT_VALUE	4 位整型	NULL	事件_值
	CREATETIME	timestamp	当前时间	非空，信息建立时间
	USER_ID	50 位字符型		用户 ID
	DEVICE_ID	30 位字符型		设备 ID

维度表包括用户表、设备表、事件表、治疗表、医生表、和医院表。主要记录了用户、设备、医生和医院等基本维度信息和异常事件、医生治疗的相关维度信息。表结构如表 3-4 到表 3-9 所示：

表 3-4 用户(患者)表

Table 3-4 User table

主键	字段	类型	默认值	备注
√	USER_ID	50 位字符型	'	非空，用户_ID
	USER_NAME	50 位字符型		非空，用户姓名
	PASSWORD	150 位字符型		非空，密码
	USER_ACCOUNT	50 位字符型		非空，用户账户
	AGE	10 位字符型	NULL	年龄
	HEIGHT	11 位整型	NULL	身高

	WEIGHT	11 位整型	NULL	体重
	BMI	精确型，5 位整数，2 位小数	NULL	身体质量指数
	ZIPCODE	20 位字符型	NULL	邮编
	PROVINCE_NAME	50 位字符型		省编码
	CITY_NAME	50 位字符型		市编码
	DISTRICT_NAME	50 位字符型		区编码
	DETAIL_ADDRESS	50 位字符型	NULL	详细地址
	ADDRESS	300 位字符型	NULL	地址
	LONGITUDE	50 位字符型		经度
	LATITUDE	50 位字符型		纬度

表 3-5 设备表

Table 3-5 Device table

主键	字段	类型	默认值	备注
√	DEVICE_ID	30 位字符型	'	非空，设备_ID
	DEVICE_NAME	80 位字符型	NULL	设备_名称
	DEV_VERSION	20 位字符型	NULL	设备_型号
	HOSPITAL_ID	150 位字符型	0	非空, 医院编码

表 3-6 事件表

Table 3-6 Event table

主键	字段	类型	默认值	备注
	EVENT_ID	3 位整型		事件 ID
	EVENT_NAME	50 位字符型		事件名称
	EVENT_DESCRIBE	200 位字符型		事件描述

表 3-7 治疗表

Table 3-7 Treat table

主键	字段	类型	默认值	备注
√	ID	整型	自增	ID
	USER_ID	50 位字符型		用户 ID
	DOCTOR_ID	50 位字符型		医生 ID
	DEVICE_ID	30 位字符型		非空，设备_ID
	START_TIME	timestamp		起始时间
	TREAT_ADDRESS	300 位字符型		治疗地点

表 3-8 医生表

Table 3-8 Doctor table

主键	字段	类型	默认值	备注
	DOCTOR_ID	50 位字符型		医生 ID
	DOCTOR_NAME	50 位字符型		医生姓名

	PASSWORD	150 位字符型		非空，密码
	DOCTOR_ACCOUNT	50 位字符型		非空，用户账户
	DEPARTMENT	150 位字符型		所属科室

表 3-9 医院表

Table 3-9 Hospital table

主键	字段	类型	默认值	备注
，√	HOSPITAL_ID	150 位字符型	0	非空，医院编码
	PASSWORD	150 位字符型		非空，密码
	HOSPITAL_ACCOUNT	50 位字符型		非空，用户账户
	HOSPITAL_NAME	150 位字符型	NULL	医院名称
	HOSPITAL_ADDRESS	150 位字符型	NULL	医院地址
	LONGITUDE	50 位字符型		经度
	LATITUDE	50 位字符型		纬度

本系统根据业务逻辑及事实表和维度表的数据结构，设计了如图 3-10 的多维数据模型作为本系统数据仓库的数据模型：

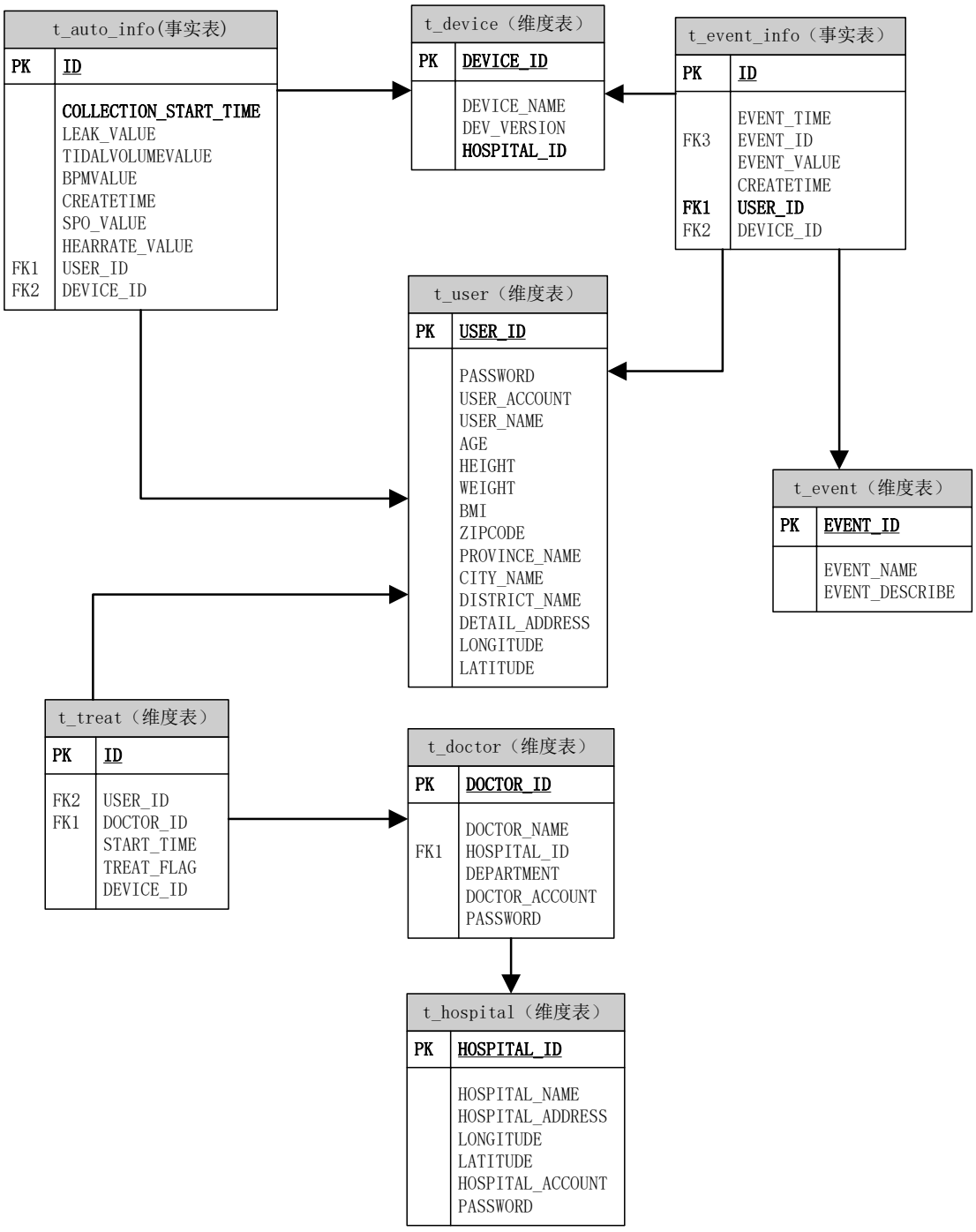


图 3-10 本系统多维数据模型

Fig 3-10 Multidimensional data model of this system

该模型结合了星型模型和雪花模型的特点，围绕两个事实表设计出了符合本系统业务逻辑的多维数据模型。两个事实表都直接与设备表与用户表两个维度表相连，并且以用户表作为扩展的“事实表”，又扩展出治疗表、医生表和医院表，形成多层次的模型。而异常事件表作为事实表，又与事件表单独相连，形成类似星型模型的结构。本系统把这种结合两种模型的多事实表和多维度表的数据模型称作事实星座模型。

3.3.2 分析引擎设计

分析引擎是统计分析模块的核心功能。由于本系统是 B/S 结构，用户在浏览器前端进行业务的交互操作，而服务器后端则根据用户的操作参数进行对应的业务统计，实现数据仓库 OLAP 的分析，其中分析引擎就是两部分的桥梁。本系统的分析引擎流程图如图 3-11 所示，主要包括两部分，第一部分是参数解析引擎，第二部分是 OLAP 查新引擎。

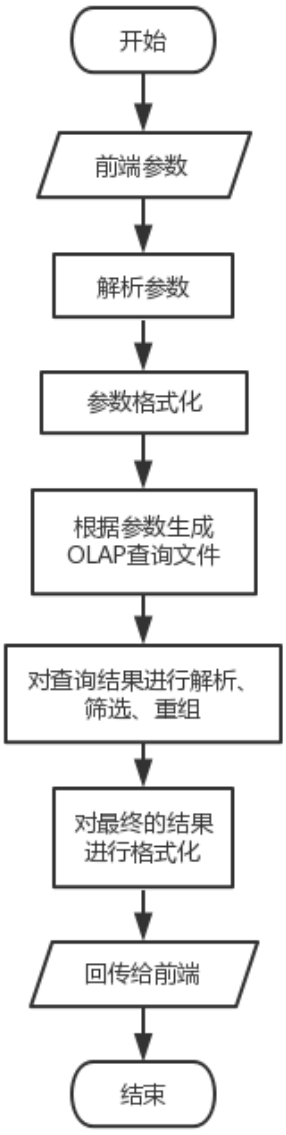


图 3-11 分析引擎总体流程图

Fig 3-11 Overall flow chart of analysis engine

参数解析引擎主要是用于接收前端用户的使用参数，进行解析判断是何种操作，将参数格式化为 OLAP 引擎可识别的参数，将参数传给 OLAP 查询引擎进行

OLAP 查询，并将 OLAP 引擎的查询结果进行格式化，转换为前端所需的参数格式进行返回。

OLAP 查询引擎主要是接收参数解析引擎传来的参数，根据参数内容生成 Druid 数据仓库的查询文件，进行 OLAP 的多维查询，并将结果进行相应的筛选和重组，形成用户所需的参数列表，回传给参数解析引擎进行格式化。

3.3.3 个体 OLAP 分析

本系统针对不同的用户类别，提供不同的统计分析功能。针对病人个体用户，他们最关注的就是自身的指标状况，以及不同时间段内的病情变化趋势。针对这些业务，本系统设计了个体 OLAP 分析模块，旨在提供个体用户可选时间段可选查询粒度的病情分析，以报表的形式展示。本系统的个体 OLAP 分析流程图如图 3-12 所示，可以提供病人进行个人的 OLAP 分析，病人可以根据选择时间维度（查询的起始时间），并且选择时间粒度进行病情查询，查询结果分为三个部分：

第一部分为病人的个人信息，包括个人的生理情况和所属医生。

第二部分为所选时间段内，以选择的时间粒度为单位，各项指标的折线图/柱状图/饼状图，并且对应各指标的最大值/最小值/平均值等信息。

第三部分为病人所用设备的信息，包括设备版本号，型号等信息。

图 3-12 个体 OLAP 分析流程图

Fig 3-12 Analysis flow chart of individual OLAP

个体 OLAP 查询是以报表的形式对病人进行各项指标的展示，并且结合各维度表输出了个人的生理指标以及设备的相关信息。

3.3.4 医生群体 OLAP 分析

对于医生用户来说，最关注的是不同病人群体的指标变化情况。针对医生用户的需求，本系统设计了针对医生的群体 OLAP 分析模块，提供医生在可选时间范围内，选择不同病人进行指标的统计和 OLAP 上卷、下钻等分析操作。本系统的医生群体 OLAP 分析流程图如图 3-13 所示，可以提供医生进行群体的 OLAP 分析，首先医生可以选择时间维度（起始时间）、选择初始的时间粒度、并且根据治疗维度表关联显示的病人，选择其中的一个或多个进行群体查询。查询结果为群体综合的各项指标的折线图，可以看到总体的变化趋势。

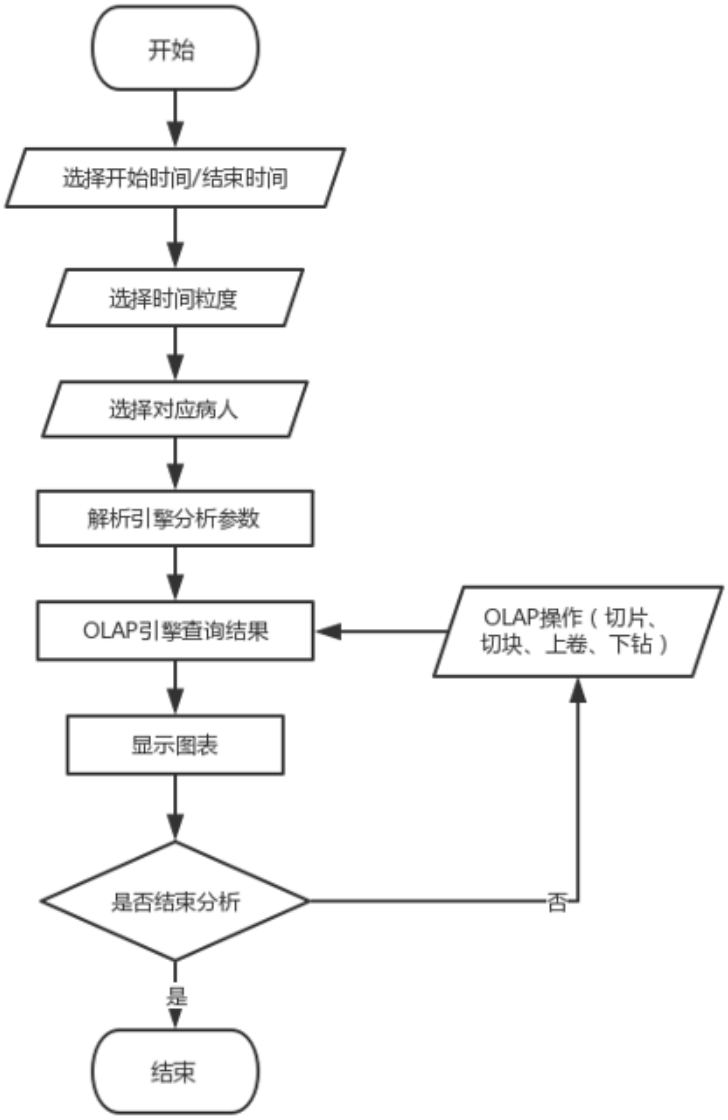


图 3-13 医生群体 OLAP 分析流程图

Fig 3-13 Group OLAP Analysis flow chart of doctor

医生看到分析结果后还可以执行更多的 OLAP 操作，如切片（可以选择其中的一项指标单独查看）、切块（可以选择其中的几项指标综合查看）、下钻（点击图表对应时间点的坐标点，进行时间维度的下钻）、上卷（点击相应的按钮，进行时间维度的上卷）等进行进一步的 OLAP 分析。

3.3.5 决策者群体 OLAP 分析

与医生用户不同，对于决策者用户来说，关注的是可能是更为宏观的病情统计分析。如病情在不同地域的发展情况、病人在不同城市的分布等等。针对以上业务，本系统设计了针对医生的群体 OLAP 分析模块，提供医生在可选时间范围内，选择不同病人进行指标的统计和 OLAP 上卷、下钻等分析操作。本系统的决

策者群体 OLAP 分析流程图如图 3-14 所示，可以提供决策者结合时空信息进行群体的 OLAP 分析，首先医生可以选择时间维度（起始时间）、选择初始的时间粒度以及地域维度进行群体查询。查询结果为群体综合的各项指标的折线图，可以看到总体的变化趋势，以及对应地域及时段的病情分布热力图。

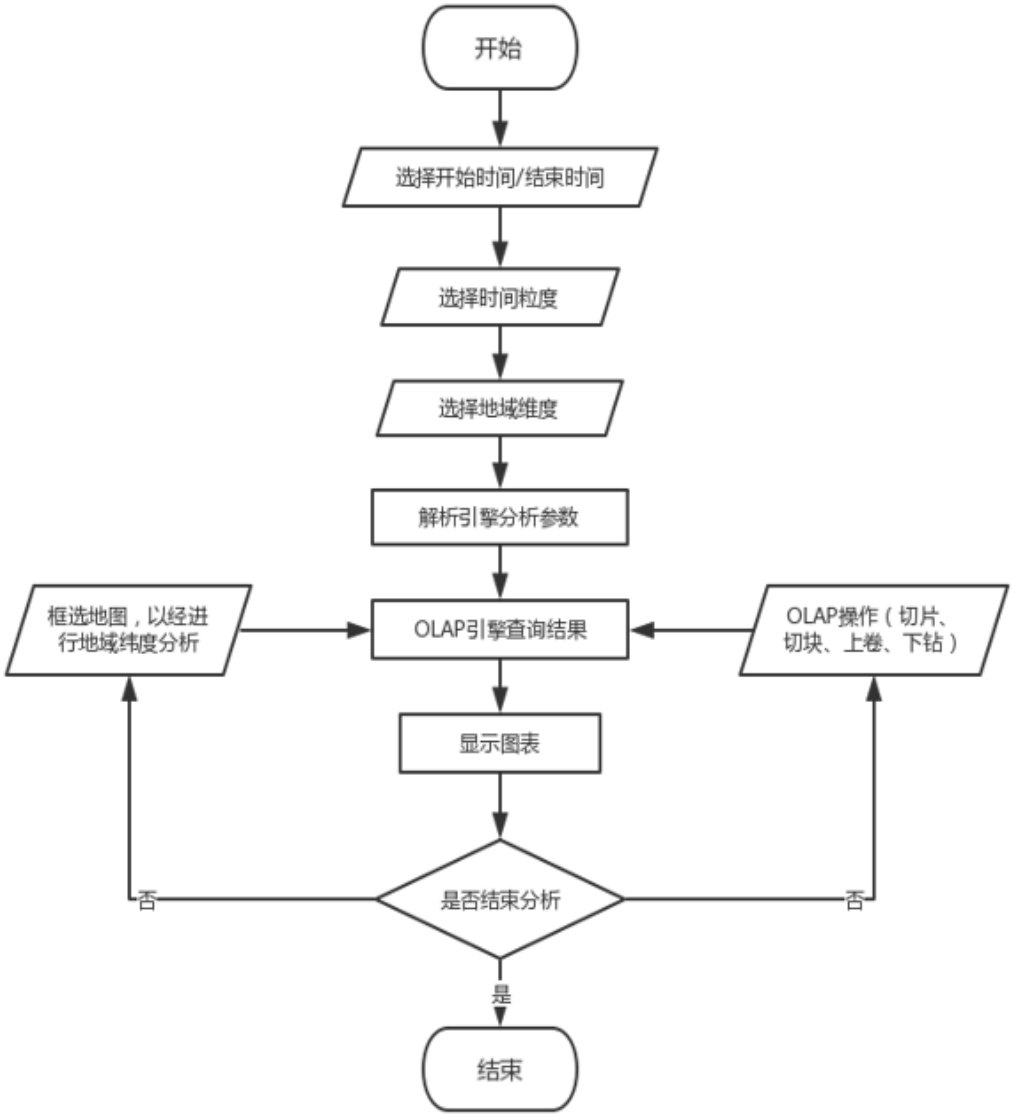


图 3-14 决策者群体 OLAP 分析流程图

Fig 3-14 Group OLAP analysis flow chart of decision makers

决策者看到分析结果后还可以对地图进行框选，系统会获取框选到范围的经纬度，将地域维度改为经纬度进行二次分析，还可以执行更多的 OLAP 操作，如切片（可以选择其中的一项指标单独查看）、切块（可以选择其中的几项指标综合查看）、下钻（点击图表对应时间点的坐标点，进行时间维度的下钻）、上卷（点击相应的按钮，进行时间维度的上卷）等进行进一步的 OLAP 分析。

3.4 本章小结

本章主要介绍了本系统的统计分析层，使用大数据实时分析框架 Druid 作为本系统的数据仓库，分析层主要运用了数据仓库的 OLAP 应用，通过对事实表、维度表的数据结构和业务逻辑的分析和总结，设计出了适合本系统的多维数据模型。

通过本系统的事实星座多维数据模型，实现了对于不同用户群体的 OLAP 分析，其中包括病人的个人 OLAP 分析、医生的群体 OLAP 分析和决策者的 OLAP 分析。

对于病人而言，更关注的自己的健康状况、病情严重程度以及相关的设备信息。所以对于病人的个体 OLAP 分析包括了病人的各项指标的具体情况，包括每项指标的峰值和平均值。

对于医生而言，关注的是自己所管理的病人的所有人的病情的变化趋势，以便及时针对病情做出相应的治疗。所以对于医生的群体 OLAP 分析包括了一到多个病人的病情汇总情况，以及各项指标的变化趋势，医生还可以针对某一项指标或者针对某一段时间的异常值进行进一步的分析观察。

对于决策者而言，关注的是不同地区、不同经纬度的病人的情况，可以根据病情分布情况来选择增加设备的数量或医疗建设。所欲所以对于决策者的群体 OLAP 分析包括了在地域维度上的扩展、经纬度与地区的转换，以及类似医生的相关 OLAP 操作。

第 4 章　基于聚类算法和关联规则算法的数据挖掘分析

4.1 引言

随着人工智能的不断发展，数据挖掘分析已经广泛应用于各行各业的分析决策中，在医疗数据分析中也不例外。传统的查询统计已经不能满足人们对于数据的分析要求，挖掘数据之间潜在的规律和关联关系，是现今数据分析的重要方向和热点。本系统完成基于联机分析处理的统计查询基础上，使用相关机器学习算法对数据进行挖掘和分析，旨在挖掘潜在的规律，为医疗人员提供更深层的数据分析，为医疗工作者提供决策依据。

4.2 数据挖掘层聚类模块

通常病人在就医过程中，随着病情变化需要获得及时的治疗方案更新，而且根据个体生理差异，更需要个性化的诊疗方案，但目前医患比例仍然无法满足如及时追踪、个性化诊疗的需求。随着大数据技术的发展，通过海量数据挖掘、建模，实现自动病情分析、诊疗方案建议，为病人提供及时的个性服务，为医生提供智能的诊断分析依据，成为可能。本文探讨了大数据医疗分析系统中，实现数据挖掘层聚类分析的技术方案，针对自动病情分析、诊疗方案建议问题，采用聚类算法对病人数据进行聚类分析[37-40], 学习典型病情模型，在此基础上提供智能的治疗方案建议。以呼吸睡眠疾病历史诊疗数据、治疗中呼吸机设置数据为基础，完成了以下两个业务的智能分析计算功能模块：

呼吸机设置推荐业务：采用呼吸机治疗，是目前睡眠呼吸障碍疾病常采用的无创治疗方案，而呼吸机的参数设置是治疗中重要因素，同时受多种因素影响，通常需要个性化配置。本课题研究了如何根据病人的生理指标（年龄、体重、身高等）、病理指标（心率、血氧等）以及常用的呼吸机参数设置，使用聚类算法对这些复杂因素进行聚类，挖掘出病情所呈现的分布规律，建立几种典型模型。根据病人生理指标以及病理指标，推测出适合的呼吸机参数设计，实现自动设置推荐。

诊疗方案推荐业务：根据病人自身指标的差异诊疗方案有所差异，实现精准医疗是目前发展方向，睡眠呼吸亦存在个体差异在治疗方案的差异。本课题提出了根据病人的各项指标及相应的代表性诊疗方案，使用聚类算法对不同的生理指标、病理指标进行聚类，挖掘出病情的典型模型，结合医生对典型病例的诊疗方

案实现有针对性的诊疗方案推荐。

4.2.1 聚类算法概述

目前，已有大量的聚类算法应用于数据分析、用户画像等业务中。综合研究了现有的主流聚类算法，可以大致分为如下几类：基于划分方法的聚类算法，如k-means 算法；基于层次方法的聚类算法，如凝聚型层次聚类算法；基于密度的聚类算法，如 DBSCAN 聚类算法；基于网格的聚类算法，如 WaveCluster 聚类算法；基于模型的聚类算法，如神经网络聚类算法等[41-46]。上述 5 类算法都属于硬聚类算法（即每个数据都只能被归为不同类别中的其中一类），除此之外，聚类算法中的另一个分支称为模糊聚类。模糊聚类算法思想是找出一个隶属函数来描述每个数据隶属于每个类的程度，而不是将数据直接硬性的归属到某一类，如FCM聚类算法[47-48]。

本节主要针对 4 种应用广泛的典型聚类方法中进行介绍及相关的公式推导。

（1）基于模糊聚类的 FCM 聚类算法

模糊 C 均值（Fuzzy C-means）算法，简称 FCM 算法，是用隶属度确定每个数据点属于某个聚类的程度的一种聚类算法。1973 年，Bezdek 提出了该算法，作为早期硬 C 均值聚类（HCM）方法的一种改进。

假定数据集为 X，如果把这些数据划分成 c 类的话，那么对应的就有 c 个类中心为 C，每个样本 j 属于某一类 i 的隶属度为u_{ij}，那么定义一个 FCM 目标函数（4-1）及其约束条件（4-2）如下所示：

$$J = \sum_{i=1}^{c}\sum_{j=1}^{n} u_{ij}^{m} ||x_i - c_i||^2 \tag{4-1}$$

$$\sum_{i=1}^{c} u_{ij} = 1, j = 1,2 \ldots, n \tag{4-2}$$

采用拉格朗日乘数法将约束条件（4-2）放到目标函数（4-1）中，并把式（4-2）的所有 j 展开，那么公式（4-1）变成公式（4-3）所示：

$$J = \sum_{i=1}^{c}\sum_{j=1}^{n} u_{ij}^{m}\|x_j - c_i\|^2 + \lambda_1\left(\sum_{i=1}^{c} u_{i1} - 1\right) + \ldots + \lambda_j\left(\sum_{i=1}^{c} u_{ij} - 1\right) + \ldots + \lambda_n\left(\sum_{i=n}^{c} u_{in} - 1\right) \tag{4-3}$$

J 对u_{ij}的求导结果并让其等于 0 得到如公式（4-4）所示：

$$\frac{\partial J}{\partial u_{ij}} = m||x_j - c_i||^2 u_{ij}^{m-1} + \lambda_j = 0 \tag{4-4}$$

将公式（4-4）简化，解出u_{ij}得到公式（4-5）：

$$u_{ij}^{m-1} = \frac{-\lambda_j}{m||x_j - c_i||^2} \tag{4-5}$$

进一步得到公式（4-6）：

$$u_{ij} = \left(\frac{-\lambda_j}{m||x_j - c_i||^2}\right)^{\frac{1}{m-1}} = \left(\frac{-\lambda_j}{m}\right)^{\frac{1}{m-1}} \left(\frac{1}{||x_j - c_i||^{\left(\frac{2}{m-1}\right)}}\right) \tag{4-6}$$

重新使用公式（4-2）的约束条件，并把算出来的u_{ij}代入式（4-2）中,再带入公式（4-6）并对c_i求导化简得到类中心的迭代公式如（4-7）所示：

$$c_i = \frac{\sum_{j=1}^{n}\left(x_j u_{ij}^m\right)}{\sum_{j=1}^{n} u_{ij}^m} \tag{4-7}$$

FCM 聚类算法流程如下：

1、确定分类数，指数 m 的值，确定迭代次数；

2、初始化一个隶属度 U；

3、根据 U 计算聚类中心 C；

4、计算目标函数 J；

5、根据 C 返回去计算 U，回到步骤 3，一直循环直到结束。

（2）基于神经网络聚类的 SOM 聚类算法,

自组织映射神经网络（Self Organizing Maps）算法简称 SOM 算法，该算法是一种无监督的学习聚类算法。SOM 算法结构相对简单，类似于一种只有输入层和隐藏层的神经网络的结构。

隐藏层中的不同节点代表不同的类，每个输入的数据训练过程中采用“竞争学习”的方式，在隐藏层中找到一个和它最为匹配的节点，称之为“激活节点”，之后采用梯度下降的方法更新激活节点的参数。同时，激活节点附近的节点根据它们之间的距离也适当的更新参数。

SOM算法模型中隐藏层的节点具有拓补关系,这个拓补关系需要人为的确定。根据需要的拓补关系模型（一维或二维），建立隐藏层的模型（节点之前连成一条线或形成一个平面）。

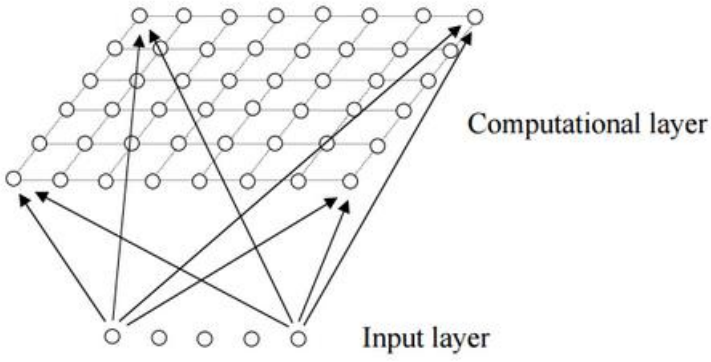


图 4-1 SOM 聚类算法神经网络示意图

Fig 4-1 Illustration of neural network for SOM clustering algorithm

根据隐藏层节点的拓扑关系，SOM 算法可以把任意维度的输入离散化到一维或者二维的离散空间上。Computation layer 中里面的节点与 Input layer 的节点是全连接的。

SOM 聚类算法流程如下：

1、确定隐藏层节点拓扑关系。

2、初始化：每个节点随机初始化自己的参数。每个节点的参数个数与 Input 的维度相同。

3、对于每一个输入数据，找到与它最相配的节点。假设输入时 D 维的， 即 X={x_i, i=1,...,D}，那么判别函数可以为欧几里得距离如公式（4-8）所示：

$$d_j(\mathbf{x}) = \sum_{i=1}^{D}(x_i - w_{ji})^2 \tag{4-8}$$

4、找到激活节点 I(x)之后，令 S_ij 表示节点 i 和 j 之间的距离，对于 I(x)临近的节点，分配给它们一个更新权重如公式（4-9）所示：

$$T_{j,I(\mathbf{x})} = \exp(-S_{j,I(\mathbf{x})}^2 / 2\sigma^2) \tag{4-9}$$

5、按照梯度下降法如公式（4-10）所示更新节点参数直到收敛：

$$\Delta w_{ji} = \eta(t) \cdot T_{j,I(\mathbf{x})}(t) \cdot (x_i - w_{ji}) \tag{4-10}$$

（3）基于层次方法的凝聚型层次聚类算法

层次聚类算法主要分为凝聚型层次聚类和分裂型层次聚类，分别对应层次的分解顺序为自底向上和自顶向下。凝聚型层次聚类在层次聚类算法中应用最为广泛。

凝聚型层次聚类的思想是将每个输入数据都作为一个类的中心，在此基础上，计算类间的距离，根据距离最小合并原则，合并相邻的原子类作为新的类，直到满足终结条件或所有对象都属于一个类。距离计算中，常用的四种类间距离度量方法如公式（4-11）-（4-14）所示：

最小距离 $$d_{min}(c_i, c_j) = min_{p \in c_i p' \in c_j} |p - p'| \tag{4-11}$$

最大距离 $$d_{max}(c_i, c_j) = max_{p \in c_i p' \in c_j} |p - p'| \tag{4-12}$$

平均值距离 $$d_{mean}(c_i, c_j) = |m_i - m_j| \tag{4-13}$$

平均距离 $$d_{arg}(c_i, c_j) = \frac{1}{n_i n_j} \sum_{p \in c_i} \sum_{p' \in c_j} |p - p'| \tag{4-14}$$

凝聚层次聚类算法流程如下：

1、将每个输入数据看作一类，计算每个类之间的最小距离；

2、将距离最小的两个类合并成一个新类；

3、重新计算新类与所有类之间的距离；

4、重复步骤 2,3 直到满足终结条件

（4）基于划分方法的 k-means 聚类算法

k-means 算法以 k 为参数，目标是把所有输入数据分成 k 个类，使类内的数据具有较高的相似度，而类间的具有较低相似度。k-means 算法的原理是：首先随机地选择 k 个数据，每个数据代表 k 个类的中心点，对其他的每个数据点计算到这 k 个点的距离，根据其与 k 个类中心的距离大小，将其赋给最近的类，然后重新计算每个类的类中心。以上为 k-means 算法一次迭代过程，不断重复此迭代过程，直到准则函数收敛或类中心不再发生变化。

通常使用采用平方误差和（SSE）准则作为 k-means 算法的准则函数来判断聚类的效果。其定义如公式（4-15）所示：

$$\mathrm{E}=\sum_{i=1}^{k}\sum_{p\in C_i}|p-m_i|^2 \tag{4-15}$$

其中：E 是所有数据点的平方误差的总和，p 是输入的数据点，m_i 是类 C_i 的平均值。该目标函数使生成的类尽可能紧凑独立，即类内的数据相似度较高。

k-means 聚类算法的流程如下：

1、选择 k 个点作为初始的 k 个类中心；

2、根据类中各个数据到类心的平均值，将每个数据重新分配到相似度高的类，常用的距离度量如公式（4-16）-（4-18）所示：

闵可夫斯基距离
$$d_{ij}=\sqrt[\lambda]{\sum_{k=1}^{n}|x_{ik}-x_{jk}|^{\lambda}} \tag{4-16}$$

欧氏距离
$$d_{ij}=\sqrt{\sum_{k=1}^{n}(x_{ik}-x_{jk})^2} \tag{4-17}$$

曼哈顿距离
$$d_{ij}=\sum_{k=1}^{n}|x_{ik}-x_{jk}| \tag{4-18}$$

3、更新类的平均值，即计算每个类中各点平均值作为新的类中心；

4、重读步骤 2、3 直到准则函数收敛或类中心不再发生变化。

4.2.2 聚类算法性能对比实验

在上一节介绍的四种聚类算法中，每种算法机理不同，因而其聚类性能亦有所差异。如 k-means 聚类算法的初始点是随机选取的，因而受数据初始化影响大，存在聚类结果的不稳定性问题；层次聚类具有不需要提前给定分类数的优点，但

是一旦合并被执行，就不能进行修正，聚类质量无法保证；FCM 算法对初始聚类中心敏感，需要人为确定聚类数，容易陷入局部最优解；SOM 算法使用神经网络，在模型上有很强的理论支持，但是处理时间较长，对于数据量大的应用支持度不够好。

为进一步比较分析 4 种聚类算法的性能差异，本节进行对比实验。选择专门用于测试聚类算法的国际通用的 UCI 数据库中的 IRIS 数据集[49] 作为基准库，进行测试分析，根据 4 种算法的聚类性能的对比分析，确定具有良好聚类性能的算法，作为本系统数据挖掘模块的聚类算法。

IRIS 数据集包含三类样本，分别为三种不同的莺尾属植物的花朵样本，总共包含 150 个样本数据。其中每个样本数据包含 4 个属性，即花萼长度、花萼宽度、花瓣长度和花瓣宽度。本实验在该数据集上分别采用上述 4 类算法进行聚类分析。考虑到 k-means 算法存在初始点的不确定性问题，本实验多次执行 k-means 算法，对相应结果取平均值作为最终结果。

表 4-1 四种聚类算法性能比较

Table4-1 Performance comparison among four clustering algorithms

聚类 算法	总错误 样本数	运行 时间/s	平均 准确率/%
FCM	17	0.471	89
SOM	25	5.267	84
凝聚层次聚类	49	0.129	67
k-means	18	0.146	89

本实验采用三种评价指标对于四种聚类算法进行比较与评估。具体评价指标定义如下。

(1)总错误样本数：所有类中聚错样本数的总和。

(2)运行时间：聚类算法从开始到结束的总时间。

(3)平均准确率：所有类中每个类的准确率的平均值如公式（4-19）所示，其中 k 为聚类数，：

$$Accuracy_{avg} = \frac{1}{k}\sum_{1}^{k}\frac{\text{类内总样本数} - \text{类内聚错样本数}}{\text{类内总样本数}} \tag{4-19}$$

如表 4-1 所示，实验结果再次验证了四种算法的性能差异。即 SOM 算法具有更高的预测准确率，但计算时间相对更长；凝聚层次聚类方法算法简单，因而计算复杂度低计算效率高；FCM 和 k-means 方法具有综合准确率和运行时间两者性能的优势。考虑到在实际的商业应用中，时效性因素更重要，同时综合运行时间及准确度两方面因素，k-means 在保证较高的准确率同时，比具有几乎相同准确率的 FCM 算法运行时间快 3 倍。

综上所述，本系统选择了 k-means 算法作为聚类分析模块的核心算法。

4.2.3 K-means 算法参数确定

k-means 算法是一种无监督的算法，适合无对无标定训练样本的数据聚类特性进行学习，适用于本类系统的基于历史就诊数据进行患者病情进行聚合分析，提取共性特性。k-means 算法中 k 值的选取决定了算法的准确性和有效性。

为了确定 k 值的选取，本文采用图谱分析进行了定性的分析，分析随着 k 的变化，聚类效果呈现的变化，确定使用类内半径（类内所有点到中心的最大值）和类内平均质心距离（类内所有点到中心的距离的平均值）的平均值为 k 值选取的依据。

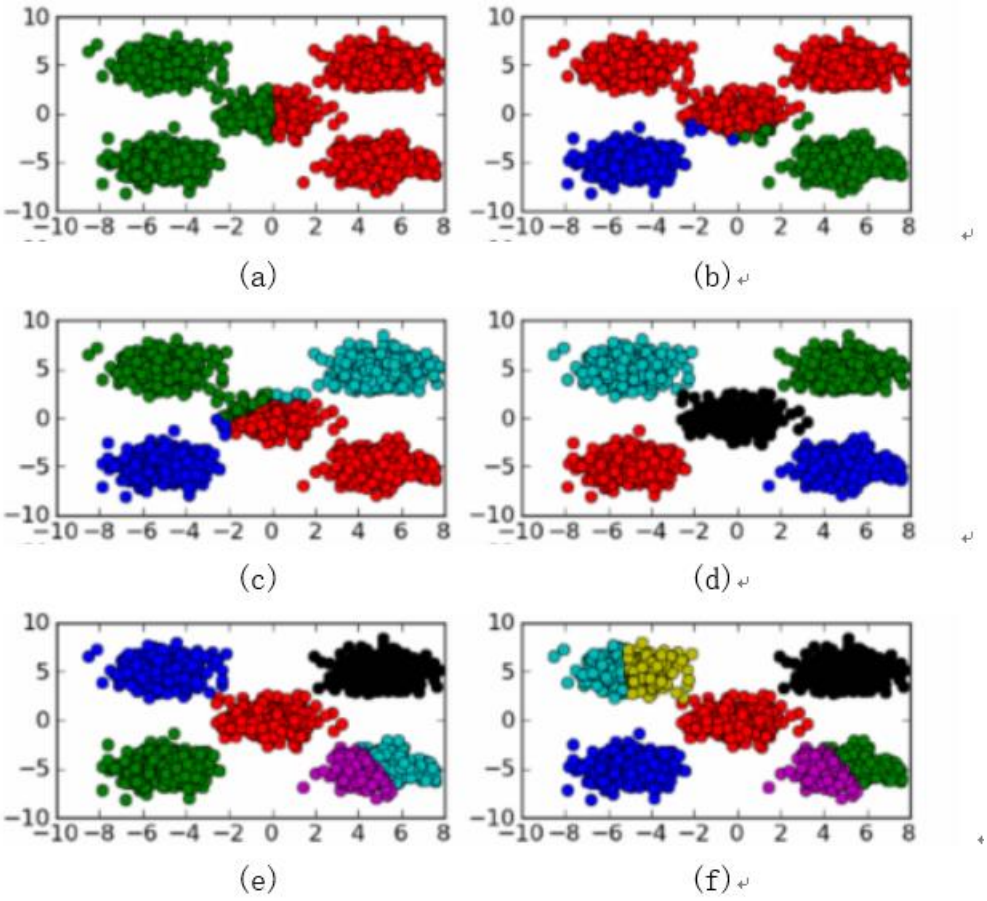


图 4-2 随 K 值变化的 k-means 聚类效果示意图

Fig 4-2 Illustration of K-means clustering algorithms effect with different K-value

如图 4-2 所示，其中子图(a)到(f)分别为 k 取值为从 2 依次增加到 7 的聚类效果示意图。从图中数据点的分布可以看出，原始数据为 5 类。子图(a)(b)(c)为 k 值小于 5 的聚类效果示意图，可以看出每个类中包含了很多空白的区域，即：类内数据相似性降低，此时，随着类数 k 的减少，类半径和平均质心距离成增加趋势。同样，子图(e)(f)为 k 值大于 5 的聚类效果示意图，可以看出，其中有一些聚集在一起的点又被划分开了，即类间数据的相似性变高，此时，随着类

数 k 的增加，类半径和平均之心距离将会缓慢减小。综上所述，当 k 值增加，类半径和平均质心距离将会缓慢减小，这时，聚类结果类内相似性增加，但当 k 值增加到一定值后，类间数据相似度亦增高，即类间距离变小。因而，随 k 值变化的类内半径的平均值的变化所绘制出折线图，该折线的拐点为适合的 k 的取值，即类内相似度高，类间聚类相似度低。

本系统根据两个业务模块的数据，取不同 k 值使用 k-means 算法进行聚类，计算各个 k 值的类内半径和平均质心距离结果如表 4-2，4-3 所示：

表 4-2 诊疗方案数据聚类结果

Table 4-2 Treatment program data clustering results

不同 K 取值	类内 半径	类内平均 质心距离
2	7.132	4.035
3	5.763	2.967
4	4.207	1.792
5	3.072	1.067
6	2.756	0.824
7	2.481	0.594
8	2.093	0.401

表 4-3 呼吸机设置数据聚类结果

Table 4-3 Ventilator set data clustering results

不同 K 取值	类内 半径	类内平均 质心距离
2	8.238	4.563
3	7.017	3.484
4	5.517	2.391
5	3.874	1.319
6	3.503	1.195
7	3.136	1.103
8	2.802	1.089

通过表 4-2 和 4-3 中数据可以看出，类内半径和类内平均质心距离变化趋势基本相同，使用表 4-2 和 4-3 中类内半径绘制折线图如图 4-3 所示：

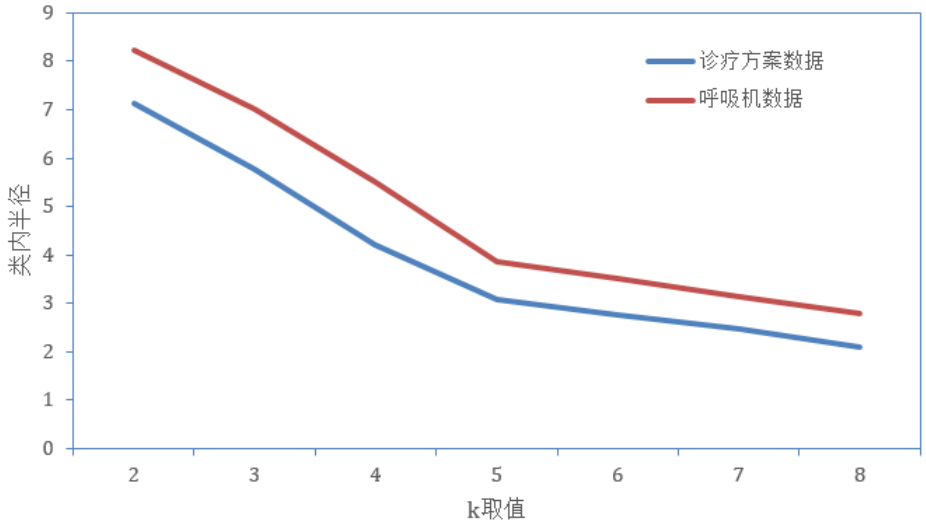


图 4-3 聚类半径趋势图

Fig 4-3 Clustering radius trend graph

根据图 4-3 中两种业务数据的聚类半径随 k 值的变化趋势，可以看出，两条折线的拐点都为 k 取 5 时，即 k 取 5 时，获得的聚类结果，具有良好的聚类性能。

4.2.4 诊疗方案推荐模块设计

本系统的诊疗方案推荐模块主要是基于聚类算法实现用户画像，在此基础上，为病人提供有针对性的诊疗建议。具体方法，对不同病人的生理及病理指标数据，通过 k-means 算法进行聚类，对病人数据进行建模，得到几种典型的用户类型，实现对病人画像。系统可以从历史数据得到典型用户类型的诊疗方案，也可以将这些典型的数据模型提供给医生，医生根据典型模型的各项指标提供相应的诊疗方案反馈给系统。用户在前端输入相关的指标信息，系统获取到不同病人当前的指标数据，通过之前训练好的聚类模型对数据进行预测，将结果进行解析和格式化后反馈给病人所属类别的最佳诊疗方案。诊疗方案推荐功能的流程图如图 4-4 所示：

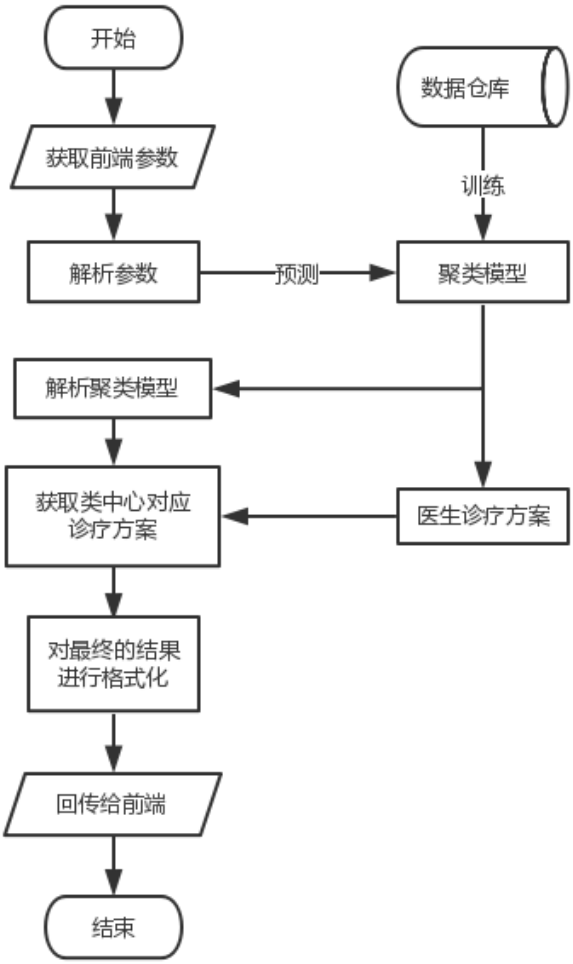


图 4-4 诊疗方案推荐流程图

Fig 4-4 Flow chart of treatment program recommended

4.2.5 呼吸机设置推荐模块设计

对于多数呼吸疾病患者，开始使用呼吸机时不知道如何进行参数设置，医生亦需要根据用户个体生理和病情状况，进行判断，确定呼吸机相关的设置。因而本文提出根据历史经验，进行聚类分析，提供综合病人个体特点的自动参数设置推荐。针对该目标，本系统完成了呼吸机设置推荐模块，主要是根据其当前的病人指标情况为其提供适合的呼吸机设置推荐。具体使用聚类算法，对病人指标和对应的呼吸机设置参数进行聚类分析，建立典型的模型。新的用户可以在前端输入自己当前的各项指标，当系统获取信息，建立对应的多维向量，使用之前训练好的模型进行预测，将得到的呼吸机配置信息返回给用户。呼吸机设置推荐模块流程图如图 4-5 所示：

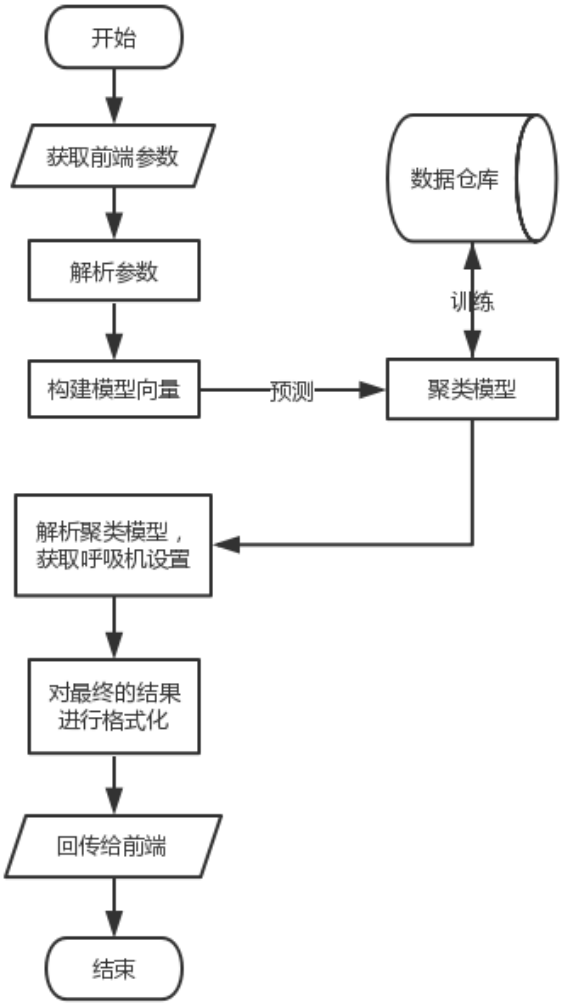


图 4-5 呼吸机设置推荐流程图

Fig 4-5 Recommended flow chart of ventilator setting

4.3 数据挖掘层关联分析模块

对于医生或医疗管理决策者来说，不仅关注病人和疾病本身的变化规律，更关心疾病的诱因、疾病的潜在症状以及疾病间是否有关联关系。这对于医生的诊疗工作具有更重要的意义。因而，本文探讨了如何在大数据医疗分析系统中，实现数据挖掘层关联分析模块的技术方案。采用关联规则算法对病人的生理指标和病理指标数据进行挖掘，得到各项指标的关联关系，为医生的诊疗提供辅助决策。在此基础上，结合了医生的诊疗经验作为先验知识，对关联规则学习算法进行了改进，获得了更加符合实际经验的挖掘结果。

4.3.1 关联规则算法概述

Apriori 算法是关联规则算法中最经典的也是应用最广泛的算法[45]。Apriori 算法的核心思想是挖掘事物频繁项集合，找出其中的强关联关系。该算法是向下封闭的，即如果一个项目集是频繁集，那么它的所有子集都是频繁集。一条关联规则的形式记为 X→Y，表示由 X 可以关联出 Y。下面介绍一下 Apriori 算法的相关 概念：

资料库（Database）—— 所有项目的集合，用 D 表示。

记录（Transaction）—— 数据库中的每条数据，用 T 表示。

项集（Itemset）—— 包含项目的集合，用 k-itemset 表示，其中 k 为项集中项目的个数。

支持度（Support）—— 用 Supp（X）表示如公式（4-20）所示，表示所有记录中 X 出现的概率。

$$Supp(X) = \frac{Count(X)}{Count(D)} = P(X) \tag{4-20}$$

置信度（Confidence）—— 用 conf（X→Y）表示如公式（4-21）所示，表示所有记录中出现 X 又同时出现 Y 的概率。

$$Conf(X \to Y) = \frac{Supp(X \cup Y)}{Supp(X)} = P(Y|X) \tag{4-21}$$

频繁集（Frequent itemset）—— 支持度大于给定的最小支持度的项集。

剪枝步（Pruning step）—— 根据 Apriori 算法向下封闭原则进行筛选。

Apriori 算法的步骤（k 从 1 开始）：

（1）列出所有的 k-itemset，执行剪枝步。

（2）计算 k-itemset 的支持度，得到 k-频繁项目集。

（3）判断 k 是否为最大或 k-频繁项目集个数<2，如果不是，k+1，重复执行步骤（1）-（3）。

（4）计算所有频繁项目集的子集之间的置信度，得到强关联规则。

以上为 Apriori 算法挖掘关联规则的整体步骤和思想，利用支持度和置信度的计算结果为依据，挖掘出项目之间的关联关系，但是该算法每次获取频繁集都需要扫描整个资料库，时间复杂度很高，且在这过程中会产生大量的候选频繁项目集，也使算法的空间复杂度很高。当资料库数据量很大，时间和空间都将开销巨大，因而该算法仍存在一定的缺陷。

4.3.2 改进的加权 FP-tree 关联规则算法

针对 Apriori 算法在大数据集时效性上的问题，产生了 FP-growth 算法[51]。下面介绍 FP-growth 算法的一些重要概念：

FP-tree —— 描述资料库中所有事物规则的树，其中根节点为 null，每个子节点为资料库中的项目并记录相应的支持度，FP-tree 的每一条路径代表资料库中每一条事物。

条件模式基 —— FP-tree 中包含某一频繁项的所有祖先路径的集合，即以某一频繁项作为后缀的前缀路径的集合。

条件 FP-tree —— 由条件模式基按照 FP-tree 原则形成的树，作为挖掘频繁项目集的目标树。

该算法在频繁项目集挖掘过程中，只需整体扫描资料库一次，将每一个项目按照支持度排序得到集合 L，将资料库中每一条事物，按照 L 的顺序插入到根节点为 null 的树种，并更新每个节点的支持度信息，构建 FP-tree 的过程如图 4-6 所示：

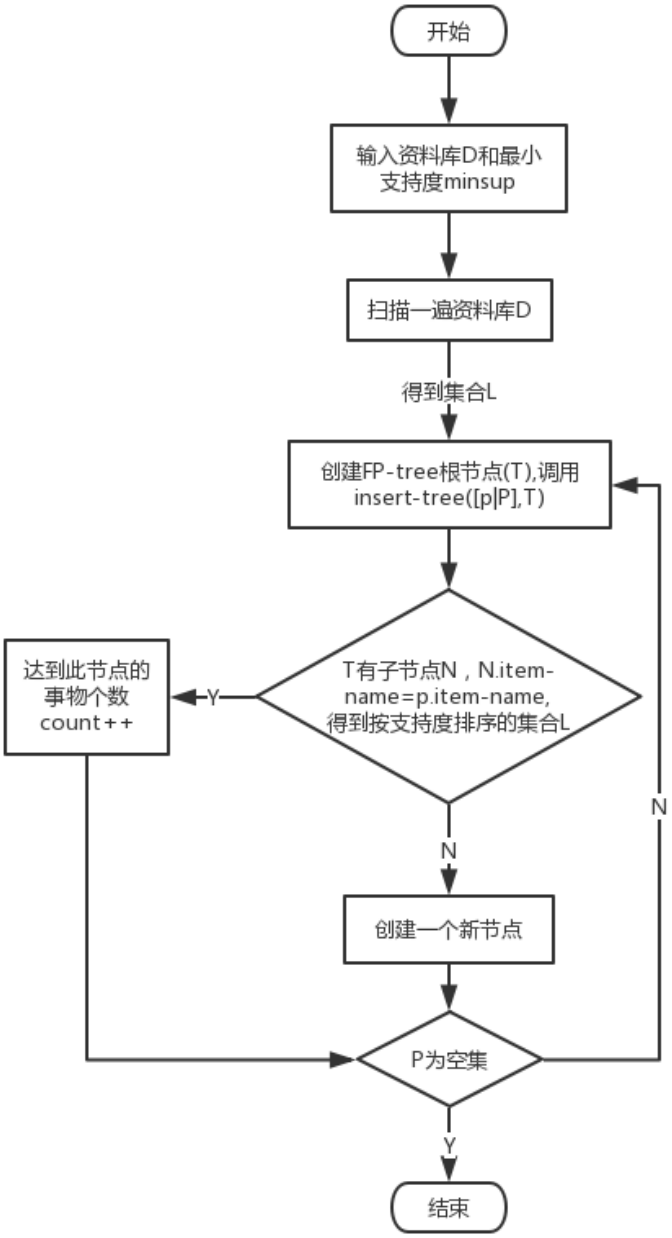


图 4-6 FP-tree 构建流程图

Fig 4-6 Flow chart of FP-tree building

为了挖掘频繁项目集，以 1-频繁集中的各个项目为后缀，从原 FP-Tree 中获得条件模式基，根据每个条件模式基构建条件 FP-tree，再对每个条件 FP-tree 进行挖掘，直到某个条件 FP-tree 只包含一个元素为止，即找到所有的频繁项目集。

FP-growth 算法利用树形结构对事物数据进行存储，利用分治法和迭代法的思想对 FP-tree 进行挖掘，在空间和时效性方面都比 Apriori 算法有了显著的提高。虽然 FP-growth 在空间和时效性方面一定程度的解决了 Apriori 的问题，但

是仍有不足。因为这两种关联规则算法都只是根据资料库中事物项目出现的次数和频率进行关联挖掘，没有考虑每个事物项目的实际意义，缺乏对关联挖掘的具体影响的描述。

基于以上问题，本节提出一种改进的 FP-growth 算法，根据医生经验和相关医疗信息作为先验知识，将资料库中的每种事物项赋予权值，这里记为 W(k)，描述其在关联挖掘中的重要程度，在使用事物构建 FP-tree 时，结点保存事务项原有的支持度，而每条连接到该节点的路径则记录该事务项的权值，这样每条表示事物的路径就变成了带权路径，而带权支持度的计算如公式（4-22）所示：

$$Sup(X) = \frac{Count(X) * W(X)}{Count(D)} \tag{4-22}$$

而权重 W 的计算对于节点数为 1 的路径来说即是该节点的权值，对于多节点路径来说，如何合理的进行权值简化是一个重要的问题。常用的简化方式有 3 种::

取最大值 —— 即取该路径中所有节点权值最大的作为该路径的权值，如公式（4-23）所示：

$$W = Max\big(W(k)\big), k \in X, Y, Z \tag{4-23}$$

取平均值 —— 即计算该路径中所有节点权值的平均值作为该路径的权值，如公式（4-24）所示：

$$W = \frac{\sum_{i=1}^{k} W_i}{k} \tag{4-24}$$

归一化[52] —— 即计算该路径中所有节点的权值占总权值的比例之和，如公式（4-25）所示：

$$W_i' = \frac{W_i}{\sum_{p=1}^{k} W_p}, \quad W = \frac{\sum_{i=1}^{k} W'_i}{k} \tag{4-25}$$

以上三种方法为常用的简化方式，取最大值的方法过于重视了权值较高的节点，而缺乏对路径的整体考虑。取平均值的方法和归一化的防范虽然可以整体考虑每个节点对于整体路径的贡献，但是归一化之后的权值总小于 1，不符合实际意义。而取平均值的方法，无法满足关联规则的向下封闭，即频繁集的子集一定也是频繁集这一原则。

为了解决上述问题，本算法采用如公式（4-26）的方式简化权值，这样既考虑了每个节点对整体路径权值的贡献，简化结果也符合实际意义，并且满足关联规则的向下封闭原则。

$$W = \frac{\prod_{i=1}^{k} W_i}{\sum_{i=1}^{k} W_i} \tag{4-26}$$

改进算法如下：

```
1.  //构建 FP-tree
2.  创建根节点 T，调用 insert(p∈P，T)方法；
3.  while (P 不为空) do
4.    if (T 含有节点 p) then
5.      节点的支持度 count++；
6.    else
7.      创建新节点，记录支持度和权值；
8.    获得按支持度排序的集合 L；
9.  end
10. //挖掘频繁集
11. while (L 不为空) do
12.   找到 l∈L 的所有祖先路径构成条件 FP-tree$T_0$；
13.   while path∈$T_0$ do
14.     计算 path 的权值 w；
15.     if w>$sup_{min}$ then
16.       path 路径的节点组成频繁集；
17.   end
18.   删除 l 节点；
19. end
```

本节将分别使用原 FP-growth 算法和本文的加权关联规则算法对病人数据的进行关联挖掘，结果如图 4-7 所示：

传统 FP-growth 算法挖掘结果：

1. 体重超重—>心率快
2. 身高较高—>心率快
3. 中年,体重超重—>心率快
4. 体重超重—>心率快,BPM 过快
5. 身高正常—>BPM 过快
6. 老年—>BPM 过快
7. 老年—>血氧过低

本文改良算法挖掘结果：

1. 体重超重—>心率快
2. 身高正常,体重超重—>心率快
3. 中年,体重超重—>心率快
4. 体重超重—>心率快,BPM 过快
5. 老年—>BPM 过快
6. 体重超重—>BPM 过快
7. 老年—>血氧过低
8. 体重正常—>血氧正常

图 4-7 关联规则挖掘结果比较

Fig 4-7 Comparison of association rule learning results

通过图 4-7 种结果可以看出，原 FP-gorwth 算法与本文提出的改良算法相比较，身高这项指标单独得到的关联规则偏多，而体重指标得到的关联规则偏少。这与实际情况相差较大，医学普遍认为身高与心率和 BPM（每分钟呼吸次数）等指标关联不大。而在本文提出的改进算法的挖掘结果中，身高这项指标由于初始的权重相对偏小，只作为关联的一部分出现在了结果中，而体重这项指标得到的关联规则更加丰富，这也更加符合现有的经验，得到的结果更加准确。

4.3.3 病情关联分析模块设计

本模块的功能主要是对病人各项指标进行关联挖掘，发现不同指标之间的关联关系。首先对病人的各项指标包括生理指标（年龄、体重、BMI 等）和病理指标（心率、血氧）进行数据的预处理，进行标定，如年龄在 25 岁以下为少年，25-40 岁为青年 40-60 岁为中年，60 岁以上为老年；心率在 60 以下为心率偏慢，60-100 位心率正常，100-120 以上为心率偏快，120 以上为心率过快等。并且根据医生经验和相关的医疗知识，对各项生理指标赋予权值，如（身高，0.9），（体重 1.1），（年龄，1.2）等。将预处理后的数据使用本系统改进的加权关联规则算

法进行关联挖掘，得到关联规则集，并记录相应的置信度。

医生或医疗决策者在前端进行操作，可以选择查看整个的关联规则集，或根据不同指标查看与该指标相关的关联规则，结果按置信度从高到低排列。病情关联分析流程图如图 4-8 所示：

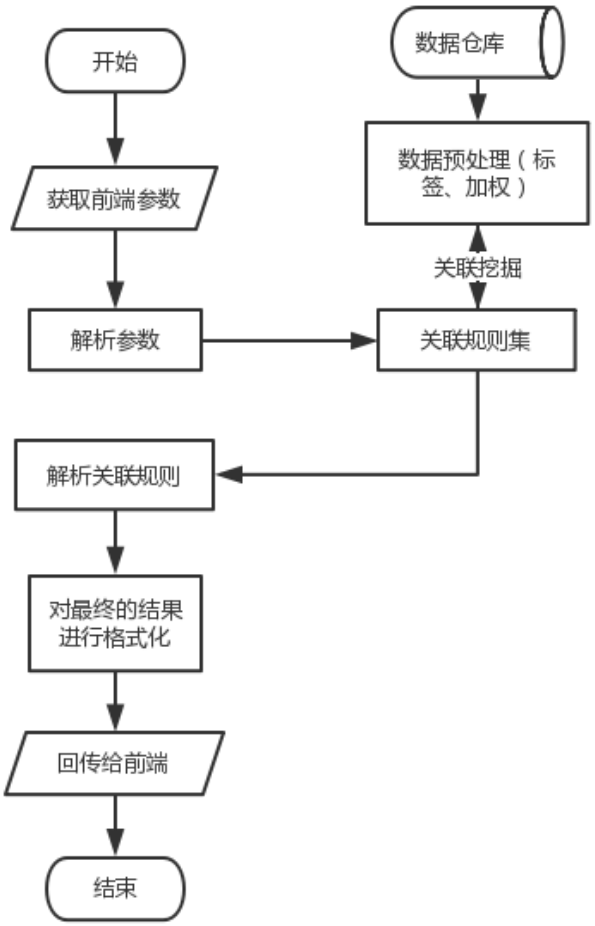


图 4-8 病情关联分析流程图

Fig 4-8 Disease association analysis flow chart

4.4 本章小结

本章主要介绍了本系统的数据挖掘层的两大模块和三大业务。两大模块分别为聚类分析模块和关联分析模块，其中在聚类分析模块中包含两大业务，即诊疗方案推荐和呼吸机设置推荐，在关联分析模块中包含病情关联分析业务。

在聚类分析模块介绍中，首先选择了应用较为广泛的几种常用的聚类算法进行了介绍，并且使用公用的测试集对这些算法进行了比较实验，结合本系统的数据特点对实验结果进行了合理的分析，最终选择了 k-means 算法作为本系统的核

心算法。之后根据本系统聚类分析模块的两大业务，对 k-means 算法的参数进行了分析和实验，找到了适合的 k 取值。后面又具体介绍了两大聚类模块业务，诊疗方案推荐是针对病情变化较快而又无法及时得到医生建议的病人，本业务通过对所有病人聚类画像，总结出几种典型的模型，对病人进行诊疗方案推荐。而呼吸机设置推荐主要是针对最初使用呼吸机的病人，他们不了解那些配置更适合自己当前的病情，本业务通过对现有的病人状况和对应的呼吸机设置进行聚类，挖掘出一些典型的模型，通过判断病人当前的病情为其推荐适当的设置。

在关联分析模块介绍中，首先介绍了关联规则最经典的 Apriori 算法以及其改进算法 FP-growth 算法，结合本系统的目标业务和实际情况，对 FP-growth 算法提出了改进，通过医生经验和相关医疗知识对路径进行加权，使得算法更加符合实际意义。后面介绍了病情关联分析业务，本业务针对的群体是医生和医疗决策者，先对原始数据进行标定和加权，之后通过本文提出的算法对原始数据进行关联挖掘，找到不同指标之间的关联规则，为其提供不同病情之间潜在的关系。

第 5 章　睡眠呼吸病情分析决策系统核心模块实现

5.1 引言

本章主要介绍本系统分析层相关业务模块的具体实现，主要包括如下几个部分：

统计分析层 —— 前后端接口设计、OLAP 分析的核心算法模块和可视化展示模块。

数据挖掘层 —— 前后端接口设计、聚类算法模块和关联挖掘算法模块。

5.2 统计分析层实现

统计分析层包括前后端接口设计、核心算法的设计和可视化展示。

5.2.1 前后端接口设计

基于统计分析层的三大核心业务，即个体、医生群体和决策者群体的分析，本模块主要实现 OLAP 的多维模型查询、OLAP 上卷及下钻等功能，主要流程为前端获取用户参数，后端获取参数进行统计分析，将结果回传给前端展示。其中前端获取的主要参数为分析的起始时间、结束时间、时间粒度、不同身份的 ID 及相关参数，而后端回传的主要参数为格式化之后的查询结果、以及前端传来的参数，本模块的接口设计如下：

（1）后端返回约定

返回结果中的 code 表示返回码，1 为成功返回数据，前端继续获取 data 和 parameters 字段数据，非 1 为返回数据失败，当大于 1000 时，前端打印错误信息给用户。

返回结果中的 message 为返回信息，记录成功或者错误信息

返回结果中的 data 和 parameters 字段为返回的数据和参数，当 code 为 1 时，这两个字段有值，如：

```
{
  "code": "1",
  "message": “成功“
  “data":{…}
  "parameters”:{…}
```

}

（2）接口定义

说明	统计分析层及 OLAP 接口			
请求方式	HTTP POST			
请求字段描述				
参数	说明	类型	是否可为空	默认值
start_time	起始时间	字符	不可空	
end_time	结束时间	字符	不可空	
granularity	粒度	字符	不可空	
user_type	用户类型	字符	不可空	
content	前端获取内容	字符	不可空	
operate	操作类别	数值	不可空	

参数示例	{
	"select_type":"button",
	"start_time":"2016-02-27T12:12:12.000Z",
	"end_time":"2016-03-31T14:00:01.000Z",
	"type":"area",
	"type_string":"北京",
	"granularity":"month"
	}			

5.5.2 核心模块代码实现

本业务层的核心业务是对数据进行 OLAP 分析，基于以上的接口设计，该模块核心代码如图 5-1 所示：

统计分析层核心算法：

```
1.  class OnlineAnalytical {
2.      String startDate,endDate;
3.      String granularity;
4.      String userType,content;
5.      int operate;
6.      private JSONObject queryJson(){…};
7.      private JSONObject drillUp(){…};
8.      private JSONObject drillDown(){…};
9.      public String resultFormat(){…};
10. }
```

图 5-1 统计分析模块核心代码

Fig 5-1 Statistical analysis module core code

图 5-1 为统计分析类，其中 startDate 和 endDate 两个成员变量表示开始日期和结束日期，代表用户选择的时间范围。granularity 代表用户在时间维度上选择的粒度，包括时、日、月等。userType 变量表示用户类别，因为本业务层分为个人 OLAP、医生群体 OLAP 和决策者群体 OLAP，所以需要区分用户类别。content 变量则表示前端获取的用户信息，主要为用户 id，医生群体 OLAP 的 content 会包括医生所选的病人 id 集合。operate 变量表示用户执行的查询操作，普通查询为 0，上卷为 1，下钻为 2，按地域查询为 3。

在核心方法中，queryJson()方法为连接数据仓库 Druid 进行查询的主方法，该方法解析 userType 和 operate 参数得到查询类型，将 startDate、endDate、granularity、content 四个参数传给 Druid 进行查询，接收 Druid 的查询结果，因为 Druid 查询结果为 json 格式，所以该方法返回 JSONObject 类型的对象。drillUp()方法为 OLAP 分析的向上钻取主方法，该方法重新给 granularity 参数赋予一个更大粒度的值，调用 queryJson 方法进行查询，返回 JSONObject 类型的对象。drillDown()方法与 drillUp()方法类似，为 OLAP 分析的向下钻取主方法，该方法重新给 granularity 参数赋予一个更小粒度的值，调用 queryJson 方法进行查询，返回 JSONObject 类型的对象。resultFormat()方法为最终格式化的主方法，解析 Druid 的查询结果，将结果重构成前端 Echart 可视化格式的 Json 字符串格式。

5.5.3 可视化展示

图 5-2 个体 OLAP 分析页面

Fig 5-2 Individual OLAP analysis page

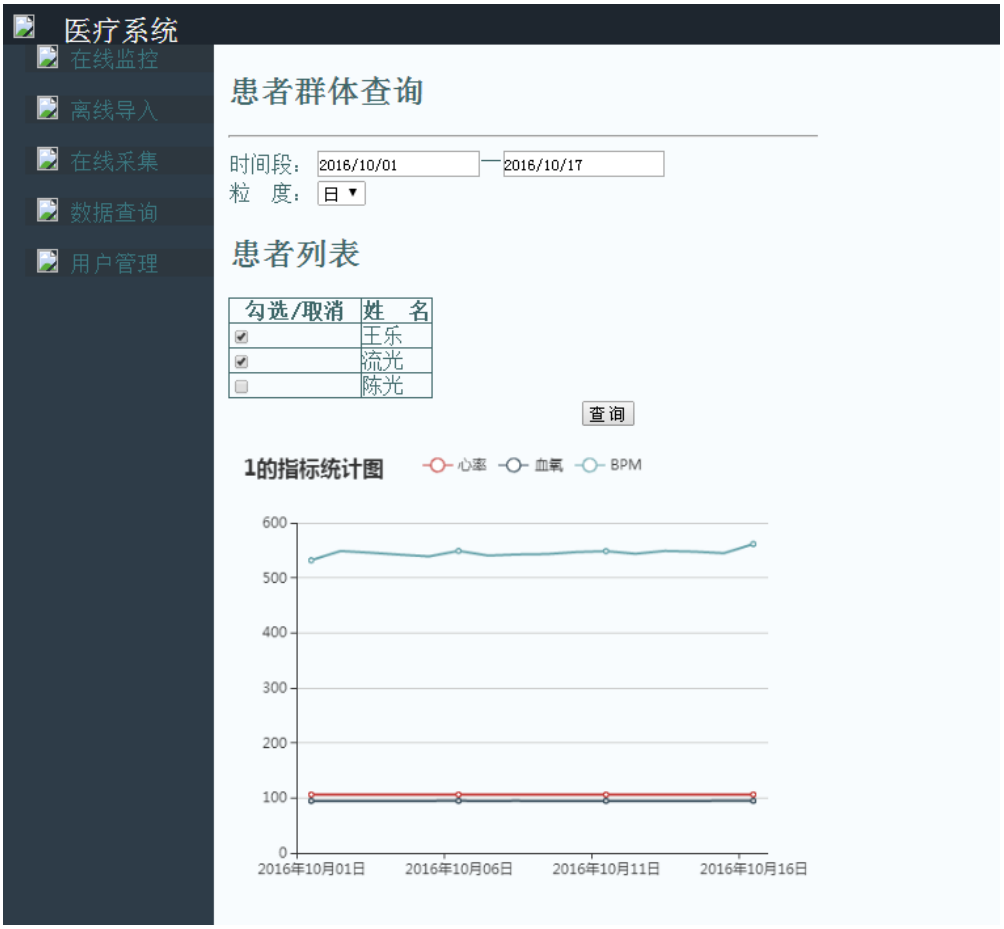


图 5-3 医生群体 OLAP 分析页面

Fig 5-3 Doctor group OLAP analysis page

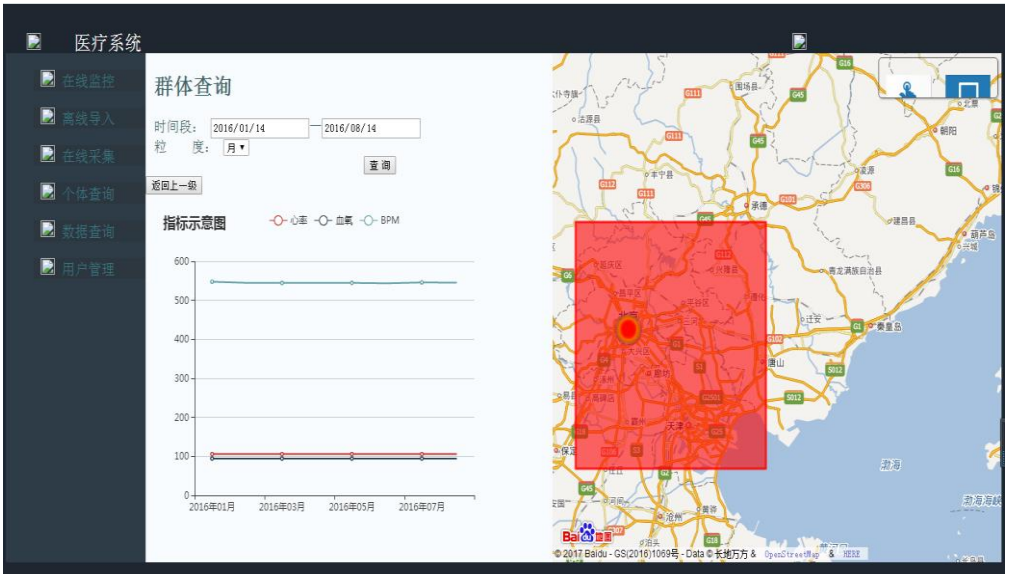


图 5-4 决策者群体 OLAP 分析页面

Fig 5-4 Decision maker group OLAP analysis page

5.3 数据挖掘层实现

本节主要介绍数据挖掘层的具体实现方法，包括前后端接口设计、聚类分析模块的核心算法设计和关联分析模块的核心算法设计。

5.3.1 前后端接口设计

基于数据挖掘层的聚类分析模块和关联分析模块的核心业务，本模块的设计流程主要为前端获取用户参数，后端获取参数进行相关算法的模型预测，将结果回传给前端展示。其中前端获取的主要参数为用户的数据参数、调用的算法类别，后端回传的主要参数为数据挖掘算法预测的结果，由于不需要进行二次分析，所以后端回传的参数中不包括前端获取到的参数。本模块的接口设计如下：

（1）约定

返回结果中的 code 表示返回码，1 为成功返回数据，前端继续获取 data 字段数据，非 1 为返回数据失败，当大于 1000 时，前端打印错误信息给用户。

返回结果中的 message 为返回信息，记录成功或者错误信息。

返回结果中的 data 字段为返回的挖掘结果，当 code 为 1 时，该字段有值，如：

```
{
  "code": "1",
  "message": “成功“
  “data":{...}
}
```

（2）接口定义

说明	数据挖掘层接口			
请求方式	HTTP POST			
请求字段描述				
参数	说明	类型	是否可为空	默认值
algo_type	调用算法类别	数值	不可空	
content	用户输入内容	字符	不可空	
select_type	关联挖掘查看类型	数值	可空	
参数示例	{ "algo_type ":3, "content":"体重 BMI 心率", "select_type":1			

	}

5.3.2 聚类分析模块代码实现

聚类分析模块是基于聚类算法进行挖掘，基于以上的接口设计，核心代码如图 5-2 所示：

聚类模块核心算法：

```
20.  class ClusteringModule {
21.      int clusteringType;
22.      String content;
23.      KMeansModel clusters;
24.      private Vector getPredict(){…};
25.      public String getAnalysisResult(){…};
26.  }
```

图 5-5 聚类模块核心代码

Fig 5-5 Clustering module core code

图 5-5 为挖掘层聚类模块类，其中变量 clusteringType 为聚类类别，包含两个业务模块，1 为诊疗方案推荐，2 为呼吸机设置推荐。变量 content 为前端用户输入的预测指标。变量 clusters 为 KMeansModel 类的实力，根据 clusteringType 的值获取对应业务的聚类模型。

在核心方法中，getPredict()方法的功能是根据 content 内容，在对应模型中进行预测，得到所属的类号和类中心的数据。如果聚类类别为呼吸机设置推荐，则在该方法中对 content 进行向量补全，以保证向量和模型维度相同再进行预测。getAnalysisResult()方法功能是获取 getPredict 方法预测的结果，通过结果获取对应类别对应的诊疗方案以及呼吸机设置推荐。

5.3.3 关联分析模块代码实现

关联分析模块是基于挖掘规则算法进行挖掘，基于以上的接口设计，核心代码如图 5-3 所示：

关联挖掘模块核心算法：

```
1.  class AssociateModule {
2.      int queryType;
3.      String content;
4.      FPGrowthModel<String> model;
5.      private ArrayList getRules(){…};
6.      public ArrayList resultFormat(){…};
7.  }
```

图 5-6 关联分析模块核心代码

Fig 5-6 Association analysis module core code

图 5-6 为挖掘层关联分析模块类，其中变量 queryType 为查询类别，1 为查询所有关联规则，2 为查询部分指标的关联关系。变量 content 为用户选择的指标，当 queryType 为 2 时，content 为用户选择的指标集合。变量 model 为 FPGrowthModel 的实例，对应关联规则算法的训练模型。

在核心方法中，getRules()方法的功能是通过 model 实例获取到最终的关联规则集。resultFormat()方法为对规则集进行格式化，获取 content 内容，返回与 content 内容相关的规则集。

5.4 本章小结

本章主要介绍了本系统两个重要业务层的实现。

从接口设计、核心算法设计和可视化展示三个方面对统计分析层的实现进行了介绍。接口设计主要介绍了前端操作数据和后端回传数据的格式，核心算法设计主要介绍了统计分析类的主要成员变量和核心方法，可视化展示介绍了前端个体查询和群体查询的界面以及交互。

从接口设计、聚类算法设计和关联挖掘算法设计三个方面对数据挖掘层的实现进行了介绍。接口设计和统计分析层类似，主要介绍了前端操作数据和后端回传数据的格式，聚类算法设计主要介绍了聚类模型类的主要成员变量和核心方法，关联挖掘算法设计主要介绍了关联模型类的主要成员变量和核心方法。

结 论

本文主要工作是完成了对呼吸睡眠病情分析决策系统，主要用于睡眠呼吸病人和医疗人员对数据进行管理、分析、决策支持和可视化的交互。其中的关键模块主要涉及到了数据仓库的联机分析处理技术和数据挖掘中的聚类算法和关联规则算法。还用到了分布式系统 Druid 和大数据框架 Spark 等框架作为本系统的一些底层支持。本文的主要工作如下：

1、基于 OLAP 的技术思想，根据本系统的不同维度的数据模型，构建适合本系统的“事实星座”多维数据模型，根据本系统的业务，完成了结合时空信息的病人个体、医生群体、决策者群体的 OLAP 分析。三种业务都实现了基于时间维度的 OLAP 上卷、下钻查询算子，决策者群体 OLAP 分析还完成了基于地区和经纬度的不同空间维度信息的统计分析。该模块的三个业务构成了本系统的统计分析层。

2、基于 K-means 聚类算法完成对病人的病情画像和诊断推荐，其中包括对病人进行呼吸机设置的推荐和诊疗方案的推荐。本文首先使用 IRIS 数据集对几种典型的聚类算法进行了性能对比实验，根据在算法运行时间和准确率等方面的综合考虑，最终确定了 k-means 算法作为本系统的聚类算法。并通过对算法和数据的分析，使用类内距离作为评判依据，对数据进行了 k 值选取的实验，确定了 k 取值为 5 时，具有更优的聚类性能。

3、提出了基于 FP-growth 算法的加权关联规则算法，并根据此改进算法实现了对病人指标的关联分析，挖掘不同指标之间的相关性，辅助医生决策。本文分析了 FP-growth 算法存在未考虑事物实际意义问题，提出了根据医生经验，不同指标诊疗时重要程度，对其赋予相应的权值的思想，使每项指标具有真实意义，并且在多结点路径的权值简化问题上通过对算法的分析和多种方法的总结，选取了适合的方法。所挖掘的规则结果表明，使用加权关联规则算法的结果比原有的 FP-growth 算法结果更加符合实际。

4、设计并实现了整个系统的整体架构。在统计分析层和数据挖掘层基础上，完成了系统的用户管理模块、数据导入模块、数据存储模块和前端模块。其中用户管理模块主要是面向不同用户的登录注册；数据导入模块主要是包括离线数据和感知数据的导入；数据存储模块主要是针对离线数据和感知数据的存储，其中 Druid 主要用于数据的缓存和存储，使用 HDFS 作为数据持久化存储；前端模块主要是使用 Echart 可视化工具和地图接口进行数据的展示，包括折线图、柱状图、热力图等形式。

由于本人能力且时间有限，本系统还存在很多可以完善和改进的方面，有待在未来的研究中进一步补充，主要表现在以下几个方面：

1、本文提出的加权关联规则的权重是根据经验值和医疗相关知识给定的值，可能存在一定的偏差，在后面的研究中可以根据大量相关数据的普通关联规则结果，根据不同指标之间的支持度和置信度综合来判定对应的权值比重。

2、本系统的数据挖掘层只有两大模块，可以在现有基础上增加分类算法分析，如使用逻辑回归算法对病人的大量数据建立回归模型，进行预测分析。

3、本系统的主要工作是根据病人的离线数据进行分析和挖掘，在后面的改进中，可以增加实时数据的分析，如对病人的病情进行实时监控，以便对病情较严重的病人及时发现问题进行医疗。

参 考 文 献

[1] 殷悦，郑钧文. 大数据时代下对数据的新认知[J]. 电子技术与软件工程, 2017(4):180-180.

[2] 夏火松. 数据仓库与数据挖掘技术[J]. 2004.

[3] 王珊. 数据仓库技术与联机分析处理[M]. 科学出版社, 1998.

[4] Herden O. Data Warehouse[M]// Taschenbuch Datenbanken. 2015. [5] Cartoux J Y, LaPresté J T, Richetin M. Face authentification or recognition by profile extraction from range images[C]//Interpretation of 3D Scenes, 1989. Proceedings., Workshop on. IEEE, 1989: 194-199.

[5] White T, Cutting D. Hadoop : the definitive guide[J]. O'reilly Media Inc Gravenstein Highway North, 2010, 215(11):1 - 4.

[6] Witten I H, Frank E, Hall M A. Data Mining: Practical Machine Learning Tools and Techniques[J]. Acm Sigmod Record, 2011, 31(1):76-77.

[7] Aswin V, Deepak S. Medical Diagnostics Using Cloud Computing with Fuzzy Logic and Uncertainty Factors[C]// International Symposium on Cloud and Services Computing. IEEE, 2013:107-112.

[8] Al-Absi H R H, Abdullah A, Hassan M I. Soft computing in medical diagnostic applications: A short review[C]// National Postgraduate Conference. IEEE, 2011:1-5.

[9] 王珊. 数据仓库技术与联机分析处理[M]. 科学出版社, 1998.

[10] Aref M, Bilal H. Exploitation Database Approach for Right On-line Analytical Processing[J]. Research Journal of Applied Sciences Engineering & Technology, 2016, 12(11):1152-1162.

[11] 裴健, 柴玮, 赵畅,等. 联机分析处理数据立方体代数[J]. 软件学报, 1999, 10(6):561-569.

[12] Proulx V K. COM 1420 / IS 1420 Principles and Methods of Interactive Interface Design[J]. 2000.

[13] 李冰, 王悦, 刘永祥. 大数据环境下基于 K—means 的用户画像与智能推荐的应用[J]. 现代计算机:上下旬, 2016(24):11-15.

[14] 张丽娟. 基于大数据分析的用户画像助力精准营销研究[J]. 电信技术, 2017, 8(1):61-62.

[15] 孙忱, 高荣. 基于二维码大数据的消费特征分析[J]. 中国市场, 2015(42):22-24.

[16] 牛温佳. 用户网络行为画像:大数据中的用户网络行为画像分析与内容推荐应用[M]. 电子工业出版社, 2016.

[17] Suzuki H, Omori S, Akiyama K, et al. Hospital information system: Springer Netherlands, US20050033603[P]. 2005.

[18] Wen C, Liu Q. Mobile remote medical monitoring system[C]// IEEE International Conference on Consumer Electronics-China. IEEE, 2017.

[19] 陈玉兵, 王加辉, 吴庆斌,等. 移动医疗系统在临床的应用与意义[C]// 华南医院信息网

络大会. 2013.

[20] Portoni L, Combi C, Pinciroli F. User-oriented views in health care information systems[J]. IEEE transactions on bio-medical engineering, 2002, 49(12):1387-98.

[21] Yang F, Tschetter E, Léauté X, et al. Druid: A real-time analytical data store[M]. 2014.

[22] 黄伟. 数据仓库中 ETL 技术的研究[J]. 中国电子商务, 2014(8):66-66.

[23] Simitsis A, Vassiliadis P, Sellis T. Optimizing ETL Processes in Data Warehouses[C]// International Conference on Data Engineering, 2005. ICDE 2005. Proceedings. IEEE Xplore, 2005:564-575.

[24] Steele B, Chandler J, Reddy S. Cluster Analysis[M]// Algorithms for Data Science. Springer International Publishing, 2016.

[25] 汤效琴, 戴汝源. 数据挖掘中聚类分析的技术方法[J]. 微计算机信息, 2003(1):3-4.

[26] Hamstra, Zaharia M /, Matei. Learning Spark: Lightning-Fast Big Data Analytics[J]. Oreilly & Associates Inc, 2015.

[27] Meng X, Bradley J, Yavuz B, et al. MLlib: machine learning in apache spark[J]. Journal of Machine Learning Research, 2015, 17(1):1235-1241.

[28] 蔡伟杰, 张晓辉, 朱建秋,等. 关联规则挖掘综述[J]. 计算机工程, 2001, 27(5):31-33.

[29] 赵志升, 李桂权. 一种基于 B/S 结构与 C/S 结构结合的新体系结构[J]. 电子技术应用, 2004, 30(8):7-9.

[30] Akinde M, Bohlen M, Johnson T, et al. Efficient OLAP Query Processing in Distributed Data Warehouses.[J]. Information Systems, 2015, 28(1–2):111-135.

[31] 赵升彬, 栾方军, 李鹏. 基于分析服务器的智能 OLAP 系统的实现[J]. 现代计算机:专业版, 2010, 2010(7):203-205.

[32] 喻钢, 周定康. 联机分析处理(OLAP)技术的研究[J]. 计算机应用, 2001, 21(11):80-81.

[33] Ho C T, Agrawal R, Megiddo N, et al. Range queries in OLAP data cubes[J]. Acm Sigmod Record, 1997, 26(2):73-88.

[34] Cao Y, Chen C, Guo F, et al. ES2: A cloud data storage system for supporting both OLTP and OLAP[C]// IEEE, International Conference on Data Engineering. IEEE Computer Society, 2011:291-302.

[35] 张同杨, 马睿, 祁滢. 星型模型在复杂数据仓库环境中应用研究[J]. 价值工程, 2015, 34(31):77-79.

[36] 严任远. 基于数据仓库的企业 OLAP 多维模型的设计与实现[J]. 情报杂志, 2006, 25(9):31-33.

[37] XU Rui, Donald Wunsch 1 1． survey of clustering algorithm[J]．IEEE．Transactions on Neural Networks, 2005,16(3)：645-67 8.

[38] YI Hong, SAM K． Learning assignment order of instances for the constrained k-means clustering algorithm[J]．IEEE Transactions on Systems, Man, and Cybernetics, Part B：Cybernetics,2009,39 (2)：568-574.

[39] 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，2007，24(1):10-13．

[40] 孙吉贵，刘杰，赵连宇．聚类算法研究[J]．软件学报，2008，19(1)：48-61．

[41] 张敏, 于剑. 基于划分的模糊聚类算法[J]. 软件学报, 2004, 15(6):858-868.

[42] 甄彤. 基于层次与划分方法的聚类算法研究[J]. 计算机工程与应用, 2006, 42(8):178-180.
[43] 苏喻, 郑诚, 封军. 文本聚类中基于密度聚类算法的研究与改进[J]. 微型机与应用, 2011, 30(1):1-3.
[44] Chen L, Yu T, Chirkova R. WaveCluster with Differential Privacy[J]. Computer Science, 2015.
[45] 叶茂, 陈勇. 基于分布模型的层次聚类算法[J]. 电子科技大学学报, 2004, 33(2):171-174.
[46] 李戈，邵峰晶，朱本浩．基于神经网络聚类的研究[J]．青岛大学学报，2001，16(4)：21-24．
[47] 李洁, 高新波, 焦李成. 基于特征加权的模糊聚类新算法[J]. 电子学报, 2006, 34(1):89-92.
[48] Bezdek J C, Ehrlich R, Full W. FCM: The fuzzy c-means clustering algorithm[J]. Computers & Geosciences, 1984, 10(2-3):191-203.
[49] FISHER R A ． Iris Plants Database//http ： //www.ics.uci.edu/~mlearn /MLRepository．Html．Authorized license．
[50] Agrawal R, Srikant R. Fast algorithms for mining association rules[M]// Readings in database systems (3rd ed.). Morgan Kaufmann Publishers Inc. 1998.
[51] Borgelt C. An Implementation of the FP-growth Algorithm[J]. Osdm Proceedings of International Workshop on Open Source Data Mining Frequent Pattern, 2010:1-5.
[52] 肖汉光, 蔡从中. 特征向量的归一化比较性研究[J]. 计算机工程与应用, 2009, 45(22):117-119.

攻读硕士学位期间发表的学术论文

1 贾熹滨，陈羿霖. 软件著作权非结构化医疗数据管理分析系统. 登记号 2017SR122214，2017 年 2 月.

致 谢

在我的硕士论文即将完成之际，我要借此机会感谢在我攻读硕士研究生的三年间，所有那些在科研、工作以及生活中帮助过我的所有人。

首先，我要衷心地感谢我的导师贾熹滨副教授。在科研方面，贾老师精益求精、勇攀高峰，面对新技术和新知识时刻保持着的好学之心、面对困难时从不轻言放弃的决心和充满坚韧的学术精神深深感染着我。在教学方面，贾老师兢兢业业、因材施教，对我给予了深刻的教导。在生活中，贾老师乐观豁达、关心同学，对于实验室的其他兄弟姐妹们十分照顾。在此次论文的选题、算法改进、实验方案设计、文章的撰写、修改以及定稿的整个毕设环节中，贾老师给与了我全方位、多角度的指导，为我的本次毕设任务的顺利完成奠定了坚实基础。不仅是对于论文用词上的修改，而是对于整个论文的关键点提出了建设性的意见，在我曾经迷茫的时候提出了宝贵的意见。在工作中，贾老师常与本人分享工作经验与心得，为我成功组织各项学生活动提供了宝贵经验，这也必将成为我未来工作中的宝贵财富。总之，贾老师的谆谆教诲与无私关怀让我受益良多，我必将铭记于心并努力实践、传承。

其次，感谢和我同实验室研究生的兄弟姐妹们。你们不仅营造了实验室欢乐的气氛，更是在我的研究中提供了很多的帮助。与你们同行，科研之路不再枯燥、孤单，前行的道路不再是满地荆棘，生活之路又是那样色彩斑斓。感谢你们，我的师兄师姐、师弟师妹以及亲爱的室友们！珍惜与你们在一起的美好时光、更期待与你们在将来的日子更美好地相处、相助。

最后，要感谢的还有我至亲的家人，你们的关怀与鼓励，是我前行中最大的信心和动力。你们用辛劳的汗水、无私的关爱，构筑了我学业顺利的物质基础与精神基础。

总之，感谢每一位关心与批评过我的人，是你们让我成为一个更好的我，一个更有益于社会的我！