

2020 届硕士专业学位论文

分类号: _____

密 级: _____

学校代码: 10269

学 号: 71174500025



華東師範大學

East China Normal University

硕士专业学位论文

MASTER' S DISSERTATION

论文题目: 基于关联规则的智能音乐 推荐平台的设计与应用

院 系: 软件工程学院

专业学位类别: 软件工程

专业学位领域: 软件工程

论文指导教师: 毛宏燕 副教授

论 文 作 者: 窦超阳

2020 年 5 月

Dissertation for master degree in 2020

Student ID: 71174500025

University code: 10269

East China Normal University

Title: Design and Application of Intelligent Music Recommendation Platform Based on Association Rules

Department :	<u>The School of Software</u>
	<u>Engineering</u>
Type :	<u>Software Engineering</u>
Domain :	<u>Software Engineering</u>
Supervisor :	<u>Hongyan Mao (Professor)</u>
Candidate :	<u>Chaoyang Dou</u>

May, 2020

华东师范大学学位论文原创性声明

郑重声明：本人呈交的学位论文《基于关联规则的智能音乐推荐平台的设计与应用》，是在华东师范大学攻读硕士/博士（请勾选）学位期间，在导师的指导下进行的研究工作及取得的研究成果。除文中已经注明引用的内容外，本论文不包含其他个人已经发表或撰写过的研究成果。对本文的研究做出重要贡献的个人和集体，均已在文中作了明确说明并表示谢意。

作者签名： 窦越阳

日期：2020 年 5 月 2 / 日

华东师范大学学位论文著作权使用声明

《基于关联规则的智能音乐推荐平台的设计与应用》系本人在华东师范大学攻读学位期间在导师指导下完成的硕士/博士（请勾选）学位论文，本论文的研究成果归华东师范大学所有。本人同意华东师范大学根据相关规定保留和使用此学位论文，并向主管部门和相关机构如国家图书馆、中信所和“知网”送交学位论文的印刷版和电子版；允许学位论文进入华东师范大学图书馆及数据库被查阅、借阅；同意学校将学位论文加入全国博士、硕士学位论文共建单位数据库进行检索，将学位论文的标题和摘要汇编出版，采用影印、缩印或者其它方式合理复制学位论文。

本学位论文属于（请勾选）

- ☐ 1. 经华东师范大学相关部门审查核定的“内部”或“涉密”学位论文*，于 年 月 日解密，解密后适用上述授权。
- ☒ 2. 不保密，适用上述授权。

导师签名： 王辰燕

本人签名： 窦越阳

2020 年 5 月 2 / 日

* “涉密”学位论文应是已经华东师范大学学位评定委员会办公室或保密委员会审定过的学位论文（需附获批的《华东师范大学研究生申请学位论文“涉密”审批表》方为有效），未经上述部门审定的学位论文均为公开学位论文。此声明栏不填写的，默认为公开学位论文，均适用上述授权）。

窦超阳 硕士学位论文答辩委员会成员名单

姓名	职称	单位	备注
万燕	教授	东华大学	主席
刘静	教授	华东师范大学	
张敏	副教授	华东师范大学	

摘 要

随着我国现代移动互联网信息技术以及新一代物联网信息技术的飞速发展，通过各种移动网络平台收听音乐已经日趋平凡，但由于歌曲词条的复杂性和缺乏有效信息，人们很难从网络中选择符合自己喜好的音乐。较为常见的方法一般仅可以基于现有的音乐库来对音乐进行检索。当用户记住目标音乐库中的各种信息时，才可以完成音乐的检索。近年来，快节奏已经成为了新时代的一种特殊的符号，所以这种方式很难满足用户的需求。用户需要得到快速及时的响应，及时为用户提供相关的音乐才能提高用户的满意度，得到用户的好评。对音乐检索而言，现阶段最重要的问题是怎样为需求个性化、多样化的用户提供最合适的音乐。

本文分析了音乐推荐平台的应用现状，对相关音乐推荐算法和数据挖掘技术进行研究后，确定将关联规则推荐算法应用于音乐推荐技术中，对比分析了本文提出的关联规则推荐算法与根据内容或是协同过滤的算法的不同，通过实验说明了关联规则推荐算法在音乐推荐上的可行性和优点。本文从性别、风格、年龄和区域等多个维度对音乐推荐策略思路进行了分析研究，给出了音乐推荐的基本流程，使用关联规则推荐算法分发歌曲和歌单，应用场景包括热门歌单、排行榜等功能组件。设计给出了音乐推荐平台的架构，其中包括业务体系建设、算法工程体系建设和数据体系建设。在业务体系中分别建立内容分析平台和内容分发平台，在算法工程中涉及到推荐搜索、流量分发、排序、分析和存储等服务框架和系统，在数据体系中建立 **Storage** 实时数据仓库和后端离线数据仓库，并在客户端和后台进行自动化实时打点。在模型构建中尝试了线性模型、树模型、大规模 **FTRL**、深度神经网络和深度时序网络等模型实践，最终选择了深度时序网络。在多目标问题上，本文采用多目标联合训练的实现方法，建立起了用户体验评估体系，根据用户反馈做出策略和机制的调整。最后本文设计实现了音乐推荐结果的可视化平台，平台采用了 **B/S** 三层架构，使用了 **MVVM** 框架 **Vue.js** 作为底层支撑。本文给出了音乐推荐平台的组件架构和用例分析，结合关联规则推荐算法实现了推荐歌单、热门歌手、排行榜等业务组件，对推荐结果的展示分别进行了性能和效果的测试。总之，本文将关联规则算法应用于音乐推荐中，设计并实现了可视化音乐推荐平台，为海量音乐的快速访问提供了高效率的技术支持。

关键词：关联规则，音乐推荐，智能音乐平台

ABSTRACT

With the rapid development of China's modern mobile Internet information technology and the new generation of Internet of Things information technology, listening to music through various mobile network platforms has become increasingly common, but due to the complexity of song entries and the lack of effective information, it is difficult for people to access the Internet Choose music that suits your preferences. The more common methods can generally only retrieve music based on existing music libraries. Only when the user memorizes various information in the target music library can the music retrieval be completed. In recent years, fast-paced has become a special symbol of the new era, so this way is difficult to meet the needs of users. Users need to get a quick and timely response and provide users with relevant music in time to improve user satisfaction and get good reviews from users. For music retrieval, the most important issue at this stage is how to provide the most suitable music for users with personalized and diversified needs.

This article analyzes the application status of music recommendation platforms. After researching related music recommendation algorithms and data mining techniques, it is determined to apply association rule recommendation algorithm to music recommendation technology. A comparative analysis of the association rule recommendation algorithm and the content or It is different from the collaborative filtering algorithm. Experiments have shown the feasibility and advantages of the association rule recommendation algorithm in music recommendation. This article analyzes and researches the music recommendation strategy from multiple dimensions such as gender, style, age and region, and gives the basic process of music CNC recommendation. The association rule recommendation algorithm is used to distribute songs and song lists. Function components such as leaderboards. The design gives the architecture of the music recommendation platform, including business system construction, algorithm engineering system construction and data system construction.

Establish a content analysis platform and a content distribution platform in the business system respectively. In the algorithm engineering, service frameworks and systems such as recommendation search, traffic distribution, sorting, analysis, and storage are involved. In the data system, establish a storage real-time data warehouse and back-end offline Data warehouse, and automated real-time management on the client and background. In the model construction, we tried linear model, tree model, large-scale FTRL, deep neural network and deep sequential network model practice, and finally chose deep sequential network. On the issue of multi-objectives, this paper adopts the realization method of multi-objective joint training, establishes a user experience evaluation system, and makes adjustments to strategies and mechanisms based on user feedback. Finally, this article designs and implements a visual platform for music recommendation results. The platform uses a B / S three-tier architecture and uses the MVVM framework Vue.js as the underlying support. In this paper, the component architecture and use case analysis of the music recommendation platform are given, and the association rule recommendation algorithm is used to implement recommended song lists, popular singers, rankings and other business components. The performance and effect of the recommendation results are tested respectively. In short, this article applies the association rule algorithm to music recommendation, designs and implements a visual music recommendation platform, and provides efficient technical support for rapid access to massive music.

Keywords: Association rules, Music recommendation, Intelligent Music Platform

目 录

第 1 章	绪论	1
1.1	研究背景与意义	1
1.2	国内外音乐推荐研究现状	6
1.2.1	国外音乐推荐研究现状	6
1.2.2	国内音乐推荐研究现状	7
1.3	本文的主要内容、研究思路和方法	9
1.4	本文整体结构	9
第 2 章	相关技术介绍	11
2.1	数据挖掘算法的理论基础	11
2.1.1	数据挖掘技术研究综述	11
2.1.2	数据挖掘基本原理	12
2.2	基于关联规则的推荐算法	13
2.3	基于内容的推荐算法	15
2.3.1	基于标注内容的推荐算法	15
2.3.2	基于音乐特征的推荐算法	16
2.4	基于协同过滤的推荐算法	17
2.4.1	基于用户的协同过滤推荐算法	17
2.4.2	基于物品的协同过滤推荐算法	19
2.5	本章小结	19
第 3 章	关联规则推荐模型	20
3.1	关联规则推荐的适用性分析	20
3.1.1	关联规则推荐的优势	20
3.1.2	关联规则推荐算法的内容	20
3.1.3	关联规则推荐算法的应用	21
3.2	构建关联规则推荐模型	23
3.2.1	数据集获取	23
3.2.2	构建数据集树结构	23
3.2.3	挖掘频繁项集	27
3.3	选择评价标准	29
3.4	实验对比分析	31

3.5 本章小结.....	33
第4章 音乐推荐平台设计.....	34
4.1 平台开发相关技术.....	34
4.1.1 B/S 项目架构.....	34
4.1.2 MVVM 开发模式.....	36
4.1.3 Vue.js 开发框架.....	37
4.2 音乐推荐策略.....	39
4.3 业务逻辑与模型实践.....	41
4.3.1 推荐逻辑与业务体系.....	41
4.3.2 模型实践与应用.....	43
4.3.3 多目标联合训练.....	45
4.4 平台功能模块设计.....	46
4.4.1 可视化组件设计.....	46
4.4.2 数据库设计.....	48
4.5 本章小结.....	50
第5章 平台实现与测试.....	51
5.1 推荐结果展示模块实现.....	51
5.1.1 推荐歌单.....	51
5.1.2 热门歌手.....	52
5.1.3 排行榜.....	53
5.1.4 搜索与播放器.....	54
5.2 平台测试.....	55
5.3 本章小结.....	61
第6章 总结展望.....	62
6.1 本文完成的主要工作.....	62
6.2 本文的主要贡献及创新点.....	62
6.3 今后的工作.....	63
致谢.....	64
参考文献.....	65

第1章 绪论

1.1 研究背景与意义

互联网的飞速发展极大的改变了人们对于信息的获取方式,信息化时代随之而来,在科学技术飞速发展的今天,网络信息的数量也随之极度扩大,当人们面对巨大的信息量时,往往很难甄别出自己真正感兴趣的信息,或者无法快速有效的找到有用的信息,信息过载成了一大难题。因此,在如今的大数据时代,对于信息的快速获取与有效筛选成为一个重要的研究方向,数据挖掘技术的日益成熟使得音乐推荐技术得到了快速发展,它让人为手动输入关键词检索音乐成为历史,获取音乐资源的方式从根本上得到了改变。

在传统模式里,人们解决信息超载问题的方式是通过输入关键词检索出想要的信息,也就是搜索引擎。用户把想要检索的关键词输入搜索引擎,然后搜索引擎会返回与之相关的检索信息,在一定程度上提高了人们检索信息的效率^[1]。但是当不同的用户输入的关键词相同时,搜索引擎返回的搜索结果列表也是相同的,并不能根据个体的不同而返回不同的信息^[46],这就成了搜索引擎的一个通病。换句话说,也就是搜索引擎并不能提供给用户个性化、多元化、定制化的服务^[2]。随着社会的发展和进步,这种信息检索方式已经不能满足人们对信息获取的个性化需求。

个性化推荐系统随之而来,个性化推荐系统的原理实际上是分析和利用每个用户的行为习惯、历史记录、爱好和品味的典型特征^[4],从这些特点中找出符合目标用户需求的信息数据,这种个性化推荐的服务既解决了数据信息超载的问题,又给了每一个目标用户定制化的服务体验。它以目标用户对产品项目的需求偏好为中心,基于用户所处的类别为用户提供个性化的推荐服务,收集、统计、分析用户的使用数据^[3],对用户偏好的产品或是信息进行预测,从而为用户提供与其分类匹配的信息。推荐系统和传统的搜索方式比较来讲,其主要优势有:通过分析目标用户的个性、行为习惯和兴趣偏好等特征数据^[44],能够主动收集目标用户的数据信息,专门为用户推荐其感兴趣的内容;同时,推荐系统也能够对用户的

动态进行跟踪,及时了解用户偏好或是兴趣的变化,结合用户的情况对信息的推荐进行优化,以更适合用户的方式精准的为用户提供相应的信息^[45]。除此之外,传统搜索的推荐具有一对多的特点^[5],而推荐推荐系统可以根据用户的需求提供个性化的内容,使用户偏好或是兴趣得到满足,其提供的服务具有一对一的特点。推荐系统还有一个明显的优势,由于可以为用户提供精准的内容,所以可以极大的减少用户搜索内容过程中消耗的能量与造成的成本,有利于增加用户的满意度。

目前,在线音乐推荐领域有很多知名的个性化音乐推荐平台。世界上最早的个性化音乐推荐平台是 Pandora 网站和 Last.fm 网站^[6]。Pandora 网站使用音乐基因组对音乐进行推荐和管理,首先对用户喜爱的音乐风格、艺术家和推荐歌曲的反馈进行收集和分析,接下来结合音乐的音色、编排、节奏以及旋律方面的特点,为用户推送相似的音乐。Last.fm 可以收集、记录、分析用户的播放与收听信息,确定用户的品味与偏好,并综合考虑用户的社交情况推送精准的音乐。我国与音乐推荐相关的研究起步相对较晚,尽管可以借鉴学习他人的成功经验,不过目前依旧处于探索阶段。早期的代表产品包括豆瓣电台、Songtasty 网站和虾米网,其中豆瓣电台最为突出。

豆瓣电台、Last.fm 以及 Pandora 等都是可以为用户提供个性化推荐服务的音乐平台,不过三者的核心技术存在一定的差异。具体技术介绍如下。

(1) Pandora 网络电台

说到 Pandora 网络电台,人们会想到“基因组计划”,这是它的标志性技术。该计划包括约一万名歌手和一万多首歌曲,所有这些都进行了详细分析。基因组计划可以全面的分析音乐库中的所有音乐,并对音乐进行合理的分类,分类的依据主要是音乐的属性,比如旋律、乐器、歌词以及和声等,这些作品由 Pandora 网络电台的特别分析师完成。

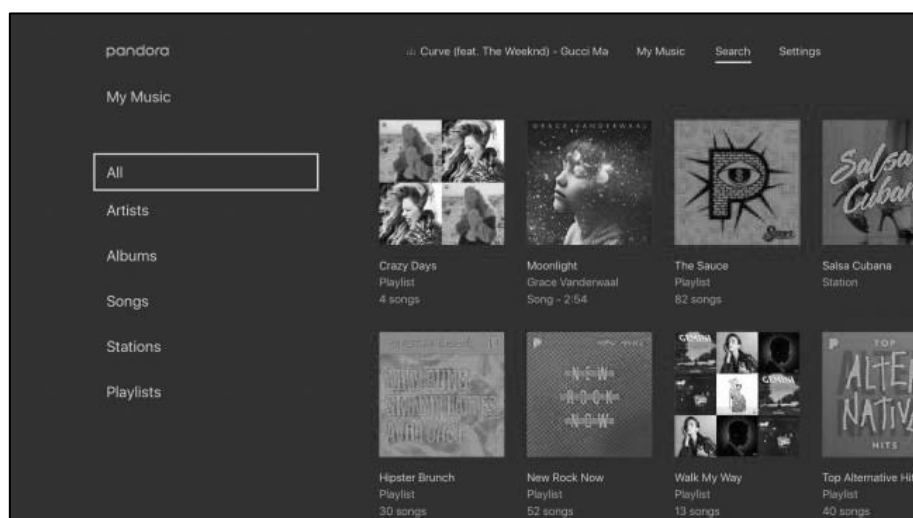


图 1-1 Pandora 网络电台界面

(2) Last.fm

与其他系统最大的区别在于，Last.fm 开发出了音乐社区，基于音乐社区可以对歌曲进行提炼，并根据大量的事实为用户提供音乐推荐服务。它基于来自数千万音乐用户的真实收听记录，因此，在用户规模不断扩大的过程中，系统中与各用户的相关的数据与信息也会随之不断增加^[42]，使系统可以更好的对用户进行分析与分类，提供更为精准的音乐。除此之外，Last.fm 系统可以根据用户对社区中朋友的探索，通过分析用户朋友的偏好对用户的情绪进行分析与预测^[43]，相对而言，这一功能的复杂程度要高于基因工程。

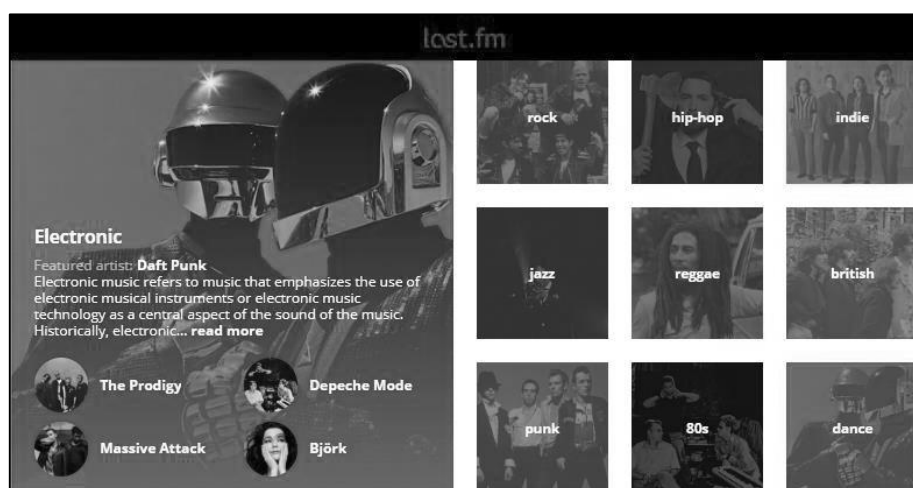


图 1-2 last.fm 界面

(3) 豆瓣电台

豆瓣平台与 Last.fm 较为相似，其属于一种可以实现音乐在线收听功能的平台，该平台可以根据用户搜索或是听过的音乐为用户筛选相关音乐并推送给用户^[41]。用户首次使用系统时，豆瓣平台会让用户选择最喜欢的艺人或输入最喜欢的歌曲或选择最喜欢的风格，这也是推荐之前的基本工作。

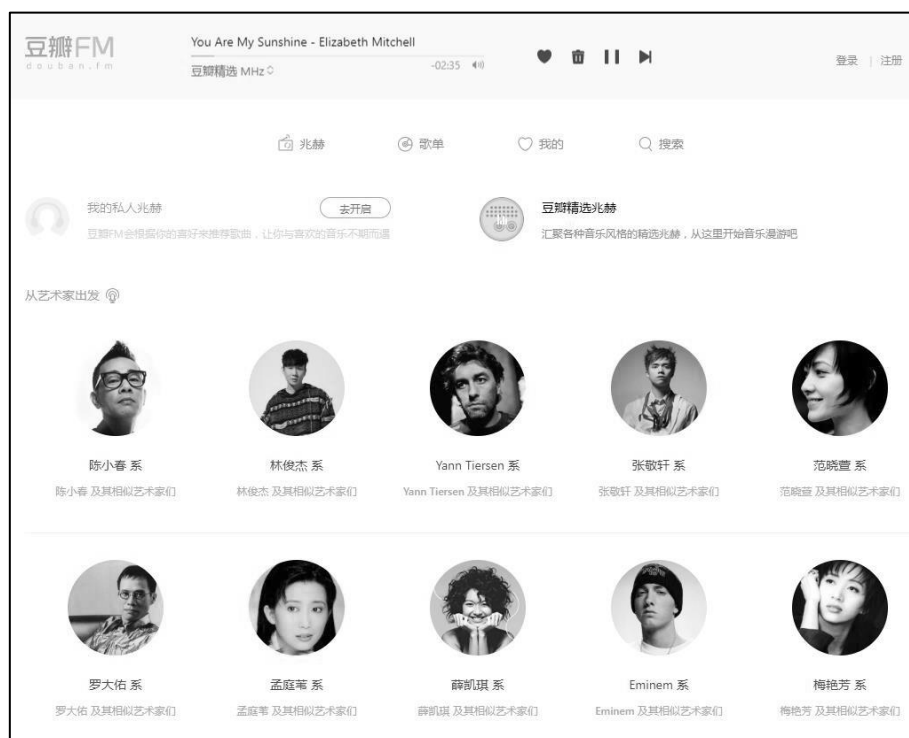


图 1-3 豆瓣电台界面

(4) 其他国内音乐网站

国内还有很多其它音乐平台，如 Songtasty，该平台可结合用户的音乐标签为用户提供相关的音乐，如闽南语、流行乐、摇滚乐等^[8]，也可以找到口味相近的人更好地分享音乐，从而提供精准的音乐推荐服务。此外，虾米音乐可分析用户的行为习惯，为用户推荐相关的音乐。



图 1-4 SongTasty 网站界面



图 1-5 虾米音乐界面

分析上述各音乐平台的特点，不难发现，无论是哪种类型的音乐推荐网站，均采用了一些推荐算法，各音乐平台大多基于用户的使用习惯以及收藏等对用户进行分析^[7]，并根据音乐的特点为用户推荐其可能偏爱或是感兴趣的音乐。通常各音乐平台均会采用不同的算法，并根据这些算法的变化来进行音乐推荐。

1.2 国内外音乐推荐研究现状

在时代不断发展的过程中,网络音乐的类型也更为丰富,同时,用户规模也呈现出了不断上升的趋势。这对音乐平台而言是一个重要的发展机遇,各平台要在有效维护现有用户的同时,主要开发潜在用户,不断提升自身的用户规模^[9]。音乐平台要设计出美观舒适的界面,为用户提供良好的操作体验,此外,还需深入了解用户的偏好与兴趣,为用户推送其可能喜欢的音乐^[40]。因此就需采用先进的音乐推荐技术,就某种意义而言,音乐推荐技术对音乐平台的发展极为重要,可以直接决定其是否可以留住现有用户,并有效吸引新的用户,从而实现更高的商业价值^[10]。

1.2.1 国外音乐推荐研究现状

西方国家很早以前就已针对音乐推荐进行了研究,现阶段一些音乐推荐技术已相对较为成熟,对我国音乐平台具有一定的借鉴价值。Man-Kwan 等根据用户情感建立一种音乐推荐架构^[11],研究出了音乐相关度图,为音乐情感模型的建立奠定了基础。Dmitry Bogdanov 根据与内容相关的技术分析用户的偏好与兴趣,具有为用户提供其可能喜欢的音乐作用,分析用户偏好或是兴趣的依据是用户提供的音乐,通过模型获得音轨,分析音乐的特点作为信息过滤的判断条件^[12]。Ziwon Hung 提出了一种基于用户本身的推荐技术,该技术可以根据用户在平台上写入的文本信息^[13],比如个人故事或是简介为用户推荐相关的音乐。可以通过两种方法基于标签为用户推荐相关音乐, Ben Horsburgh 在创新、改进标签表示法后,研究出了一种混合表示法,该方法可以在不引入内容的情况下,通过稀疏标签实现表示的功能,并通过动态加权对拟标签量进行限制^[14]。Ja-Hwung Su 则提出,由于可能存在多种不同的标签,所以基于标签推送音乐可能存在一定的风险,也就是无法为用户推荐其真正喜欢的音乐,所以其研究出了可以基于标签分析不同音乐相似性的音乐推荐技术^[15],并通过研究证明,与仅基于评级或是评价建立的推荐技术相比,其提出的算法更具优势。Chen-Che Lu 综合采用根据情感、协作与内容的推荐技术,提出了一种混合推荐技术^[16],该方法可以基于用户偏好

或是兴趣对上述方法的权重进行计算,并综合分析结果为用户提供更为精准的歌曲。

1.2.2 国内音乐推荐研究现状

国内音乐推荐技术的起步较晚,不过随着相关研究的推进,我国各音乐网站也对音乐推荐进行了探索,并尝试对相关技术进行创新。现阶段,我国学术界与音乐推荐技术相关的研究主要是基于国外学者的研究进行创新与改进。刘珊珊设计并实现了一种有机融合社会标签与音乐特征的推荐系统^[17],该系统可以收集音乐的节奏以及音色等方面的特征,接下来根据用户选择的社会标签进行降维,从而确定音乐的相似性,呈现出可视化的推荐结果。建立该系统时,首先需要根据音乐特征建立数据库,该数据库由音乐的音频特征与社会标签构成,方式主要有数据挖掘与信号处理技术^[18],接下来针对音乐特征进行预处理与降维,处理特征矢量,从而在二维空间中将歌曲的相似性呈现出来。张燕等结合用户的点击与内容分析,并且结合描述、提取的音乐特征与相关的推荐技术^[19],在大范围内对音乐片段进行搜索,可以有效的减小相关投票、表演者以及因为人的限制。王君等学者基于协同与内容的相关技术,实现了一种层次推荐技术,并针对层次推荐的概念进行了介绍。其一,基于层次推荐技术实现的系统可以根据各用户兴趣的相似度实现协同过滤;其二,音乐的相似度可以分为多个维度,比如歌词、音色、情感、旋律以及节奏等,将上述两个方面内容有机融合能够更好的体现二者的优势,从而为用户提供更为精准的音乐,使用户的满意度得到提升。

国内也有部分研究具有明显的创新性,比如有学者在研究的过程中,融合了相关领域的技术与音乐推荐技术。肖利民等学者设计出一种可以为规模较大的音乐库提供支持的推荐技术^[19],基于相似算法,考虑到图像处理单元可以实现多线程处理的功能,将索引机制与图像处理单元结合提出了一种音乐推荐方法,该方法在数据处理规模方面具有一定的优势。孔令志等学者基于频率倒谱系数,通过高斯混合模型构建出了一种音乐推荐模型^[20],首先该模型需要利用相关技术对音乐的特征进行提取,接下来通过相关算法生成音乐模板,最后根据音乐模板库确定各音乐的相似性,从而为用户推送精确的音乐,使用户的需要得到满足。

印晶晶等学者根据概率图研究出了一种音乐推荐技术,首先,该方法需要分析各音乐间的关系,也就是确定音乐间的相似性,接下来建立音乐网络,基于主题模型确定主题的概率分布并以此构建概率图模型。该模型包含了基于主题概率分布的属性与基于音乐网络的全局结构,进行转化后获得因子图,通过推理算法得到各主题内包含的音乐文件,也就是推荐列表。

国内也有部分学者针对音乐推荐技术的应用进行了研究,林鸿飞等学者首先指出各音乐平台在音乐推荐方面的问题^[21],如数据集稀疏,评分信息、项目属性以及用户描述缺失等,接下来针对协同标注的优势进行了介绍,提出由于协同标注可以描述丰富的信息,而且也不会缺失项目属性,所以通过协同标注为用户更好的推荐音乐,林鸿飞等在研究的过程中引用了音乐基因组^[22],用户在添加标签或是标注后,系统可以对此类信息进行采集,以此分析用户更为关注的音乐特征,了解用户的偏好或是兴趣,筛选与用户偏好的音乐具有相似性的音乐风格或是音乐类型,从而为用户提供更好的推荐服务,满足用户的个性化需求。

基于关联规则针对音乐推荐进行的研究中,张二芬等学者经研究提出^[23],在音乐推荐中应用关联规则较易实现,而且可以便捷的更新推荐结果。王纯等学者首先分析了彩铃推荐问题^[24],接下来基于关联规则进行了研究,并应用了推荐权重。王中原等学者综合应用信息过滤与数据挖掘技术提出了一种音乐推荐技术,以此设计并实现了一种推荐系统^[25]。潘兴德等针对常见的推荐技术进行了总结与分析^[26],并指出了各方法的不足,针对音乐推荐技术的发展趋势进行了展望。

通过以上分析可以发现,现阶段各国学者在针对音乐推荐进行研究后,主要提出了几种相关技术,分别为基于协同、标签与内容的推荐技术^[38],与这些技术相关的研究相对较多,而基于关联规则针对音乐推荐开展的研究相对较少,此类研究尚处于初级阶段。而基于关联规则的广告推送与数据挖掘技术在购物领域发挥了巨大的作用,因此在音乐推荐领域探索关联规则算法的应用存在一定的必要性,相信其也可以在应用方面做出极大的贡献,针对该算法的应用效果与可行性进行研究与验证,探索其具体的应用方法,可以在精准营销方面促进音乐商业的发展^[39]。

1.3 本文的主要内容、研究思路和方法

根据研究目的与内容,本文首先针对现阶段数据挖掘领域中的常见推荐算法进行了介绍,并着重分析了关联规则算法中的 FP-growth 算法,将该算法与基于内容、协同的算法进行对比,揭示该算法的优势。最后基于该算法设计并实现了一种移动音乐推荐平台,并测试了本文实现的音乐平台。

1.4 本文整体结构

本研究共由五部分构成,具体为:

第一章 绪论,针对本文的研究目的、意义与背景进行介绍,遍历与音乐推荐技术相关的研究,总结研究现状,对本文的研究思路、方法与内容进行阐述。

第二章 简要介绍数据挖掘技术,分别介绍了目前较为常见的推荐算法,具体为基于协同过滤、内容与关联规则的算法,由于本文拟采用关联规则算法设计一种音乐推荐系统,因此对其进行了重点的介绍。

第三章对关联规则算法中的 FP-growth 算法做了关于应用在音乐推荐中的适用性分析,选取了 QQ 音乐数据作为数据源,分别由多样性、新颖性、覆盖率以及准确度几个维度对基于关联规则、基于协同过滤与基于内容的算法进行了对比分析。

第四章将 FP-growth 算法应用与实际项目音乐平台的开发中,对音乐平台开发所使用的架构模式,技术栈做了简单介绍,并对平台的功能模块设计以及各个模块的实现过程做了说明。

第五章对音乐推荐平台的实现过程中的核心代码部分做了说明,并对推荐结果的展示效果和性能做了测试和介绍。

第六章总结。针对全文进行总结,梳理了本文的主要工作,展望了音乐平台今后的发展趋势。

本文的研究框架如图 1-6 所示:

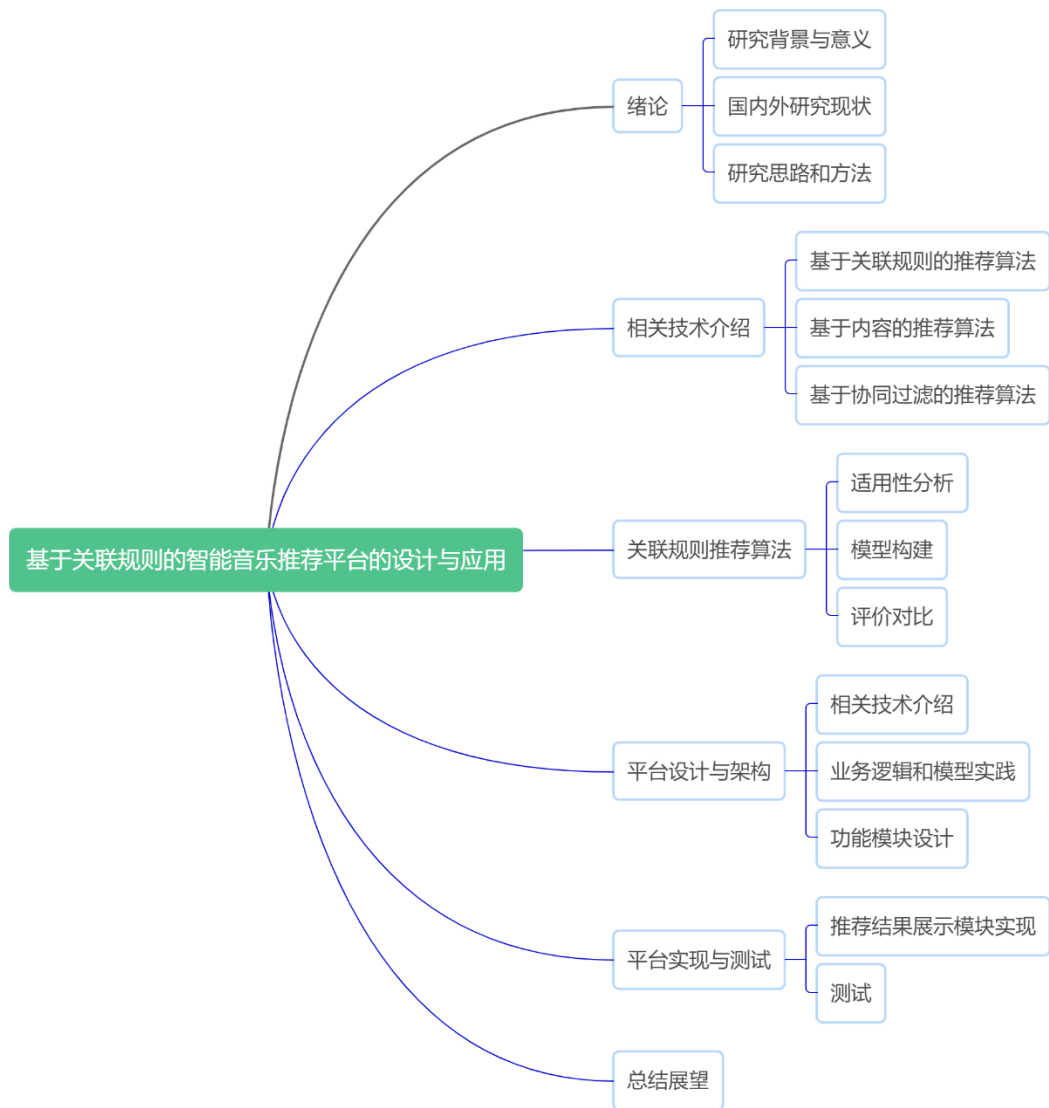


图 1-6 本文的研究框架

第2章 相关技术介绍

2.1 数据挖掘算法的理论基础

需要是创造的源泉，由于互联网的飞速发展，数据信息愈来愈庞大，从巨大的信息中检索出人们想要的信息引起了广泛关注，数据挖掘应运而生，所谓数据挖掘就是从庞大的数据库中找出隐性的、未可知关联的并存在价值的信息的过程。

2.1.1 数据挖掘技术研究综述

数十年前就已出现了数据挖掘这一概念。上个世纪八十年代就有学者针对数据挖掘与知识发现进行了研究^[27]，并提出了关联规则算法，该算法可以发现海量数据各项集间的相关关系或是关联。近年来，关联规则算法已成为了学界的一项重要课题，学者们对其进行了大量的研究，在研究推进的过程中^[28]，逐步提高了该算法的应用范围。在各关联规则算法中，最具代表性的是 FP-growth 与 Apriori 算法。1993 年 Agrawal 等学者提出了一种根据项集间的联系对关联规则进行挖掘的构想^[30]，在针对这一构想进行探索与实践的过程中，Agrawal 等于次年研究出了 Apriori 算法，Apriori 算法可以根据频繁项在两个阶段中的情况进行迭代。该方法要对数据库进行多次扫描，如数据库规模较大可能就需要较长的时间，因此该方法具有一定的局限。此后 J.Han 等学者基于 Agrawal 等的成果提出了 FP-growth 算法^[29]，该算法仅需对数据库进行两次扫描，因此极大的降低了扫描成本与时间，使算法的适用性得到了提升。

现阶段西方学者在针对数据挖掘进行研究的过程中，主要关注的是与数据挖掘相关的理论、数据挖掘的方法以及相关方法的改进等方面的问题^[31]。比如有学者提出了 BID 框架，该框架的侧重点属于基础的底层框架。而 Jure Leskovec 等学者针对网络内容与信息传播问题开展了相关的研究^[32]，并在信息传播领域取得了一定的成绩。韩家玮等学者针对网络对象的关系与排序挖掘进行研究，极大的推动了这一领域相关研究的发展。此外也出现了一些可用于商业领域的挖掘系统，比如 DBMiner 等。Oracle 等数据企业也积极针对数据挖掘进行更全面深入的研究^[33]，并成功将研究成果转换为实际的产品后推动了数据库的智能化发展。

2.1.2 数据挖掘基本原理

近年来,在各种相关技术发展迅速的同时,信息技术也得到了进一步的发展。很多领域均可以产生海量的数据,不过在各领域信息呈现爆炸性增长的同时,与知识挖掘相关的技术却相对较为匮乏。对于很多领域来讲,怎样可以快速有效的挖掘海量信息中的有价值的信息,已成为了一个重要的问题^[34]。为应对这一问题,在学者以及研究人员的努力下出现了数据挖掘技术,该技术是一种知识发现过程中的重要环节,不过现阶段数据挖掘技术依旧处于初级阶段,并未发展成熟,仅可以为相关应用的研发以及实践等提供框架方面的支持。

通过数据挖掘技术所获取的信息具有实用性、有效性与未知性的特征,数据挖掘主要涉及偏差分析、实训模式、预测、分类、聚类以及关联分析等方面的内容^[35]。关联规则算法对音乐数据具有一定的适用性,通过该技术可以确定音乐数据的关联,从而为用户提供其偏好或是感兴趣的音乐,使用户的需求得到满足,达到精准推送的目的,从而提高用户的满意度。

如图 2-1 所示为数据挖掘的一般流程,在数据挖掘的过程中,通常首先需要对问题进行定义,接下来对选择的数据进行预处理,并在对数据进行转换后将有价值的信息抽取出来,最后对知识进行同化并应用知识。在定义问题的过程中,要针对客户需求与项目目标进行明确,接下来清洗规范所需挖掘的数据,最后基于合适的算法完成数据的挖掘工作,并对比分析挖掘所得的结果,通过成果转换在实际的学习或是工作中应用同化后的知识。

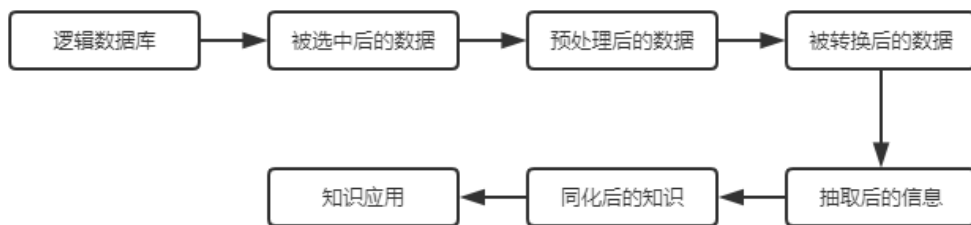


图 2-1 数据挖掘的一般流程

2.2 基于关联规则的推荐算法

与事件间关联或是依赖相关的知识就是关联，关联可以理解为关联性与相关性，两者均能够针对某一事件与其他事件的关联程度进行描述。

假定 I 代表全部项目的集合， A 集合的各元素均为项目，也就是项集。而事物 T 属于项目子集，所有事务均存在一个与之对应的 Tid ，也就是事务标识。 A 包含于 T ，如 A 中的项目数量为 k ，此时 A 就是 k 项集。事务数据库为 D ， D 中 A 数量占总体的比例为项集支持度，如该数值大于支持度阈值下限，那么 A 就被叫做大项集或是频繁项集。

XY 的蕴含关系可以理解为关联规则，如 XI, YI ，同时满足 $XY=\Phi$ ，那么 X 与 Y 分别为规则前件与结果。关联规则 XY 具有信任度与支持度。

规则中某一模式出现的频率就是支持度，如包含 XY 的事务占比为 $s\%$ ，那么可以说明 D 中 XY 有 $s\%$ 的支持度，也可通过 $P(XY)$ 表示，也就是 $\text{support}(XY) = P(XY)$ 。

蕴含强度就是信任度，也就是 $c\%$ 的事务同时包含 X 与 XY ，如 X 有 $\text{support}(x)$ 的支持度，那么可以通过 $\text{support}(XY)/\text{support}(X)$ 描述信任度，可以表示为 $(Y|X)$ ，其属于一种条件概率。

在关联规则中，信任度与支持度是两种重要的度量依据，确定高于预设信任度与支持度最小值的规则的过程就是关联规则挖掘^[36]。

(1) Apriori 算法理论

通过前文介绍可以了解，早在 1994 年就已出现了 Apriori 算法，如图 2-2 为 Apriori 算法的基本流程，该算法属于一种开创性的算法，其可以通过 K 项集的情况挖掘 $K+1$ 项集的关联规则，具体过程为：

针对数据库进行整体的扫描，对所有项出现的次数进行计算，确定符合最小支持度要求的项，找出 L_1 ，接下来基于 L_1 确定 L_2 、 L_3 ……，直至无法确定频繁 K 项集，此时完成一次整体的扫描。这一过程涉及的步骤主要有两个：其一为连接步，也就是基于 L_{k-1} 找出 C_k ，其二为剪枝步， C_k 属于 L_k 中的一种超集，即 C_k 中的集合可以并非频繁集，不过 C_k 需要包含任意频繁 K 项集。基于先验性可压缩

C_k ，剔除不满足最小信任度或是支持度的项，便可获得全部的频繁子集，此时便可确定强关联规则，也就是找出同时符合最小信任度与支持度要求的项的规则。

通过上述分析能够发现，Apriori 算法要对数据库进行数次扫描，所以如数据库的规模较大，那么就会增加扫描成本，在扫描的过程中耗费大量的时间，因此算法具有一定的局限性，为应对这一问题，一些学者基于该算法提出了多种改进算法。

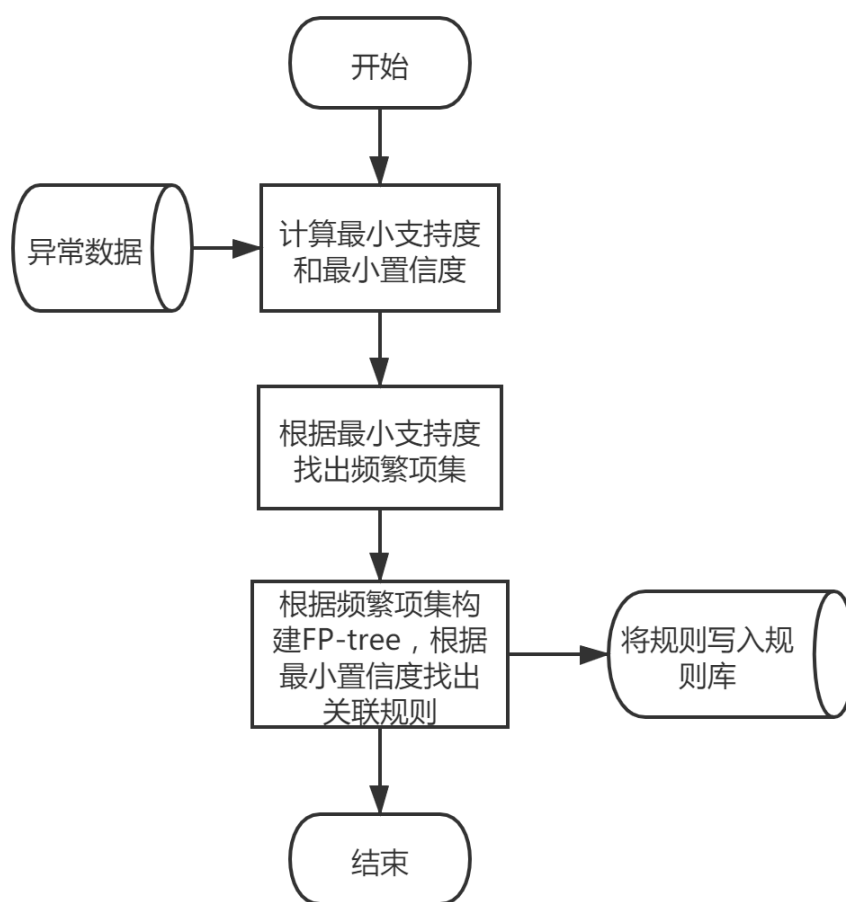


图 2-2 Apriori 算法基本流程

(2) FP-growth 算法理论

J.Han 等学者基于 Apriori 算法，提出了 FP-growth 算法，该算法基于分支策略可压缩符合最小支持度要求的事物获得频繁模式树^[48]，保留相同事物的关联，基于表中所有频繁 1 项集确定频繁模式树的条件，对平面项集进行挖掘，直至挖掘完全部条件频繁模式树。

(3) 其他算法说明

除 FP-growth 以外，也有很多学者基于经典算法提出了其他的算法，比如 1995 年 Park 等学者研究出了散列算法，Savasere 等学者提出了一种数据集划分算法^[49]，也就是 Partition 算法。此外 Brin 等学者又研究出了 DIC 算法^[50]，在基于 DIC 算法进行深入研究后，Mannila 也提出了相关的算法。现阶段分层算法主要有 AprioriHybrid、AprioriTid 以及 Apriori 法等。绝大部分增量更新算法的基础均为 Apriori 法^[51]，比如 PIUA、IUA 以及 FUP 算法等。而并行算法主要包括 CAD、DD、CD、PDM、FDM 以及 DMA 等算法。

总体来讲，现阶段应用较为广泛的关联规则算法主要有多层与多维关联规则算法，基于加权支持度与约束的算法也在一些领域有所应用。

2.3 基于内容的推荐算法

2.3.1 基于标注内容的推荐算法

该算法是指以标签化的方式处理音乐，通过用户或是音乐发行者以标签的方式对音乐进行描述，可将此类描述信息作为确定音乐相似程度的依据，比如风格、演唱者、乐器及所处的年代等。

在使用该算法的过程中，首先需要基于平台用户注册过程中提供的基本信息或是以调查的方式了解用户的偏好与兴趣，分析各类关键词对用户的影响权重，接下来对所有音乐进行描述，确定有关于某一音乐的关键词。接下来以向量化的方式分别处理音乐与用户的描述文件，计算前者与后者的相似性，从而为音乐推荐提供依据。可以通过多种方式计算相似度，比如相关系数以及关联系数等。现阶段应用较为广泛的为余弦相似度，如果将音乐向量与用户向量分别记为 \mathbf{a} 与 \mathbf{b} ，那么相似度具体计算方法如下所示。

$$\cos(\mathbf{a}, \mathbf{b}) = \frac{\mathbf{a} \cdot \mathbf{b}}{|\mathbf{a}| |\mathbf{b}|} \quad (2-1)$$

2.3.2 基于音乐特征的推荐算法

与普通商品相比，音乐具有一定的特殊性，所以有学者提出了根据特征推荐音乐的算法，此类算法与根据标注的算法相比^[37]，其通过音乐特征计算相似度。描述音乐的文件主要为音频信号的各种参数，在通过转换获得特征向量可实现数据库中音乐与用户偏好音乐相似度的计算。确定数据库中各音乐与用户偏好音乐的相似度后，可以根据相似度的数值为用户推荐相关的音乐。

对该算法来讲，选择底层参数是最重要的内容，现阶段频率倒谱系数在这一方面的应用较多，可以通过如下方式计算该系数：

一，以分帧、加窗的方式处理音频信号，接下来通过转换获得频率信号，获得如下所示的频谱。

$$X(k) = \sum_{n=0}^{N-1} x(n)e^{-j\frac{2\pi n k}{N}}, \quad 0 \leq k \leq N-1 \quad (2-2)$$

上式中， N 、 x 、 $x(n)$ 分别代表的是傅利叶变换点数、信号强度与输入信号。

二，针对能量谱进行计算，将能量谱通过一组 Mel 尺度的三角滤波器组，并假定三角滤波器组中滤波器的数量为 $M(M=100)$ ， $M=1,2,3,...M$ 为中心频率。

对所有滤波器组能量对数的输出进行计算，可以得到

$$S(m) = \ln \left(\sum_{k=1}^{N-1} |X(k)|^2 H_m(k) \right) \quad 0 \leq m \leq M-1 \quad (2-3)$$

上式中， $H_m(k)$ 代表的是频率响应。

最后，通过离散余弦变换获得频率倒谱系数。

$$C(n) = \sum_{m=1}^{N-1} S(m) \cos(n\pi(m - 0.5m)), \quad n=1,2,...,L \quad (2-4)$$

上式中， n 代表是的阶数，在式(2-3)中代入(2-2)所得的数据可以对离散余弦变换进行计算，获得 L 阶参数。。

带宽、频率中心以及零率等也属于应用较为广泛的底层音频特征。

2.4 基于协同过滤的推荐算法

2.4.1 基于用户的协同过滤推荐算法

该算法需要基于用户对音乐的打分确定各用户群的相似度^[52]，从而确定最近邻，接下来根据最近邻进行预测，分析目标音乐可以获得用户的评分，根据评分结果为用户推荐相关音乐，算法的具体过程如图 2-2 所示：

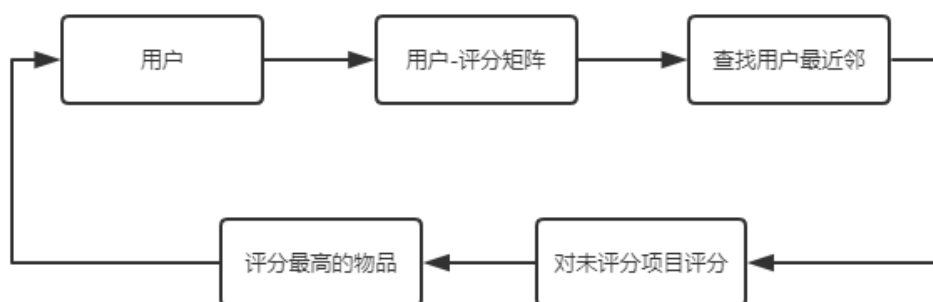


图 2-2 基于用户的协同过滤算法

可通过表 2-1 对推荐流程进行简单的说明，该表中的用户 B、C 可以定义为相似用户，在下表情形下，可为用户 C 推荐音乐 4。

表 2-1 用户音乐标签

	音乐 1	音乐 2	音乐 3	音乐 4
用户 A			○	
用户 B	○	○		○
用户 C	○	○		

基于协同过滤的算法包括下述几个重要步骤：

(1) 描述用户-评分矩阵

处理原始数据后或获得阶数为 $m \times n$ 的矩阵，矩阵的行与列分别代表的是用户与音乐，用户的评分为 r_{ui} ，具体如下。

$$R = \begin{bmatrix} r_{11} & r_{12} & \cdots \\ r_{21} & r_{22} & \cdots \\ \cdots & \cdots & \cdots \end{bmatrix} \quad (2-5)$$

(2) 相似度计算

对各用户的相似度进行计算，确定最近邻，该步骤的主要目的是确定最近邻组合，一般选择 k-近邻法进行分析， $N(b) = \{b_1, b_2, b_3, \dots, b_k\}$ 为最近邻集合，可基于 $\text{sim}(b_m, b)$ 对用户进行排列，通常该系数的取值范围为 $[-1, 1]$ ，如系数与 1 较为接近，那么可知 b 和 b_m 较为接近，预设 k 的数值后可判断最近邻集的数量。

确定最近邻集合时要计算相似度，一般采用余弦相似性或是 Pearson 法等。

通过 Pearson 法可以确定某直线与两组数据的拟合程度，在推荐系统中应用 Pearson 系数可以确定用户间的相似度，具体计算方法如下。

$$\text{sim}(i, j) = \frac{n \sum_{u \in U} r_{ui} r_{uj} - \sum_{u \in U} r_{ui} \sum_{u \in U} r_{uj}}{\sqrt{n \sum_{u \in U} r_{ui}^2 - (\sum_{u \in U} r_{ui})^2} \sqrt{n \sum_{u \in U} r_{uj}^2 - (\sum_{u \in U} r_{uj})^2}} \quad (2-6)$$

上式中，用户集合为 U ， $[-1, 1]$ 为 $\text{sim}(i, j)$ 的取值，如该数值较高，那么说明相似度也相对较高。

通过余弦相似度可以对向量夹角进行计算，对于音乐推荐系统而言，可以 n 维空间向量的方式看待用户评分矩阵，可通过计算向量夹角确定相似度，具体计算方法为：

$$\text{sim}(i, j) = \frac{\sum_{u \in U} r_{ui} r_{uj}}{\sqrt{\sum_{u \in U} r_{ui}^2} \sqrt{\sum_{u \in U} r_{uj}^2}} \quad (2-7)$$

对协同过滤算法来讲，确定相似度是最重要的内容，相似度的计算结果对音乐推荐的效率与准确性具有决定性的影响。

(3) 推荐结果

在计算相似度确定最近邻集合之后，需要基于最近邻分析所有音乐的分数。对用户评分进行预测的过程中，通常采用两种方法，一种是基于最近邻进行评分，

对用户某个音乐的评分进行预测^[63]，另一种是基于最近邻对用户音乐的评分进行排序，据排序结果将相关音乐推荐给用户。

2.4.2 基于物品的协同过滤推荐算法

在移动网络用户规模不断增加的过程中，各音乐平台的用户数量也随之有了大幅的增长，同时数据库中的音乐数量也越来越多，因此用户评分矩阵在持续增大，所以就增加了相似度的计算难度^[53]。在空间与时间具有较大复杂性的情况下，基于用户的协同算法产生了更明显的局限，所以有研究人员提出了基于物品的协同方法，该算法可以通过对物品间的相似度进行计算，为用户推送相似的物品。在音乐推荐领域中可以将这种思想理解为，如某一用户对梁静茹的作品存在兴趣，系统就可能将范玮琪的作品推送给该用户，推送的依据是两位歌手较为相似。除此之外，系统有可能会将张信哲的作品推荐给该用户，推送的依据是两位歌手的作品主要为情歌。系统具体采用何种方式为用户推送音乐，主要取决于各系统计算相似度的方式。

不过现阶段在应用该方法进行音乐推荐的过程中，相关平台并未结合自身的特点改进该算法，仅在音乐推荐系统中套用了该方法^[58]。该方法的步骤主要有三个，首先需要通过计算构建矩阵，接下来对各音乐的相似度进行计算，此时一般会采用余弦相似度以及相关系数等方法确定最近邻^[54]。最后，确定评分排序，根据排序情况为用户推荐相关的音乐。

2.5 本章小结

本章主要介绍了数据挖掘技术的研究现状和基本原理，阐述了基于关联规则的推荐算法，介绍了基于标注内容的推荐算法和基于音乐特征的推荐算法，以及协同过滤推荐算法中的基于用户的协同过滤算法和基于物品的协同过滤推荐算法，并对以上算法的实现过程及其原理做了简要说明。

第3章 关联规则推荐模型

3.1 关联规则推荐的适用性分析

3.1.1 关联规则推荐的优势

在使用 Apriori 算法时,首先需要扫描数次数据库,此时会由于记录扫描数据而形成大量的数据,因此消耗的时间较多,同时也会占据较大的空间。除此之外,该算法在对较长频繁模式进行挖掘的过程中,通常无法表现良好的性能,所以学者们对该算法进行了改进,基于该算法又提出了多种其他算法,比如 Jiwei Han 研究出的关联规则算法 FP-growth^[58]。

FP-growth 与 Apriori 法相比,其可以有效的节约空间与时间,该算法在构造频繁项集的过程中采用了 FP-tree,仅需要扫描两次数据库,因此并不会形成大规模的候选集,极大的降低了搜索空间^[59],与 Apriori 法相比在空间以及时间效率方面均有了明显的提升。

音乐作为一种重要的艺术,人类对其存在着强烈的创作欲,因此在时代不断发展的过程中产生了海量的音乐数据^[55],所以需要一种可以针对海量数据进行处理的算法才可以为用户提供有效的音乐推荐。音乐推荐属于一种关联推荐,通过挖掘关联规则可以由用户的历史活动确定用户的偏好,所以在音乐推荐领域关联规则算法 FP-growth 也具有一定的适用性。

3.1.2 关联规则推荐算法的内容

通过 FP-growth 算法可以压缩最小支持度要求的事物,获得 FP-tree 并保留相同事物的关系,具体过程为:

- a. 针对数据库进行整体扫描确定频繁 1 项集并进行降序排列,构造频繁模式树,基于频繁 1 项级的顺序,重新排列数据库中的各项事务。
- b. 将根节点定为 Null,建立频繁模式树,可以获得数个条件数据库,所有数据库均存在一个与之具有关联的频繁项集,分别对所有数据库进行挖掘。
- c. 频繁模式树构造完成后,可基于条件挖掘所有的评分项集,找出强关联规则。

在推荐系统中应用强关联规则，可以为音乐的推荐提供依据，将最适合用户的音乐推送给用户。

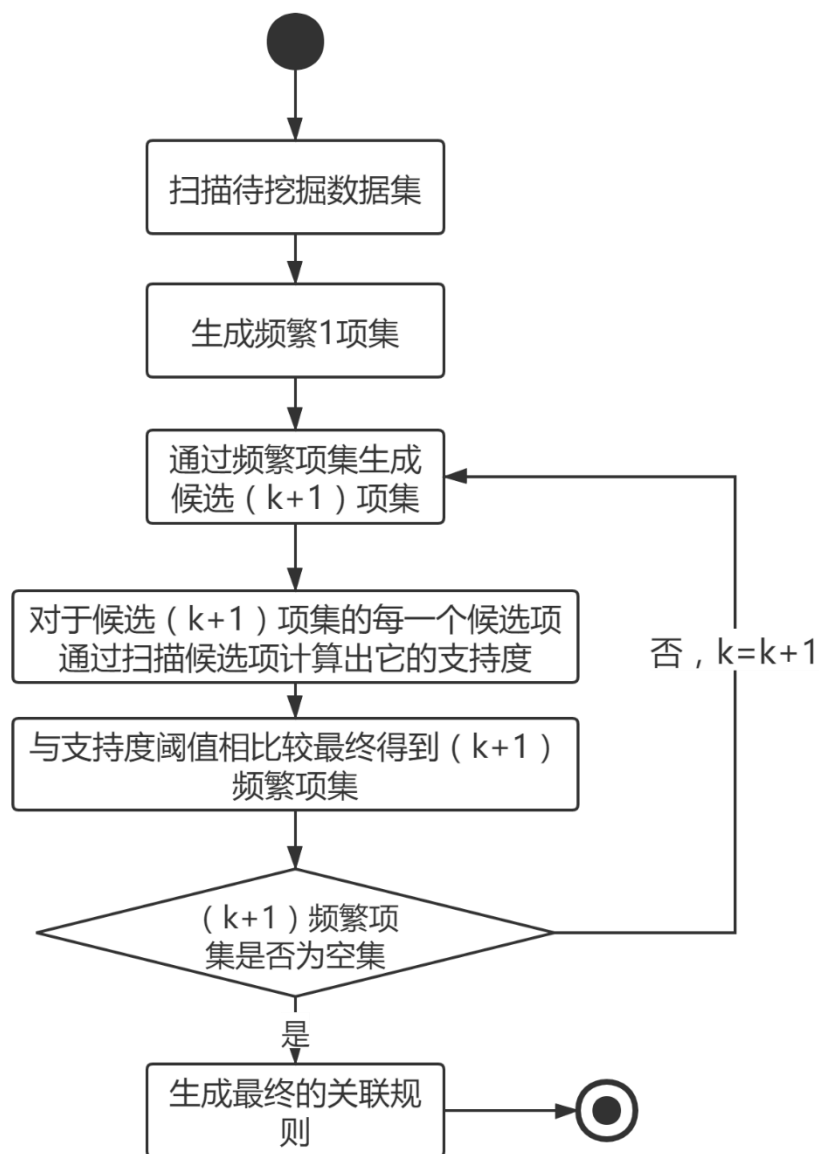


图 3-1 关联规则算法基本流程

3.1.3 关联规则推荐算法的应用

现阶段已有多种领域应用了关联规则算法，比如农业、通讯、营销、医学、生物以及金融领域等^[62]，确定关联规则是数据挖掘中的核心工作，确定关联规则可以为多领域的研究提供支持，比如在医学领域可以通过数据挖掘确定肿瘤数据库中各种属性的联系。周晓辉等在针对药物在临床中的应用进行研究时，通过关

联规则算法研究出合理的治疗方案，并基于该算法科学的总结归纳了临床经验。

关联规则算法在营销领域也得到了广泛的应用，例如，该算法现阶段已被运用于关联规则挖掘以及机器学习等领域中^[56]。刘正华针对该方法进行了改进，保留了项头表，以优先队列的方式代替了数结构中较为复杂的部分，并在电商领域应用该算法，分析了商品推送的具体过程，基于其改进后的算法设计并实现了一种为用户推荐相关商品的系统，该系统可以为电商企业的运营提供支持，增加各类商品的销量，推动我国电商的发展。杨丽华等学者基于该算法设计并实现了一种销售决策系统，该系统的应用领域为卷烟商品，之所以可以在该领域中应用关联规则算法，主要原因是烟草销售过程中会产生大量的数据^[60]，而该算法可以为挖掘产品的销售关联提供支持，为制定销售策略提供依据，使营销人员获得重要的参考信息，从而促进卷烟的销售。

关联规则算法在医药领域也有所应用，比如董辉在研究的过程中，结合要领域的特性改进优化了关联规则算法，通过改变后的算法对药方剂的规律进行了挖掘，发现了各类方剂中隐藏的珍贵知识，不但促进了中医学信息化建设的发展，同时也为我国中医的国际化发展提供了技术的支撑。在中药方剂挖掘方面，关联规则算法的应用存在一定的必要，并且在初步应用的过程中已做出了一定的贡献。

在音乐推荐方面，关联规则算法的研究相对较少，主要原因是现阶段研究人员并未在音乐推荐领域发挥该算法多样性与精确性的优势。通过关联规则算法较为难以抽取音乐领域关联规则，噪声数据对知识发现造成了极大的干扰^[57]，不过在音乐推荐的领域中应用该算法存在一定的适用性，可以极大的提高音乐推荐的准确性与科学性。基于此，本文针对音乐推荐中关联规则算法的应用进行了研究，具体研究过程为：

- 1) 对文本数据进行预处理，获得十或是二进制的的数据；
- 2) 因为存在海量的用户数据，如针对用户的所有数据挖掘关联规则，那么将会为研究造成极大的不便，因此本文仅选择一段期间内用户的行为进行记录，从而降低数据转换所消耗的时间以及空间^[61]，并对挖掘后所获得的用户兴趣进行记录，作为音乐推荐的参考。

以上述两种方式针对数据进行处理,可以更为简单的抽取关联规则,在音乐推荐领域应用,本课题的研究有利于丰富相关研究,处理音乐推荐新颖性与多样性的问题,在提高用户满意度的同时,促进各音乐平台的发展。

3.2 构建关联规则推荐模型

3.2.1 数据集获取

现阶段我国已出现了多种音乐网站,其中 QQ 音乐可以为用户提供收听、添加标签以及发表评论的服务,此外 QQ 音乐也对用户欣赏歌曲的次数以及歌曲的名称进行了记录。本文收集了 QQ 音乐中的部分数据,抽取的收听过的歌曲数量在 9 首以上的用户进行研究,抽取的歌曲与用户的数量分别为 43,764 与 4126,获得的用户信息为 26,419 个,此外还采集了用户的好友关系,具体数量为 6805 个。未完成上述工作后,本文基于所抽取的语料进行了算法的研究。

对语料集当中的歌曲进行划分,具体如下图所示,其中 A、B、C 分别代表的是训练集、测试集与用户未收听过的歌曲集,A 与 B 分别为历史记录 80% 与 20%。



图 3-2 语料集划分

3.2.2 构建数据集树结构

关联规则算法通过构造树结构对采集到的数据记录进行压缩,因此挖掘频繁项集只需要扫描数据库记录两次,而不需要在采集过程中生成候选集,在这构造数据集机构上表现出非常高的效率,以下面表格中所展示的音乐数据集为例,每

个字母代表不同的歌曲。

表 3-1 音乐数据集示例

TID	Items
T1	{A,B}
T2	{B,C,D,E}
T3	{A,C,D,F}
T4	{B,A,C,D}
T5	{B,A,C,E}

接下来需要构造 FP-Tree，FP-tree 是一种树结构，在其中需要定义每个项集的唯一标识 id，当前节点的孩子结点和父结点，以及当前结点在树中的下一结点标识。初始化当前项集所出现的次数。所以树结构的定义如下：

```
public class FpItemNode {  
    String id;  
    ChildList;  
    FpItemNode parent;  
  
    FpItemNode next;  
    long count;  
}
```

树中的每个结点就是数据集中的每一个项集，在此模型构建的过程中，假设当前的最小绝对支持度是 3，接下来详细介绍 FP-Tree 树结构的构造过程。

1) 首先需要扫描数据集中的所有记录，第一次扫描过程结束后，频繁一级项集就会从这个过程中产生，然后将生成的项集从出现的次数由大到小进行降序

排序，然后生成有序的一级频繁项集，如表 3-2 所示：

表 3-2 频繁一级项集

Item	Count
A	4
B	4
C	4
D	3

从上表中可以看出，E 和 F 并没有出现在第一次的扫描过程结果当中，原因是因为模型中定义的最小绝对支持度为 3，E 在数据集中出现了 2 次，而 F 在数据集中出现了 1 次，这两个项集的出现次数都小于定义的最小绝对支持度，因此 E 和 F 不属于频繁项集，根据 Apriori 算法所提供的定理可以知道，当一个项集不是频繁项集，那么它的超集也一定不是频繁项集，所以在接下来的扫描过程中将不会再考虑非频繁项集 E 和 F。

接下来再次扫描数据集记录，对每条记录中出现在 1 过程中产生的表中的项，按照出现在表中的数据进行排序，初始化时，新建一个标记为 null 的结点，并且把它作为根结点。

第一条记录：{A,B}，按照 1 过程生成的表过滤排序后得到的依然为{A,B}，此时新建一个结点，id 设置为{A}，并将其插入到根节点 null 的下面，将它的 count 设置为 1，然后再新建一个结点，id 设置为{B}，将其插入到{A}结点下面，生成的树结构如图所示。

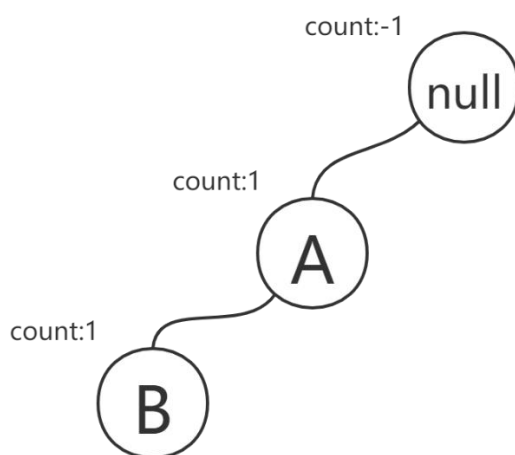


图 3-3 第一条记录树结构

第二条记录: {B,C,D,E}, 经过过滤并且排序之后为: {B,C,D}, 在树结构中, 根节点并没有 id 为 {B} 的子结点, 所以在当前树结构中新建一个 {B} 结点作为根结点的另一个孩子结点, 然后新建 {C} 结点插入到新建的 {B} 结点的下面, 再新建 {D} 结点插入到刚才的 {C} 结点下, 所以扫描第二条记录后生成的树结构如下图所示:

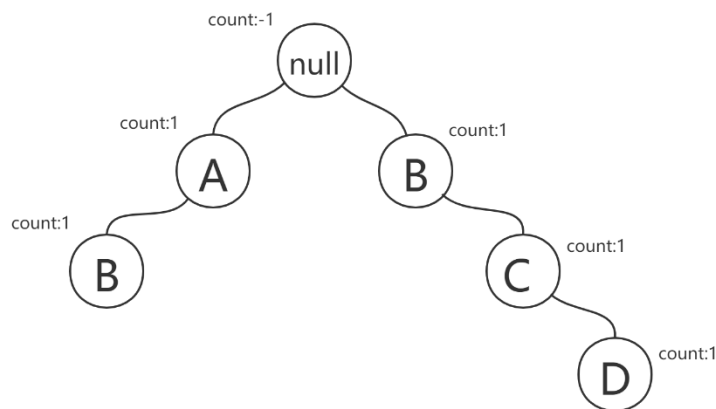


图 3-4 第二条记录树结构

重复以上步骤, 直至扫描完所有的记录, 就构造完成了一颗完整的 FP-Tree, 在树结构中, 每一条路径就代表一个项集, 许多的项集且都有公共项, 出现次数越多的项越有可能是公共项, 因为对出现的项按照出现次数由大到小进行排序, 就可以实现有效的压缩存储, 在存储的过程中, 我们需要一个表头和对每一个 id

相同的结点做一个线索，这样在后面的数据挖掘过程中方便使用，其中线索的构造过程就是在树建立的时候插入进去的，最后生成的 FP-tree 如下图所示。

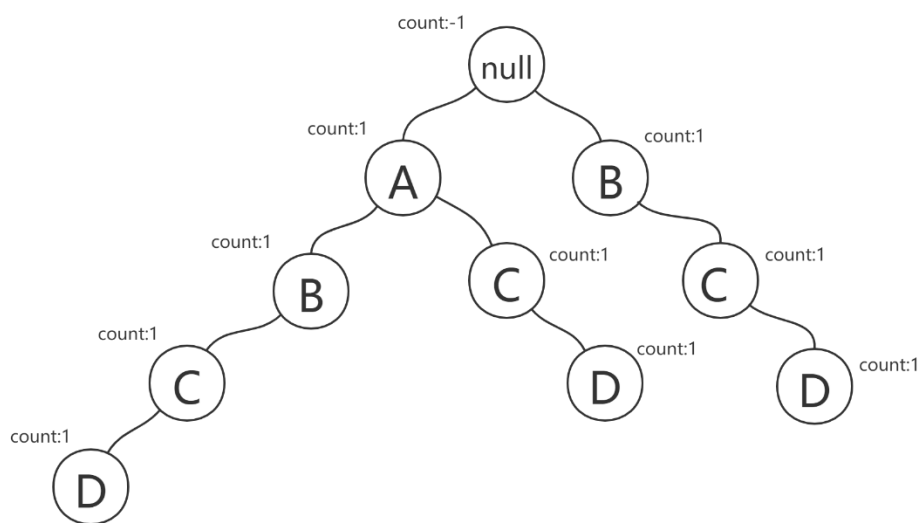


图 3-5 最终的 FP-Tree 树结构

3.2.3 挖掘频繁项集

FP-Tree 树结构建立好之后，就可以通过 FP-growth 算法进行频繁项集的挖掘，挖掘的过程从表头 header 的最后一项开始。此处的最后一项就是通过 {D} 开始挖掘，根据 {D} 的线索链找到所有的 {D} 结点，然后通过 {D} 结点找到其所有的分支：{A,B,C,D:1}，{A,C,D:1}，{B,C,D:1}，其中“1”表示出现了 1 次，根据前缀路径，最后可以生成一个条件树，建立的方式和之前的构造方式一样。这里的数据集为：

表 3-3 频繁项集数据集

TID	Items
T1	{A,B,C}
T2	{A,C}
T3	{B,C}

在绝对支持度依然为 3 的前提下，我们得到的新的树结构为：

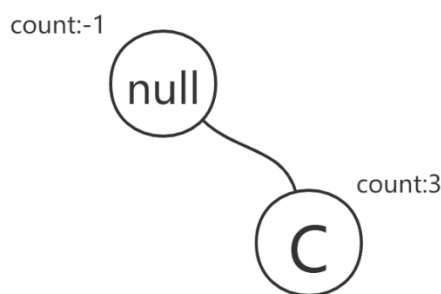


图 3-6 条件树结构

在条件树构造完成之后，需要对条件树进行挖掘，当构造好的条件树只有一条路径时，路径上的所有组合即是条件频繁集，假如 $\{D\}$ 的条件频繁集为 $\{S1, S2, S3\}$ ，那么对应的 $\{D\}$ 的频繁集就是 $\{S1+\{D\}, S2+\{D\}, S3+\{D\}\}$ ，可以看出，只要是 $\{D\}$ 的频繁集都会有相同的后缀 $\{D\}$ ，当前的条件频繁集为 $\{\{\}, \{C\}\}$ ，于是 $\{D\}$ 的频繁集为 $\{\{D\}, \{C, D\}\}$ 。

在上述过程结束后，接下来需要寻找 header 表头中的倒数第二项 $\{C\}$ 的频繁集，同上可以得到 $\{C\}$ 的前缀路径为： $\{B:1\}$ ， $\{A:1\}$ ， $\{A,B:2\}$ ，生成的 FP-Tree 的数据集为：

表 3-4 $\{C\}$ 的频繁项集

TID	Items
T1	$\{B\}$
T2	$\{A\}$
T3	$\{A,B\}$
T4	$\{A,B\}$

根据以上数据集所构造的条件 FP-Tree 树结构为：

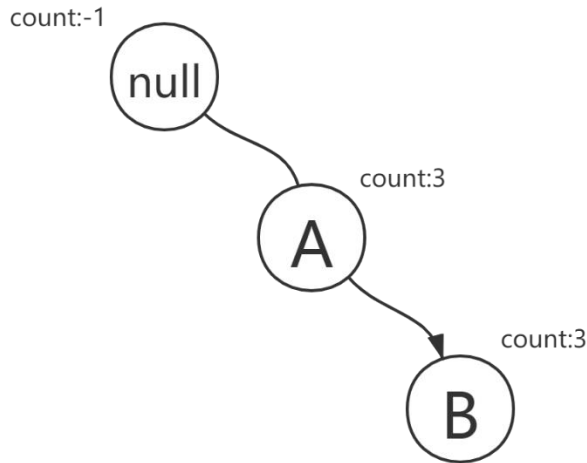


图 3-7 根据条件树生成的 FP-tree 树结构

这个树结构可以看出已经是单一路径,所以路径上涉及到的所有组合即为条件频繁集 $\{\{\}, \{A\}, \{B\}, \{A,B\}\}$, 加上 $\{C\}$ 之后, 又得到一组新的频繁项集 $\{\{C\}, \{A,C\}, \{B,C\}, \{A,B,C\}\}$, 在这里可以看出, 这组频繁项集中一定包含相同的后缀 $\{C\}$, 并不包含 $\{D\}$, 所以当前这一组频繁项集不会与上一组相重复。

从对获取到的 QQ 音乐数据集来看, 数据量较大, 相比于 Apriori 算法, FP-growth 算法的平均效率远远领先。

3.3 选择评价标准

因为本文的目的是针对在多样性、精确性、稀疏性以及冷启动方面对三种算法的优势进行分析, 所以本文将基于这几项问题建立评价指标体系, 具体为:

(1) 多样性

如为用户推送过多相似度较高的音乐, 便可能会导致听觉疲劳, 为解决这一问题, 周涛等学者研究出了一种可以针对多样化程度进行度量的方法, 具体如下:

$$H_{ut}(L) = 1 - \frac{Q_{ut}(L)}{L} \quad (3-1)$$

上式中, 推荐表表长为 L , 系统为 u 与 t 用户推送的相同项目的数量为 $Q_{ut}(L)$ 。

(2) 新颖性

尽管推荐系统为用户推送了近期较为流行的歌曲，具有较高的准确度，不过也可能在新颖性上有所欠缺，因此有时也可能无法得用户的认可，可通过如下公式计算新颖性：

$$U_a = \log_2 \frac{M}{K_a} \quad (3-2)$$

上式中， a 、 K_a 、 M 、 U_a 分别代表的是项目、用户对 a 的使用数次、项目数与 a 的自信息，该方法可以对自信息的平均值进行计算从而度量新颖性。

(3) 覆盖率

推荐项目在总体中的占比就是覆盖率，分析该指标可以确定长尾项的挖掘能力，其具体的计算方法为：

$$COV(L) = \frac{N_d(L)}{N} \quad (3-3)$$

上式中， $N_d(L)$ 、 N 与 L 分别代表推送的总数、项目的总数与推送列表长度。

(4) 准确度

现阶段，应用较为广泛的度量准确度的方法有两种，分别为对评分预测与 TopN 列表准确度进行度量的方法，因为本文主要是针对音乐推荐列表进行研究，所以选择后者进行介绍。该方法的具体公式为：

$$P@N = \frac{|Items_{rel}@N|}{N} \quad (3-4)$$

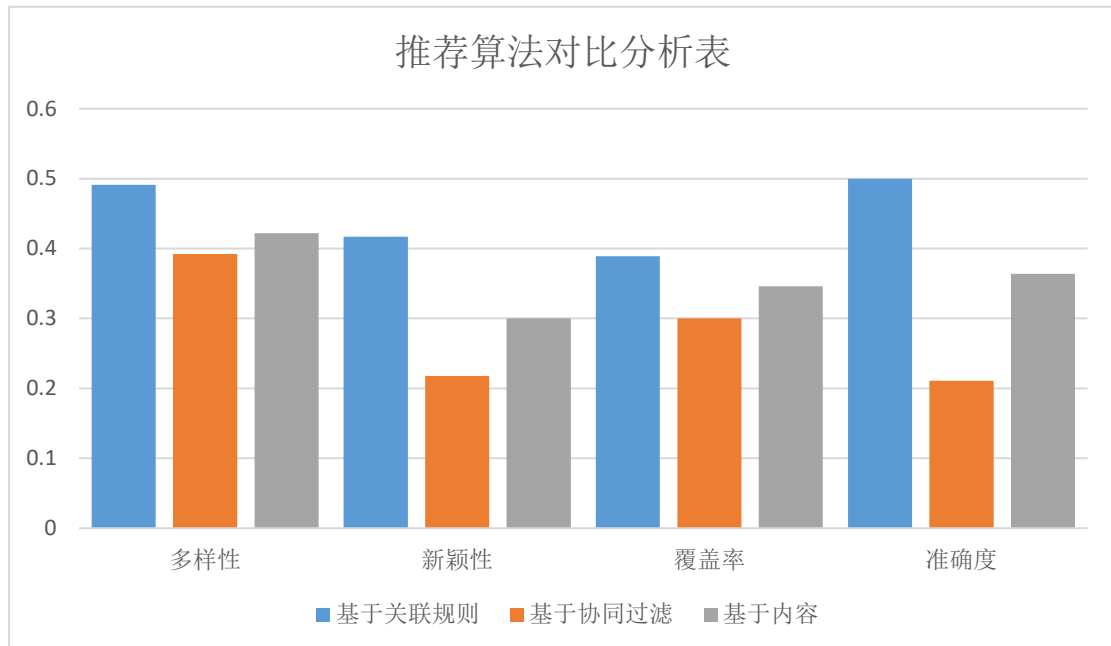
上式中的分子代表的是用户对前 N 项推荐项目中感兴趣的集合。

评价推荐列表准确度，可以确定算法是否可以为用户提供精准的音乐推荐，从而通过对比揭示算法的优势。

在针对算法进行评价的过程中，要考虑多种不同的因素，同时，各算法的优势也存在一定的差异。现阶段，学术界在针对某一算法进行评价的过程中通常选择一种或是数种因素开展评价工作。为了更好的揭示关联规则算法的优势，本文选择了以上评价指标对三种算法进行分析。

3.4 实验对比分析

表 3-5 推荐算法对比分析表



分析上图能够发现，在多样性、覆盖率、新颖性以及准确度等方面，关联规则算法均具有明显的优势，尤其是关联规则算法的准确度明显高于另外两种算法。出现这种情况的原因是，该算法为用户推荐音乐的过程中，并非仅仅关注用户的历史收听数据，同时还结合了用户近期的兴趣为其推送相关的音乐，所推荐的音乐并非用户收听列表中的内容，而是由数据库中挑选的歌曲，因此可以极大的提高推荐的准确度，并使推荐的内容更为丰富与新颖。

关联规则算法的步骤主要有：

第一，基于用户的历史行为建立数据库，一般来讲，数据库中的所有内容具有大致相同的重要程度，如存在特殊的情况，也可以根据用户欣赏某一类歌曲的次数对权重进行调整，从而使推荐的精确度得到加强；

第二，基于关联规则算法确定各项目的关联。

最后，对用户的历史数据与关联规则进行匹配，获得推荐列表，因为平台的用户一般在针对曲风或是心情进行描述时采用了大致相同的描述信息，如可以考虑用户的历史记录确定用户偏好的音乐与数据库中其他音乐的属性间的关联，那么通常可以更准确的为用户推荐极感兴趣的音乐。

关联规则算法的优点主要有：

(1) 对数据源并无特殊的要求，仅需要了解用户的行为即可进行关联规则的挖掘。

(2) 可以针对用户的行为进行推测，确定潜在的兴趣，从而将相关的音乐推荐给用户，提高推荐列表的多样性。

(3) 无需确定音乐的内容属性，因此，基于该算法推荐的音乐类型更为丰富。

因此，本文的音乐平台的实现将采用基于关联规则推荐算法中的 FP-growth 算法。

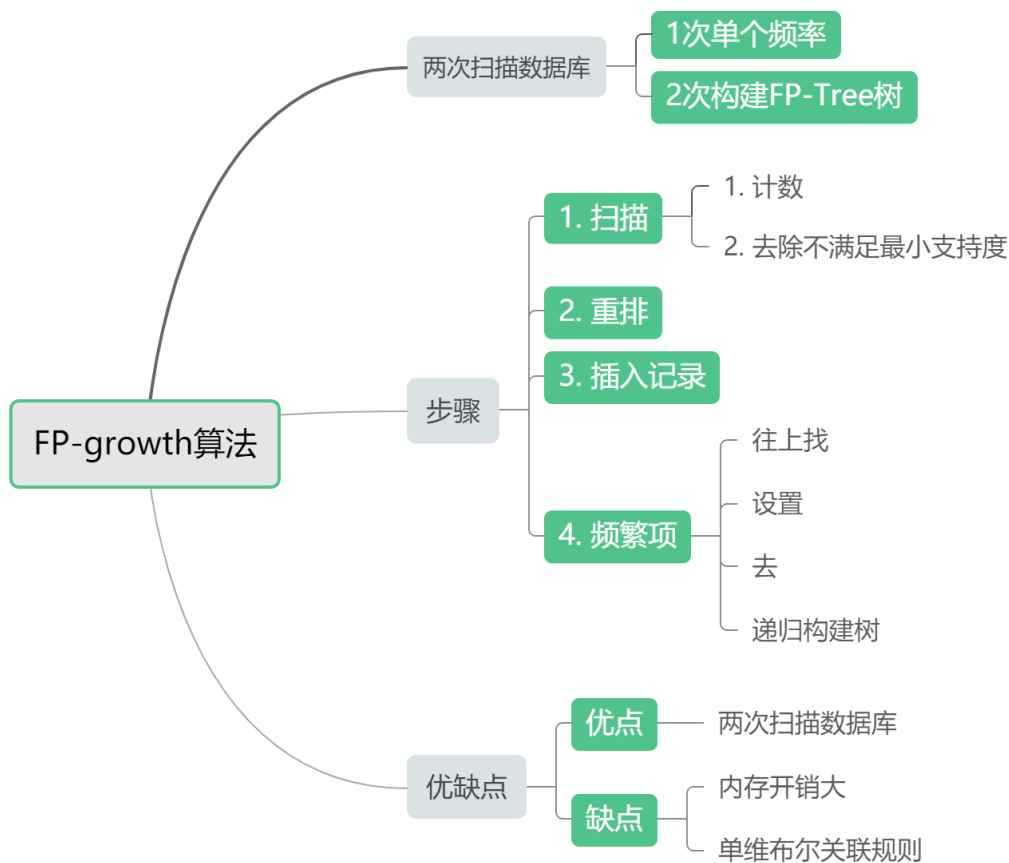


图 3-8 FP-growth 算法

3.5 本章小结

本章对关联规则算法在音乐推荐领域的适用性做了分析说明,介绍了关联规则算法的基本流程,并对其在各个领域的应用做了简要说明。收集了 QQ 音乐的部分数据并抽取收听歌曲数量到达 9 首以上的用户进行分析研究,划分出构建模型所需的数据集。详细介绍了 FP-tree 树结构的构造过程。通过多样性、新颖性、覆盖率、准确度等评价标准,分析对比了基于关联规则的推荐算法与基于协同过滤的推荐算法和基于内容的推荐算法的结果,说明了关联规则推荐算法在音乐推荐上的优势。

第4章 音乐推荐平台设计

4.1 平台开发相关技术

4.1.1 B/S 项目架构

B/S 三层架构的应用极为广泛，该架构主要由数据库、应用与浏览器构成，其中服务器所需处理的事物逻辑相对较多，而浏览器所需处理的事务逻辑相对较少，在该架构下开发的应用无需进行下载配置，仅需要安装浏览器即可运行应用。浏览器客户端需要负责视图的展示，服务器需要负责处理各类事务，因此能够降低浏览器的负担，使浏览器相对较为简单，由于浏览器仅需处理较少的事务逻辑，因此也被称之为瘦客户端。

C/S 架构属于一种典型的二层架构，其主要由服务器与客户端构成，需要在电脑中安装的应用。其服务器主要有两种类型，分别为 Socket 与数据库服务器，对于前者来讲，应用需要通过客户端与 Socket 实现数据的交互，而后者可以在连接数据库后使用数据库中的数据。C/S 与 B/S 相比，其客户端更为复杂，因此叫做胖客户端。主要原因是，C/S 客户端所需处理的事物逻辑相对较多，同时还需要负责视图的展示，为了实现数据的持久化，数据库需要与客户端进行数据的交互，从而达到项目的要求。

图 4-1 为 B/S 的结构示意图：

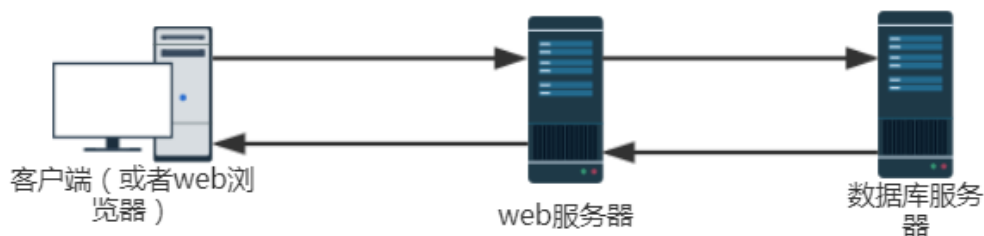


图 4-1 B/S 架构示意

C/S 为 B/S 的前身，前者属于改进拓展后者而形成的一种产物，B/S 与 C/S 相比具有更多的优势，具体为：

1. 具有更高的安全性。因为 C/S 结构下的应用具有数据分散的特点，所以

存在黑客、病毒、地震以及火灾等方面的风险。此外对于一些规模较大的应用来讲，由于基于 C/S 结构开发的应用需要安装的服务器数量较多，同时客服服务器需要同步数据，因此也会影响到数据的安全性。数据的安全性对于规模较大的应用而言极为重要，基于 C/S 开发出的应用在安全性方面并不理想，所以存在明显的局限性。而基于 B/S 开发的应用可以在总部服务器中储存所有数据，而数据库连接信息以及其他业务数据均无需保存于客户端中，因此不存在数据同步的问题，可以很好的避免 C/S 结构应用中的问题，从而提高数据的安全性。

2. 具有良好的一致性。基于 C/S 开发的应用需要在各地设置服务器，并对各服务器的数据进行同步，完成数据的同步后，中央服务器才可以获得各地的信息，在同步的过程中如发送网络故障，那么就可能使部分服务器中的数据无法同步，或是数据同步的时间不一致，导致数据无法为决策提供支撑。而基于 B/S 结构开发的应用可以集中存放各地的数据，所有客户端中的数据均可以在中央数据库中直接存储，因此并不会引发一致性的问题，使集团总服务器可以及时的了解相关数据，从而做出有效的决策。

3. B/S 结构的应用具有更好的实时性。对于一些集团应用来讲，基于 C/S 开发的应用仅可以为总部提供事后数据，而总部无法查询实时数据。而基于 B/S 结构研发出的应用，由于中央数据库可以直接存入所有客户端中的数据，因此能够使总部及时了解所有业务的情况，使总部快速做出相应的决策，避免企业遭受损失。

4. 溯源性的比较。基于 C/S 开发的应用，为降低数据的通讯量一般只将中间报表上传到总部服务器，因此总部无法针对各门店、分公司等分支机构的原始单据进行查询。而因为基于 B/S 开发的应用可以集中存放所有数据，所以集团总部能够更好的针对各地的原始单据进行查询。

5. 服务器响应比较。在时代发展的过程中，企业的业务模式以及流程等也会发生一定的变化，所以企业需要根据内外部环境的变化及时调整自身的业务模式以及流程。而当企业业务模式或者流程发生改变后，企业原有的软件也需进行升级或是维护。因为 C/S 的结构分布性特点，必须要在各地分别安装相应的程

序，因此即便极小的问题，可能也需耗费大量的时间进行部署，在部署的过程中为了确保各地应用具有一致性，一般需要在更新的过程中停止所有业务，也就是常说的休克更新，在为企业造成不便的同时，也必然会使企业遭受一定的损失，所以在服务器响应时间方面基于 C/S 开发的应用存在明显的问题。但基于 B/S 开发的应用，各应用节点之间并不存在程序，更新某一地方中的软件后就可以实现所有程序的更新，因此服务响应极为迅速。

6. 应用限制方面的比较。基于 C/S 开发出的系统一般仅可以为宽带用户或是局域网用户提供相关的服务。而基于 B/S 结构开发出的应用能够在所有网络中应用，比如为拨号入网的企业提供服务，可以覆盖宽带未普及的地方。

7. 存储方面的比较。基于 C/S 开发的软件一些数据为本地的临时文件，也有一些数据源于数据库，所以其实响应时间稍微较短。而基于 B/S 结构开发的软件需要由数据库提供所有的数据，因此响应时间相对较长。

4.1.2 MVVM 开发模式

MVC 为 MVVM 的前身，在 MVC 不断发展的过程中，研究人员对其进行了改进与拓展，从而产生了 MVVM，MVVM 可以抽象化视图的行为与状态，分离业务逻辑与视图。微软公司开发出的 WPF 为人类提供了大量新的体验，比如动画、视频、音频等，不过也对软件 UI 层提出了更多的要求，比如可定制化以及系列化要求等。除此之外，WPF 也在技术层面表现出了 DataTemplate、Routed Events、Binding、Command 以及 Property 等新特性。在 WPF 与/VF 融合发展后，形成了 MVVM，其基于 MVP 框架柔和进了 WPF 的特性，具有更丰富的功能，可以满足大量复杂的需求，为各类功能的实现提供支持。

MVVM 旨在利用 WPF 中的数据绑定函数，通过从视图层中几乎删除所有 GUI 代码（代码隐藏），更好地促进视图层开发与模式其余部分的分离。不需要用户体验(UX)开发人员编写 GUI 代码，他们可以使用框架标记语言(如 XAML)，并创建到应用程序开发人员编写和维护的视图模型的数据绑定。由于可以实现角色的分离，因此研究人员可以在开发的过程中将更多的精力用于用户需求的满足，而无需耗费大量的时间与精力编辑业务逻辑，从而可以提高程序的研发效率。即

便某一程序仅由一名研发者进行开发,适当的分离模型与视图也能够提高开发效率,节约研发成本,因为基于最终用户反馈,用户界面通常在开发周期中经常发生变化,而且处于开发周期后期。

图 4-2 为 MVVM 的实现模式:

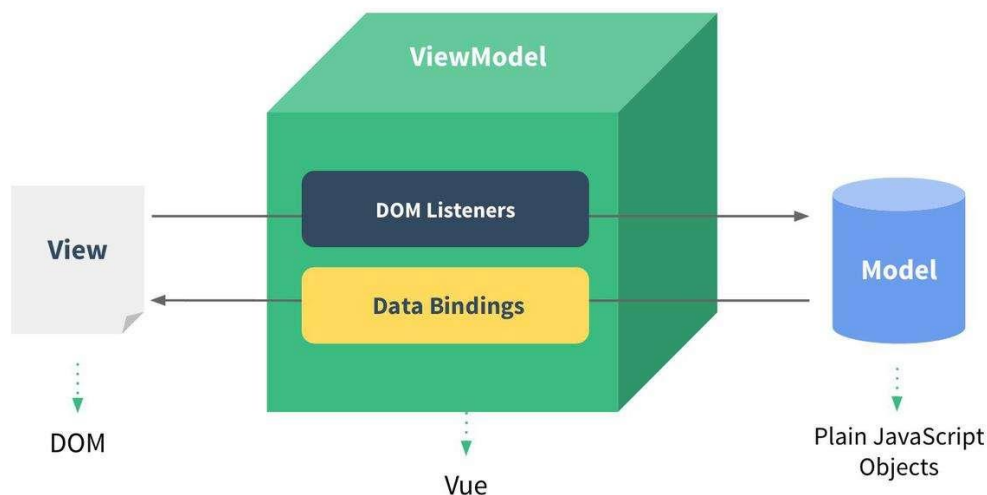


图 4-2 MVVM 的实现模式

与 MVC 模式相同, MVVM 的主要功能也是将模型与视图分离,其主要优势有如下几种,

1. 低耦合。能够实现视图的独立修改与变化,视图发生变化的情况下,可以使 Model 不随之作出调整,同样 Model 在变化后也可以不修改视图。
2. 可重用性。可以在 ViewModel 中放入部分视图逻辑,使视图重新使用该视图逻辑,所以 MVVM 模式具有良好的可重用性。
3. 独立开发。在程序开发的过程中,研发者可以在数据业务逻辑开发方面投入更多的精力,而设计者仅需负责页面的设计工作,通过 Expression Blend 能够针对界面进行高效的设计,自动生成 xaml。
4. 可测试。界面的测试相对较为困难,可基于某某 ViewModel 测试。

4.1.3 Vue.js 开发框架

在开发界面的过程中经常会应用 Vue,其属于一种渐进式框架。Vue 与其他框架相比,其可以自下而上进行设计,其核心库关注的内容仅为视图层,相对更

为简单，与既有项目或者第三方库进行整合也更为方便。除此之外，综合使用其他支持库或是工具链后，Vue 框架可以为复杂性较高的单页应用提供驱动。

单独来讲，Vue.js 是一种针对 Web 界面进行开发的前端库，属于一种轻量级工具，其主要特点有组件化与响应式编程。

所谓响应式编程，即为保持状态和视图的同步，这个在大多数前端 MV*（MVC/MVVM/MVW）框架，不管是早期的 backbone.js 还是现在 AngularJS 都对这一特性进行了实现（也称之为数据绑定），但这几者的实现方式和使用方式都不相同。相比而言，Vue.js 使用起来更为简单，也无需引入太多的新概念，对事例声明后，可以在视图上绑定 data 中的数据，对 data 中的数据进行修改后，视图可以自动更改绑定的相关数据。

Vue.js 的组件化理念和 ReactJS 异曲同工——“一切都是组件”，可以将任意封装好的代码注册成标签，例如：Vue.component('example', Example)，可以在模板中以<example></example>的形式调用。如果组件抽象得合理，这在很大程度上能减少重复开发，而且配合 Vue.js 的周边工具 vue-loader，我们可以将一个组件的 CSS、HTML 和 js 都写在一个文件里，以模块化的方式开发应用。

Vue.js 也可以配合其他工具使用，比如 vue-resource 或 vue-router 等，为路由与异步请求提供支持，这样就满足了开发单页面应用的基本条件。

在 Vue 中组件系统属于一种核心的概念，由于其较为抽象，所以可以选择可复用的、独立的、小型的组件建立规模较大的应用，大致所有类型的界面均可以通过抽象转化为组件树如图 4-3：

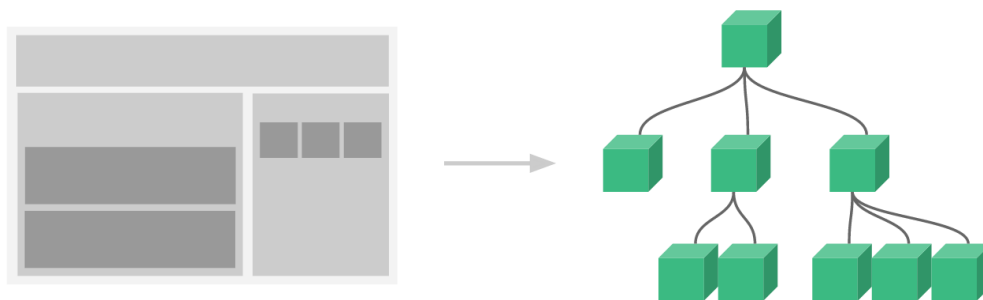


图 4-3 Vue 中的抽象组件树

4.2 音乐推荐策略

在音乐内容方面,本文实现的音乐推荐平台主要把音乐内容分为歌曲和歌手两个部分,通过对歌曲的关联分析,找出相关程度比较高的歌曲,进而找到更大的目标用户群。对曲风、年代、语言等维度的分析过滤,生成不同的排行榜单,从而找出对各类排行榜单比较敏感的用户群,作为榜单的推荐目标群。通过对歌手之间的关联分析找出存在关联的歌手,涉及的维度包括性别、风格、年龄和区域等因素。

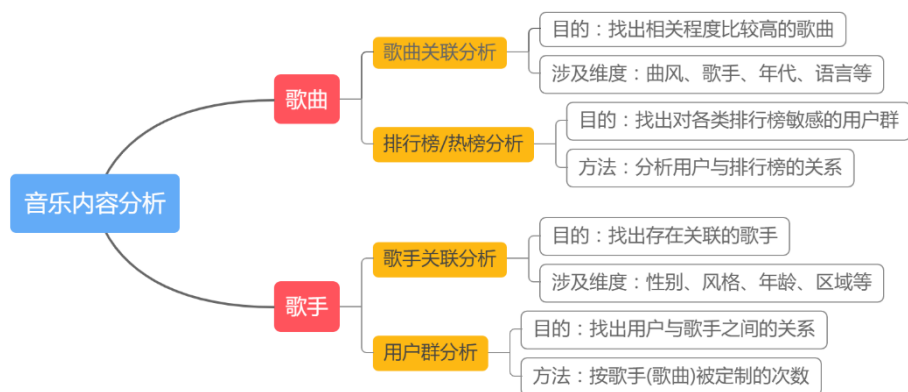


图 4-4 音乐内容分析思路

在用户群体方面,主要可以分为音乐消费人群和音乐专业人士两种。音乐消费人群主要是追求音乐的娱乐性,体验感,对于专业需求较低。这种人群更加贴近实际的应用场景。专业人士体现在对旋律、风格等知识层面的需求,在鉴赏层面上表达对音乐的需求更加强烈。所以,本文研究的音乐平台对于用户的行为信息搜集主要分为两类,一种是模糊行为数据的收集,主要表现为对榜单的敏感度,对某一歌曲或者歌手的定制次数,以及对不同维度下的歌曲或者歌手的喜爱程度。另一种是精确行为数据的收集,主要表现为用户的检索行为,对已知对象的明确需求。不同类型的用户行为数据对用户意图的反应也不一样,在推荐过程中体现出不同的价值。

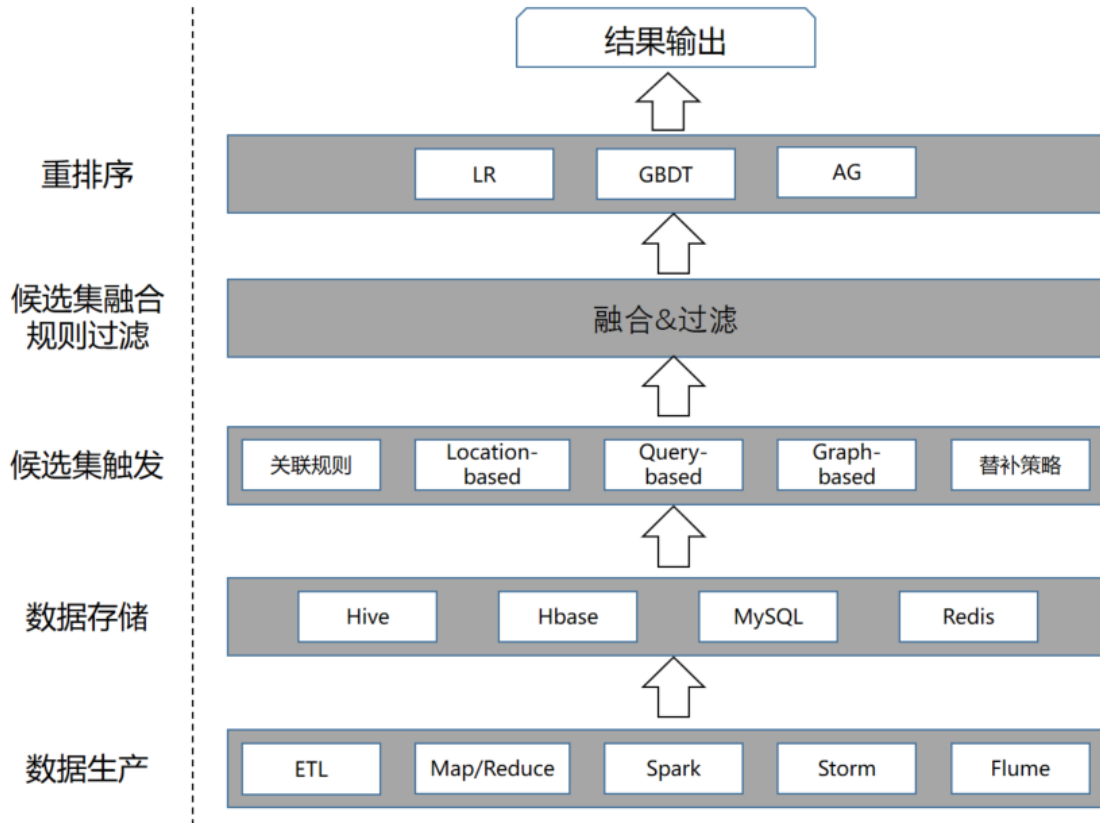


图 4-5 音乐数据推荐过程

用户在音乐平台的不同链接和页面中的行为记录在用户的活动行为数据中。一方面，这些数据用于候选集触发算法的离线计算。另一方面，这些行为代表着不同的意图。因此，在训练重排序模型时，可以针对不同的行为设置不同的回归目标值，以详细描述用户的行为。此外，用户的这些行为还可以作为在线预测和离线训练的重排序模型的交叉特征。在某些方面无法满足用户的需求的地方可以通过负反馈数据收集。所以，对特定因素进行滤波或降权的这种考虑，在候选集的触发过程中，有助于减少负因素重复出现的次数，以此提高用户体验。另外，在重排序模型的训练过程中，负反馈数据以负样本数据参与模型训练，比起那些没有排序或点击的样本要明显很多。对于用户属性的统一描述一般是通过用户画像表示，而大部分的数据都是直接获取的，少部分的数据是二次处理后的。在候选集的触发过程中，这些数据不仅可以对音乐进行加权还能进行减重，还可以作为用户维度特征应用于重排序模型中。通过对 UGC（User Generate Content）数

据的多重处理分析,我们可以提取一些标签,然后利用这些标签对音乐进行标注,实现音乐的个性化显示。

4.3 业务逻辑与模型实践

4.3.1 推荐逻辑与业务体系

在音乐推荐的实际应用场景中,我们使用关联规则算法分发歌曲和歌曲列表。典型的组件有:热门歌单和排行榜,可以根据个性化场景推荐相关曲目。

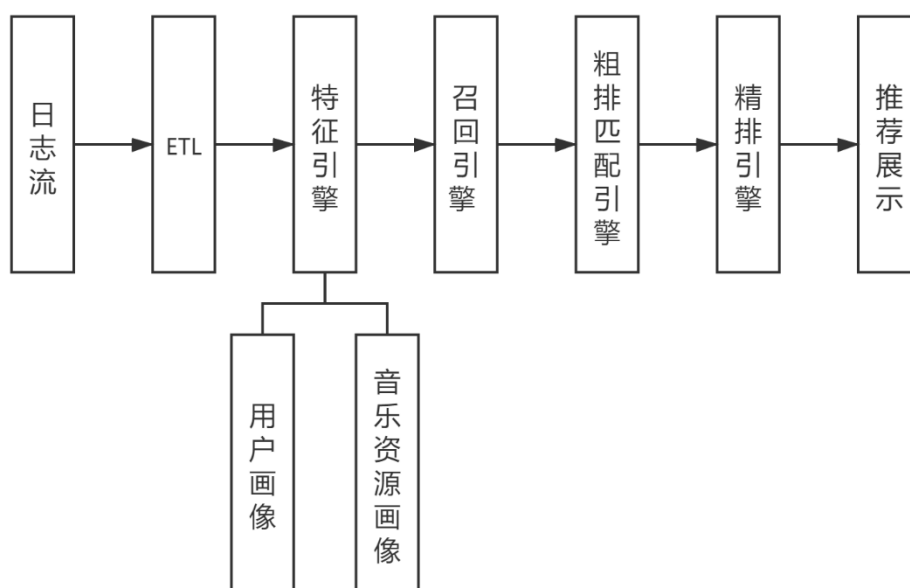


图 4-6 推荐平台系统逻辑图

图 4-6 是整个音乐推荐平台的系统逻辑图,其中包括用户行为的日志流,ETL,特征引擎,召回引擎以及排序匹配到最后展示推荐内容。

对于音乐推荐平台来说,最重要的是如何了解用户的画像,即通过前端数据的整合,了解用户喜欢什么样的音乐。在这些基础之上对用户进行精准个性化推荐。所以对于本文所设计的推荐平台的体系建设主要分为业务体系建设、算法体系建设、算法工程体系建设和数据体系建设。

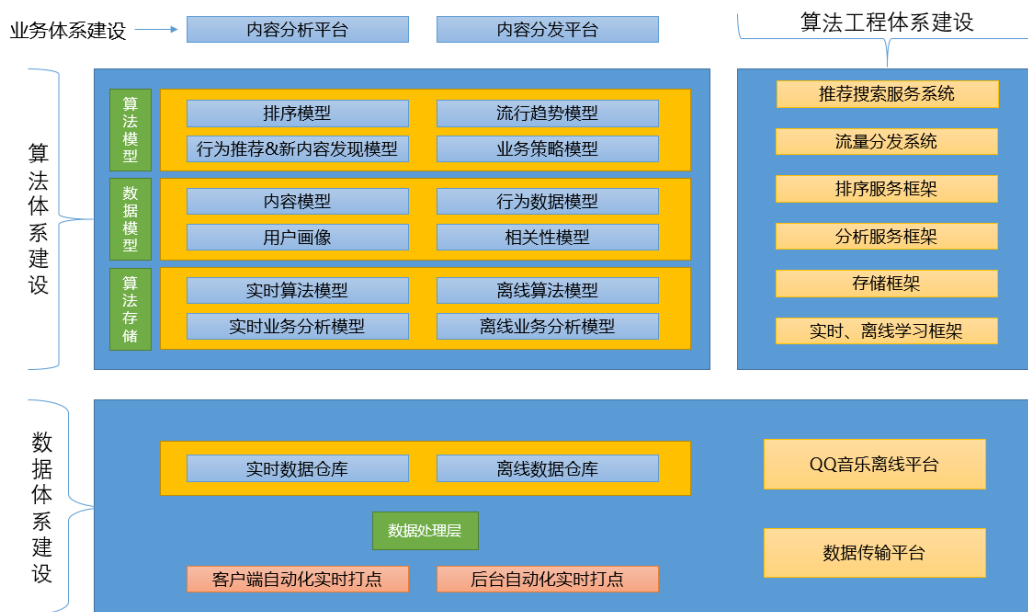


图 4-7 推荐平台业务体系建设图

从图 4-7 可以看出，业务体系建设主要分为内容分析平台和内容分发平台两部分，算法体系建设中有算法模型层、数据模型层和算法存储层，算法工程体系涉及到的模块较多，主要为推荐搜索服务、流量分发、排序、存储、实时与离线学习等。在数据体系建设中，分为实时数据仓库，通过 Storage 存储，离线数据仓库，通过后台数据表存储，通过对 QQ 音乐离线数据的获取，然后通过数据传输平台链接给后台系统中，然后对数据进行关联规则算法过滤与处理，最后通过内容分发平台展现出最终的推荐内容。为了记录用户行为数据，客户端使用自动化实时打点系统 Pstat 进行打点，然后将打点数据传输到后台中分析处理，通过内容分析平台对用户行为数据进行分析，对用户进行精准个性化音乐推荐，以满足用户的需求。

在整个音乐推荐过程中涉及到很多业务平台、数据模型还有算法工程，这是由于音乐推荐相对于其他推荐有较高的复杂性，主要表现在一下几个方面：

1. 因为我们可以短时间内分辨出自己的喜好，我们可以在 10 秒内浏览 10 多件衣服。但音乐需要时间去体验。我们经常听一首歌 10 秒钟或更长时间，然后才发现它不是我们喜欢的。因此，音乐不能通过直接看来理解。在开发音乐推荐平台的过程中，要以用户体验为指导，真正了解音乐本身。

2. 衣服在单位时间内只能消费一次，但人们可以通过循环播放列表和单曲在一个单位时间内反复欣赏音乐。因此，这是一种重复消费行为，在整个推荐过程中还需要把握这一规律。

3. 由于音乐消费成本相对较高，我们需要更加关注用户体验，以及用户消费过程中存在的强时序相关性。同时，给用户 10 秒、30 秒还是 60 秒的时间听一首歌，对他们来说，其表达的意义是截然不同的。因此，我们需要提供的是真正有意义的消费，在这种情况下，这些关联才能反映出有效的行为意义。

鉴于以上提到的有关音乐推荐的复杂性，当有新的音乐被收录时，我们可以通过一些语言与之进行关联，建立起一个新的解释性描述。如图 4-8 所示，例如有一首《演员》的歌曲，当它被收录时，就建立起与之相关的描述性短语信息，通过新增不同的关键词来还原音乐画像，从而产生对该音乐的相关特征描述，最终生成歌曲信息的可视化。

歌曲 → 选择关联歌单 → 自动生成特征词 → 过滤词表 → 生成短语描述 → 歌曲短语可视化

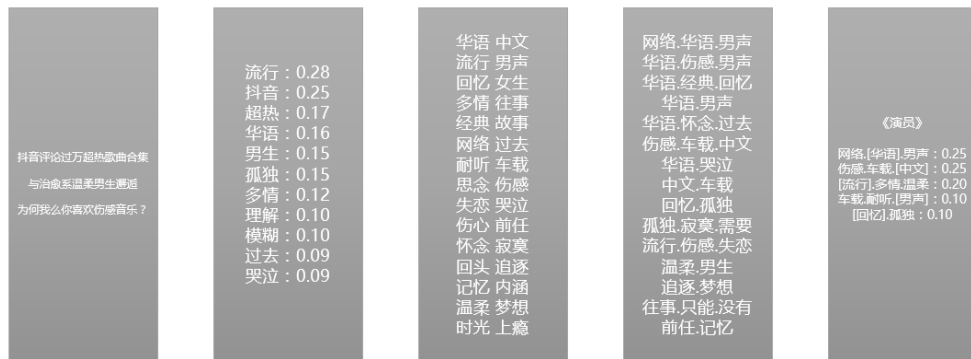


图 4-8 歌曲可视化信息生成过程

4.3.2 模型实践与应用

在歌曲应用场景中，我们可以直接向模型提供用户行为生成的与歌曲相关的丰富数据。通过关联规则算法，将各种音乐维度上的信息抽象成标签，然而，虽然我们有足够的音乐资源和行为样本，但是行为序列往往不是线性的，因此我们遇到了过度拟合和特征时间穿梭（特征记忆）的问题。对于有效的利用数据，根据不同行为的关联性进行学习来提升模型的拟合能力，在模型构建上本文尝试了

不同的数据集模型。

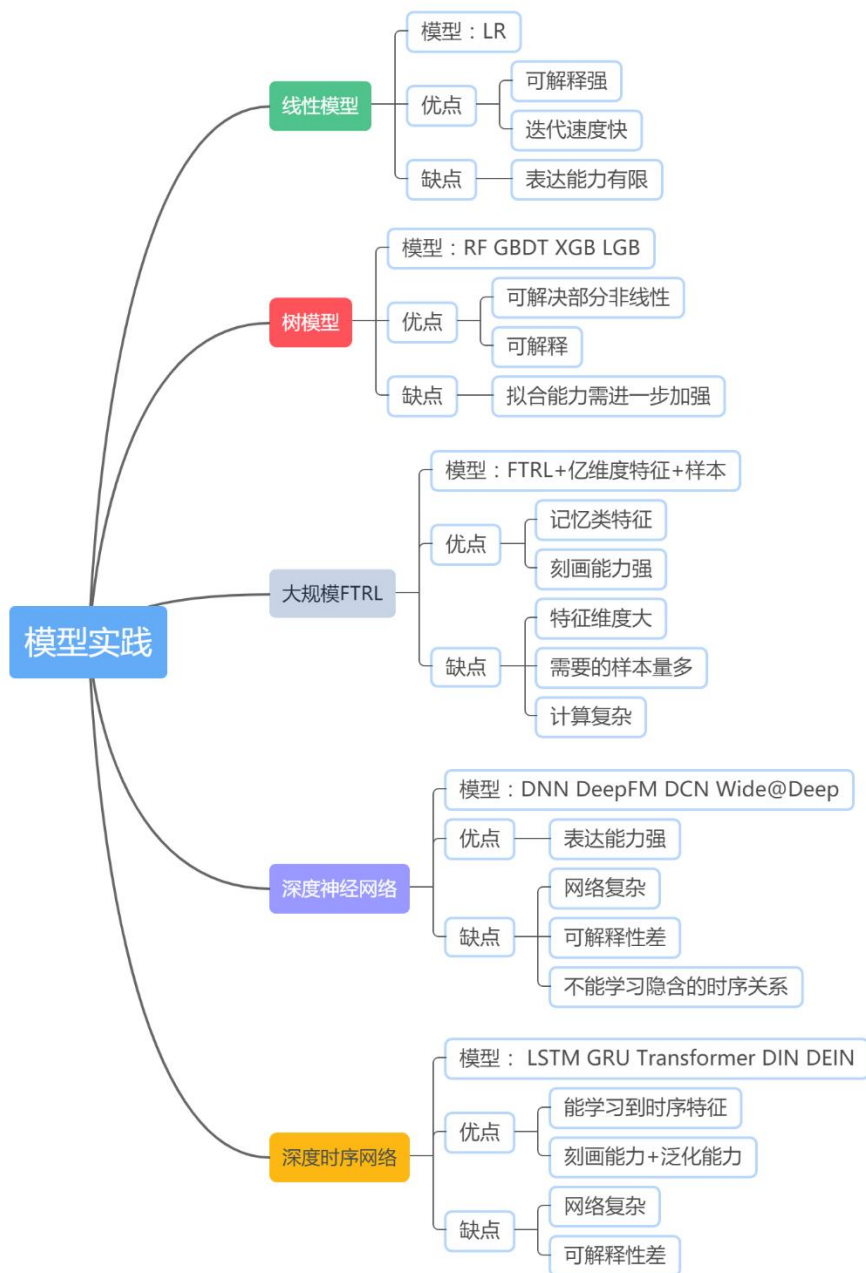


图 4-9 推荐模型对比

本文首先从线性模型出发，虽然线性模型的可解释性强，且迭代速度快，但是其表达能力有限。后来转入到树模型，树模型相比较而言可解决部分非线性的问题，但其拟合能力还需要进一步的加强。对于大规模的 FTRL 来说，它的记忆类特征刻画能力强，但是特征维度较大，且需要巨大的样本数量作为支撑，计算过程也复杂。为了增强表达能力，本文采用了深度神经网络，包括 DNN 模型，

DCN 模型等，但是深度神经网络模型的网络复杂，可解释性差，对于隐含的时序关系也不能学习。最后，本文选择了深度时序网络，它虽然也有深度神经网络的缺点，但是深度时序网络的刻画能力和泛化能力突出，能够学习到不同的时序特征。

4.3.3 多目标联合训练

推荐系统是典型的统计应用，但统计只能解决 95% 的问题，剩下的 5% 是关于个人喜好的。在实际的推荐过程中，我们往往会遇到各种不同的问题，其中包括点击率与在线时长不是同步提升的情况，有时还会出现反比的气势，针对此时的多目标问题的解决方法有多个，如图 4-10 所示，样本加权、Weight Loss、以及部分网络共享都可以解决多目标的问题，所以本文采用多目标联合训练，这是一种简单的实现方法。

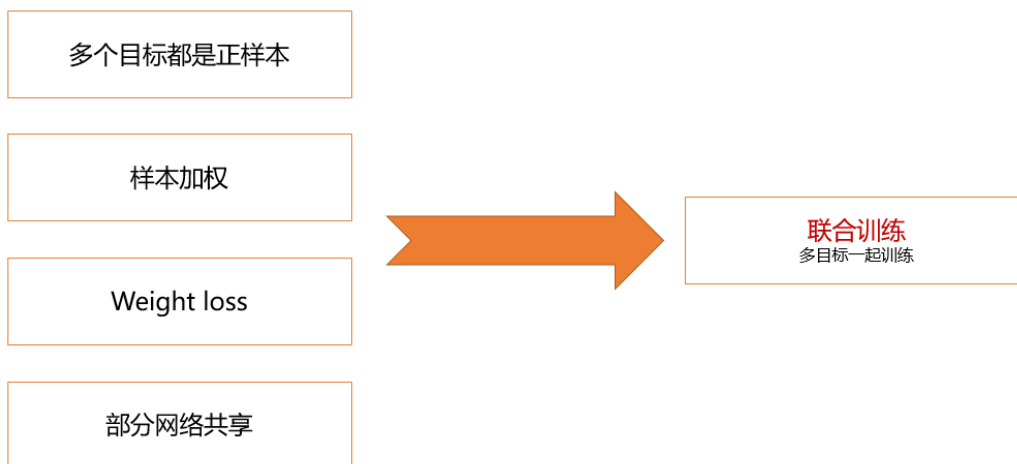


图 4-10 多目标联合训练

联合训练的有时就在于通过在浅层共享多个目标任务，在任务间添加噪声数据。这不仅减少了网络过拟合，而且提高了泛化效果。在多目标任务学习中，通过保持不同任务的局部极小值在不同的位置并相互作用，有助于避免局部极小值。通过多目标任务的联合训练，该模型能尽可能地求解共同的最优解。采用类似“窃听”的方式，跟踪用户的音乐收藏等操作，然后做出相应的判断。

通过基于关联规则的音乐推荐平台，我们可以不断提供强大的长尾探索能力

和精准匹配能力，进而推动他们在不断提升用户体验的同时，在音乐平台上主动分享和探索更多的歌曲资源。

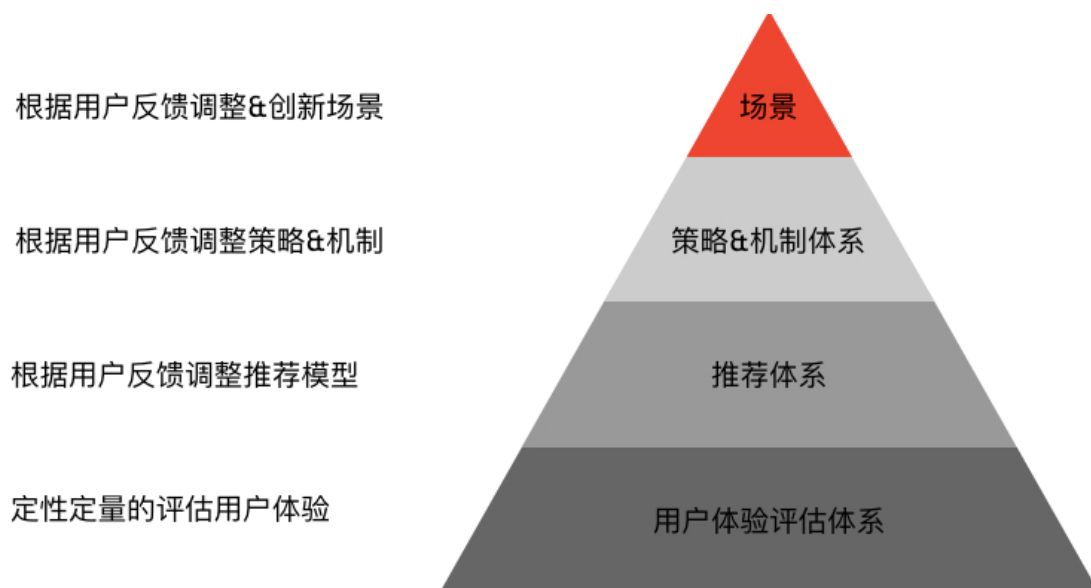


图 4-11 用户体系结构

为了达到让用户自发分享的目的，本文建立了如图 4-11 所示的体系结构，具体包括对用户的行为、认知和态度的分析，接入用户调研平台对用户做线上问卷调查反馈，分析用户和用户群对歌单、歌手和排行榜等推荐元素的使用行为，建立完整的数据反馈体系，从而不断优化平台。这样不断地进行统计学习，构建强大的用户体验评估体系，其中包括收藏、喜欢、切歌等用户正向和负向的反馈行为，让推荐平台带来更高的用户体验。

4.4 平台功能模块设计

4.4.1 可视化组件设计

本文把前端可视化组件划分为 2 大部分，第一是支撑层，支撑层分为架构支撑和底层框架，架构支撑主要包括基础组件库、状态管理、路由、服务器通讯和第三方插件，底层框架基于 MVVM 框架 Vue.js。业务层主要是页面架构，其中包括推荐歌单、歌手列表、歌手详情页、播放内核、排行榜、搜索和用户中心等

页面。使用 Vue-cli 脚手架对文件资源进行模块化管理，并且通过 webpack 实现前端的自动化项目构建。

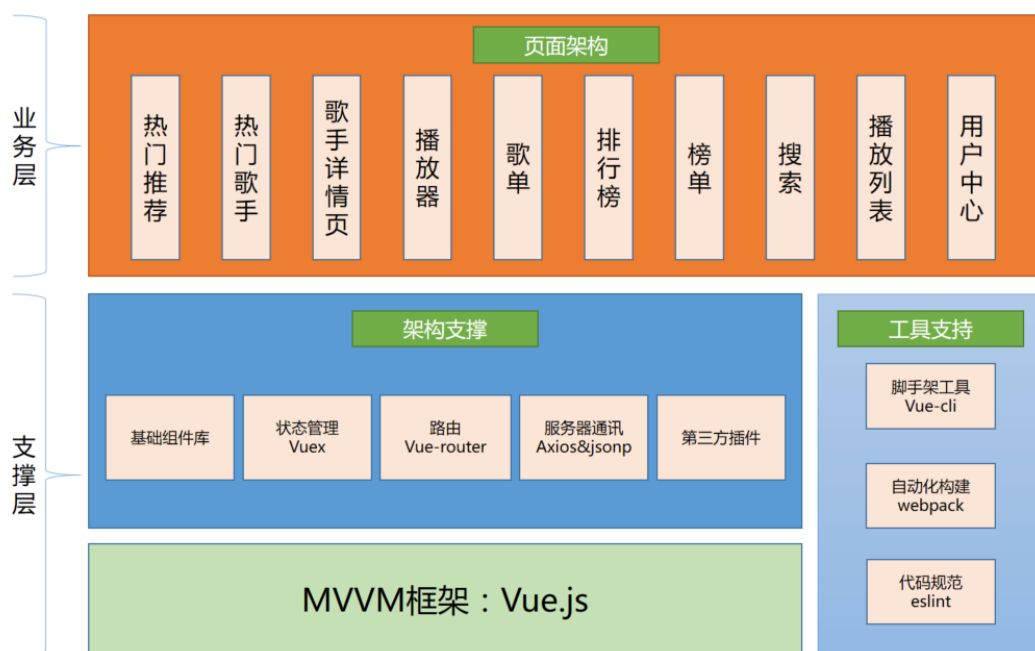


图 4-12 推荐结果可视化组件架构图

(1) 推荐歌单页面通过分析个人信息与音乐收听行为，结合关联规则的 FP-growth 算法，为用户展现一个热门歌单推荐，随着用户收听歌曲数量的增加，推荐的准确性也越来越高。

(2) 热门歌手页面会通过分析当前所有用户的收听频率与播放次数将热门歌手以字母排序返回一个列表，并展现给用户。

(3) 排行榜是对音乐数据的整体分类与整合，通过分析歌曲的标签、发行时间、收听频率以及热度上升速度，将歌曲分类成热歌、新歌、飙升等排行榜，每个榜单会筛选出 100 首歌曲。

(4) 搜索模块根据歌曲、歌手的信息进行模糊匹配，也会根据用户搜索的关键词分析出当前搜索的最热门的歌曲、歌手列表，并推荐给用户。播放器组件是包括歌曲播放、加入喜爱、播放暂停等功能控件的一个页面。

本文设计的针对推荐结果的可视化前台是一个移动端的 Webapp，Webapp 最大的优点就是平台无关性，只要在浏览器端访问就能正常使用，用例分析如图 4-13 所示。

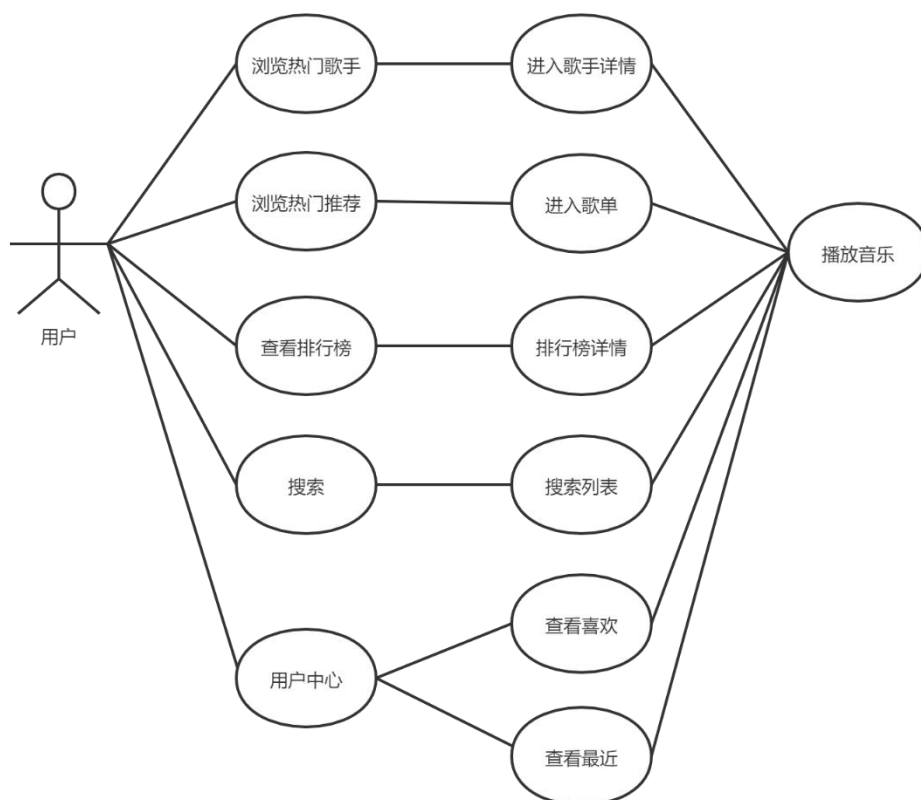


图 4-13 可视化平台用例分析

4.4.2 数据库设计

数据库的建立主要考虑各个表中数据的唯一性、关联性和可处理难易度。数据库主要数据表如表所示。

表 4-1 热门歌单数据表

属性名	属性类型	属性说明
diss_id	BIGINT(20)	歌单 ID
create_time	DATE	创建时间
commit_time	DATE	提交时间
diss_name	VERCHAR(64)	歌单名称
img_url	VERCHAR(128)	歌单缩略图
introduction	VERCHAR(128)	歌单介绍
listen_num	BIGINT(20)	收听数量
score	BIGINT(20)	评分

表 4-2 歌手数据表

属性名	属性类型	属性说明
area	VARCHAR(64)	地区
name	VARCHAR(64)	名字
id	BIGINT(20)	ID
tag	VARCHAR(128)	标签
other_name	VARCHAR(64)	别名

表 4-3 排行数据表

属性名	属性类型	属性说明
id	BIGINT(20)	ID
type	BIGINT(20)	类型
pic_url	VARCHAR(128)	排行榜缩略图
listen_count	BIGINT(20)	收听数量
top_title	VARCHAR(64)	排行标题

表 4-4 排行数据表

属性名	属性类型	属性说明
id	BIGINT(20)	ID
tag	VARCHAR(128)	标签
url	VARCHAR(128)	歌曲地址
name	VARCHAR(64)	名称
singer	VARCHAR(64)	歌手
file_name	VARCHAR(64)	文件名称
type	VARCHAR(128)	歌曲类型
tips	VARCHAR(128)	附加信息

4.5 本章小结

本章主要介绍了音乐推荐平台的设计与架构,介绍了平台开发应用到的相关架构模式,开发模式以及开发框架,对音乐推荐内容进行了分析,介绍了推荐逻辑和平台涉及到的业务体系,对推荐模型的实践应用过程做了说明,用联合训练解决多目标的问题,建立用户反馈体系,提高用户体验。

第5章 平台实现与测试

5.1 推荐结果展示模块实现

5.1.1 推荐歌单

通过 Promise 封装接口调用的 api 组件，然后通过 jsonp 解决跨域请求数据的问题。通过 axios 实现服务端代理，处理数据库所返回的数据，把返回的数据渲染到 recommond 组件中，封装移动端的 scroll 插件实现页面 touch 滚动。核心代码如下所示。

```
renderRecommendPlay(list) {  
  var btm = list.length > 0 ? '80px' : ''  
  this.$refs.scrollListContent.style.bottom = btm  
  this.$refs.scrollListContent.refresh()  
}  
handlerImageLoadingPro() {  
  if (!this.isLoadedChecked) {  
    this.isLoadedChecked = true  
    this.$refs.scrollListContent.refresh()  
  }  
}  
setItemSelected(item) {  
  this.$router.push({  
    path: `/recommend/${item.playerDissListId}`  
  });  
  getAllRecoListContent() {  
    getAllRecoListContent().then((res) => {  
      if (res.code === ERR_OK) {  
        this.myRecoListContent = res.data.getSliders  
      }  
    })  
  }  
  getAllDiscListContent() {  
    getAllDiscListContent().then((res) => {  
      if (res.code === ERR_OK) {  
        this.myDisListContent = res.data.getLists  
      }  
    })  
  }  
}
```

5.1.2 热门歌手

开发实现 listView 基础滚动组件，通过右侧导航中的字母点击切换到具体的歌手分类位置，页面的顶部自动切换分类标题并固定显示。将处理好的数据通过接口调用后实时渲染到页面中，当点击某一具体歌手后，显示当前歌手详情，并推荐其当前最热门的歌曲列表并通过热度降序排序的方式依次渲染。核心代码如下所示。

```
handlerSingleSingerSelect(singleSinger) {
  this.$router.push({
    path: `/mySinger/${ singleSinger.id}`
  })
  getAllSingleSingerListContent() {
    getAllSingleSingerListContent ().then((res) => {
      if (res.code === ERR_OK) {
        this.singerList = this._formatSingerData(res.data.allListData)
      }
    })
  }
  _formatSingerData (l) {
    let arr = {
      big: {
        title: TITLE_NAME,
        lists: []
      }
    }
    l.forEach((i, e) => {
      if (e < LENGTH_FOR_SINGER) {
        arr.big.lists.push(new Singer({
          name: i.fistSinger_name,
          id: i.firstSinger_mid
        }))
      }
      var key = item.firstIndex
      if (!arr[key]) {
        arr[key] = {
          title: key,
          items: []
        }
      }
    })
  }
}
```

```

arr[key].items.push(new Singer({
  name: item. fistSinger_name,
  id: item. fistSinger_mid
}))
})
var ret = []
var hot = []
for (let key in arr) {
  var val = arr[key]
  if (val.title.match(/[a-zA-Z]/)) {
    ret.push(val)
  } else if (val.title === TITLE_HOT_NAME) {
    hot.push(val)
  }
}
ret.sort((a, b) => {
  return a.title.charCodeAt(0) - b.title.charCodeAt(0)
})
return hot.concat(ret)
}

```

5.1.3 排行榜

这个模块核心点主要是将排行榜二维数组渲染为多个嵌套列表，通过 Vuex 实现路由数据通讯，点击某个分类的排行详情时跳转到排行详情页面，并将当前分类下推荐的歌曲数据渲染到列表中。核心代码如下所示。

```

rankListPerViewContent(item) {
  this.$router.push({
    path: `/rankView/${item.id}`
  })
  this.renderListTopView(item)
}
_getListView() {
  getListView().then((res) => {
    if (res.code === ERR_OK) {
      this.content = res.data.content
    }
  })
}
}

```

```

setListViewBeforeRender() {
  setTimeout(() => {
    this.$Lazyload.handlerRenderLazyLoad()
  }, 20)
}

```

5.1.4 搜索与播放器

搜索页主要实现数据的模糊查询，触发方式通过 input 的 onChange 事件，实时将检索的数据渲染至 suggest 组件中，数据携带歌曲的唯一标识符，当点击搜索结果时跳转至播放器模块。播放器模块通过 Html5 中的 audio 组件实现歌曲播放，封装自定义 controlbar 实时控制歌曲的切换、播放暂停、拖动定位时间等功能。核心代码如下所示。

```

_handlerKeyGetter() {
  handlerKeyGetter ().then((res) => {
    if (res.code === ERR_OK) {
      this.singerKey = res.data. singerKey.slice(0, 10)
    }
  })
}
formData(afterList) {
  if (!afterList) {
    setTimeout(() => {
      this.$refs.shortcut.refresh()
    }, 20)
  }
}
handlerLeaveCost() {
  this.$refs.endContent.style.transition = ""
  this.$refs. endContent.style[transform] = ""
}
ready() {
  this.songReady = true
  this.savePlayHistory(this.currentSong)
}
updateTime(e) {
  this.currentTime = e.target.currentTime
}

```

5.2 平台测试

推荐歌单模块主要包含两部分，顶部为热度最高的 5 个歌单，并通过轮播组件自动滚动显示，列表部分为热门的歌单推荐，用户可通过 touchslide 滑动浏览其余歌单，点击歌单会进入歌单详情页，详情页中为当前分类歌单中热门歌曲列表。

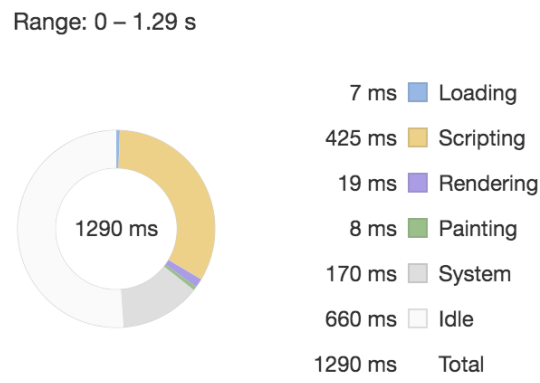


图 5-1 推荐歌单模块性能测试

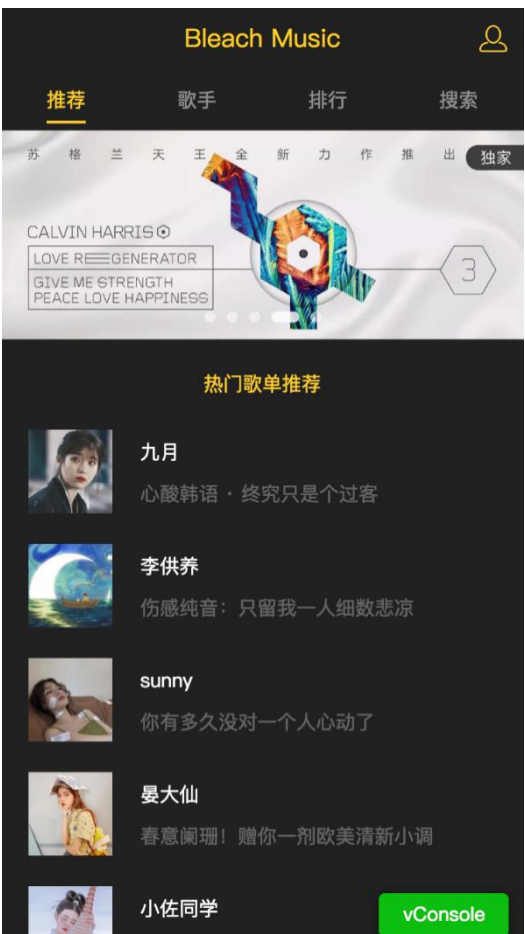


图 5-2 推荐歌单模块展示结果

热门歌手模块是一个类似通讯录展示方式的列表，并且以字母排序的方式推荐出当前首字母分类下的热门歌手，最顶部是所有歌手中排名最靠前的歌手。点击歌手会进入歌手详情页，详情页中包含当前歌手的热门歌曲列表。

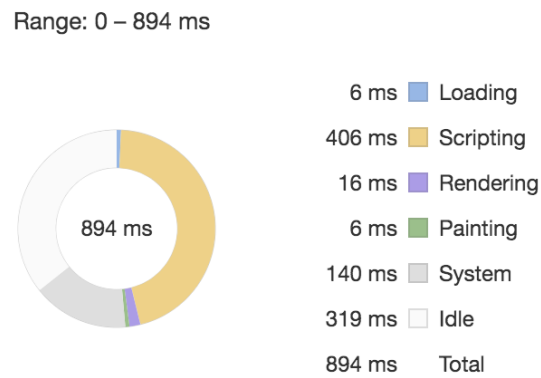


图 5-3 热门歌手模块性能测试

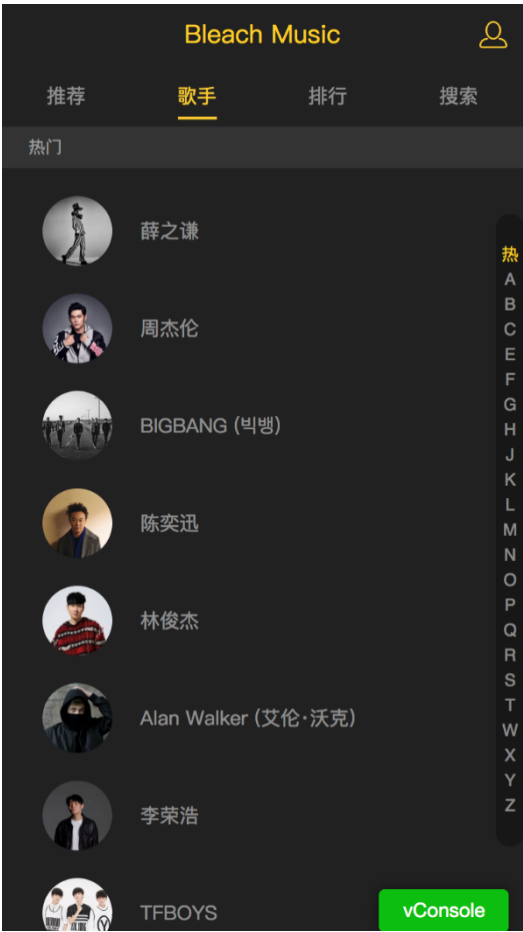


图 5-4 热门歌手模块展示结果

排行榜模块中首先以歌曲标签进行了歌单分类,再以热度进行分类,然后将热度与标签进行分类关联规则算法下的筛选与整合,生成每一歌单下的二级列表,例如以流行标签和热度结合,生成“流行指数”歌单,以歌曲发行时间和热度结合生成新歌榜。点击分类会进入二级排行榜页面,列表中为当前排行榜下的热门歌曲排行。

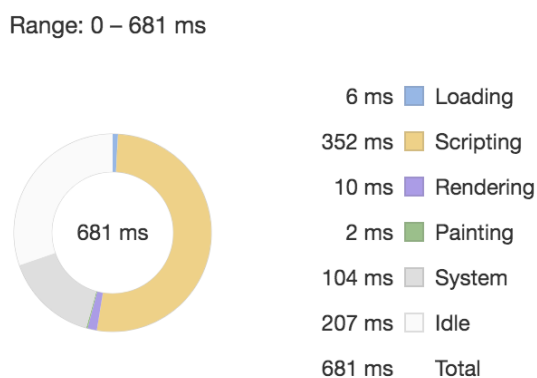


图 5-5 排行榜模块性能测试



图 5-6 排行榜模块展示结果

搜索模块中未填入关键字时，会有 suggest 推荐歌曲组件列表，当键入关键字时，会监听搜索框的 `change` 事件，并实时渲染当前关键字模糊匹配到的歌曲及歌手。

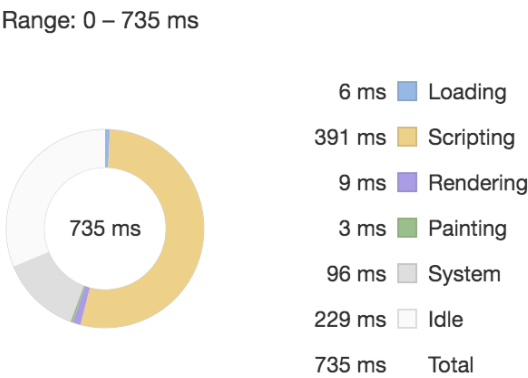


图 5-7 搜索模块性能测试



图 5-8 搜索模块展示结果

个人中心模块分为两部分，最近播放歌曲列表以及我喜欢的歌曲列表，这两个模块的设计从用户体验出发，方便了用户查找自己喜欢的歌曲，定制化的功能满足了用户的个性化需求。而且列表的顶部添加了随机播放全部歌曲的功能，一键将自己喜欢或者最近听的歌曲列表加入当前播放列表中。

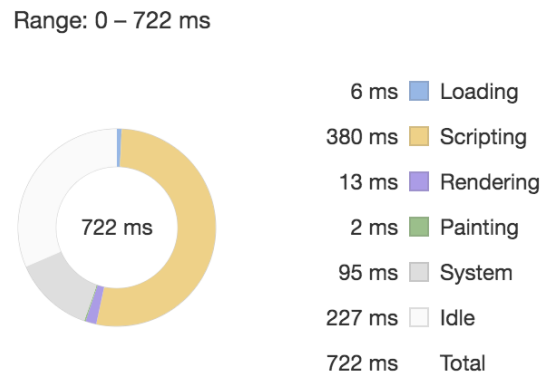


图 5-9 个人中心模块性能测试

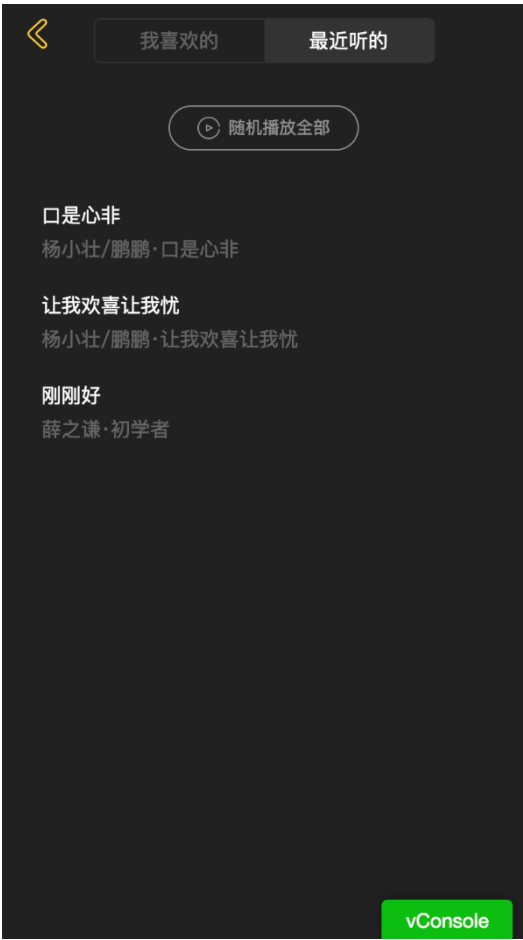


图 5-10 个人中心模块展示结果

播放器模块中主要有以下功能，播放模式切换，可以切换随机播放，列表循环，单曲播放等播放模式，切换上一曲下一曲，播放暂停，加入喜欢，左右滑动可以在歌曲封面和歌曲的歌词两部分内容做切换，最突出的一个亮点是，当用户浏览其他功能模块时，可以将播放器最小化至页面的底部，而不会停止播放。

Range: 0 – 775 ms

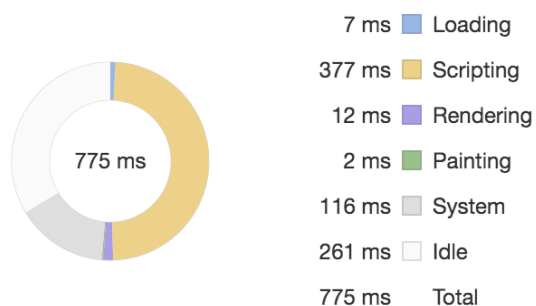


图 5-11 播放器模块性能测试



图 5-12 播放器模块展示结果

5.3 本章小结

本章介绍了推荐结果展示的实现，包括推荐歌单、热门歌手排行榜、以及搜索与播放器核心部分的代码。邀请十几位同事和同学进行了各种机型和系统的测试，让用户对推荐展示的表现效果进行真实体验，对展示结果模块的性能和表现效果做了不同网络情况下的测试，并得出平均加载时间，在推荐结果的总体展示中，通过 vConsole 测试插件对所出现的 bug 进行了复现，找出了其出现的原因并修复。

第6章 总结展望

6.1 本文完成的主要工作

本文在针对现存与音乐推荐相关的技术进行了简单的介绍与分析后,总结归纳了目前各音乐平台在音乐推荐方面的不足,并分析了各推荐算法的应用情况。接下来阐述了与数据挖掘相关的理论,针对关联规则算法的形成与发展进行了梳理,全面介绍了该算法的优势,并就该算法在音乐推荐领域应用的可行性以及可创造的价值进行了分析与探讨。通过与根据内容与协同过滤的算法的对比,揭示了关联规则算法的优势。在对比分析三种算法的过程中,本文首先确定了相关的评价标准,并分别由精度、新颖性以及多样性等方面对三种方法进行了对比与分析,同时实现了音乐推荐平台,最终验证了算法可以应用于音乐推荐,为用户提供个性化的音乐推荐服务,使用户需求得到满足。

6.2 本文的主要贡献及创新点

通过各渠道收集与本文相关的资料以及文献,本文发现在音乐推荐方面关联规则算法的应用相对较少。本研究基于关联规则算法设计的音乐推荐系统可以丰富这一领域的内容,推动音乐推荐技术的发展,为用户提供知心的服务。此外本文也通过与基于内容的推荐算法和基于协同过滤的推荐算法的对比,对关联规则算法的适用性与有效性进行了验证,通过本文设计和实现的音乐推荐平台对关联规则算法在实际应用中可能起到的作用进行了展示与介绍。

本文基于 B/S 三层构架设计了一种音乐推荐平台,在设计的过程中选择了现阶段较为流行的 MVVM 开发框架 Vue.js,在推荐模型的选择上进行了多次实践,然后对不同的推荐模型做了对比与分析。对音乐推荐设计的业务平台进行了整合架构,并在实现平台的开发后邀请同事和同学对该平台进行了测试,通过分析测试反馈结果可以发现,本研究设计的系统大致可以满足用户的要求,在用户体验设计上展现出较高的前端水平。

6.3 今后的工作

另一方面,个性化推荐属于一种持续创造、学习、观察与积累的过程,涉及了海量的用户数据,同时用户需求等也会发生一定的变化,因此要不断改进相关的推荐方法。受到知识储备、研究设计能力以及时间精力的影响,本文定然存在一定的不足,比如本文采集的数据相对较少,同时算法的应用也较为简单,不过笔者将在今后的工作、生活中进一步改进完善该推荐算法,使其发挥更大的作用。

致谢

光阴荏苒，不知不觉间硕士生活已将结束，再回首一切都恍如昨日，学习与生活中的种种均历历在目，尽管时光一去不返，不过却在我心中烙下了幸福的回忆，这将成为我人生中最宝贵的一笔财富。

首先我要感谢毛宏燕老师，可以说没有毛老师的精心指导与近乎严苛的要求，本文就无法顺利完成。毛老师在本文撰写的所有过程中均留下了一抹抹浓墨重彩的身影，投入了大量的精力，为我指明了论文的撰写方向，并指出了很多不足之处，提高了本文的设计水平。毛老师的精神与态度都将对我未来的人生产生深刻的影响，在此，我谨向毛老师致以我最衷心的祝愿，祝毛老师身体安康，万事顺利。

其次我要感谢郭辰为我提供的帮助，感谢龚潇在工作中为我提供的支持。感谢 A 班可爱的同学们，谢谢你们的陪伴。

最后我要感谢我的父母，谢谢你们为我赋予的宝贵生命与 20 余年来的关心与支持，感谢所有亲人与朋友，你们的关心帮助与陪伴我都将铭记于心，余生不敢或忘。

参考文献

- [1] Man-Kwan Shan, Fang-Fei Kuo, Meng-Fen Chiang, et al. Emotion-based music recommendation by affinity discovery from film music[J].Expert Systems with Applications,2009,36:766-7674.
- [2] Dmitry Bogdanov, Martin Haro, Ferdinand Fuhrmann, et al. Semantic audio content-based music recommendation and visualization based Management,2013,49: I 3-33.on user preference examples[J].Information Processing and Management,2013,49: I 3-33.
- [3] Le Q., Mikolov T., Distributed Representations of Sentences and Documents[C], International Conference on Machine Learning — ICML 2014, vol. 32, p. 1188-1196
- [4] DENG JJ, LEUNG CHC, MILANIA, et al. Emotional states associated with music : classification , prediction of changes , and c onsideration in recommendation [J]. ACM tran-sactions on interactive intelligent systems, 2015, 5 (1): 1-36.
- [5] CHOIK, LEE JH, HUX, et al. Music subject classification based on lyrics and user interpretations [J]. Proceedings of the association for information science & technology, 2016, 53 (1): 1-10.
- [6] SON L H, TIEN N D. Tune up fuzzy C -means for big data: some novel hybrid clustering algorithms based on initial selection and incremental clustering [J]. International journal of fuzzy systems, 2017, 19 (5): 1585-1602.
- [7] Yu K, Xu X, Ester M, et al. Feature Weighting and Instance Selection for Collaborative Filtering: An Information-theoretic Approach[J]. English: Knowledge and Information Systems, 2003:201-224.
- [8] 李东勤. 电子商务中个性化推荐系统的研究[J]. 现代计算机月刊, 2012(10): 3-8.
- [9] 王海艳, 杨文彬, 王随昌,等. 基于可信联盟的服务推荐方法[J]. 计算机学报, 2014, 37(2): 301-311.
- [10] 项亮,推荐系统实战[M],2 人民邮电出版社出版发行 2012 年 6 月第 1 版, ISBN 978-7-115-28158-6/,page 22-23

- [11]Jiang M, Cui P, Han J. Data-driven behavioral analytics: Observations, representations and models [J]. ACM CIKM (tutorial), 2016.
- [12]JOSEPH S, ABDU J E. Real-time retail price determination in smart grid from real - time load profiles [J]. International transactions on electrical energy systems, 2018, 28 (3): 2509-2515.
- [13]刘珊珊.音频特征与社会标签相结合的音乐推荐系统 [D].湖北武汉:华中科技大学, 2011
- [14]王然.移动用户行为感知的音乐推荐系统研究与实现 [D].北京:北京工业大学,2016.
- [15]钟伟 . 基于音乐基因的混合音乐推荐系统的设计与实现 [D]. 安徽 : 安徽大学,2014.
- [16]Jaime Selection Janice Shack-M William L. Wascher. and Research Experience: A Multinomial of Statistical Inference, Model Data Mining[J].Economics Letters,1985,18(1):39-44
- [17]Wang F, WenY, Guo T, et al. Personalized Commodity Recommendations of Retail Business Using User Feature Based Collaborative Filtering[C]//2018 IEEE Intl Conf on Parallel & Distributed Processing with Applications, Ubiquitous Computing & Communications, Big Data & Cloud Computing, Social Computing & Networking, Sustainable Computing & Communications (ISPA/IUCC/BDCLOUD/SocialCom/SustainCom). IEEE, 2018: 273-278
- [18]Takács G, Pilázy I, Németh B, et al. Scalable collaborative filtering approaches for large recommender systems [J]. The Journal of Machine Learning Research,2009.10.623-656.
- [19]Ghaisas S. Recommendation system for agile software development [P]: U.S. Patent 9,262,126. 2016-2-16.
- [20]Hieu V. Nguyen., Cosine Similarity Metric Learning for Face Verification [M]. Part of the Lecture Notes in Computer Science book series (LNCS, volume 6493)
- [21]李博,陈志刚,黄瑞,基于 LDA 模型的音乐化推荐算法[J],计算机工程,2016.6,第 42 卷第 6 期,175-180
- [22]NAJI M, FIROOZABADI M, AZADFALLAH P. Emotion classification during music listening from forehead biosignals [J].Signal image & video processing,

- 2015, 9 (6): 1365-1375.
- [23]BANIYA B K, LEE J. Importance of audio feature reduction in automatic music genre classification [J]. Multimedia tools & applications, 2016, 75 (6): 1-14.
- [24]WANG B, DAN C, SHI B, et al. Comprehensive association rules mining of health examination data with an extended FP-Growth method [J]. Mobile networks & applications, 2017, 22 (2): 1-8.
- [25]KHONGLAH B K, PRASANNA S R M. Clean speech/speech with background music classification using HNGD spectrum [J]. International journal of speech technology, 2017, 20 (6): 1-14.
- [26]Sarwar B, Karypis G, Konstan J, et al. Item-based collaboration filltering recommendation algorithms[C]. Americ: International Conference on World Wide Web, 2001:285-295.
- [27]黄创光,印鉴,汪静.不确定近邻的协同过滤推荐算[J].天津:计算机学报,2010:350-410.
- [28]朱丽中,徐秀娟,刘宇.基于项目和信任的协同过滤推荐算法[J],广州:计算机工程,2013:58-62.
- [29]张良均,云伟标,王路,刘晓勇.R 语言数据分析与挖掘实战[M].广州:机械工业出版社,2015:220.
- [30]Si L, Jin R. Flexible Mixture Model lor Collaborative Filtering[C]. Machine Learning Proceedings of the Twentieth International Conference, 2003:704-711.
- [31]Meng A, Shawe-Taylor J. An Investigation of Feature Models for Music Genre Classification Using the Support Vector Classifier[A]. Ismir 2005, International Conference on Music Information Retrieval[C]. DBLP, 2005: 604-609.
- [32]Yang B D, Lee W. Disambiguating Music Emotion Using[A]. Software Agents[C]. ISMIR. 2010.
- [33]王守涛.一种基于多维时间序列分析的音乐推荐系统研究与实现[D].江苏:南京大学,2014.
- [34]李瑞敏.基于社会化网络的个性化音乐推荐算法研究 [D].辽宁:大连理工大学,2013.
- [35]隋占丽.基于协同过滤算法的音乐推荐系统[D].福建:华侨大学,2013.
- [36]金蕾.个性化音乐推荐算法的研究与实现[D].山东:山东大学,2017.

- [37]谭学清,何珊.音乐个性化推荐系统研究综述[J].现代图书情报技术,2014(09):22-32.
- [38]Azzalini A. Statistical inference based on the likelihood [M]. Routledge, 2017.
- [39]Edward I, George Casella Biometrics,. George,Explaining the Gibbs Sampler[D], Unit, Cornell University , Ithaca , NY , 14853 , USA
- [40]Sauro J, Lewis J R. Quantifying the user experience: Practical statistics for user research [M]. Morgan Kaufmann, 2016.
- [41]Edelman B, Lai Z. Design of search engine services: Channel interdependence in search engine results [J]. Journal of Marketing Research, 2016, 53(6): 881-900.
- [42]Richards M. Software architecture patterns [M]. O'Reilly Media, Incorporated, 2015.
- [43]Hieu V. Nguyen., Cosine Similarity Metric Learning for Face Verification [M]. Part of the Lecture Notes in Computer Science book series (LNCS, volume 6493)
- [44]Paul Jaccard, THE DISTRIBUTION OF THE FLORA IN THE ALPINE ZONE[J]. translated from the French original (which appeared in the Revue géiérale des Sciences, 15th December, 1907, pp. 961 - 967)
- [45]Ghaisas S. Recommendation system for agile software development [P]: U.S. Patent 9,262,126. 2016-2-16.
- [46]Miller B N, Riedl J T, Konstan J A. Experience with Grouplens: making Usenet useful again[C]. In: Proceedings, 1997:219-231.
- [47]Adomavicius G, Tuzhilin A. Toward the next generation of recommender systems: a survey of the state-of-the-art and possible extensions [J]. IEEE Transactions on Knowledge and Data Engineering, 2005, 17(6):734-7491.
- [48]陈如明 .大数据时代的挑战价值与应对策略 [J]. 天津 : 天津大学出版社 2012,36(17):15-36.
- [49]许海玲,吴潇,李晓东,等.互联网推荐系统比较研究[J].软件学报,2009, 20 (2): 350-362.
- [50]毛宇星,陈彤兵,施伯乐.一种高效的多层和概化关联规则挖掘方法[J]. 软件学报 2011, 22(12): 2965-2980.
- [51]董辉.基于改进 FP Growth 算法的中药方剂配伍规律挖掘研究[(J].赤峰学院学报(自然科学版),2011,09:198-200.

- [52] 葛覃, 刘正华. 基于 FP-Growth 关联挖掘算法的商品推荐系统研究 [A]. TCGNC, CSAA, Proceedings of IEEE CGNCC2014 [C].//TCGNC, CSAA, Science and Technology on Aircraft Control Laboratory, 2014:5.
- [53] Yu K, Xu X, Ester M, et al. Feature weighting and instance selection for collaborative filtering: An information-theoretic approach*[J]. Knowledge and Information Systems, 2003, 5(2): 201-224.
- [54] 张洪顺. 推荐系统中矩阵稀疏性问题的研究[D]. 北京交通大学, 2018.
- [55] 王猛, 叶西宁. 音乐个性化推荐算法 RR-UBPMF 的研究[J]. 华东理工大学学报(自然科学版), 2017, 43(01): 113-118.
- [56] 陈运文. 谈谈“推荐系统”和“搜索引擎” [J]. 计算机与网络, 2017, 43(24): 43-44.
- [57] 王升升, 赵海燕, 陈庆奎, 曹健. 个性化推荐中的隐语义模型[J]. 小型微型计算机系统, 2016, 37(05): 881-889.
- [58] Koren Y, Bell R, Volinsky C. Matrix Factorization Techniques for Recommender Systems[J]. Computer, 2009, 42 (8) :30-37.
- [59] 吕晨, 姜伟, 虎嵩林. 一种基于新型图模型的 API 推荐系统 [J]. 计算机学报, 2015, 38(11): 2172-2187.
- [60] 朱郁筱, 吕琳媛. 推荐系统评价指标综述 [J]. 电子科技大学学报, 2012, 41(02): 163-175.
- [61] 陈雅茜. 音乐推荐系统及相关技术研究[J]. 计算机工程与应用, 2012(18): 9-16.
- [62] KOJI M, MICHAEL J P. Collaborative filtering with the simple Bayesian classifier[C]. Proceedings of the 6th Pacific Rim international conference on Artificial intelligence, Melbourne, Australia, 2000: 679-689.
- [63] Zhong E, Fan W, Yang Q. Contextual collaborative filtering via hierarchical matrix factorization. In Proceedings of the 2012 SIAM International Conference on Data Mining. Society for Industrial and Applied Mathematics, 2012: 744-755.