

2017 年度专业硕士学位论文

学校代码：10269

学 号：63141500348

華東師範大學

基于推荐系统的一款在线音乐网站的设计 与实现

院 系：计算机科学与软件工程学院

类 别：工程硕士

领 域：软件工程

指导教师：周 俊 副教授

申 请 人：龚 婧

2017 年 1 月完成

2017 Professional Master's Degree Thesis

University Code: 10269

Student ID: 63141500348

East China Normal University

Design and implementation of an online music website based on recommendation system

Department: School of computer science

and Software Engineering

Type: Master of Engineering

Domain: Software Engineering

Supervisor: Zhou Jun Associate Professor

Applicant : Gong Jing

2017.1

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《基于推荐系统的一款在线音乐网站的设计与实现》，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： 龚婧

日期： 年 月 日

华东师范大学学位论文著作权使用声明

《基于推荐系统的一款在线音乐网站的设计与实现》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 周俊

本人签名 龚婧

年 月 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。

龚婧硕士学位论文答辩委员会成员名单

姓名	职称	单位	备注
安平	教授	上海大学	主席
郭建	副教授	华东师范大学	
章玥	副教授	华东师范大学	

摘 要

随着物质生活的提高人们对于精神需求的追求越来越崇尚，音乐作为一种能够陶冶情操、放松心情的娱乐，在人们的生活中占着比较重要的一个部分。互联网的发展为音乐的展现形式提供了一个新的载体，音乐在线网站已经逐渐走进人们的生活中，推荐系统是目前新兴的一个领域，将其应用在音乐网站中，成为音乐网站发展的新的趋势。在用户听音乐的过程中，记录用户历史的数据信息，利用推荐技术挖掘用户的喜好然后为用户推荐歌曲，让用户在使用的过程中更加享受。本文根据系统中的用户在听歌的过程中所产生的日志信息，利用该日志信息整合相应的推荐指标，利用个性化推荐技术来为用户推荐其喜欢的歌曲、可能感兴趣的专辑、可能喜欢的歌手等。

在推荐算法的领域中，协同过滤算法是其中提出的比较早也是目前应用比较广泛的一种推荐算法，该算法能够基于内容对于一些数据信息进行过滤，根据所需要的内容信息进行推荐。将协同过滤算法与一些其他的算法相结合，混合产生了一些比较优秀的算法，例如 SVD 算法、Slope One 等，在本文中所研究的主要的内容就是对协同过滤相关的算法做了一些相关的研究以及实现工作。

在对算法进行测试上面是以 Last.fm 数据集作为所开发的算法的一个测试集，通过该测试集合的测试结果来为算法的效果提供实验的依据。在论文中探讨了音乐评分的方法，对基于所研究的推荐算法进行了研究，并在此基础上设计并开发了音乐在线网站。

【关键字】 推荐系统，个性化，Slope One，SVD

【论文类型】 应用研究

Abstract

With the improvement of material life, the spiritual needs of people's lives become more and more important. Music play an important role of people's lives which as a way to cultivate sentiment and relaxation of entertainment. The development of Internet provides a new carrier for the form of music. Therefore, music online website has gradually entered people's lives. Recommendation system is a new field, and its application in the music website has become a new trend in the development of music websites. In the process of listening to music, the user records the history of the data information, the use of recommendation technology to tap the user's preferences and then recommend songs for the user, allowing users to enjoy the use of the process. This paper is based on the log information system user generated in the listening process, using the log information integration recommended corresponding index, to recommend their love songs, may be interested in the album, love singer may use personalized for users by recommended technology.

Collaborative filtering algorithm is one of the most widely used in the field of recommendation algorithm. The algorithm can filter some data information based on the content, and recommend it according to the content information. Collaborative filtering algorithm combined with some other algorithms, resulting in a number of relatively good algorithms, such as SVD algorithm, Slope One, etc. The main content of this paper is to do some research on the collaborative filtering algorithm and design the corresponding website.

The algorithm is tested on the Last.fm data set as a test set of the developed algorithm, the results of the test can provide experimental basis for the effectiveness of the algorithm. This paper mainly discusses the method of converting the implicit feedback data into the score, designs the music system based on recommendation, and realizes the related functions.

【Keywords】 Recommendation System, Personalization, Slope One, SVD

【Paper Type】 Application Study

目 录

1 绪 言	1
1.1 研究的背景与意义	1
1.2 国内外研究现状分析	2
1.3 主要研究内容	4
1.4 本文主要创新点	4
1.5 本文的主要结构	5
2 相关技术分析	6
2.1 推荐系统简介	6
2.2 基于内容的推荐系统	7
2.3 基于协同过滤的推荐系统	8
3 系统需求分析	9
3.1 功能需要分析	9
3.1.1 登陆注册模块	9
3.1.2 专辑模块	9
3.1.3 艺术家模块	10
3.1.4 歌曲排行模块	11
3.1.5 歌曲标签	11
3.1.6 音乐推荐模块	12
3.2 非功能需求分析	12
3.2.1 性能要求	13
3.3 数据需求分析	14
3.4 本章小结	15
4 系统总体设计	16
4.1 系统架构设计	16
4.2 总体功能结构设计	17
4.3 推荐模块设计	18
4.4 数据库设计	20
4.4.1 数据库设计原则	20
4.4.2 E-R 图设计	21

4.3.3 逻辑结构设计	22
4.5 本章小结.....	26
5 系统实现.....	27
5.1 系统的实现环境	27
5.1.1 硬件环境	27
5.1.2 软件平台	27
5.2 主要功能实现	27
5.2.1 登录注册模块实现	27
5.2.2 专辑模块实现	28
5.2.3 艺术家模块实现	29
5.2.4 歌曲排行模块实现	30
5.2.5 歌曲标签	30
5.2.6 推荐模块实现	31
5.3 本章小结	44
6 系统测试.....	45
6.1 测试方案	45
6.2 测试用例	45
6.3 性能测试	47
6.4 测试结果	49
6.5 本章小结	49
7 总结与展望.....	50
7.1 总结	50
7.2 展望	51
参考文献.....	52
致 谢.....	54

1 绪 言

1.1 研究的背景与意义

21 世纪以来,网络快速的发展给人们的生活带来极大方便的同时也在每天产生了巨大的数据信息,对于用户而言,仅凭视觉、个人能力很难在海量的数据中获取到对于自己有用的数据信息,这样最直接的后果就是用户对此会感到厌倦并将降低对该数据信息的使用频率。对于大多数的用户来说,如何从这些数据量巨大的信息中来发现那些对于自己有帮助的数据部分,是一个比较让人头疼而且费时间的问题。在一般情况下,有两种不同的方法可以对信息过载问题进行解决,这两种方法分别是信息检索以及信息过滤。所谓信息检索就是进行数据信息查询,海量数据信息通过一定的组织方式被进行梳理,然后在用户需求的基础上来对所需要的数据信息进行查找,而对于该查找动作中需要查找的数据信息以及该相关查找的具体的过程就把它叫做一个信息检索过程^[2]。通常用户在网上过程中使用较多的一种信息检索的操作就是那些搜索引擎,例如 Google、baidu 等。而信息过滤指的是在海量数据中根据用户的需求对该海量数据信息进行有选择性的筛选,对于用户来说不需要的以及没有任何作用的数据信息进行过滤,这些过滤出来的信息即为能够满足用户个性化需求的数据信息。

所谓个性化服务指的就是在上面的背景下所出现的一个能够用于在巨大的数据信息面前进行信息过滤操作的衍生服务,是对信息超载问题比较好的解决的一种方式。而对推荐系统而言,他属于个性化服务中的一种,推荐系统主要通过对用户的行为信息进行分析,通过构建模型的方式来发现从这些巨大的信息中找出一些对用户有用的数据信息,然后在后面的操作中可以通过该模型来对相关数据信息做挖掘操作,借此来发现该用户可能感兴趣或者喜欢的信息,并在一些时间点或者符合某些条件的情况下为这些用户推荐这些有用的信息^[3]。推荐系统在一方面中能够从巨量的数据信息中通过一些算法或者其它技术来发现用户的一些重要的喜好信息,然后根据用户的这些喜好信息来对项目中的其它信息做相应的预测,一般通过上述这种方式就可以为该用户产生一个属于他们自己的个性化推荐服务,并且可以帮助该用户在巨量的数据信息中找到其所需要的信息;此外,还可以对公司提供个性化营销,对公司在网上的销售能力进行提升,与此同时可以通过该技术手段来获取更多的忠

实用户，能够为公司创造更多的利润价值。对于推荐系统而言，在系统中的应用目前来说还是处于起步阶段，最早也是最成功的一个应用就是将推荐算法应用在了电子商务领域，通过在该领域中应用推荐算法，能够更好的为用户推荐所感兴趣的物品，为电子商家创造更多的利润。就目前而言电子商务的规模发展的很快，越来越多的企业涉足到电子商务中，更多的商品通过电子商务进行交易，而用户的选择也越来越多，面对越来越多的商品品牌，用户选择自己喜欢的商品变得越来越困难。同时，顾客的需求往往又是不清晰的。在这个过程中，如果电子商务中的商家能够借此机会，为用户推荐其所感兴趣的物品来满足用户的实际的需求，那么就可以将这些需求转为商品的交易额，从而完成提高产品销售量的目标^[4]。随着相关的推荐技术的不断的发展以及成熟，它也逐渐的应用到了其他的领域中。

随着网络媒体的不断的发展以及成熟，人们的生活中已经慢慢的充满着数字音乐，但是在海量的音乐数据信息面前，目前一些音乐库中所涉及到的音乐曲数已经以百万甚至千万记，在这样巨大的音乐库面前，用户不可能一首一首的将这些歌曲听完，然后来决定自己喜欢哪些歌曲，这就需要通过一些信息技术来辅助我们找到自己可能感兴趣的歌曲，而音乐推荐系统就是能够为我们解决这一问题。

1.2 国内外研究现状分析

系统过滤算法是目前使用的比较多的而且也是使用的比较早的一种算法，该算法最先提出者是由 Goldberg 在上世纪 90 年代正式提出^[1]，该算法正式的发表，从一个层面来说，表示推荐系统领域的初步诞生。MIT 实验室在 1995 年的时候，根据该推荐算法理论进行了实现，并在此基础上开发了全世界第一个音乐推荐系统 Ringo^[2]，Ringo 的一个主要的目的就是向系统中的用户推荐他们可能感兴趣、可能喜欢的歌曲，并根据用户平常的听歌习惯来对歌曲进行评分，再根据这些评分来作为推荐算法的一个主要的指标。在此之后，个性化推荐系统如春笋般发展起来，到现在为止个性化推荐系统以及相关的技术在生活中随处可见。在 1995 年开始，ACM SIGKDD 在每年中都会进行一次 KDD Cup，而这年度一次的会议成为了在数据挖掘科学领域中的一项比较重要的活动之一，并在很大的程度上对数据挖掘的研究工作进行了推动。DVD 电子商务公司 Netflix 早在 2006 年 10 月份就在全世界发起了一项技术竞赛，该项竞赛拿出了 100 万美金来奖励竞赛过程中所实现的推荐系统比该公司的效果好 10% 以上的竞赛者。该项竞赛活动在全世界范围内吸引了大量的科学研究者，

在一定程度上对推荐系统的研究发展起到了很大的促进作用。在 2012 年进行举办的 **Million Song Dataset Challenge** 是用来专门针对音乐方面的推荐系统竞赛的, 在该竞赛的过程中为竞赛者提供了一个包含了 100 万个用户的相关音乐记录数据集, 让竞赛者用来对其所实现的音乐推荐算法进行测试以便检验效果^[3, 4]。个性化推荐系统除了在活动竞赛中有着发展之外, 在日常的学术研究方面也有不少的学者将许多的精力投入到其中, 许多学者通过一些一系列的会议的形式来对该方面的研究情况进行探讨, 例如 **ACM**、**WWW** 等会议中发表了许多相关的高质量学术论文。

基于内容的过滤根据其实现的原理以及所实现的目标在整体上来说还是属于信息检索领域中的一种^[5], 该过滤在最开始的时候是用于文本中挖掘出对于用户需求有用的一些基本信息, 到目前的发展情况而言, 该过滤被许多的研究者应用到了音频、视频等领域, 出现这种现象的主要原因在于机器学习技术的快速的发展。该推荐主要通过所给定的数据集来提取所需要的物品的信息, 然后根据用户在历史偏好情况来对用户可能喜欢的物品进行预测、推荐。该推荐方法一个比较好的优势就在于他不用依赖系统中其他用户的评分情况, 而仅仅只需要系统中的物品的某些信息, 根据物品的数据信息来对物品的属性情况进行提取、标注, 最终进行标签化。

本文中所开发的音乐推荐的相关的算法就是在基于内容推荐的基础上进行的, 并根据音乐网站中的特点以及特殊的基本数据信息, 将与用户习惯相近的其他用户所喜欢的歌曲按照一定的规则推荐给该用户。该过程中首先做的就是需要对系统中的音乐数据特征信息进行抽取, 然后选择相应的相似度计算公式来对这些数据信息之间的相似度情况进行计算。该领域对系统中的音乐数据信息进行社会化标签是使用比较多的一种做法, 该种做法中使用最多的数据信息就是音乐特征信息^[6-8]。

关联规则也是推荐算法中使用比较多的一种处理手段, 该方法的使用最早是在数据挖掘领域, 其最开始是用来对数据信息中的频繁项集信息进行挖掘, 而推荐规则是该关联规则中的最为重要的组成部分之一, 对于目前所有的机遇关联规则的推荐算法都有着属于各自的规则, 而这些规则基本上是相互不一样的^[9]。

协同过滤推荐在大体上可以划分为基于用户、基于物品、基于模型的集中协同过滤算法^[10]。而与基于内容不同的是基于用户的是在推荐算法领域中最早出现算法之一, 早在上世纪 90 年代中所诞生的世界上第一个推荐算法中所使用的算法就是该推荐算法^[11-12]。而基于物品的算法首次由美国的亚马孙公司进行提出的, 在该公司

中主要是利用该算法来为其所拥有的以及潜在的用户推荐该用户在历史的购物中所喜欢物品相似的物品数据信息^[13]。无论是基于用户还是基于物品的推荐，他们一个最共同的特点就是需要计算用户与用户之间或者物品与物品之间的相似度，然后根据所计算得到的这些相似度来对用户进行推荐^[14-15]。在当前的基于音乐的推荐算法中还存在着一些不足的地方，例如对于用户的收藏音乐的行为没有过多的重视，而用户的收藏往往是推荐过程中所需要占的比较重的比重。

1.3 主要研究内容

系统的研究和开发工作主要是来源于本人在学习和工作中的需要，在实际的开发过程中，主要工作内容是：

（1）对本课题中需要使用到的推荐算法、音乐系统等需求进行调研，在此分析的基础上初步确定开发的技术以及实施方案。

（2）通过与相关的使用用户做比较详细的了解沟通，然后根据对这些了解沟通以及调研的结果来对系统所需要的用户需求情况做了确定。

（3）在经过研究之后确定系统的开发方式为敏捷开发，并按照相关规范来制定规划以及按照规划来分配任务，并对任务的实际完成情况进行跟进。

（4）根据系统需求分析的结果来对系统进行设计、系统总体框架的搭建工作，设计系统数据库、系统架构、系统接口标准等。

（5）以系统设计情况为基础，利用相应的开发技术来对系统中的功能模块进行实现工作，并在系统完成之后对其功能进行测试，以验证系统的使用可行性。

1.4 本文主要创新点

（1）针对基于内容的推荐音乐较为单一，缺少用户意见的问题，通过在推荐算法中加入用户评分、意见指标，来增加其推荐过程中的权重。

（2）对音乐的特点以及用户的习惯进行分析，而不仅仅只是向用户推荐热门、流行等音乐，增加向用户推荐音乐的成功率，使得在推荐的过程中更加的个性化。

（3）在推荐的过程中将用户的收听历史记录、收藏的音乐等信息进行收集，来作为一个新的指标，进一步的分析用户兴趣点。

1.5 本文的主要结构

本文中主要研究推荐算法在音乐系统中的应用，为音乐推荐提供更加优良的服务，论文的主要结构如下。

第一章绪论，本章主要对系统的相关背景知识做了简单的介绍，且对国内外在这方面的研究情况、应用情况进行了阐述。

第二章相关技术介绍，该部分主要是对系统中所需要用到的、比较重要的技术做出说明以及所使用的相关推荐算法知识进行了简单说明。

第三章系统需求分析，该部分中主要是分为功能以及非功能两部分的需求分析。功能需求分析中主要是对系统的主要模块的实际需求进行了分析，而非功能需求分析主要是对系统中例如性能等需求进行分析。

第四章系统总体设计，该部分主要是对系统的设计工作进行阐述，首先对其架构做出说明，然后根据架构的详细设计情况和对用户的详细需求分析来对所开发的系统中功能做设计，最后对系统中所涉及到的数据库的设计工作做了简单的描述。

第五章系统核心模块实现，该部分主要是对系统比较核心模块的实现工作进行阐述。

第六章系统测试，本部分主要对系统进行测试，利用黑盒测试方法测试了各个功能模块是否能正常运行，并对系统的响应时间等性能测试较为详细说明。

第七章总结与展望，在本部分中主要是对全文的研究的情况做一个总结，并根据在研究的过程中所存在的问题以及不足提出一些展望以及有针对性的建议。

2 相关技术分析

2.1 推荐系统简介

推荐系统在严格的意义上来说一种能够给用户在进行购买某些东西的时候提供一些有价值的建议的技术或者系统软件，更通熟一点来说，推荐系统就是使用现在先进的信息技术知识来将所需要购物的用户与其所需要购买的物品信息联系起来^[16]。在本处需要注意的是，提到的物品的概念只是一个抽象并不具体的概念，他可以泛指推荐系统为用户所推荐的所有的东西信息，例如食品、音乐、家具等信息。系统中所推荐的物品的来源以及提供者想要通过该推荐进行获取利益价值的主要原因在于以下的一些方面：

（1）通过这种推荐的方式能够增加商家的销量，而商品销量的提高也是使用推荐系统的一个比较重要的原因。

（2）通过向用户推荐其可能喜欢或者满意的东西，这样可以让用户在使用服务的过程中时常感觉到惊喜，从一方面而言这样能够使该用户更加的使用我们所提供的服务。

（3）通过对数据信息的挖掘能够更好的发现用户的实际的需求，从另一方面来说就是商家能够更加的了解用户的实际需求，这样可以根据这些来产生更多的销量。

在相关的推荐系统中一般所需要实现的最为核心的功能就是为所需要的用户推荐他们可能喜欢的物品，那么就这方面来看推荐系统的必须要拥有为用户推荐预测某些物品信息的能力^[17]，而且在准确率上不能过低。通常在推荐系统中所涉及到的预测方法就是用所制定的某些标准去向用户推荐他们可能喜欢或者感兴趣的物品^[18]。总体来说的话，可以根据所使用的预测方法的差异差别，将相关的推荐系统划分分为以下的集中不同的类型：

（1）基于内容的推荐（Content-based）：该类型的推荐所实现的一个最基本的原理就是通过对系统中的用户历史的购买数据信息等，按照所计算出来的信息进行排序之后向用户推荐其可能喜欢的物品^[19,20]。物品之间是否相似或者说相似的程度如何，主要是通过计算物品中所选择出来的属性通过相似度计算公式进行计算得到的^[21]。因此在推荐系统中对这些有用属性的选择也是比较难但同时也是比较重要的一个步骤。

(2) 基于协同过滤的推荐 (Collaborative filtering): 该类型中所实现的基本原理是为系统中的相关用户推荐与其兴趣爱好、购物习惯等信息比较相近的用户所喜爱的信息^[22]。而该推荐算法中相似度的计算主要是通过计算用户对所购买的物品的评分信息。

(3) 基于人口统计学的推荐 (Demographic): 该类型所实现的基本的原理就是根据人口统计学知识来为信息上比较近似的用户推荐物品^[23]。人口统计学信息如用户的性别, 年龄, 国家等。该类推荐算法相比其他的推荐算法而言是较为粗糙的, 该类算法的一个比较重要的应用就是在现今的广告领域中, 主要是用户挖掘出一些一些广告词。

(4) 混合的推荐系统 (Hybrid): 对于上述 1、2、3 中所描述的推荐系统来说, 都不是那么的完美, 因此可以考虑将这些算法混合起来并结合他们中的优点^[24], 而混合推荐系统就是使用这种形式来进行。

2.2 基于内容的推荐系统

在前面的部分中已经对基于内容的推荐算法做了一些基本的介绍, 对于该部分中的详细的设计情况以及基本结构情况如图 2-1 所示。可以从该图中得到一些信息, 对于该推荐中最重要的 3 个因素就是对所涉及到的数据信息的一些重要的特征属性进行分析、提取等操作、对系统中的所有的用户画像基本信息情况进行学习、对该系统中的各自的物品之间的相似度按照所选择的计算公式进行结果的计算^[25]。通过分析计算来提取用户的画像, 这些所有的操作都是为了对相似度的计算做铺垫。

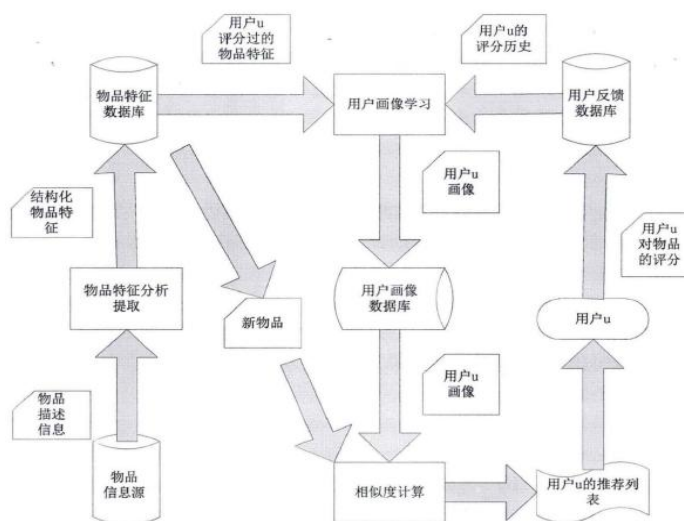


图 2-1 基于内容的推荐系统基本流程图

2.3 基于协同过滤的推荐系统

系统过滤方法的选择是对于各个推荐划分的一种比较重要的属性，与基于内容在该方面选择不同的是，基于协同的系统过滤却是通过借助于群体的力量来解决问题的一种方法^[26]，他是推荐系统应用的比较早的一种方法之一，对于该方法最早的出现一般认为是由 Goldberg 在他所发表的论文中正式提出。在一般的情况下，我们需要对一件事情做一个决定的时候，一般都是会在对该问题进行解决之前向与自己关系比较熟或者比较好的朋友进行咨询，而自己的朋友一般都会根据自己的以往的经验来为其做一些相应的推荐^[27]。在一般的情况下，人们不仅仅只会选一个朋友，有时候会同时向多个朋友进行这方面的信息咨询，然后综合这些朋友的推荐意见进行综合的选择，再来决定购买某些东西^[28]。在推荐的过程中，我们如果需要为一个用户推荐一些信息的时候，则需要先找到与这个用户有着相似购物习惯的用户^[29]，然后再把这些有着相似购物习惯的用户所喜欢的但是该用户却没有购买过的物品推荐给该用户。

对于基于协同过滤的推荐系统而言，其主要的原理就是，如果用户 u 、 v 在物品都是一样的条件下，相互的评分比较靠近，那么用户 u 对物品 i 的打分在很大的程度上面与用户 v 对物品 i 所打的分比较靠近，因此对于用户 v, u 而言他们可能就拥有比较一样的购物喜好^[30-32]，这样就可以为用户 u 推荐 v 在购买的过程中所评分比较高的一些物品信息。

3 系统需求分析

3.1 功能需要分析

本课题中，主要研究的目的是通过使用推荐算法来为音乐网站中的用户提供比较个性化的音乐推荐服务，在推荐的过程中对音乐网站中的用户行为信息进行分析，通过推荐算法的应用，为其推荐一些感兴趣的歌曲、专辑、艺术家等信息。总的来说，本系统主要包括登陆注册、专辑、艺术家、歌曲分类、歌曲标签、歌曲排行、音乐推荐等几大功能模块。

3.1.1 登陆注册模块

登陆注册模块主要是用于用户在进入系统时的登陆以及注册功能操作，用户可以通过在系统中进行注册来成为系统中的用户，这样用户就可以使用私人馆以及其他个性化推荐服务，系统可以更加容易的对用户在系统在线听歌、搜索等行为为其进行更加准确的音乐推荐。对于该功能模块的详细情况如下图 3-1 所示：

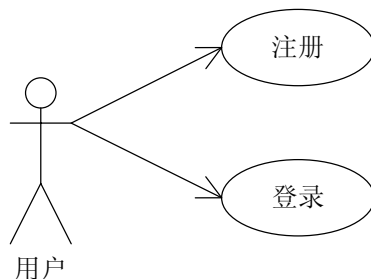


图 3-1 登录注册模块用例图

3.1.2 专辑模块

该模块主要是将系统中的音乐歌曲进行分类，通过不同的专辑来对歌曲进行区分，以使用户能够很好的对所喜欢的歌曲类型进行查看。在本专辑模块中主要包括全部专辑、国语专辑、欧美专辑、日韩专辑、DJ 舞曲等几大专辑，全部专辑中主要是收集了所有的歌曲信息，国语专辑主要是对国语歌曲进行了收录，欧美专辑主要是对欧美主要国家中的歌曲进行了收录，日韩专辑主要是对日本韩国的歌曲进行了收集，总的来说，在该模块中的详细的用例情况如图 3-2 所示：

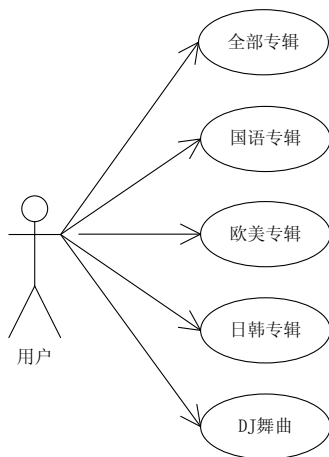


图 3-2 专辑模块用例图

3.1.3 艺术家模块

艺术家模块中将所有的歌曲按照所唱的艺术家的分类,用户可以根据所喜欢的艺术家来快速的找到其所有的歌曲曲目,在该模块中主要包括华语男歌手、华语女歌手、华语组合、日韩男歌手、日韩女歌手、日韩组合、欧美男歌手、欧美女歌手、欧美组合等几种子功能模块,其详细的用例情况如图 3-3 所示:

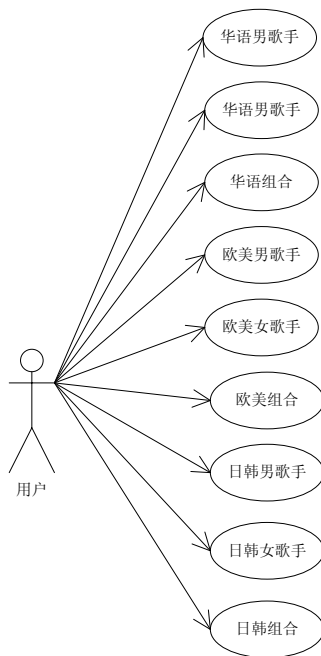


图 3-3 艺术家模块

3.1.4 歌曲排行模块

歌曲排行模块主要是对所有的歌曲按照其点击的热度、所听的人数等进行由高到低的一个排序，用户可以通过该模块清晰的对歌曲的排行情况进行了解，可以通过该功能来进行最新、最热的歌曲。在该模块中主要包括了人气排行、下载排行、收藏排行、最新上传排行等子功能模块，对于其详细的用例情况如图 3-4 所示：

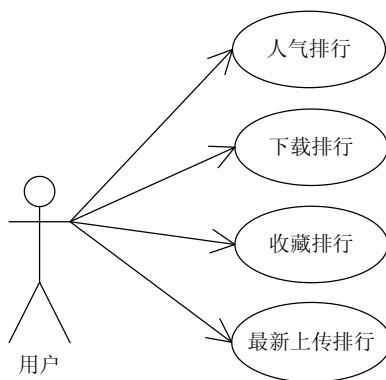


图 3-4 歌曲排行模块用例图

3.1.5 歌曲标签

歌曲标签模块中通过将歌曲的风格情况将其进行归类，可以跨艺术家、专辑等进行组合在一起，歌曲标签的制定不是简单的凭开发者的经验进行制定的，而是根据用户的喜欢以及在音乐网站中实际的使用情况进行制定，用户可以根据自己所喜欢的风格来进行选择，能够很快的找到自己喜欢的歌曲。在该功能模块中主要包括热门、年代、场景、心情、节日、语言等几个子功能模块，其详细的用例情况如图 3-5 所示：

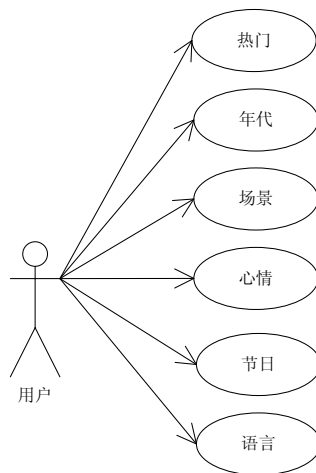


图 3-5 歌曲标签模块用例图

3.1.6 音乐推荐模块

音乐推荐模块是重点功能模块，网站中的许多的功能都是可以通过该模块来进行推荐，在页面中的一些链接中会出现推荐的一些信息，例如今日推荐、猜你喜欢等，用户在访问其他功能模块的时候，可以在页面中的适当位置发现这样的链接，通过需要进行获取。在该模块中主要是通过一些推荐算法来对该功能模块进行实现，其主要的骨架就是基于这些算法，对于这些算法的详细情况，主要在实现章节进行讲解。在该功能模块中主要包括每日推荐、猜你喜欢、热门推荐等几个子功能模块，其详细的用例情况如图 3-6 所示：

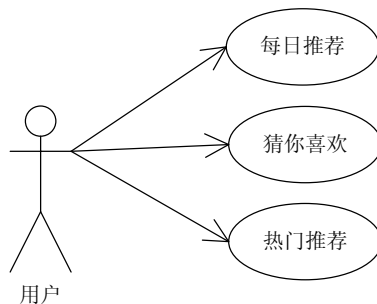


图 3-6 推荐模块用例图

3.2 非功能需求分析

在本小节中主要是通过性能要求来对非功能需求分析进行阐述。

3.2.1 性能要求

在实际的系统开发过程中，通过一些技术手段来让系统在使用能够更加的流畅，这就需要对系统的性能要求在一定范围内可控，所设计的性能的要求也是在合理的范围内。

这些经验是能够得以按客户性能要求需求顺利实施的有利保障。本项目建成后能满足用户对性能的要求，如表 3-1 所示：

表 3-1 性能要求

功能名称	性能要求
数据检索	在系统的操作过程中，比较简单的查询响应时间<3 秒；比较复杂的组合查询响应时间<30 秒
数据转换	在进行数据更新、数据删除等操作的更新的速度不应该超过 5s
数据分析	能够对有关联的数据信息检查出结果，并在建立关联的时间不应该超过 5s
音乐检索	能够对数据库中的完整性以及对于数据的一些合理的范围值进行查看，并对一些数据的重复情况、想死情况进行检查；对于这部分的时间要不超过 5s
数据录入编辑	该部分要求能够从数据库中生成数据以便录入表格来适应用户在网站中的使用的需求的变化情况
数据上载接收	在该部分中的整体的时间处理上面要小于 10s，对于比较大文件最长的处理时间不应该超过 5 min。
数据汇总	系统中简单数据的汇总的处理时间要小于 30 秒钟；比较复杂的数据汇总处理时间要小于 10 分钟，并有进度显示；对于极为复杂的数据汇总时间应要小于 1 小时。
数据库备份与恢复	处理时间最好不大于 30 分钟。在系统的运行过程中，通过对数据库的数据进行备份，以便防止数据库出现故障或者突然停电等突发情况的时候对数据进行恢复，对数据库中的相关数据信息进行备份是很有必要的操作。
用户及权限管理	根据系统中对于用户功能的划分，不同的用户权限对应于不同的功能模块。在时间的处理上应该不能够超过 30s。

3.3 数据需求分析

对于推荐系统而言其主要需要的一些信息就是相关的数据，这些数据涉及的面比较宽、比较多、比较复杂，但是他们的价值都是用来服务推荐系统中的算法，这些数据主要包括以下方面：

(1) 用户评分数据

该部分的数据信息主要是包括音乐网站中的用户对网站中艺术家、歌曲、专辑等信息进行评分，这些评分信息可以从侧面中反应该用户对于该对象的喜欢程度，是一个比较重要的指标；

(2) 用户收听记录

在音乐系统中，用户的一个最基本也是最重要的行为就是对网站中的音乐进行收听，正常的情况下，用户对于网站中音乐的评分都是隐性的，所以收听记录相比较评分而言也是比较重要的一个指标参数；

(3) 用户标签

对于该方面的数据需求是系统中一个较为重要的点，该方面数据信息的范畴主要包含音乐网站中用户对系统所有的歌曲、专辑、歌手、艺术家等进行标记的一个标签，从侧面中来看一个用户对于一个标签所打的次数的多少可以反映该用户的一些偏好的情况；

(4) 用户的收藏、分享数据

网站中的用户在看到自己喜欢的专辑、歌手、歌曲、艺术家的时候，一般情况下会对其进行收藏或者通过一些分享的途径来分享给其他的人；

(5) 朋友和近邻

网站中的用户与用户之间可以通过某种方式产生朋友或者其他的关系，如果这些关系相似度比较靠近，那么就两者之间就可以形成一种近邻的关系；

(6) 相似歌手、歌曲

该信息主要是利用相似度计算公式来计算得到的比较相似的歌曲、专辑、艺

术家、歌手。

3.4 本章小结

在本章中主要对系统的需求做了比较简单的阐述，主要从系统的功能性、非功能性析、数据等几个部分来展开。

4 系统总体设计

4.1 系统架构设计

本课题中，所研究的在线音乐网站的总体架构设计按照最新的软件工程思想，采用层次化设计方法进行设计，可以概括为存储层、基本信息收集层、数据模型层、推荐引擎、接口层、反馈收集层、前端展现（界面层）等，系统总体架构如图 4-1 所示：

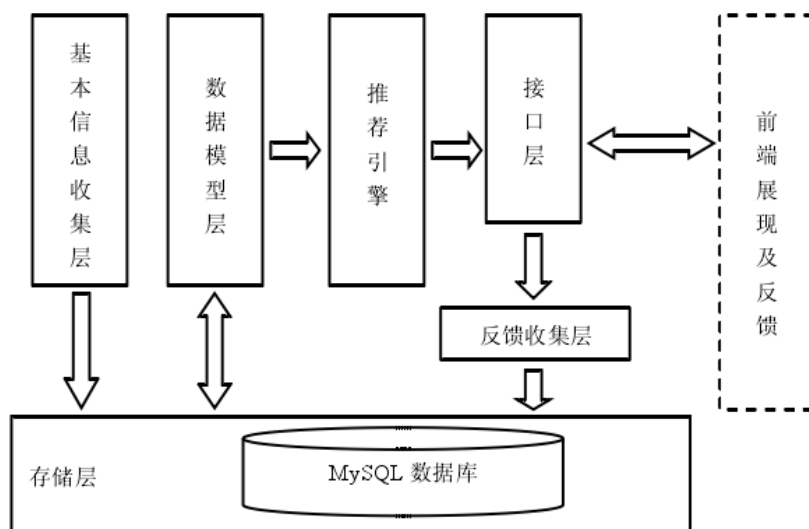


图 4-1 系统总体架构图

（1）存储层

该层主要的作用就是用来对系统中所产生、修改、删除的数据信息进行存储，这些数据信息在音乐网站中主要是包括艺术家、歌曲、专辑、用户等。目前使用的比较多的一个数据库就是 MySQL，他是一个免费并且开源的数据库，而且在性能上面有着比较好的优势，所以在本音乐网站在数据库的选择上也选择了 MySQL 数据库。

（2）基本信息收集层

该层主要的作用就是对音乐网站中的数据信息通过收集的方式来对这些数据信息进行保存，对于这些基本信息而言，其主要是包含了用户、艺术家、歌曲等基本信息。

（3）数据模型层

该层主要是通过将数据库中的原始数据信息转换为音乐网站中可以读取的数据信息，包括数据的格式以及数据信息。但是这些数据信息不能够直接的为用户产生推荐信息，而需要通过所建立的数据模型来进行解析转换。

（4）推荐引擎层

该层主要是在所建立的数据模型上面进行一些计算，通过这些计算来为音乐网站中的用户产生一些歌曲、艺术家、专辑等推荐列表。在本系统中主要采用的就是前面章节中提到的协同过滤推荐算法，采用 Slope One 以及 SVD 来对推荐引擎进行构建。

（5）接口层

该层主要的目的是用于对前端程序以及系统后台程序之间进行通信的一个入口。因为在本文中所开发的音乐网站是运行在服务器后台，在计算中所产生的数据信息需要通过在前端中进行展现，而界面中与用户之间的交互的数据信息也是需要通过前端界面传递给后台程序，而接口的作用就是用来承接上述的数据传递的一个连接。

（6）反馈收集层

该层主要的作用就是用来对音乐网站中的推荐的结果信息进行收集、反馈，以便在后续对程序进行维护以及升级。

（7）界面层

该层主要是用于对所开发的系统的功能通过可视化的界面的形式来为用户进行展现，对于不同的用户提供一些不同的工作台。而所开发的操作界面主要需要通过浏览器的形式进行展现，目前市面上的浏览器比较多，这就需要所开发的程序能够对于不同的浏览器进行相互的兼容，以便能够应对这些不同的浏览器展现操作界面时不出现兼容问题。

4.2 总体功能结构设计

在前面章节的分析以及描述中，对系统中的功能以及性能需求情况进行一个比较大概的了解，在此基础上结合本人在相关的学习过程中所积累的一些经验，对音

乐网站的整体功能结构情况设计如图 4-2 所示：

本音乐网站主要包括了专辑、艺术家、歌曲标签、歌曲排行、音乐推荐等几大功能模块。其中专辑模块中主要包括了 DJ 舞曲、全部专辑、国语专辑、欧美专辑、日韩专辑等几大子功能模块；艺术家模块中主要包括国语男歌手、国语女歌手、国语组合、欧美男歌手、欧美女歌手、欧美组合、日韩男歌手、日韩女歌手、日韩组合等几大子功能模块；歌曲标签模块主要报刊款热门、年代、场景、心情、节日、语言等几大子功能模块；歌曲排行模块主要包括人气排行、下载排行、收藏排行、上传排行等几大子功能模块；音乐推荐模块主要包括每日推荐、猜你喜欢、热门推荐等几大子功能模块。

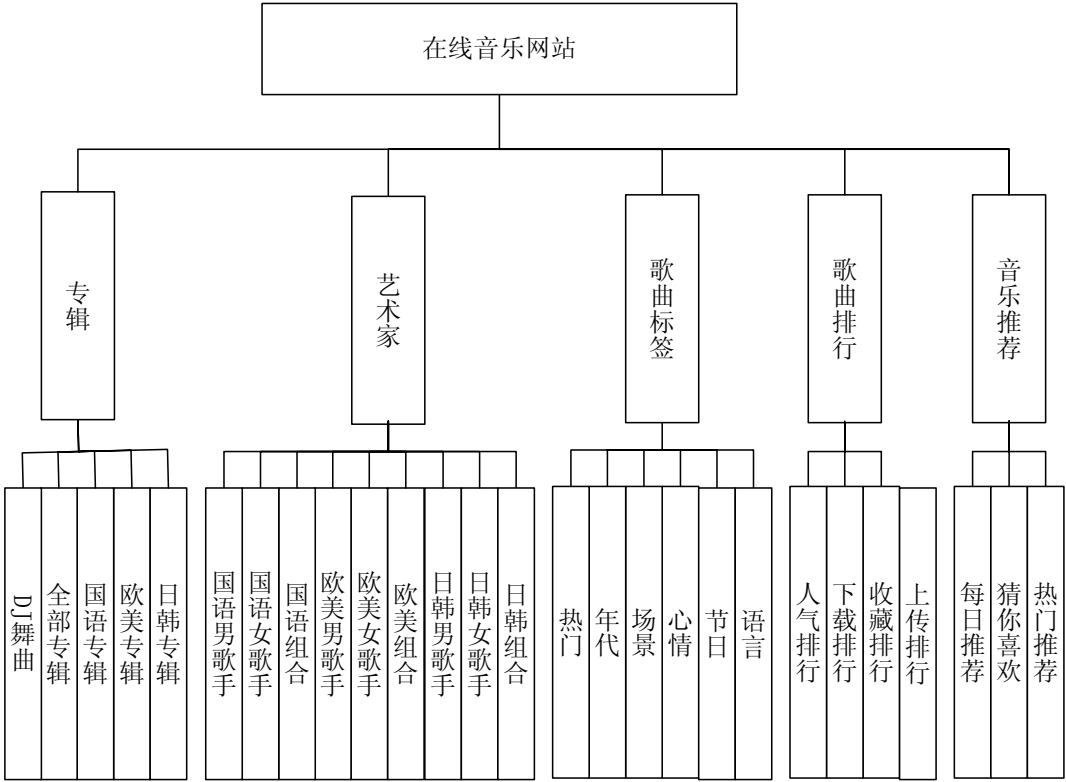


图 4-2 功能结构图

4.3 推荐模块设计

该模块主要的作用就是用来对系统中所有用户以及网站中还没有进行注册但是有一些访问行为的用户的行为的日志信息做处理，在该部分中对推荐系统模型的建立最主要的就是数据模型，然后根据选择的数据信息来产生预测模型，为系统中的用户推荐他们可能喜欢的艺术家、歌曲、专辑等。图 4-3 是推荐模块设计图。

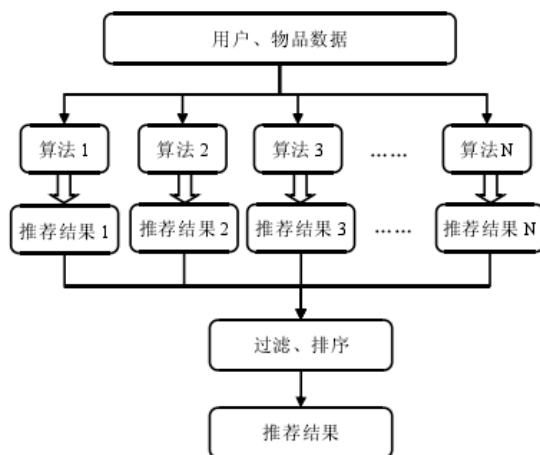


图 4-3 音乐推荐模块设计图

在音乐网站中需要对用户的行为记录信息进行收集，一般来说这些数据信息会因为算法选择的不同而收集的信息可能侧重点有些许不同，本文中主要收集的信息包含音乐网站中已经注册或者没有注册但是有访问行为的用户进行收听音乐、收藏、评分等方面的数据信息，将所收集到的这些数据信息存入到数据库中进行备份使用，然后需要根据这些收集到的行为信息转换为用户对于音乐的评分的数据信息，这些数据信息主要是通过采用评分矩阵的形式进行给出。再需要结合系统中该用户所配置的歌曲属性、文件等信息做一些推荐，通过系统中的程序将这些所推荐的信息推送给该用户。最后系统中该用户在收到推荐系统所给他推荐的这些信息之后会做出一些反应，这些反馈的结果会被系统记录下来后进一步形成该用户在音乐网站中的行为记录。

音乐网站中，推荐算法的最终目的就是需要为网站中的用户推荐一个与音乐相关的推荐列表信息，这些推荐列表一般是有序的，而进行有序排列的一些常见方式就是对推荐算法中得到的推荐信息进行排序以及过滤。

一种方法就是按照推荐算法中所预测的用户得分情况进行排序，但是这种方式的一种比较大的缺点就是可能会把比较冷门以及网站中的该用户可能并不知道的音乐信息排序在前面。在音乐的收听过程中，有一些艺术家或者歌曲的收听次数并不多，但是却有一小部分的用户特别的喜欢。

另一种方法及时按照音乐网站中音乐的热门程序来对推荐算法中所推荐的结果进行排序，在一般情况下而言，用户都喜欢许多人喜欢的音乐，但是如果按照热门的情况来进行排序或者推荐的话，那么就与个性化推荐服务有点自我矛盾。因此，

在对推荐结果进行过滤以及排序的过程中需要对两者因素进行一个相互的平衡，例如采用加权的方式来对结果进行处理就是一个比较好的方式。

系统用户在对音乐网站中的音乐进行收听操作的同时，该用户在收听该音乐之前的搜索音乐的行为，以及最终确定对该音乐进行收听之后的收听音乐和各种操作行为可以反映该用户对该音乐信息在一定程度上的喜欢。对用户推荐结果的反馈信息时，需要对用户历史的操作的行为信息进行综合的分析，就可以得到该用户对该音乐的兴趣程度情况。每一步操作过程中我们可以为每一步操作的权重情况赋予相应的权重分值，然后进行推荐的时候，我们就可以将用户在进行每一类操作过程中按照其权重分值的不一樣来赋予该类操作不同的分值，用户对每首歌曲、每个歌手产生行为过后加上相应分值，在进行推荐的时候可以将这些权重分值直接作为评分值，对于系统中推荐算法的设计详细框架情况如图 4-4 所示。

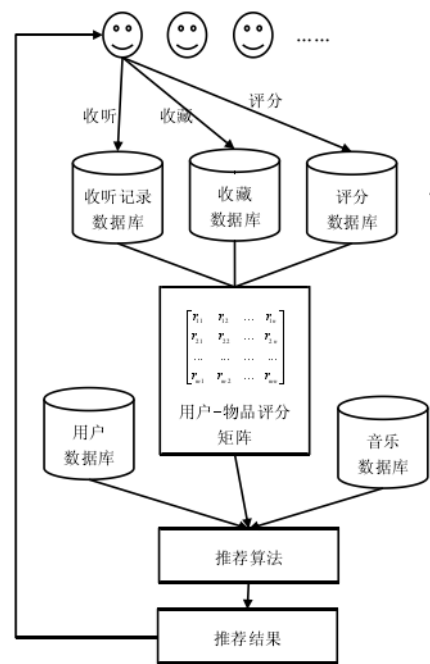


图 4-4 音乐推荐算法框图

4.4 数据库设计

4.4.1 数据库设计原则

根据音乐网站中数据操作的特点以及运营成本等多因素考虑本系统选用了 MySQL 数据库作为音乐网站的数据库，该数据库为免费并且开源的数据库，而且在性

能上面有着比较好的优势，所以本音乐网站在数据库的选择上也选择了 My SQL 数据库。MySQL 数据库能够实现对事物的控制处理，进行事务处理能够在很大的程度上实现对数据操作的安全。MySQL 数据库的安全机制也是比较有竞争力的一个因素。

对数据进行处理主要涉及到后台处理、中间处理、前置处理等三个部分，其中在中间处理部分就是数据仓库所在的位置，操作系统中数据存储器中的数据，经过数据的一系列转换和清理等行为，再保存到数据仓库中，然后系统根据业务的需要再从数据仓库中进行相关数据的获取，以便对业务功能提供数据的支撑。

本部分将会首先通过 ER 图来对系统中的数据之间的实体关系情况做一个大概的介绍，然后再通过数据库的逻辑结构设计对数据库中的一些重要的表格的信息进行详细的说明。

4.4.2 E-R 图设计

根据 3.3 小节中对系统所需要的数据分析，可以得到对系统所需要的一些数据的基本的要求，就整体而言，对于系统的数据需求以及数据之间的关系情况如下图 4-5 所示：

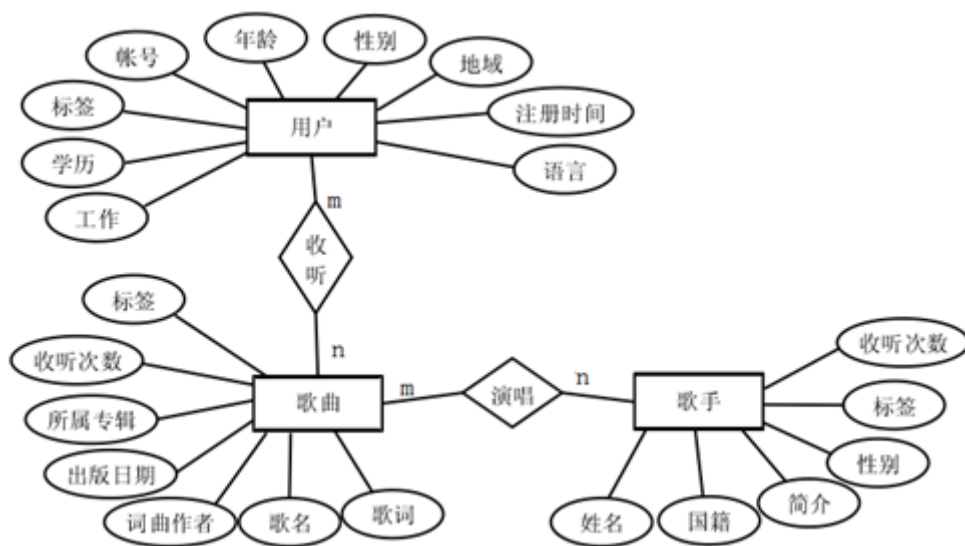


图 4-5 数据对象 E-R 图

4.3.3 逻辑结构设计

该部分中将对系统中的几张主要的表的设计情况进行简单的描述，对于这些表的具体结构如下：

用户基本信息表中主要包括音乐网站中用户相关的一些基本信息，例如编号、注册时间、姓名、性别、国籍、联系方式等数据信息，对于该表的详细的设计情况如表 4-1 所示。

表 4-1 用户基本信息表

名称	CODE	类型	LENGTH	是否允许空	是否主键
用户编号	userId	Char	5	否	是
注册时间	SignTime	date	3	否	否
性别	Sex	Char	1	是	否
生日	birthday	date	3	是	否
姓名	Name	Varchar	20	是	否
国籍、地域	Country	Varchar	80	是	否
学历	Diploma	Varchar	20	是	否

歌手信息表中主要是用来对系统中的歌手的基本信息进行存储，其中主要包括了歌手 id、性别、歌手姓名、歌手国籍地域、歌手所有歌曲播放次数、歌曲数、歌手简介等基本属性，如表 4-2。

表 4-2 歌手信息表

名称	CODE	类型	LENGTH	是否允许空	是否主键
歌手id	Id	Int	10	否	是
性别	Gender	Char(1)	1	否	否
歌手姓名	Name	Varchar (20)	20	是	否
歌手国籍、地域	Country	Varchar(30)	30	是	否
歌手所有歌曲播放次数	Plays	Int	20	否	否
歌曲数	Tracks	Int	20	否	否
简介	Intro	Varchar(255)	255	是	否

歌曲信息表主要是用来对系统中的所有的歌曲信息进行保存, 主要包括歌曲 id、歌手 id、歌名、所述专辑、歌曲播放次数、简介等基本属性, 表 4-3 为其对应的详细的数据情况。

表 4-3 歌曲信息表

名称	CODE	类型	LENGTH	是否允许空	是否主键
歌曲id	Id	Int	10	否	是
歌手id	Aid	Int	10	否	否
歌名	Name	Varchar(120)	120	否	否
所述专辑	Album	Varchar(100)	100	是	否
歌曲播放次数	Plays	Int	20	否	否
简介	Intro	Varchar	255	是	否

歌曲标签信息表主要是用来对系统中的歌曲的标签信息进行保存, 主要包括记录 id、歌曲 id、标签、歌曲 id 被打该标签的次数、该标签的同义词等基本属性, 表 4-4 为其对应的详细信息。

表 4-4 歌曲标签信息表

名称	CODE	类型	LENGTH	是否允许空	是否主键
记录id	Id	Int	10	否	是
歌曲id	Tid	Int	10	否	否
标签	Tag	Varchar	20	否	否
歌曲id被打该标签的次数	Cnt	Int	20	否	否
该标签的同义词	Synonym	Varchar	20	是	否

用户-歌曲关系表主要是用来对系统中的用户以及相对应的所听歌曲的基本信息进行关联的一个表, 其主要包括了记录 id、用户 id、歌曲 id、操作类型、操作产生时间等基本属性, 对于其详细的设计情况如表 4-5 所示。

表 4-5 用户-歌曲关系表

名称	CODE	类型	LENGTH	是否允许空	是否主键
记录id	Id	Int	10	否	是
用户id	Uid	Int	10	否	否
歌曲id	Tid	Int	10	否	否
操作类型	Type	Int	2	是	否
操作产生时间	Time	Date	3	是	否

用户-歌手关系表主要是用来对系统中的用户所收听的歌手的相关关联信息进行保存，其主要包括记录 id、用户 id、歌手 id、操作类型、操作产生时间等基本属性，其详细的设计情况如表 4-6 所示。

表 4-6 用户-歌手关系表

名称	CODE	类型	LENGTH	是否允许空	是否主键
记录id	Id	Int	10	否	是
用户id	Uid	Int	10	否	否
歌手id	Aid	Int	10	否	否
操作类型	Type	Int	2	是	否
操作产生时间	Time	Date	3	是	否

用户-歌曲标签表主要是用来对系统中的用户以及所听歌曲之间的信息进行相关联，主要包括了记录 id、用户 id、歌曲 id、标签、操作产生时间等基本属性，对于其详细的设计情况如表 4-7 所示。

表 4-7 用户-歌曲标签表

名称	CODE	类型	LENGTH	是否允许空	是否主键
记录id	Id	Int	10	否	是
用户id	Uid	Int	10	否	否
歌曲id	Tid	Int	10	否	否
标签	Tag	Int	20	是	否
操作产生时间	Time	Date	3	是	否

歌曲收听统计表主要是用来对系统中的用户收听的歌曲的情况进行统计，主要包括用户 id、歌手 id、歌曲 id、单曲重复播放次数、最后一次收听时间等基本属性，对于其详细的设计情况如表 4-8 所示。

表 4-8 歌曲收听统计表

名称	CODE	类型	LENGTH	是否允许空	是否主键
用户id	Id	Int	10	否	是
歌手id	Aid	Int	10	否	否
歌曲id	Tid	Int	10	否	否
单曲重复播放 次数	Plays	Int	20	否	否
最后一次收听 时间	Time	Date	3	是	否

歌手评分表主要是用来对系统中用户对歌手的评分信息进行保存，其主要包括用户 id、歌手 id、用户收听该歌手的次数、评分、评分时间等基本属性，对于其详细的设计情况如表 4-9 所示。

表 4-9 歌手评分表

名称	CODE	类型	LENGTH	是否允许空	是否主键
用户id	Uid	Int	10	否	否
歌手id	aid	Int	10	否	否
用户收听该歌 手的次数	Plays	Int	20	否	否
评分	Rate	Float	4	是	否
评分时间	Time	Date	3	是	否

推荐日志表主要是用来对推荐过程中所产生的一些日志信息进行保存，主要包括记录 id、用户 id、为用户推荐的 id、推荐产生的时间等基本属性，其详细的设计情况如表 4-10 所示。

表 4-10 推荐日志表

名称	CODE	类型	LENGTH	是否允许空	是否主键
记录id	Id	Int	10	否	是
用户id	Uid	Int	10	否	否
为用户推荐的id	Aid	Int	10	否	否
推荐产生的时间	Time	Date	3	是	否

用户对推荐结果的反馈日志表主要是用来对系统中的用户反馈推荐算法的日志信息进行保存，这些数据信息主要包含记录 id、用户 id、推荐 id、反馈等基本属性，对于其详细的设计情况如表 4-11 所示。

表 4-11 用户对推荐结果的反馈日志表

名称	CODE	类型	LENGTH	是否允许空	是否主键
记录id	Id	Int	10	否	是
用户id	Uid	Int	10	否	否
推荐id	Rid	Int	10	否	否
用户对推荐结果采取的行为	Action	Int	2	是	否
操作产生时间	Time	Date	3	是	否

4.5 本章小结

在本章中主要是对系统的设计工作做了简单的说明，主要是从系统架构设计、推荐模块设计、数据库设计等三个方面对其做了简单的描述。

5 系统实现

5.1 系统的实现环境

5.1.1 硬件环境

本系统中的硬件环境主要是由数据库、应用服务器、网络、客户端等部分构成。对于相关硬件的环境的详细的情况如下：

- (1) 数据库服务器：内存为 64G，最大支持 256G；4CPU；硬盘：10T*4 网卡；千兆网卡，带光钎接口。
- (2) 应用服务器：2CPU；内存为 32G；PC 服务器。
- (3) 客户端：对于客户端无过多的要求，对于一般的办公电脑以及移动终端设备都可以使用。

5.1.2 软件平台

- (1) 操作系统：Windows Server 2010。
- (2) 浏览器：在操作系统中安装浏览器。
- (3) 开发工具：在本系统中采用 eclipse+tomcat 的开发工具组合。
- (4) 其他软件方面：采用 JAVA 开发包，JDK 1.7，Android SDK。

5.2 主要功能实现

本节是对系统中主要功能模块的实现进行阐述，下面从登录注册、专辑、艺术家、歌曲排行、歌曲标签、推荐等部分来展示。

5.2.1 登录注册模块实现

登录注册功能是一般系统网站的最基本的功能，通过登录入口可以进行登录操作。首先，用户进行登录的过程中需要输入用户名以及该用户名所对应的密码信息来进行登录操作，用户可以选择“记住我”来对密码进行保存，这样在下次登录的时候就不需要进行密码的输入了。其次，在功能模块中还为用户提供了忘记密码功

能，如果用户因为某种原因忘记了自己的登录密码，那么就可以选择所提供的密码找回功能来进行密码的召回操作。最后，对于没有在音乐网站中进行注册的用户可以通过功能中所提供的新用户注册功能来做注册操作。对于该模块的实现的情况如图 5-1 所示。



The image shows a user login interface. At the top, there is a back arrow icon and the text '用户登录'. Below this, there are two input fields: '用户名' (Username) and '密码' (Password). Under the password field, there are two links: '记住我' (Remember me) and '忘记密码?' (Forgot password?). Below these links, there is a red button labeled '登录' (Login). To the right of the login button, there are two social media icons: a blue one for QQ and a red one for Weibo. At the bottom, there are two links: '新用户注册?' (New user registration?) and '申请一个新帐号' (Apply for a new account).

图 5-1 登录功能操作界面

5.2.2 专辑模块实现

专辑模块主要是对所有的歌曲通过其所属的专辑来进行归类，通过该模块就可以很方便的找到所需要的音乐专辑。在专辑列表中会对专辑的名字、发行时间、唱片公司、上架的时间等进行说明，用户可以根据其需要来选择播放该专辑。对于该模块所实现的详细的操作界面如图 5-2 所示。



图 5-2 专辑模块主操作界面

5.2.3 艺术家模块实现

该模块中主要是对音乐网站中的歌曲信息按照其所唱的艺术家的这一属性做相应的分类操作，每个用户都有自己所喜欢的歌手，通过该功能可以很方便的找到自己所喜欢的艺术家，然后点击该艺术家图标就可以进入到该艺术家下面艺人的所有歌曲曲目，通过选择所需要播放的歌曲进行播放，用户也可以对该艺术家的所有的歌曲进行播放，对于该模块所实现的操作界面如图 5-3 所示。

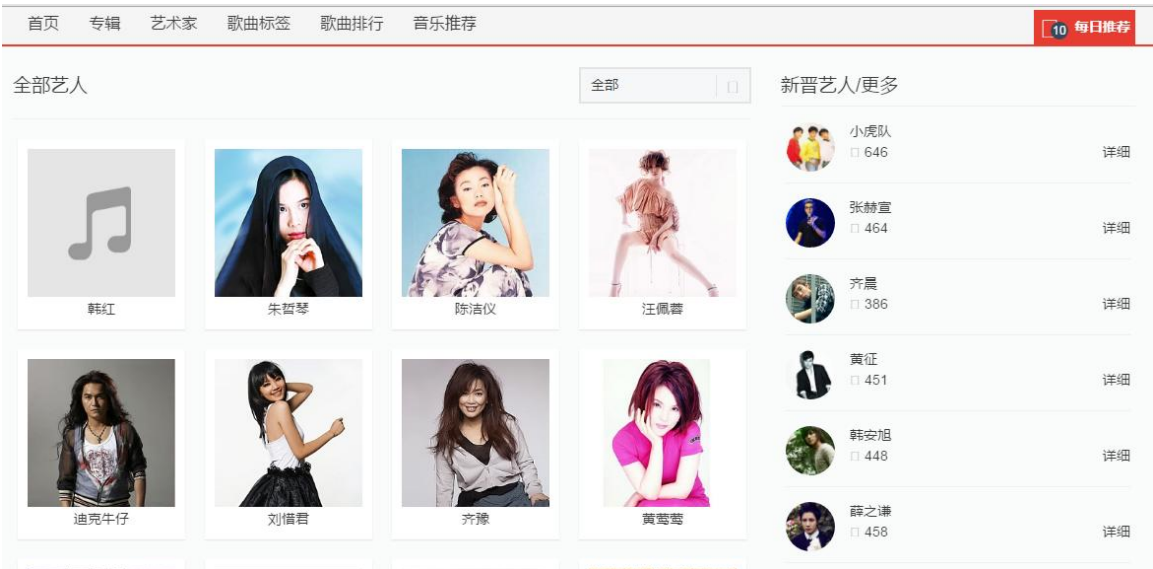


图 5-3 艺术家模块全部艺人操作界面

5.2.4 歌曲排行模块实现

对于用户而言，歌曲过多，想要找到自己喜欢的歌曲比较困难，还有的用户不知道自己喜欢什么歌曲，只是想看看最热门、听得最多的歌曲是什么，这个时候，歌曲排行模块就可以为用户实现该功能。在该模块中，为用户展示人气排行、下载排行、收藏排行等实用功能，用户通过人气排行就可以清楚的知道最近的歌曲人气情况，来选择性的听取好歌，对于该模块所实现的详细的操作界面的情况如图 5-4 所示。

榜单排行(TOP100)

人气排行

最新推荐

下载排行

收藏排行

最新上传

人气排行

1	会得	郑秀文	20,510	<div><div></div><div></div><div></div></div>
2	心不设防	陈慧琳	6,329	<div><div></div><div></div><div></div></div>
3	爱 (Live)	小虎队	4,705	<div><div></div><div></div><div></div></div>
4	以后的以后	庄心妍	3,515	<div><div></div><div></div><div></div></div>
5	光明	谭艳	2,888	<div><div></div><div></div><div></div></div>
6	如烟 (Live)	五月天	2,380	<div><div></div><div></div><div></div></div>
7	君と僕、風かぬ想い	佐橋俊彦	2,070	<div><div></div><div></div><div></div></div>
8	California Dreaming	原声带	2,036	<div><div></div><div></div><div></div></div>
9	我们不该这样的	张赫宣	1,807	<div><div></div><div></div><div></div></div>

图 5-4 歌曲排行模块中的人气排行操作界面

5.2.5 歌曲标签

歌曲标签模块中通过将歌曲的风格情况将其进行归类，可以跨艺术家、专辑等进行组合在一起，这些标签都是根据用户的喜欢的歌使用情况进行制定，用户可以根据自己所喜欢的风格来进行选择，能够很快的找到自己喜欢的歌曲。对于该模块中，用户选择相应的标签之后就会对符合该标签的歌曲的列表进行展现，在歌曲列表中，会有歌曲的名字、演唱者、点击次数、分享、播放、收藏等功能，其详细的实现情况如图 5-5 所示。



图 5-5 歌曲标签中操作界面

5.2.6 推荐模块实现

推荐算法主要的一些操作步骤包括对音乐网站中与用户相关基本信息的收集，根据所收集到的数据信息按照推荐模型来进行推荐，根据用户对推荐结果的反馈信息进行收集处理等几个部分。用户基本信息的收集主要是产生在系统用户在进行新用户注册、相关数据信息输入、数据信息修改等操作中产生，本部分中将不对这些进行阐述。而据用户对推荐结果的反馈结果信息进行收集处理，是音乐网站系统中的最为主要的数据信息的源头，该部分主要是由系统中的数据预处理部分进行完成，然后根据处理的结果建立所需要的数据模型。而推荐模块中最为重要的推荐算法主要是用于根据数据信息来产生推荐接口，该部分中所实现的程序可以通过接口的形式进行使用。在本音乐网站的开发过程中主要用到了 Slope One 以及 SVD 两个算法，本小节中将会主要对系统中的数据预处理模块以及这两种算法的实现的情况进行说明。

(1) 数据预处理模块实现

目前市面上大多数的推荐系统主要是利用系统中的评分数据来对推荐结果进行评分预测，但是在线音乐网站中的用户却比较少对网站中的歌曲、艺术家、专辑等做评分，大部分的用户都是仅仅进行收藏、收听等有关于非显性的数据，所以在对数据的处理过程中通常是需要对歌曲、专辑、艺术家的收听次数信息进行统计，然后按照所指定的一定的规则进行评分形式的转化，在音乐推荐系统中通常会根据歌

曲、歌手、专辑、艺术家来进行层次的推荐。本系统中所实现的代码对于数据预处理的类主要包括 CMySQL Opea 和 CPre Processing 两个类。在线音乐网站中相关的数据信息一般都是非常大的，基本上在百万级别之上，这样系统频繁的对数据进行查询检索的过程中将会浪费掉比较多的时间。所以本在线音乐网站中引入 B 树来对这些数据信息存储并穿件索引，该程序中所涉及到 CBTTree 和 CData Type For Btree 两个不同的数据类，对于系统中所开发的 B 树的类主要是按照通用类的形式进行处理开发，后面的程序中还会用到该类。对于数据缓存的处理情况如图 5-6 所示。

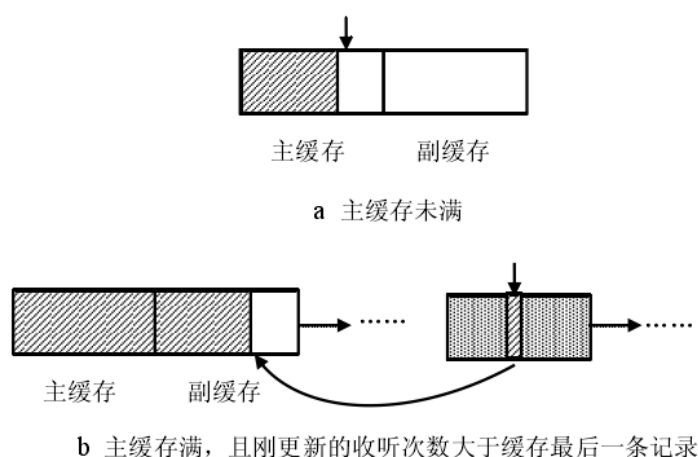


图 5-6 缓存使用示意图

在实际的处理过程中，用户与系统中歌曲、艺术家、专辑等关系数据信息主要是按照数组链表的形式存在于系统的内存中，为系统的每一个用户创建属于各自的链表。对该链表中的首个字节用来做为整个链表中的缓存信息进行存储。该链表中将会对该用户所收听次数最多的歌曲、专辑、艺术家等信息进行保存，然后按照降序的处理形式进行保存，这样就可以在查找过程中及缓存中对该数据信息进行查找，提高了系统中相关数据信息的查询速度。在缓存实现过程中我们将会把该部分的缓存平分为两个部分，这两个部分中分别叫做主、副两个缓存形式。在主缓存还没有存满的时候，当有新的数据信息需要存入的时候将会直接将数据信息加入到该末尾的部分，可以参照图 5-1(a) 部分。当在主缓存部分使用完了之后，系统将会创建新的节点以便能够对新的数据信息进行保存，在系统中如果又有新的数据信息需要增加的时候就可以将该数据加入到副缓冲区中，对于该步骤的详细的情况可以参照图 5-1(b)。

对于系统中所实现的 B 树的数据存放数据结构体的定义情况如下所示：

在该 B 树中定义了数中的子树、父节点、值、树中所添加的关键字的个数，通过

定义该 B 树的数据结构，后面的数据可以直接将数据存储在该 B 树中，然后通过父节点、子节点等来进行链接。

```
struct tree Node
{
    struct tree Node** children; //子树地址,动态开辟 order + 1 个空间
    void** key;
    struct tree Node* parent; //父节点
    unsigned int key Num; //当前已添加的关键字个数
};
```

下面展现的是对所指定的关键字信息进行查询的主要函数所实现的代码情况，以及函数的参数值以及返回值：

key: 待查询的关键字；

sub Tree: 在程序查找失败的条件下，关键字所需要插入到的节点的地址；

at: 该参数的值表示的是所需要插入或者删除的关键字在所对应的数据信息的数组中的所对应的下标中；

返回值: 在程序查询成功的时候所返回的关键字的地址，如果查询失败的话就返回 null 值。

```
void* CBTree::Query(void *key, unsigned int* at, struct tree Node** sub Tree)
{
    //首先对于 key 以及 m_p Root 根节点为空的情况，直接返回空值
    if(NULL == key | NULL == m_p Root)
        return NULL;
    //赋初始值
    struct tree Node *ptree = m_p Root;
    int l, h, m, r;
    //对关键字信息进行查询，查询的方式是通过二分法算法来进行查询，当查询到所指定
    //定的关键字之后，中断该循环，返回查找到的值
    while(1) {
        l = 0;
        h = ptree->key Num - 1;
        m = (l + h) / 2;
        //通过二分法方式来查找结点
```

```

while(l <= h)
ptree{ m = (l + h) / 2;
r = m_pclass Cmp->Compare(key, ptree->key[m]);
if(r < 0)h = m - 1;
else if(r > 0)
l = ++m;
else{
if(NULL != sub Tree)(
*sub Tree) = ptree;
if(NULL != at)
*at = m;
return ptree->key[m]; //查找成功，返回关键字地址
} }
if(NULL != ptree->children && NULL != ptree->children[m])
ptree = ptree->children[m]; //还未找到，查找下一结点
else
break;
}
if(NULL != at)
*at = m;
if(NULL != sub Tree)
(*sub Tree) = ptree;
return NULL;
}

```

对于新入的数据需要在 B 树中进行插入操作，而代码中所提供的插入数据的过程主要包括了三个函数：Insert、Insert At、Split。其中函数 Insert 主要是用来对所需要插入到树中的关键字是否已经在原有的树中存在，如果没有存在的话，则返回该树中的位置地址。函数 Insert At 的作用主要是用来对关键字的信息进行插入的操作，如果该节点中的信息已经满了，则需要对该节点进行分裂操作，而函数 split 的作用就是用来对节点进行分裂操作。前两个函数实现如下：

Insert 函数的参数及返回值：

key：待插入关键字指针；返回值：如果所输入的参数值无效，那么返回 null 值，

如果所需要插入的关键字在 B 树中已经存在了，那么就会返回该位置的节点指针信息。插入成功完成之后将会返回该关键字信息。

```
void* CBTree::Insert(void *key)
{
    //首先对于 key 以及 m_p Root 根节点为空的情况，直接返回空值
    if(NULL == key | NULL == m_p Root)return NULL;
    //如果该树无节点，则直接返回 key 值
    if(0 == m_p Root->key Num)
    { m_p Root->key[0] = key;
      m_p Root->key Num++;
      m_n Depth = 1;
      return key;
    }
    //构造树，进行初始化
    struct tree Node *p Tree;
    unsigned int at;
    void* key Addr;
    //查询树中 key 值的位置
    key Addr = Query(key, &at, &p Tree);//
    //如果 key 为空，则将该位置地址信息进行返回，然后调用 Insert At 函数在该位置中插入 key 值
```

```
if(NULL == key Addr)
{ Insert At(p Tree, key, NULL, at); //执行插入操作
return key;
}
return key Addr;
}
```

函数 Insert At 无返回值，参数意义如下：

p Tree: 待插入数据的子树节点；

key: 待插入关键字指针；

child: 关键字

key 的右子树;

at: 表示的是所应插入到的数组中的下标的具体位置

成后 key 应该在 p Tree->key[at]。

```
void CBTree::Insert At(struct tree Node* p Tree, void* key, struct tree Node*
child,unsigned int at)
```

```
//对于所传入的树 p Tree 为空树或者 key 值为空，则直接返回
```

```
{ if(NULL == p Tree | NULL == key)
```

```
return;
```

```
unsigned int n = at, j;
```

```
//如果所对应的结点没有满，则直接插入
```

```
if(p Tree->key Num < m_cn Order)
```

```
{ for(j = p Tree->key Num; j > n; --j)
```

```
p Tree->key[j] = p Tree->key[j - 1];
```

```
if(NULL != p Tree->children)
```

```
{ for(j = p Tree->key Num; j > n; --j)
```

```
p Tree->children[j + 1] = p Tree->children[j];
```

```
p Tree->children[j + 1] = child;
```

```
}
```

```
p Tree->key[j] = key;
```

```
p Tree->key Num++;
```

```
return;
```

```
}
```

```
//如果关键字已满，则需要对树(p Tree,进行分裂，然后调用分裂树的函数 Split 来对
树进行分裂
```

```
Split(p Tree, key, child, n);
```

```
}
```

在节点的分裂过程中，该函数所实现的过程比较长，所以在此处对所实现的伪代

码信息进行说明，该程序涉及到的具体的参数意义的详细情况如下：

p Tree: 待分裂的子树；

key、child、at 的意义同函数 Insert At。

Split(struct tree Node* p Tree, void* key, struct tree Node *child, unsigned int at)

//如果所需要分裂的树就是根节点，则需要新建立一个 node，然后取值 mid

if(p Tree == m_p Root)

new Root = new struct tree Node;

m_p Root = new Root;

mid = (m_cn Order + 1) / 2;

p New RChild = new struct tree Node;

if(at < mid)

p Tree 中下标[mid, order)的关键字及子结点移入 p New RChild;

key 插入到 p Tree->key 数组中 at 下标处

child 插入到 p Tree->children 数组中 at+1 下标处

else

p Tree 中下标[mid + 1, order)的关键字及子结点移入 p New RChild;

key 插入到 p New RChild->key 数组中 at-mid 下标处

child 插入到 p New RChild->children 数组中 at-mid+1 下标处

把 p Tree->key[mid]、p New RChild 插入到父结点

对于数据的预处理相关的程序在一次处理的过程中只能对一个用户的数据进行处理，在处理的过程中会将音乐网站中的用户对歌曲、专辑、艺术家的收听次数进行计算，然后将处理之后的数据存在上一章中所设计的对应的数据表中，对于该程序的处理的流程如图 5-7 所示。

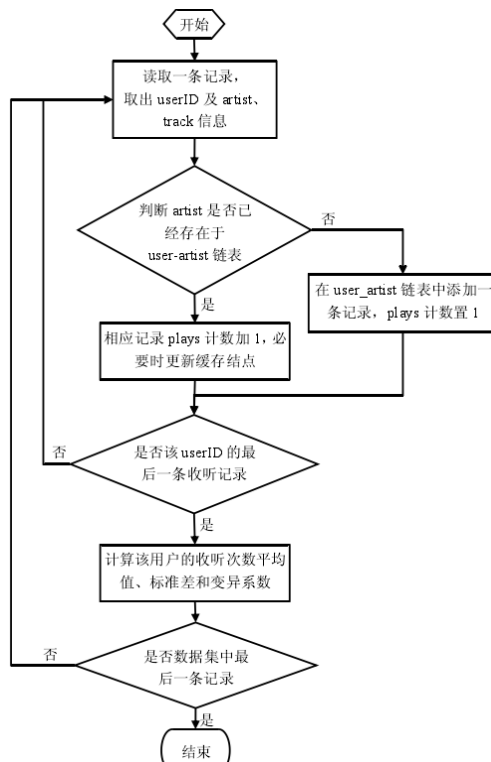


图 5-7 数据预处理流程图

用户与系统中歌曲之间收听的统计情况，对于同一个系统用户而言，其所对应的统计记录数也许会比较大大，这样在后面处理完之后也许就不会再进行使用了，所以在处理完成这样的一个数据之后则需要将这些数据信息进行保存。对于上述情况而言，也按照之前的情况创建索引树的话，那么将会花费比较多的时间以及计算机资源在该树的维护中。在本部分的实现过程中，我们将会按照前面部分所提到的算法中对数据存储方式的在本部分中进行实现，在使用代码进行实现的过程中主要是通过一个单向的链表信息来对这样的一个结构信息进行保存，对于该链表的数结构情况可以参照图 5-8 所示。在该结点中最多可以保存 N 首歌曲或者 N 个专辑或 N 个艺术家的相关的收听信息。

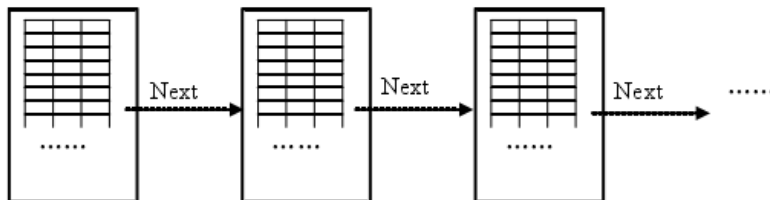


图 5-8 user-track 结构示意图

(2) Slope One 模块实现

Slope One 模块的实现包括类 CRec Base 、 CSlopeone Predict 、 CBTree 和 CDatatypeDiffForBTree。其中 CRecBase 类是属于相关的数据模型类，其主要功能就是负责对预处理之后所生成的各种数据模型。在系统各个算法模块中都会在计算的过程中使用到相关的数据模型，因此该部分中需要设计一个专门用户这方面管理的类，使得它能够独立于其它的算法模块，当其它算法需要根据数据信息进行计算的时候，该类会负责将所对应的数据地址信息传递给所对应的数据模块，通过这种方式可以避免对一些已经读取过的数据进行再次读取。而当实际的数据发生一些修改变化之后，我们仅仅只需要在数据的相关类中进行相应的修改即可。

Slope One 的一个比较大的优势在于其不需要对预测模型进行相应的训练，而只需要对评分差矩阵 D 进行相应的计算即可。表 5-1 展现的是一个评分差矩阵相关的具体的例子，其中主要包括了 Moby、Pearl Jam 、Portishead 以及 Incubus 歌手与其相关的评分差信息。

表 5-1 歌手评分差距表

	Moby	Incubus	Portishead	Pearl Jam
Moby		-0.40234 256	0.18323 322	-0.10887 248
Incubus	0.40234 256		0.64224 232	0.09865 223
Portishead	-0.18323 322	-0.64224 232		-0.43498 223
Pearl Jam	0.10887 248	-0.09865 223	0.43498 223	

从该矩阵中可以看到对角线中两侧的先关的数据信息是互为相反数，在实际的数据的保存操作中我们可以只对上三角的数据信息进行保存，这样可以对实际的存储空间进行不少节约。需要注意的是，对于系统中的艺术家而言并不是每一位都会有一些共同收听或收藏或下载过的系统中的用户，也就是说在该矩阵只中有一部分的数值可能为 0，对于这部分的数据在实际的计算中没有作用，但是该部分数据却在使用中占用了比较的空间。如果对系统中所有的艺术家都采取两两之间做比较，那么对于系统内存而言将会占用很大，所以在实际的操作中我们需要对这些稀疏的矩阵进行处理以便节省不必要的内存浪费。在图 5-4 中展现的就是我们经过改动之后的矩阵的系统结构情况。

在评分差矩阵中，需要做一些如下的规定以便在后面的计算中更加方便： D_i 记为 i 的具体评分差结点，在实际的矩阵中所展现的就是一行数据信息， d_{ij} 表示的是物品 i 对 j 的相应的评分差具体值，因此对于 D_i 而言， $j > i$ 是必然存在的一种关系；而且对于 D 中的每一行数据，对 ID 大的物品都是在最后的行中进行保存。

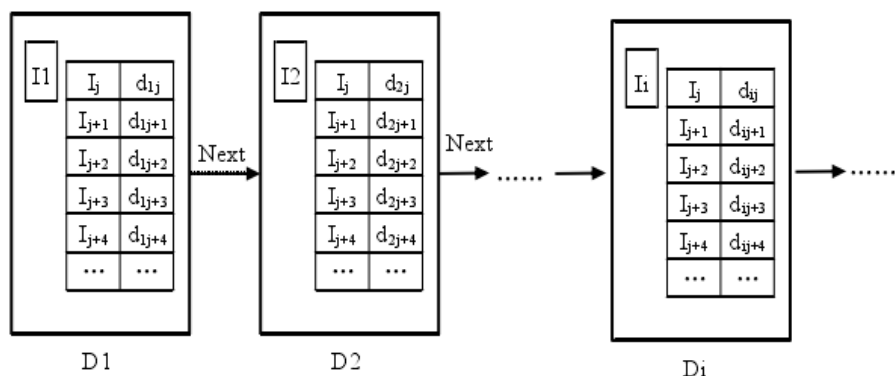


图 5-9 评分差距结构图

在计算过程的实际操作中，因为在进行推荐的过程中需要对相互有评分信息的用户具体数量作为一个乘子，所以在矩阵中实际保存的是两个艺术家相互之间评分的所有差值的总和。对于图 5-9 中所提到的评分差矩阵数据结构方面实现代码情况如下面展示：

```
struct diff_cell
{ unsigned int allUseriid; // 所计算的物品中对应的 id 值
  unsigned int all_num_user; // 对该评分差做出贡献的系统中的用户的数量
  int score_sum_ofDiff; // 评分差的求和
};

struct diff_chain
{ struct diff_cell dif_one_item[512]; // 使用数组的形式来对 diff_cell 进行保存
  unsigned int num_cell; // 数组中已经使用过的数据的多少
  struct diff_chain *pre, *next; // 使用两个指针来保存 pre 以及 next 数据信息
};
```

在上述结构体中的 `diff_cell` 所需要保存的主要数据信息是与评分差相关的数据信息，而 `allUseriid` 是用来对所计算的物品中对应的 `id` 值进行保存，结构中的 `score_sum_ofDiff` 是所计算的评分差的求和，当然在该部分中主要考虑的是使用整数来进行评分，所以保存的数据类型是 `int`，如果在实际的计算中需要使用的评分是小

数，那么可以将该类型改变 `doube`，该结构中的 `all_num_user` 则是用于保存对该评分差做出贡献的系统中的用户数量。在表 5-1 中展示中，所描述的歌手 Moby 相比较 Incubus 的具体的评分差为(754, -103, 256)，而对于 Incubus 而言，其所对应的 ID 的具体的值为 754，如果将这个具体的结构信息在数据集中进行解析的话，那么表示的就是在系统中有 256 个用户在历史的收听过程中对上述的两个歌手的歌都收听所，但是相比于 Moby，评分差的值为-103。

在结构体 `diff_chain` 中通过链表的方式来对一些 `diff_cell` 的结构信息进行了保存。但是为了能够更好的提高运算的效率，在每隔结构中都会申请一个 512 大小的存储空间数组来对该数据信息进行保存，而 `num_cell` 所代表的具体信息就是表示该数组中已经使用过的数据的多少。

对于上述所描述到的评分差矩阵的具体的创建相关流程情况可以见图 5-10 所示。在该矩阵建立好之后，需要取物品 i 所对应的所有的评分差数据信息，就可以通过以下两个相关的步骤进行执行：第一步需要对 i 前面所有的数据信息记性查询，如果在查询的过程中发现有相关与 i 的评分差的时候，则需要在计算中将该值赋值为其相反数。第二步需要在矩阵中取出 i 行的具体的数据信息，可以使用 Iu 来代表 u 在历史的时间中所评论过的全部物品，使用 $r[u][k]$ 来代表 u 对物品 k 在进行评分过程中的具体的评分值。

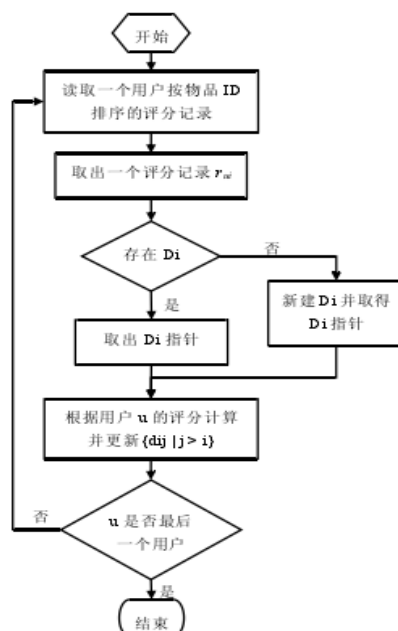


图 5-10 评分差矩阵创建流程

(3) SVD 模块实现

接下来介绍 SVD 算法的实现。除了公共部分以外,该模块只有一个类 CSvd Rec。该算法开始的标志是通过在系统中选用随机的特征向量作为初始值,然后进入到程序中进行循环迭代,这就需要一个结束的标志来对整个的循环迭代进行一个终止操作,在计算迭代的过程中如果算法的准确率不再发生变化的时候,就将作为迭代的终止条件,在本文所开发的算法中主要是采用 RMSE 来作为对准确性评价的一个标准。

与 Slope One 相比不一样的是, SVD 在进行计算的过程中则是需要根据所需要的数据集对预测模型进行预测,然后再使用训练得到的相关预测模型来对数据集中的数据信息进行预测,计算表其 RMSE 值,然后通过不断的改进所开发的模型以便来使得预测的准确度更好,对于该算法的一个比较简单的流程情况如图 5-7 所示。

在迭代的过程中可以设置 step 为程序迭代的最大值,如果程序在进行迭代的过程中达到该值时还没有找到最优的解的话,那么程序将会自动的终止,使用符号 U 来表示运行的用户的数据集合,使用符号 I_u 来表示 u 曾经在系统中评论过的所有的歌手的信息集合,使用符号 d 来代表衰减速率,使用符号 b_u 、 b_i 分别代表用户、物品偏置,使用符号 p 、 q 来分别代表用户、歌手特征矩阵,具体该算法的计算过程可以参照下面的图 5-11 所示:

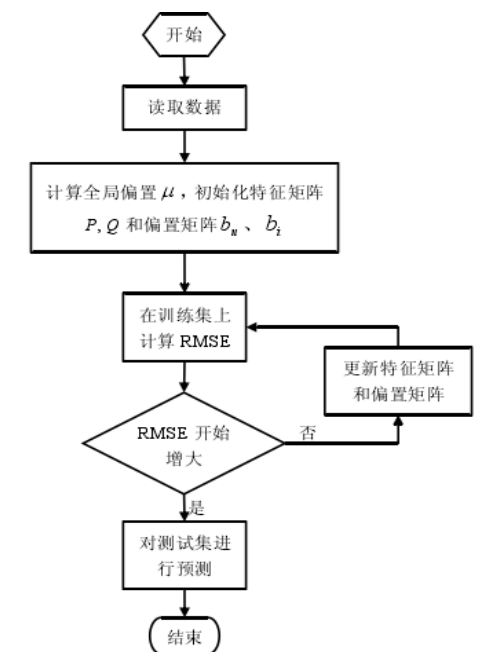


图 5-11 SVD 算法流程图

```

for iter = 0 to step
    n = 0
    rmse = 0
    for u in U
        for i in Iu
            pui = Predict Rate(u, i) //估算用户 u 对歌手 i 的评分
            eui = rui - pui
            n++
            rmse += eui * eui
            bu[u] += alpha * (eui - lambda * bu[u]) //对偏执进行更新
            bi[i] += alpha * (eui - lambda * bi[i])
            //通过从 0 到 F 循环来更新特征向量
            for k = 0 to F
                p[u][k] += alpha * (q[i][k] * eui - lambda * p[u][k])
                q[i][k] += alpha * (p[u][k] * eui - lambda * q[i][k])
        alpha *= d

```

函数 Predict Rate 原型为：float Predict Rate(unsigned int user, unsigned int item);

对于该函数而言，其主要的目的就是预测 user 对 item 的评分值，其详细的参数情况以及相对应的返回值的说明如下：

user: 用户

ID; Item: 项目 ID;

返回值: user 对 item 的评分估值。

函数实现如下

```

float CSvd Rec::Predict Rate(unsigned int user, unsigned int item)
{
    //初始化所需要的参数值
    int i;
    double d;
    float ret;
    if(user >= (unsigned int)m_n User Num | item >= (unsigned int)m_n Item Num)
        return 0.0;
    //均值加用户、物品偏置
    ret = (float)(m_f Ave + m_p Bu[user] + m_p Bi[item]);
    d = 0;

    if(1 < m_p User Rate[user].num_filled)

```

```

{
    for(i = 0; i < m_n Dimension; ++i)
    {
        d += m_pp User Pro[user][i] * m_pp Item Pro[item][i];
    }
    ret += (float)d;
}
if(0.1f > ret)
    ret = 0.1f;
else if(6.0 < ret)
    ret = 5.9f;
return ret;
}

```

(4) 界面实现

在上述对于推荐模块中的核心的实现进行阐述，根据其进行实现得到操作界面如图 5-12 所示。音乐推荐模块中主要是对相关的音乐信息进行综合的推荐，可以对按照推荐进行整体的排序、对单曲进行推荐、对专辑进行推荐以及每日推荐。



图 5-12 音乐推荐模块操作界面

5.3 本章小结

在本章中主要是对系统的核心模块的实现工作进行了大致的阐述，首先是对系统实现的软硬件平台进行了描述，然后依次对数据预处理、Slope One、SVD 等几个主要的模块的实现工作进行了阐述。

6 系统测试

6.1 测试方案

网络页面是软件 and 用户达成一致的一个展示界面，网页界面的好坏决定这用户对软件体验效果的满意程度。好的界面效果能够让使用者心情舒畅，检测的时候包括每一个按钮，菜单和各种热键。

测试目标：（1）简单性（2）标准性（3）辅助措施（4）合适性（5）友好与协调（6）主页位置（7）热键组合（8）可靠性（9）网页间的交流

功能测试是检测系统时最核心的步骤，它会检测出系统是不是能满足使用者要求，能不能正常运行。功能测试要全面，覆盖各个界面。

系统测试过程：

系统在进行检查的时候，就是模拟系统正常指挥生产的正常程序，每一个生产过程的系统反应都要逐一排查，每一个硬件的按键都要排查，按键以后功能能不能顺利的实现。还有检查系统分析错误信息的能力，故意输入错误的信息，看系统能不能提示我们输入的错误，还有就是输错以后，系统的指令会不会发生错误。

6.2 测试用例

（1）利用黑盒测试方法

通过对系统各模块功能测试，具体内容和结果测试如表 6-1 所示。

表 6-1 系统功能模块测试表

测试模块	测试内容	测试方法	测试结果
登录注册	登录	模拟用户登录操作，看在用户名以及秘密出错、正确等情况下的运行情况，看能否通过忘记密码、注册等来进行链接跳转。	测试结果正常
	注册	在注册界面中输入相应信息，能否进行注册。是否对已经存在的账号进行提示	测试结果正常
	每日推荐	每天能否在各个页面中根据推荐算	测试结果正常

推荐模块		法来对歌曲列表进行推荐	
	猜你喜欢	通过不同的点击、收听、收藏等操作，然后点击猜你喜欢功能，看能否根据这些信息给用户一些智能预测	测试结果正常
	热门推荐	查看系统中所推荐的音乐的热门程度	测试结果正常
专辑模块	国语、欧美、日韩、DJ舞曲	选择这些功能查看能否展现相应的专辑；点击播放专辑之后能否对该专辑中的所有歌曲进行播放；点击该专辑之后能否进入到专辑的介绍界面	测试结果正常
艺术家模块	华语、欧美、日韩男女歌手	选择这些功能查看能否展现相应类别下的歌手；点击歌手之后能否进入到该歌手的界面中；能否对该歌手下的歌曲进行收藏、播放等	测试结果正常
歌曲标签模块	热门、年代、场景、心情、节日、语言	选择所对应的标签之后，在界面中能够展现相应歌曲；能否对该标签下的歌曲进行收藏、播放、下载等	测试结果正常
歌曲排行模块	人气排行	根据歌曲的播放、收藏、下载等情况进行综合的排行；能够对列表中的歌曲进行播放、下载、收藏等	测试结果正常
	下载排行	根据歌曲的下载量进行排行；能够对列表中的歌曲进行播放、下载、收藏等	测试结果正常
	收藏排行	根据歌曲的收藏量进行排行；能够对列表中的歌曲进行播放、下载、收藏等	测试结果正常

(2) 用户界面测试

该方面的测试目的是用于检测所开发的系统的操作界面能否向用户提供正常的输入等操作，例如用户能否在操作界面中的输入框中输入编辑信息，能够在点击所对应的按钮之后实现所需要的功能等等，对于该部分的测试步骤以及所选择的测试的功能的结果情况如表 6-2 所示。

表 6-2 界面测试表

序号	测试内容	测试方法	测试结果
1	界面输入	是否正常	光标可以正确定位到输入域中，输入数据有效，输入顺序合理
2	界面显示	在软件的测试运行中检查界面显示是否正常	在800*600分辨率下，界面显示正常
3	界面文字与提示	检测系统运行中是否有文字异常	文字清晰，表述正常

6.3 性能测试

一个系统能不能正常的工作，质量能不能过关，都要取决于系统性能检测，它会对各种条件下系统的运行状况进行检测，保证系统的运行平稳。性能测试有负载测试和压力测试，首先说负载测试，负载测试主要是在原有的基础加一些工作，然后看看在众多任务执行的情况下，系统还能不能正常运行。压力测试主要是确定系统工作的一个及限度，看大于这个极限以后能不能正常，让系统拥有更大的服务范围。

因为该网站使用的实际场合是音乐歌曲方面，涉及到大量音乐文件的上传、加载、下载等操作，对性能要求比较高。在操作界面中还涉及到各个浏览器的兼容性问题，在每个不同的浏览器中界面需要能够正常的显示，所以系统应该具有兼容性，用测试工具 LoadRunner8.0 同时测试反应时间和负载能力，实际测试结果如表 6-3 所示。

表 6-3 系统性能测试表

测试项	测试事务	测试内容	测试结果
响应时间	查询歌曲时间	最长响应时间(s)	4.547
		最短响应时间(s)	1.132
		平均响应时间(s)	1.339
	下载/上传歌曲时间	最长响应时间(s)	3.821
		最短响应时间(s)	3.364
		平均响应时间(s)	3.359
	推荐歌曲列表生成时间	最长响应时间(s)	10.352
		最短响应时间(s)	3.686
		平均响应时间(s)	5.862

	推荐专辑生成时间	最长响应时间(s)	8.689
		最短响应时间(s)	3.236
		平均响应时间(s)	4.623
负载	30个用户	系统工作72小时	17%
	60个用户		26%
	100个用户		43%

注：本处的负载指的是用户同时下载大量歌曲所使用的资源占比

对于上述查询歌曲、下载/上传歌曲、推荐歌曲列表生成、推荐专辑生成等的响应时间情况可以详细见图 6-3，在该图中纵坐标表示的是时间(s)。

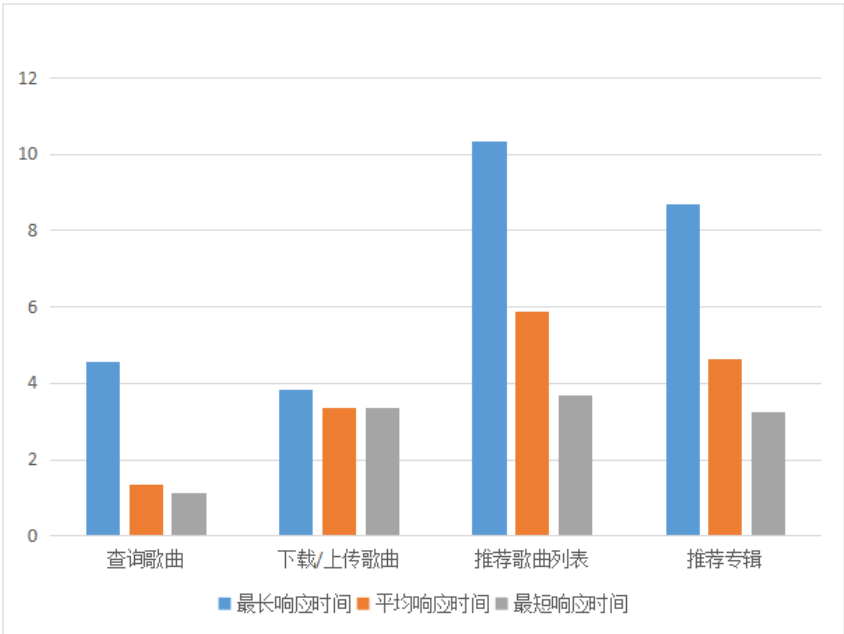


图 6-1 响应时间图

对于负载的情况可以详细见图 6-2，在该图中横坐标表示的同时进行下载、上传歌曲的用户数量，纵坐标表示的是系统资源使用的情况（百分制表示），从该图中可以看到随着并发数的增多，资源的使用逐渐增多。

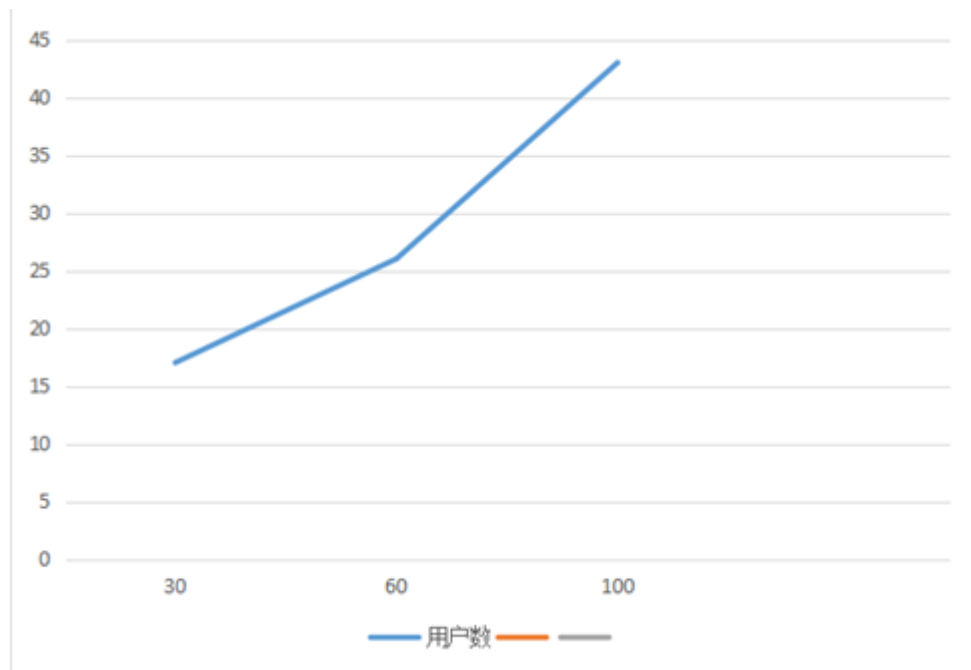


图 6-2 负载图

测试结论，系统各项性能均能满足设计要求。

6.4 测试结果

监测的结果证明系统的功能是符合我们设想的功能要求的，并且系统在数据分析，查找的过程中十分的迅速。但是时间不允许，我们不能面面俱到，系统的安全，兼容性，整体性，导航能力这些能力还没有经过仔细的检查。总的来说，系统还是基本上达到了音乐网站的具体要求。

6.5 本章小结

在本章中主要是对系统的测试工作进行了比较详细的阐述，在对系统的测试方法过程进行讲解之前，首先对测试的方案的设计情况进行了说明，然后根据测试方案来对系统的测试用例进行了说明，期间还对系统的性能测试情况以及结果情况作了比较简单的说明。最终通过对系统测试的结果进行分析，系统符合生产运行的需要。

7 总结与展望

以上章节对系统的研究现状、需求分析、系统设计、系统实现、系统测试等进行了阐述，在本章中将对在系统开发过程中存在的问题以及今后的优化、研究方向等进行说明，并对系统的总体开发做了总结。并且将对全文以及课题的研究做出总结，在此基础上对以后的研究方向做了相应的展望。

7.1 总结

本文详细的阐述了音乐系统的设计与实现工作，主要是通过绪论、相关技术介绍、系统需求分析、系统设计、系统实现等几大模块来进行阐述。本文作为音乐在网站开发的一个研究的概括，重点通过使用推荐系统相关的算法来对将其应用音乐网站中，使得在系统中能够实现根据用户在线网站中的收听歌曲、收藏、下载等行为来预测其可能喜欢的艺术家、歌曲、专辑等信息，下面将对全文的相关研究工作进行一个总结：

本文中主要是对在线音乐网站的开发设计工作作了比较详细的说明，由于所研究的课题是将推荐算法与音乐网站相结合，所以在文章中对推荐系统相关的算法知识以及相关的研究情况进行了简单的说明，同时，对推荐系统中使用的比较多的协同过滤算法以及关联规则算法的相关知识也做了比较详细的说明。进行讲解的过程中分析了他们在使用过程一些优缺点以及各自在使用的场景中的优点以及缺点情况。论文中还研究了音乐网站的相关特点，在音乐在线网站的基础上结合推荐算法知识，实现了基于推荐系统的一款在线音乐网站的设计以及实现工作，但是在实现以及研究的实际过程中，鉴于知识的欠缺，本文中还有着一些不足以及需要改进的地方：

(1) 系统推荐算法的准确度有待进一步提高。在对算法的实现过程中所实现的算法模型使用了随机数的形式作为了系统的初始值，这样的做法会使得算法的准确度会因为系统所选择的随机数的不同而产生不一样的差异值，但经过本文具体的实验的统计在本文所开发的算法中的 RMSE 值在[0.96,1.22]这个区间范围内，从这个数值来看误差还是有点大。

(2) 系统的冷启动问题、扩展性等有待改善。在算法设计以及实现的过程中对启动问题做了一定的预料和准备，但是在实际的实现过程中对于效果不是很好。

(3) 需要对混合推荐算法进行进一步的研究。在本文中对混合推荐算法进行了研究和实现,但是具体的效果值不是特别的满意,所以在后期的研究和对音乐网站的改进中需要对该算法进行进一步的研究,以便能够对该算法进行进一步的改进来实现更好的推荐效果。

7.2 展望

鉴于以上几点原因,在未来的研究中还需要进一步发现对上述描述中的不足进行改善,研究过程中需要结合音乐网站的特性情况进行研究,具体一点讲,今后的研究的过程中我将会从以下的一些方面进行展开:

(1) 学习研究推荐算法,研究用户行为数据的利用,提高推荐效能。本文中所实现的推荐算法相对于目前的所有的推荐算法而言,仅仅只是使用了相关的两个算法,在推荐算法领域中的算法还有需要的没有应用到,具体这些算法在音乐网站中的具体的推荐效果怎么样,还需要对这些算法进行具体的实现,以便将这些算法应用到该系统中。对于后面系统中所有的用户的收藏行为、分享歌曲的行为也是用户用于表现其对该音乐有兴趣的一种形式,这是一种相对于显性而言的比较隐性的比较形式,而这些隐性的数据在一定程度上也会对推荐的效果产生影响,所以后期将要进一步的加大对这些隐性的数据进行研究。在算法上面要充分的结合各个推荐算法的优缺点情况,然后通过混合推荐算法的形式来对其进行实现。

(2) 对于冷启动、系统扩展性等问题进行进一步的考虑。在现有的一些研究成果中,有相关的研究人员在研究的结论中表示,只要相关的用户对你产品产生了兴趣,那么这个 *yoghurt* 就会源源不断的为你贡献新的数据信息,而你则不必要去担心以及关心冷启动的问题。但是在实际的操作过程中如果一个推荐系统需要让用户获得更好的体验,那么我们还是要重视该问题。

(3) 研究音乐推荐的特点和音乐推荐系统架构设计。对这方面的研究是一个持久并且效果不是能够在短时间内就可以清晰的看到,音乐推荐算法的研究是一个比较新也一个比较特比特的领域,我们需要对其的研究进行持续的关注,从中发现更多的属于该系统相关的用户的特点信息,以便能够开发出更加符合用户口味的音乐推荐系统。

(4) 除此之外,在该研究中需要考虑到音乐网站中的用户的隐私保护问题,这些问题在一个程度上也反应了对客户的重视的程度。

参考文献

- [1] 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,(1):12-26.
- [2] 郭峰;张珣.家庭背景音乐系统大数据的分析[J].物联网技术,2015,(07):18-26.
- [3] Brian McFee, Thierry Bertin-Mahieux, Daniel P.W. Ellis, Gert R.G. Lanckriet. The Million Song Dataset Challenge[C]. The International World Wide Web Conference 2012, Lyon France, 2012. 04.
- [4] 郑人杰.实用软件工程.北京:清华大学出版社,2008.10
- [5] Jean-JulienAucouturier, Francois Pachet. Finding Songs that Sound the Same[C]. IEEE MPCA-2002, Belgium, 2002.
- [6] 许旭英.论网络信息化教学环境与信息化数字校园建设[J].企业家天地.2008(02).
- [7] 朱银芳;李晓婷.电子校务系统的理解与构建[J].科技经济市场.2008(07).
- [8] George Tzanetakis, Perry Cook. Musical Genre Classification ofAudio Signals[C]. IEEE Transactions on Speech andAudio Processing. 2002.10(5), 293-302.
- [9] 金芝. 需求分析与系统设计.北京:机械工业出版社,2013.06.
- [10] 邹益. 基于内容的音频检索研究[D]. 南京:南京理工大学. 2007.
- [11] 罗文. 协同过滤推荐算法综述[J]. 科技传播,2015,07(01):12-28.
- [12] Rakesh Agrawal, Tomasz Imielinski, Arun Swami. Mining association rules between sets of items in large databases[C]. ACM SIGMOD Conference. 1993.
- [13] 隋占丽;李影;于娟;王波. 基于协同过滤技术的音乐推荐系统的研究[J].2015,(02):38-50.
- [14] 徐淮杰;张二芬. 基于关联规则与奇异值分解的音乐推荐系统[J]. 电子设计工程,2013,(01):28-36.
- [15] 陈雅茜.公共环境下的混合型音乐推荐系统的关键技术研究[J].2015,(11):36-42.
- [16] 周强;李曦. 基于推荐技术的中国音乐数据库系统的设计[J].2015,(07):39-62
- [17] 刘治宇. 基于混合推荐算法的情境感知音乐推荐系统研究与实现[D].成都:电子科技大学,2015.
- [18] 赵营. 基于协同过滤算法的习题推荐[J]. 科技经济导刊,2016,(05):136.

- [19] 管鹏,张键,顾杰. 大数据背景下个性化音乐推荐方案探究[J].无线互联科技,2016,(11):99-100.
- [20] 闫俊,刘文飞,林鸿飞. 基于标签混合语义空间的音乐推荐方法研究[J]. 中文信息学报,2014,(04):117-122.
- [21] 黄治移. 融入音乐子人格特质和社交网络行为分析的音乐推荐方法[D].杭州: 浙江工商大学, 2015.
- [22] 罗文. 协同过滤推荐算法综述[J].科技传播,2015,(07):115-196.
- [23] 刘涛. 一个面向论坛用户的个性化阅读推荐系统研究与实现[D].南京: 南京大学, 2013.
- [24] 於骏洁. “避风港”原则在音乐侵权行为中的应用现状与建议[J]. 音乐传播,2015,(01):12-26.
- [25] 刘治宇. 基于混合推荐算法的情境感知音乐推荐系统研究与实现[D].成都: 电子科技大学, 2015.
- [26] 郭宇红; 杨金然. 一种基于 AHP 的智能电影推荐方法[J]. 南京邮电大学学报(自然科学版),2015,(03):36-39.
- [27] 周强,李曦. 基于推荐技术的中国音乐数据库系统的设计[J]. 计算机技术与发展,2015,(07):162-165.
- [28] 刘瑾; 詹玲超. 基于 Struts 的音乐网站的设计与实现[J]. 数字技术与应用,2016,(03):22-28.
- [29] 郭莉敏. 基于时间感知和社交网络信任度的协同过滤算法研究[D].南京: 南京信息工程大学, 2016.
- [30] 李博; 陈志刚; 黄瑞; 郑祥云. 基于 LDA 模型的音乐推荐算法[J]. 计算机工程,2016,(06):12-16.
- [31] 李艳,李葆华,王金环. 一种新的基于 LDA-MURE 模型的音乐个性化推荐算法[J]. 吉林大学学报(理学版),2017,(02):371-375.
- [32] 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,(01):113-118.

致 谢

时光如梭，转眼间研究生学习生涯即将结束，在这两年半的求学过程中我收获的不仅是快乐的学习时光、知识的财富，还得到导师、家人及朋友的支持和帮助。

首先，最想感谢的是导师-周俊博士。本论文是在您的悉心指导下一点点完成的，从论文的选题、定稿、到今天的答辩，您不仅给予知识上、技术上耐心的指导，还给予我精神支持，一步步走来，艰难的过程只有您和我才知晓。虽然您是一位年轻的硕导，但您拥有渊博的专业知识，严谨的治学态度，精益求精的工作精神。本着对学生负责的态度，您从来都以最快的时间给出最专业的论文修改建议，让学生有更充分的时间修改论文，本论文能够按时的完成并进行答辩，您给予我巨大的帮助。在此，谨向导师表示崇高的敬意和衷心的感谢！

其次，要感谢我的家人，是你们在我的求学过程中给予精神支持和物质支持，让我义无反顾的投身到研究生学习中，是你们的无私的付出使我有今天小小的成绩。

最后，感谢学校研究生处的老师，是你们细心、认真的工作态度，让我能顺利进入答辩环节。衷心的感谢在场的答辩专家对我的论文进行评审并给出宝贵意见，在此，祝您身体健康、万事顺心！

龚婧

2017 年 03 月 12 日

答辩材料 2

学位论文导师评阅意见表

论文作者姓名	姜婧	学 号	63141500348
导 师 姓 名	周俊	工作单位	华东师范大学计算机科学与软件工程学院
从事学科专业	密码与网络安全系	专业技术职称	副教授

该学位论文根据系统中用户在听歌过程中所产生的日志信息作为推荐指标，利用个性化推荐技术来为用户推荐其喜欢的歌曲、可能感兴趣的专辑、可能喜欢的歌手等。在系统设计中，利用协同过滤算法，基于内容对于一些数据信息进行过滤，根据所需要的内容信息进行推荐，实现了一个基于推荐系统的在线音乐网站设计。最后，以 Last.fm 数据集作为所开发的算法的测试集，为所开发的推荐系统效果提供实验的依据。

作者开发的基于推荐的在线音乐网站，针对基于内容的推荐音乐较为单一，缺少用户意见的问题，通过在推荐算法中加入用户评分、意见指标，未增加其推荐过程中的权重。对音乐的特点以及用户的习惯进行分析，而不仅仅只是向用户推荐热门、流行等音乐，增加向用户推荐音乐的成功率，使得在推荐的过程中更加的个性化。在推荐的过程中将用户的收听的历史记录、收藏的音乐等信息进行收集，未作为一个新的指标，进一步的分析用户兴趣点，具有较高的创新性。

该学位论文工作对国内外应用研究现状论述清晰准确，发展趋势判断合理；研究资料与数据全面、可靠。研究思路清晰，方案设计可行。所设计的基于推荐的在线音乐网站系统综合运用了软件工程的基本理论及专业知识；设计方法科学、合理、可行。论文结果明确，可信度高，具有较高的工程应用价值与社会意义。工作量饱满且具有一定难度。行文表述简洁、规范；能够反映工程设计的核心内容；具有较强的系统性与逻辑性；文字表述清晰，图表、公式规范、清晰。

综上所述，是一篇合格的工程硕士学位论文。

导师签名：

周俊

2017年 3月 23日

華東師範大學

硕士学位论文评阅意见书 (专业学位)

论文题目 基于推荐系统的一款在线音乐网站的设计与实现

作者姓名 龚婧

作者学号 63141500348

专业学位
类别领域 软件工程

所在院系所 计算机科学与软件工程学院

导师姓名 周俊

2017年5月2日

致 函

尊敬的 薛庆水 专家：

我校 软件工程 专业学位硕士研究生 龚婧 的学位论文已经完成，特聘请您为该生硕士学位论文的评阅人，请您在百忙之中对该论文进行评阅，并请您将评阅意见于 2017 年 5 月 17 日 前密封寄送：

华东师范大学 计算机科学与软件工程 学院 研究生秘书收，联系电话 62233261

请您在填写评阅意见表的同时对论文提出修改意见，我们将转达学位申请者本人，进一步对论文加以完善。对您的支持，我们深表感谢！

此致

敬礼！

华东师范大学专业学位评定分委员会

2017 年 5 月 2 日

评价指标	评价要素	得分 (满分 100 分)	权重
选题与综述	来源于工程实际；系所属工程领域的研究范畴。目的明确；具有必要性；具有应用前景。	85	15%
内容	对国内外应用研究现状论述清晰准确，发展趋势判断合理；研究资料与数据全面、可靠。研究思路清晰，方案设计可行；资料与数据分析科学、准确。	87	30%
工作量	设计工作量饱满；具有一定难度。	82	10%
成果	具有工程应用价值；具有经济效益或社会效益。体现作者的新思路或新见解。	84	30%
撰写及规范性	表述简洁、规范；能够反映应用研究的核心内容。具有较强的系统性与逻辑性；文字表达清晰，图表、公式规范。引用文献的真实性、权威性、规范性。	83	15%
注 1. 各单项评价指标请按照满分 100 分给分，如某项得分为 80 分，权重 20%，请打 80 分、勿打 16 分；综合得分在各单项评价指标得分的基础上产生，为各评价指标的加权平均。		综合得分 (满分 100 分)	84.7
2. “异议”指下列情况之一：①任一评价指标得分低于 60 分或综合得分低于 60 分（即评价等级为“不合格”）；②违反学术道德规定。		是否异议 (打√)	是 否 <input checked="" type="checkbox"/>
是否推荐参评优秀论文（打√）	不推荐	<input checked="" type="checkbox"/>	推荐参评校级优秀论文
	推荐参评省市级优秀论文		推荐参评国家级优秀论文
评阅专家职称	副研究员	专业领域	计算机软件与理论
所在单位	上海应用技术大学	联系电话	15900739251
您对论文涉及的领域熟悉程度（打√）	很熟悉	比较熟悉	<input checked="" type="checkbox"/> 一般熟悉 (建议退回)
您是否指导专业学位论文（打√）	指导	<input checked="" type="checkbox"/> 不指导	曾指导过

评阅人对学位论文的评阅意见

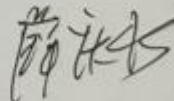
1. 请参照评价指标体系，全面评价论文，并指出存在的问题及建议；
2. 对于单项评价指标或综合得分未达到 60 分（即“异议”）的论文，请逐条列举出存在的具体问题、或直接给出修改意见；
3. 若违反学术道德规定，请具体指出，必要时请附相关佐证资料

在线音乐网站是信息系统的广泛应用之一。作者基于推荐系统设计和实现了一款在线音乐网站。其主要内容包括：对推荐系统相关的算法知识以及相关的研究情况进行了简单的说明，同时，对推荐系统中使用的比较多的协同过滤算法以及关联规则算法做了比较详细的说明。分析了他们的一些优缺点。研究了音乐网站的相关特点，在音乐在线网站的基础上结合推荐算法知识，实现了基于推荐系统的在线音乐网站的设计以及实现。论文选题尚可，具有一定的现实意义。

论文论点比较明确，思路比较清晰，结构合理，内容详实。理论分析正确，实验设计合理。

从理论论述以及方案设计可以看出作者具有较扎实的计算机理论基础，阅读了一定数目的相关论文，掌握了比较严谨科学的研究方法，具备了独立从事科研工作的能力。

评阅人签字：



2017 年 5 月 4 日

華東師範大學

硕士学位论文评阅意见书 (专业学位)

论文题目 基于推荐系统的一款在线音乐网站的设计与实现

作者姓名 龚婧

作者学号 63141500348

专业学位
类别领域 软件工程

所在院系所 计算机科学与软件工程学院

导师姓名 周俊

2017 年 5 月 2 日

致 函

尊敬的 曾鹏 专家：

我校 软件工程 专业学位硕士研究生 龚婧 的学位论文已经完成，特聘请您为该生硕士学位论文的评阅人，请您在百忙之中对该论文进行评阅，并请您将评阅意见于 2017 年 5 月 17 日 前密封寄送：

华东师范大学 计算机科学与软件工程 学院 研究生秘书收，联系电话 62233261

请您在填写评阅意见表的同时对论文提出修改意见，我们将转达学位申请者本人，进一步对论文加以完善。对您的支持，我们深表感谢！

此致

敬礼！

华东师范大学专业学位评定分委员会

2017 年 5 月 2 日

评价指标	评价要素	得分 (满分 100 分)	权重
选题与综述	来源于工程实际；系所属工程领域的研究范畴，目的明确；具有必要性；具有应用前景。	74	15%
内容	对国内外应用研究现状论述清晰准确，发展趋势判断合理；研究资料与数据全面、可靠。研究思路清晰，方案设计可行；资料与数据分析科学、准确。	76	30%
工作量	设计工作量饱满；具有一定难度。	76	10%
成果	具有工程应用价值；具有经济效益或社会效益。体现作者的新思路或新见解。	70	30%
撰写及规范性	表述简洁、规范；能够反映应用研究的核心内容。具有较强的系统性与逻辑性；文字表达清晰，图表、公式规范。引用文献的真实性、权威性、规范性。	72	15%
注 1. 各单项评价指标请按照满分 100 分给分，如某项得分为 80 分，权重 20%，请打 80 分、勿打 16 分；综合得分在各单项评价指标得分的基础上产生，为各评价指标的加权平均。		综合得分 (满分 100 分)	73
2. “异议”指下列情况之一：①任一评价指标得分低于 60 分或综合得分低于 60 分（即评价等级为“不合格”）；②违反学术道德规定。		是否异议 (打√)	是 否
是否推荐参评优秀论文 (打√)	不推荐	√	推荐参评校级优秀论文
评阅专家职称	副教授	专业领域	密码与网络安全
所在单位	华东师范大学	联系电话	021-62231530
您对论文涉及的领域熟悉程度 (打√)	很熟悉	比较熟悉	√ 一般熟悉 (建议退回)
您是否指导专业学位论文 (打√)	指导	√	不指导 曾指导过

评阅人对学位论文的评阅意见

1. 请参照评价指标体系，全面评价论文，并指出存在的问题及建议；
2. 对于单项评价指标或综合得分未达到 60 分（即“异议”）的论文，请逐条列举出存在的具体问题、或直接给出修改意见；
3. 若违反学术道德规定，请具体指出，必要时请附相关佐证资料

龚婧同学论文基于音乐网站中推荐系统的发展需求和趋势，设计开发了一个音乐在线网站。论文选题具有现实应用价值以及潜在的理论研究意义。

论文首先对相关的推荐算法、音乐系统等需求进行调研，在此分析的基础上初步确定开发的技术以及实施方案。其次根据系统需求分析的结果对系统进行设计，搭建系统总体框架。最后以系统设计结果为基础，利用相应的开发技术来对系统中的功能模块进行实现工作。实现的音乐推荐模块界面美观，功能较为全面。反应了作者掌握了软件工程专业的基础理论和分析方法，具有一定的实践能力。

论文内容较充实，层次分明，逻辑结构较清晰，反映了作者在本学科具有一定的理论基础，能够应用所学知识解决实际的工程问题。论文达到了软件工程硕士学位论文的研究水平，建议提交答辩。

一些修改建议如下：

- 1：不同章节内代码格式应统一。
- 2：部分图片拉伸比例失衡，显示不清晰。
- 3：参考文献格式不规范。

评阅人签字：

2017 年 5 月 4 日

市盲审成绩



评阅盲审总评

学位类别：工程硕士 专业学位领域：软件工程 姓名：龚婧 学号：63141500348

评阅类型	职称	是否博导	选题	创新	学术	应用	准确	状态
市盲审			C	C	C	C	C	通过
评阅	副教授	否	85	87	82	84	83	通过
评阅	副教授	否	74	76	76	70	72	通过

评阅：	通过
市盲审：	通过
校盲审：	未校盲审
总评：	通过：请打印答辩材料，参加答辩
总评：	<input type="radio"/> 通过 <input type="radio"/> 不通过
备注：	<input type="text"/>

打印答辩材料