分类号 TP391.41　　密级
U D C　　编号 10741

兰州财经大学
LANZHOU UNIVERSITY OF FINANCE AND ECONOMICS

硕 士 学 位 论 文

论文题目　结合时间效应的音乐推荐方法研究

研 究 生 姓 名：李 欣
指导教师姓名、职称：米红娟 教授
学 科、专 业 名 称：管理科学与工程
研 究 方 向：信息管理与信息系统

提 交 日 期：2021 年 5 月 15 日

独创性声明

本人声明所呈交的论文是我个人在导师指导下进行的研究工作及取得的研究成果。尽我所知，除了文中特别加以标注和致谢的地方外，论文中不包含其他人已经发表或撰写过的研究成果。与我一同工作的同志对本研究所做的任何贡献均已在论文中作了明确的说明并表示了谢意。

学位论文作者签名：李欣 签字日期：2021.5.15

导师签名：[illegible] 签字日期：2021.5.15

关于论文使用授权的说明

本人完全了解学校关于保留、使用学位论文的各项规定，同意（选择“同意”/“不同意”）以下事项：

1.学校有权保留本论文的复印件和磁盘，允许论文被查阅和借阅，可以采用影印、缩印或扫描等复制手段保存、汇编学位论文；

2.学校有权将本人的学位论文提交至清华大学“中国学术期刊（光盘版）电子杂志社”用于出版和编入 CNKI《中国知识资源总库》或其他同类数据库，传播本学位论文的全部或部分内容。

学位论文作者签名：李欣 签字日期：2021.5.15

导师签名：[illegible] 签字日期：2021.5.15

Research on Music Recommendation Method Combined with Time Effect

Candidate: Li Xin

Supervisor: Mi Hongjuan

摘 要

随着互联网和数字音乐的迅速发展，各类音乐平台为用户提供了大量的音乐作品。然而随着音乐作品数量的急剧增加，用户面对大量的歌曲信息，很难快速找到自己感兴趣的音乐。为了给用户提供良好的使用体验，同时增加用户对音乐平台的满意度，各类音乐平台使用推荐系统来为用户提供个性化推荐服务。由于用户的兴趣是不断变化的，且随着时间的推移，会出现遗忘现象，进而对用户当前兴趣产生影响。然而常见的推荐系统进行个性化推荐时，很少考虑时间因素，为了使用户在数据庞大的音乐数据里找到自己感兴趣的音乐作品，本文将遗忘现象对兴趣产生的影响纳入个性化的音乐推荐算法中，以提高推荐质量。

本文考虑到时间因素的影响，提出了两种基于时间衰减函数的音乐推荐模型，一是基于时间衰减函数的协同过滤音乐推荐模型：首先以艾宾浩斯遗忘曲线为基础，拟合指数型时间衰减函数与幂函数型时间衰减函数，其次根据用户听取歌曲频数的分布，建立了合理的评分机制，然后根据修正余弦相似度公式计算评分相似度、歌曲相似度，将两者融合得到歌曲的综合相似度，再引入时间衰减函数得到音乐综合相似度，进行评分预测；二是基于时间衰减函数的 LightGBM 音乐推荐模型：首先通过时间衰减函数对用户评分进行衰减修正，然后运用 LightGBM 模型进行评分预测，当评分大于等于阈值时进行推荐。

在公开音乐数据集 Last.fm 上，通过实验对提出的两种基于时间衰减函数的个性化音乐推荐算法进行评估。实验表明，引入了时间衰减函数的协同过滤推荐算法（TDF-CF）优于传统的协同过滤算法（CF），引入幂函数型时间衰减函数的协同过滤推荐算法效果更好；引入了时间衰减函数的 LightGBM 音乐推荐算法（TDF-LGBM）的推荐效果优于未引入时间效应的传统的协同过滤算法（CF），并且引入幂函数型时间衰减函数的 LightGBM 音乐推荐算法效果更好。最后两种模型的实验结果进行对比分析表明，融入时间效应的 TDF-LGBM 音乐推荐算法的推荐效果优于 TDF-CF 的音乐推荐算法的推荐效果，且幂函数型的遗忘曲线对于推荐更有优势，引入幂函数型衰减函数的 TDF-LGBM 算法的推荐结果最佳。因此，结合时间效应与 LightGBM 算法建模音乐推荐，能够提高音乐推荐的准确性，为目标用户提供更符合其偏好的音乐作品。

关键词：LightGBM 音乐推荐 协同过滤 遗忘曲线 时间衰减函数

Abstract

With the fast development of the Internet and digital music, various music platforms provide users with a large number of music works. However, with the quick increase in the number of music works, users face a large amount of song information, and it is difficult for users to fast find the music they are interested in. In order to provide users with a good experience and increase user satisfaction with music platforms, various music platforms use recommendation systems to provide users with personalized recommendation services. Because the user's interest is constantly changing, and over time, there will be a forgetting phenomenon, which in turn affects the user's current interest. However, when common recommendation systems make personalized recom- mendations, they rarely consider the time factor. In order to enable users to find music works they are interested in in the huge music data, this article incorporates the impact of forgetting on interests into personalized music recommendations. Algorithm to improve the quality of recommendations.

The paper takes into consideration the impact of time components, and proposes two music recommendation models based on time decay functions. One is a collaborative filtering music recommendation model based on time decay functions: first, based on the Ebbinghaus forgetting curve, fitting an exponential type time decay function and power function time decay function. Secondly, a reasonable scoring mechanism is

established according to the distribution of the frequency of the songs listened to by users, and then the scoring similarity and song similarity are calculated according to the modified cosine similarity formula, and the two are merged to obtain the song's Comprehensive similarity, and then introduce the time decay function to obtain the comprehensive similarity of music, and then make the score prediction; the second is the LightGBM music recommendation model based on the time decay function: firstly, the user's score is attenuated and corrected by the time decay function, and then the LightGBM model is used to predict the score , Recommend when the score is greater than or equal to the threshold.On the public music data set Last.fm, the two proposed personalized music recommendation algorithms based on time decay function are evaluated through experiments. Experiments show that in model 1, the collaborative filtering recommendation algorithm (TDF-CF) with the introduction of a time decay function is better than the traditional collaborative filtering algorithm (CF), and the collaborative filtering recommendation algorithm with the introduction of a power function-type time decay function is better . In the second model, the recommendation effect of LightGBM music recommendation algorithm (TDF-LGBM) with time decay function is better than that of music recommendation algorithm (CF) without time effect, and LightGBM music recommendation algorithm with power function time decay function is

introduced Better results. Finally, the experimental results of the two models are compared and analyzed. The recommendation effect of the TDF-LGBM music recommendation algorithm incorporating the time effect is better than that of the TDF-CF music recommendation algorithm, and the power function type forgetting curve is more advantageous for recommendation.

On the public music data set Last.fm, the two proposed personalized music recommendation algorithms based on time decay function are evaluated through experiments. Experiments show that the collaborative filtering recommendation algorithm (TDF-CF) that introduces the time decay function is better than the traditional collaborative filtering algorithm (CF), and the collaborative filtering recommendation algorithm that introduces the power function time decay function is better; the time decay function is introduced The recommendation effect of the LightGBM music recommendation algorithm (TDF-LGBM) is better than that of the traditional collaborative filtering algorithm (CF) that does not introduce time effects, and the LightGBM music recommendation algorithm that introduces a power function time decay function is better. The comparative analysis of the experimental results of the last two models shows that the recommendation effect of the TDF-LGBM music recommendation algorithm incorporating the time effect is better than that of the TDF-CF music recommendation algorithm, and the power

function type forgetting curve is more advantageous for recommendation. The recommendation result of the TDF-LGBM algorithm that introduces the power function decay function is the best. Therefore, combining time effect and LightGBM algorithm modeling music recommendation can improve the accuracy of music recommendation and provide target users with music works that are more in line with their preferences.

Keywords: LightGBM; Music recommendation; Collaborative filtering; Forgetting curve; Time decay function

目 录

1 绪论.....1
1.1 研究背景和意义.....1
1.1.1 研究背景.....1
1.1.2 研究意义.....4
1.2 国内外研究现状.....5
1.2.1 协同过滤方法.....5
1.2.2 基于内容的推荐方法.....6
1.2.3 混合推荐方法.....6
1.3 研究内容和创新点.....9
1.4 论文的组织结构.....11
2 相关理论基础.....12
2.1 艾宾浩斯遗忘曲线.....12
2.2 时间衰减函数.....13
2.3 LightGBM 算法.....14
2.3.1 集成学习的思想.....14
2.3.2 梯度提升决策树算法.....15
2.3.3 LightGBM 算法的原理.....17
2.3.4 LightGBM 对 XGBoost 的改进优化.....18
2.3.5 LightGBM 的参数设置.....19
2.4 推荐算法.....20
2.4.1 基于内容的推荐.....20
2.4.2 协同过滤的推荐.....21
2.4.3 混合推荐.....26
2.5 推荐算法总结.....27
2.6 本章小结.....28
3 引入时间衰减函数的协同过滤音乐推荐算法.....29
3.1 问题解决和思路描述.....29

3.2 时间衰减函数 29
3.2.1 指数函数型衰减函数 29
3.2.2 幂函数型衰减函数 31
3.3 综合相似度 31
3.3.1 获取评分 32
3.3.2 评分相似度 32
3.3.3 歌曲相似度 33
3.3.4 歌曲综合相似度 35
3.3.5 融入时间衰减函数的综合相似度 35
3.3.6 评分预测 35
3.4 评价指标 36
3.5 实验设置 37
3.5.1 实验环境 37
3.5.2 实验数据 37
3.5.3 对比方法与参数描述 38
3.5.4 实验结果与分析 39
3.6 本章小结 41
4 引入时间衰减函数的 LightGBM 音乐推荐算法 42
4.1 问题描述和解决思路 42
4.2 引入时间效应的 LightGBM 的音乐推荐算法 42
4.2.1 融入时间衰减函数的评分修正 42
4.2.2 基于 LightGBM 的歌曲评分预测 43
4.3 实验设置 44
4.3.1 实验数据 44
4.3.2 实验结果与分析 44
4.4 两种模型实验结果对比 47
4.5 本章小结 47
5 总结与展望 49
5.1 总结 49

5.2 展望 50

参考文献 52

致 谢 59

1 绪论

1.1 研究背景和意义

1.1.1 研究背景

在大数据时代，人们面临着信息过载的问题，如何及时、准确地获取用户所需要的信息成为研究的热点。因此，解决信息过载的方法相继出现，比如分类目录[1][2]与搜索引擎[3]错误!未找到引用源。。各类网站的分类目录把网站信息整理归类，每个类别包括网站名称，URL 链接，内容提示等信息。用户可以通过目录找到自己喜欢的网站，比如通过搜狗网站，用户能够迅速查找到自己所需要的网站。但是分类目录也有局限性，它只能为用户提供大致的信息，不能对用户所需信息进行精准定位。基于关键字搜索的搜索引擎的出现为用户提供了更快捷的信息查询方法，用户可以通过关键字来查询所需要的信息。虽然通过关键字可以使用户快速获得相关信息，但是用户对搜索的结果仍需要进行大量的筛选过滤。同时，搜索引擎是一种用户根据自己的需求，主动进行信息检索的方式，然而它不能通过主动地挖掘用户的需求，这导致无法为用户进行个性化的推荐。随着信息推送的科技水平越来越高，传统的“人找信息”的服务模式正在被“信息找人”的信息推送服务逐渐代替，因此可以结合用户的需求与 Context 进行个性化推荐[4]。于是“个性化服务”被提出，通过用户的个人需求，为其提供符合其需求的服务，刚好推荐系统是“个性化服务”的一种载体被研究并应用到各个领域。推荐系统根据用户的历史行为记录，挖掘符合用户品味的内容，进而探索用户与项目之间的关系，向用户提供个性化的推荐。推荐系统与分类目录、搜索引擎的区别是用户不需要主动描述自己的需求，而是基于“大数据”与用户模型对用户进行推荐，这使得推荐系统在研究领域的地位越来越高。

推荐系统的出现对信息超载的问题提供了有效的解决方案，它以用户之前的行为信息作为特征，对用户推荐最符合其偏好的项目，进而推送个性化服务。推荐系统的优点是可以主动收集用户的特征信息，从中挖掘用户的行为习惯与需求偏好，为用户尽可能地推荐符合其需求的信息[6]；同时可以根据用户需求的变化，及时地调整信息推送的服务内容与方式。推荐系统被广泛的应用于各类网络平

台。一个相对完整的推荐系统，主要有行为记录、用户建模、推荐算法这三个必不可少的模块，通过搜集用户的行为数据，建立其偏好模型，使用推荐算法为用户提供满足其需要的信息，其通用模型如图 1.1 所示。

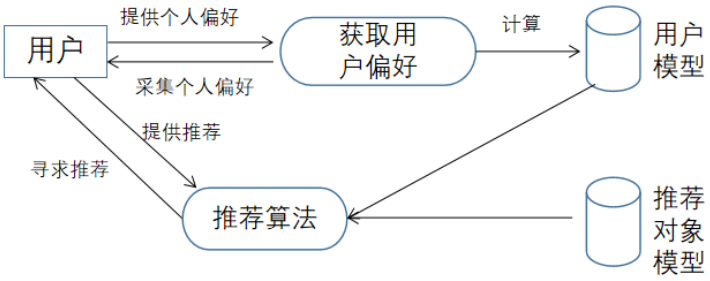


图1.1 推荐系统的通用模型

行为记录模块是收集用户的某些使用行为信息，如登录、点击、订阅、评论等；用户建模模块是输入用户的行为记录信息，在建立的推荐模型中获取用户的偏好；推荐算法是将建立的用户模型与具体推荐物品的模型相结合，为用户尽可能准确地挑选其喜欢的物品，并把这些物品通过排序的形式进行推荐[13]。推荐系统因为其个性化并主动为用户进行推荐的特点，在日常生活中的应用领域极为广泛。一些比较常用的推荐系统如表 1.1 所示。

表 1.1 推荐系统应用实例

应用领域	推荐系统
电子商务	Amazon, taobao, Alibaba
音乐	Last.fm, Yahoo, Pandora
电影	MovieLens, Netflix, Moviefinder
视频	Youtobe, aiqiyi, bilibili
阅读	Google Reader, dangdang, zite

音乐是人们表达和赋予情感的重要方式之一，也是人们生活中最重要的娱乐活动之一。手机的基本需求和存储能力已经无法承受音乐数据库中庞大的异质音乐数据，这将导致用户信息疲劳[7]，当面对音乐库中的大量音乐数据时，用户通常无法快速找到自己感兴趣的歌曲，并且其他人推荐的音乐库也无法满足用户的

的个性化需求。随着音乐流媒体服务的兴起，比如 AppleMusic，Spotif 和 KKBox 等，以及各类音乐专辑的不断发行与音乐平台的迅速产生，各大音乐平台使用个性化的推荐系统，为用户尽可能地找到符合其偏好的音乐。

在音乐推荐系统中，对音乐的描述可以从音乐的内容、上下文环境、用户的个人特征以及听音乐时的上下文环境这几个部分来区分。音乐的内容可以直接从音乐文件的音频信号中获得，音乐的上下文环境包括了演唱者的个人信息、音乐的发布时间、曲风等信息。对音乐数据进行分析，提取音乐的信息特征，通过歌手或者演唱者的名字，获取上下文环境的特征，然后在音乐数据库或者相关数据中查找与音乐有关的信息。除此之外，音乐的信息特征相比其上下文的环境，更具有客观性，比如音乐的旋律不会因为人们的主观意识来发生改变。用户的个人特征是指听众的个人信息、品味、喜好特征等。用户听音乐时的上下文环境主要指其听音乐时的自然环境等。因此，推荐系统的本质是提供一种信息过滤机制。该系统可以自动分析用户的历史行为并挖掘用户的喜好，不需要人工干预就能够过滤掉庞大信息中没有用的信息，进而推荐用户感兴趣的信息。目前，各大领域都在使用推荐系统，比如电子商务领域（淘宝，亚马逊，京东等）、信息检索领域（360 网站、搜狗等）、图书、旅行、电影和音乐等。其中，听音乐是人们生活中最常涉及的活动之一，研究表明，在日常生活中，人们听音乐的行为记录远超于阅读、购物等其他行为记录，这种现象的出现，其一是因为音乐一直以来都是人们生活中不可或缺的娱乐方式，其二是因为现在的生活节奏太快，人们需要一种缓解压力的方法，刚好音乐这种形式可以舒缓人们的压力，而且也易于被接受。所以音乐推荐系统的应用方式，与其他推荐系统一样，通过分析用户的行为偏好和音乐本身的特征信息，向用户进行个性化的推荐。

从推荐的技术层面分析，推荐过程主要被分为挖掘用户的偏好信息、推荐音乐以及对推荐结果进行筛选这几个过程。最主要的推荐方法有协同过滤、基于内容的推荐和混合式推荐等。与其他领域的推荐相比，音乐推荐具有如下特点：

（1）有关音乐的用户信息易于收集，且音乐作品的时间不长，可以在较短的时间内收听多首歌曲。

（2）没有固定的音乐分类标准。音乐可以从多个角度进行划分，例如场合，流派、乐器和情绪等。

（3）用户在听音乐时具有一定的顺序，音乐的顺序有重要的意义，并且用户的喜好会随着环境和时间的变化而变化。

根据音乐丰富多变的特性以及用户的即时偏好等因素，可以看出音乐推荐在推荐领域的特殊性，它不仅需要分析用户与相似用户历史行为中表现出的偏好，还需要挖掘用户的即时信息，为该用户推荐更符合其偏好的音乐。随着各种风格的音乐越来越多，如何向用户更准确地推荐其感兴趣的音乐成为目前音乐推荐系统中研究的热点问题。

1.1.2 研究意义

在大数据环境下，推荐系统主要面临的挑战有以下几点：（1）推荐时需要挖掘更多的信息，多维度的用户信息导致了高维稀疏性问题，而且会产生冗余数据与噪声数据；（2）需要收集显式数据与隐式反馈数据，将显式数据和隐式反馈数据相结合才可以更好地提高推荐系统的性能；（3）需要推荐系统具有较强的数据存储与处理能力，产生新数据的速度越来越快，数量也越来越多，以前的数据库几乎都是关系型，无法满足现在的推荐系统，而且风险性比较高[错误!未找到引用源。]。近些年，互联网技术发展的速度越来越快，一些较早的推荐算法已经无法支持用户兴趣的动态变化与系统及时性的要求，尤其是在一些具有较高时效性的系统中，若不能根据用户兴趣的变化与系统的时效性及时做出改变，那么为用户推荐结果没有任何意义[9]。用户的兴趣会随着时间的变化而改变，这要求推荐系统也要根据用户的兴趣变化相应地做出响应，即推荐系统不是一成不变的系统，它应该是一个动态且适应性较强的系统。因此，将用户的兴趣偏好与音乐的时效性合理地融入到推荐算法中，是音乐推荐领域中非常重要的研究课题。

随着音乐服务越来越受欢迎，出现了许多关于音乐服务方面的音乐推荐技术的研究成果。国内外学者一直高度重视音乐推荐算法的研究，在行业中也出现了大量成功的音乐平台，比如国外有著名音乐网站 Paodnra，国内有网易云音乐等[错误!未找到引用源。]，但在大数据环境下，这些平台也在动态的发生变化。由于传统单一的推荐算法可扩展性差，难以提高用户的满意度，也不能充分有效地利用音乐的特征，而在推荐系统中，时间因素是用来反映用户兴趣变化的非常重要的一种信息[错误!未找到引用源。]，因此，本文在传统的推荐算法基础上，从时间效应与用户兴趣变

化的角度，融入了反映人们遗忘规律的艾宾浩斯遗忘曲线[11]，基于艾宾浩斯遗忘曲线理论，提出改进的基于时间衰减函数的协同过滤音乐推荐算法（TDF-CF：Collaborative Filtering based on time decay function）和基于时间衰减函数的LightGBM（Light Gradient Boosting Machine）音乐推荐模型（TDF-LGBM：LightGBM music recommendation model based on time decay function）。从用户行为习惯的周期性重复性的角度来看，它整合了音乐的数据特征和用户的行为特征，针对用户兴趣与时间效应的关系，对用户进行更准确地推荐，从而提高音乐推荐的准确性，这要求音乐个性化推荐系统要尽可能地满足目标用户的需求，并及时根据用户偏好随时间的变化进行有效地调整，以此来提供个性化的服务，音乐推荐作为推荐系统中非常特别的领域，本文的研究具有实际意义。

1.2 国内外研究现状

从上世纪 90 年代，针对推荐系统展开的研究便逐渐起步。它涉及的知识领域较广，比如认知诊断[13]、信息搜索[14]、机器学习[15]、普适计算[16]等。Resnick 等学者于 1997 年对推荐系统的概念作出了具体定义[17]，即通过特定的方法以及技术，以已经产生的历史记录为依据针对用户的兴趣爱好构建相应的模型，从而为用户带来能够达到其预期的物品以及相关信息等。推荐系统可以被视为一种信息过滤工具，而推荐算法也曾经被称为过滤算法。以推荐算法之间存在的差异性为依据，可以将其划分为三种类型，即基于内容的推荐方法[18]、协同过滤[19]和混合推荐[20][21]。以上种类中应用频率最高的是协同过滤这一方法，本文内容针对音乐推荐算法的研究现状展开介绍与讨论。

1.2.1 协同过滤方法

协同过滤方法[22][23]实际上利用的是矩阵分解，而矩阵是以用户提供的反馈信息为基础建立起来的。用户与音乐之间存在的交互可通过多种形式进行展现，主要表现为搜索历史、听歌记录以及评论等。尤其是与音乐相关的历史记录，属于一种隐性反馈，这是由于这些记录是通过系统进行全自动搜集得到的。协同过滤一般主要通过对用户给出的评分以及评价内容展开文本处理，得出评分矩阵，从而将其进一步转化为显性信息，再经过分解处理便能够用于推荐[24]。然而协同过

滤也能够通过已经经过处理的隐性反馈信息来进行推荐，由此产生的推荐结果一般具有较高的准确性，因此能够提升用户的满意度。Nabizadehd 等人[25]通过研究提出，可以通过协同过滤这一方式将用户、乐队、乐器种类等多种音乐要素间存在的内在联系进行挖掘，此方法能够将数据大量汇聚于用户以及音乐之间产生的历史记录，并转化为用户对音乐给出的评分。Su 等人[26]创新性地设置了某一固定的阈值，此方法利用对用户与音乐之间存在的听歌历史以及频次以预设好的阈值为依据进行转换，最终得到评分，从而以音乐间存在的相似性为依据为用户提供准确的推荐。除此之外，还有相关学者对推荐算法展开了优化，深度学习的出现为推荐算法提供了全新的研究思路。Dieleman 等人利用 deep learning 中的卷积神经网络来对回归模型展开训练[27]，此模型能够以音频中存在的内容信号为依据展开深入分析，再进一步根据分析结果对用户可能产生兴趣的音乐类类型进行预测，由此一来能够有效缩减通过专家标记的知识复杂程度，从而可以大量缩减人工成本。Focuss 等[错误!未找到引用源。]以图模型有关的理论推出了一种以随机游走节点相似度为基础的计算方式，能够在一定程度上解决协同过滤系统内部数据稀疏性这一弊端。Koren[29]等人针对协同过滤算法展开了适当的优化，在矩阵分解模型中增加时间因素，进一步展开综合分析，此方法可以顺利解决数据不够密集这一问题。协同过滤的方式怎样才能以已有的数据和可利用的资源为用户进行合理地推荐，此问题在音乐推荐方面仍然有待解决，同时冷启动以及稀疏性在协同过滤中仍然属于有待进一步解决的问题。

1.2.2 基于内容的推荐方法

基于内容的推荐算法[30]属于推荐系统中最为常用的推荐算法，该方法为推荐系统中最基础的算法之一，目前在文本分类等领域的应用较为广泛。主要可以概括如下：将所有物品的特征进行抽取，对物品进行表述，通过用户过去感兴趣以及不感兴趣的物品特征进行学习，对用户的偏好特征进行模拟，借助与上一层级的用户偏好以及候选特征展开全面对比，从而为用户推荐数据集内存在的物品。而音乐具有一定的特殊性，其特征提取存在较高的难度，例如音调、音色、旋律等各种要素。所以，现阶段与音乐特征相关的各种处理以及使用通常以标签来进行替代。标签的应用领域也十分广泛[31]，比如，以标签为基础的检索[32]、以标签

为基础的图像处理[33]、以标签为基础的音乐信息检索[34]。从音乐推荐的角度上来说，标签能够较为准确地将音乐的本质特征进行描述。国内外许多相关学者对标签进行整合处理后合并到音乐的推荐中都得到了不错的效果。Negar Hariri[35][36]通过社交媒体网站对标签进行下载，同时利用 LDA（Latent Dirichlet Allocation）模型以一定的逻辑预测下一首歌，同时对 LDA 中的话题使用各种类型的标签进行组合，以预测的下一个话题来完成标签的筛选过程，同时为目标用户进行音乐推荐。王兴茂等**错误!未找到引用源。**以非共同评分项目集为起点，以历史偏好记录为基础来搜集近邻用户的有关信息，从而优化推荐质量。Liu 等人[39]表示，可以把用户对音乐标签标注的行为特点作为自动进行音频分类的依据。Dmitry Dolgikh[39]在建立模型时利用了歌手所有的标签，再进一步通过社区检测算法对各种不同音乐的类型展开偏好划分，从而形成各种偏好社区，再以各个子社区的标签为依据构建起推荐列表。不仅如此，Aaron 等人[40]也推出了一种以潜在因素模型为基础的推荐方式，利用深度学习过程中的卷积神经网络进行模型训练的过程，将音频直接转化为潜在因素展开对用户偏好的分析以及预测。基于内容的推荐方法也具有不少缺点，例如对于新用户的偏好以及新物品存在的特征不能进行定义，通常大部分特征都必须通过专家来进行标注，从而所需时间很长，导致人工成本较高。

1.2.3 混合推荐方法

混合推荐这一方式能够有效弥补以上两种方法的缺陷，也就是通过联用多种方式来对单个方式存在的缺点进行补充，针对推荐的准确性进行优化。个性化推荐的过程中通常需要对部分外界因素进行转化，将其转变为推荐对象所具备的特征，进一步将特征融入模型，此类外界因素就叫做上下文环境，能够有效优化推荐的最终结果[41]。以数据集中当前已经存在的上下文环境以及推荐目标所处的环境为依据，设置相应的推荐方式，目前已经发展为国内外音乐推荐的一大研究热点。Gorgoglione[42]表示通过上下文信息进行推荐更容易使得用户产生信任，而信任能够决定用户的消费行为，因此在推荐算法中引入时间上下文信息可以有效提升用户的信任，从而进一步促进用户产生消费行为。Hariri[43]以用户经常收听的列表名为依据对其偏好进行预测，从而得到相应的音乐列表。Wang[44]等相关学者推出了车载音乐推荐方式，以时间、车速以及噪音等因素为依据，从已经标注

完毕的音乐范围中挑选与场景环境相契合的音乐。Kapoor[45]等学者以各种环境的区别为依据推出了他们的方法，即通过天气、温度以及光照等自然环境因素与标注完毕的音乐进行匹配，从而产生相应的音乐列表。以上音乐具有的内容特征以及上下文环境都可以添加到推荐模型中，从而有效提升推荐算法的准确性。从现阶段的发展来看，与用户音乐偏好相关的研究已经取得较为丰富的研究成果，此类用户的音乐偏好向量中主要含有新颖性[46]、多样性[47]以及主流性[48]。Markus Schedl 等人[49]针对各种特征展开了全面研究后，推出以多种特征进行简单线性组合的方法进行推荐。通过研究显示，通过上述方法推荐音乐，比利用单一特征与算法进行结合的效果更为显著，然而并不能将各种特征的具体性质进行考虑，也没有通过算法将各种特征进行融合，因此这一部分内容仍然要进一步展开研究。

用户的兴趣是处于不断变化的过程中的，主要取决于知识背景、社会角色以及生活环境等方面，除此之外，物品的流行程度也会发生一定的变化，在考虑因素中添加了“时间”后，整个考虑维度都获得了提升，音乐的发展与时代的发展相适应，因此就可以对潮流以及音乐流行程度发生变化的原因有所把握。Campos[49]表示时间对用户的行为习惯与兴趣有一定的影响，它是非常重要的上下文信息，用户在不同时段看到同一列表可能会产生各不相同的反映，例如白天、晚上的偏好、节假日以及工作日的偏好等，而且物品的流行程度也与时间的变化相互联系。以协同过滤的推荐算法为基础，Koren Y 等人[29]推出了一种以 SVD 为基础的协同过滤推荐算法，同时将时间动态因素增加到模型中，以便于更准确地把握用户的收听偏好和音乐流行度之间存在的联系，从而有效提升算法的稳定性以及准确性。此方法于 2011 年 KDD SVD 的音乐评分预测中受到了广泛关注；Chen 等[51]推出了一种能够以时间变化为基础，同时对用户接受能力进行综合分析的推荐方式，对于数据较为稀疏的环境来说，能够有效提升推荐的准确度；汪静等[52]通过共同评分以及相似性权重对协同过滤推荐算法展开优化，能够提升用户之间进行共同评分的权重，在最大限度上优化了推荐准确度；Baltrunas[53]所提出的方法实质上等同于时间物品的事件过滤法，也就是以特定目标时间为依据来选择与目标时间具有一致性的评分记录作为训练集，对模型展开训练。Panniello 表示，时间信息能够优化推荐质量[53]。朱思丞等[55]表示，在影响因子中添加时间因素，进一步应用于算法，能够有效提升用户的偏好信息实效性，优化推荐结果；

郭晶晶等[55]通过李雅普诺夫模型和信任度高度融合的方式，在物联网信任推荐领域中引入推荐系统，能够有效优化系统效益；孙光福等[56]相关学者推出了一种以用户间存在的时序行为为依据的推荐方式，同时将近邻集合通过奇异值分解的协同过滤算法展开处理过程，从而有效提升准确性。Ding 等人[57]以基于近邻的协同过滤算法为依据，提出了时间权重这一概念，在进行预测评分的过程中针对近邻评分展开加权求和，所有邻居评分的权重都等于相似度以及时间权重相乘的结果，有效提升了推荐的准确性。文献[59]在针对计算机相似度以及通过近邻评分展开预测的过程中，对近期数据赋予了相应时刻更高的权重，从而提升了算法的精准度。文献[60]以近邻模型为基础的协同过滤算法中，通过逻辑斯特函数来对时间权重含义进行定义，结果表明用户兴趣与时间存在紧密度联系，Logistic 函数能够对人类的记忆曲线进行模拟。通过试验进一步表明，将时间权重应用于推荐过程能够提升最终结果的准确性。Hariri Neger 等人[61]针对音乐原本具有的属性增加了社会化标签，同时利用用户的播放列表来对各种音乐主题下的用户情境信息进行挖掘，由此产生了一种全新的主题分类方式，能够对收听列表进行有效优化；Ricardo Dias 等人[37]建立起了以时间上下文为基础的音乐推荐系统，对传统的协同过滤推荐算法进行优化，同时利用两种具有差异性的特点来获取用户的收听偏好：一是对用户行为的时间属性以及对话类型进行大范围提取，同时对会话之间的相似性进行对比分析，二是通过主体建模算法进行与时间信息相关的模拟，由实验结果可知，应用了上下文感知技术能够明显提升推荐的准确性。以上方法没有将用户特征的时间以及偏好因素进行有效利用，因此对推荐的准确度产生了不利影响。

1.3 研究内容和创新点

基于以上背景，本文针对传统的音乐推荐算法在音乐推荐系统中存在的问题进行了一些改进和创新探索，提出了引入时间衰减函数的协同过滤推荐算法，将两种时间衰减函数对音乐推荐的影响程度进行了对比，结合用户评分机制与时间效应，提高了对用户推荐的准确率。但是随着用户与音乐作品的数量不断增多，协同过滤算法的性能有所下降，于是提出了引入时间效应的 LightGBM 个性化音乐推荐方法，为目标用户提供更符合其偏好的推荐项目。总的来说，本文的研究

内容总结如下：

（1）介绍了推荐系统中最常用的推荐算法，描述了这些算法的优缺点，并介绍了当前一些主流推荐算法存在的问题。

（2）通过播放次数和频率线性函数计算用户评分，解决了用户评分稀疏性问题。首先根据公式将播放次数转换为播放频率，然后通过频率线性函数计算用户评分，并在此基础上，进行建模。

（3）为了缓解在音乐推荐系统中存在的问题，提出了改进的协同过滤推荐算法，考虑到时间效应对用户偏好的影响，该算法在使用传统协同过滤算法的基础上融入了时间因素，对目标用户的偏好和评分相似度进行了预测。根据用户听歌的最近一次的时间与最初的时间，计算用户的收听时间，进而得到两种时间衰减函数；通过修正余弦相似度公式计算歌曲和评分的相似度，在计算音乐相似度的时候融入两种时间衰减函数，从而进行推荐预测。这样不仅考虑了时间因素对用户的影响，而且评分数据也得到了充实，可以更好地挖掘用户偏好，从而更精准地对目标用户进行推荐。

（4）提出了基于改进的 LightGBM 个性化音乐推荐算法。首先根据人们的遗忘规律，将艾宾浩斯遗忘曲线用于时间衰减函数的拟合，然后结合了评分机制与时间衰减函数，在 LightGBM 算法上对音乐进行评分预测。在实验中对比了两种时间衰减函数对目标用户偏好的影响程度，评估了 LightGBM 算法的推荐性能。该算法通过用户评分进行衰减修正，结合 LightGBM 算法的特点对目标用户进行音乐推荐，可以更准确地为用户提供其感兴趣的音乐。

（5）在公开音乐数据集 Last.fm 上验证本文提出的模型。基于改进的协同过滤音乐推荐方法 TDF-CF（TDF-CF：Collaborative Filtering based on time decay function）实验结果表明，以 RMSE（Root Mean Square Error，均方根误差）、MAP（Mean Average Precision，平均准确率）等方法作为评估标准，幂函数型的时间衰减函数更符合对目标用户的偏好预测，而且引入时间效应的 TDF-CF 比传统的协同过滤推荐方法效果好。将提出的基于时间衰减函数的 LightGBM 音乐推荐方法（TDF-LGBM：LightGBM music recommendation model based on time decay function）进行分析，引入时间效应的 LightGBM 音乐推荐方法的推荐效果（TDF-LGBM：LightGBM music recommendation model based on time decay

function）比未引入时间衰减函数的 LightGBM 推荐方法的推荐效果好，并且对比两种方法，引入时间效应的 TDF-LGBM 推荐模型的推荐效果优于引入时间效应的 TDF-CF 推荐模型。因此，将时间效应引入到 LightGBM 算法中可以更准确地预测用户偏好，提高音乐推荐的质量。

1.4 论文的组织结构

第一章：绪论。介绍本文的研究背景与意义，阐述推荐系统在国内外的研究现状以及研究内容。

第二章：相关理论基础。主要介绍了遗忘曲线、LightGBM 算法以及推荐系统中常用的推荐算法等相关理论。

第三章：引入时间衰减函数的协同过滤音乐推荐算法。具体阐述了时间衰减函数的原理以及改进算法在音乐推荐中的步骤，通过实验对比，分析两种时间衰减函数对推荐结果的影响程度，并将融入了时间衰减函数的协同过滤推荐方法与传统的协同过滤方法进行比较，验证改进的方法推荐效果更好。

第四章：引入时间衰减函数的 LightGBM 音乐推荐算法。具体介绍了该算法在音乐推荐中的步骤，通过实验验证提出的方法比传统方法的效果更好，并将实验结果与上一章的实验结果进行对比分析。

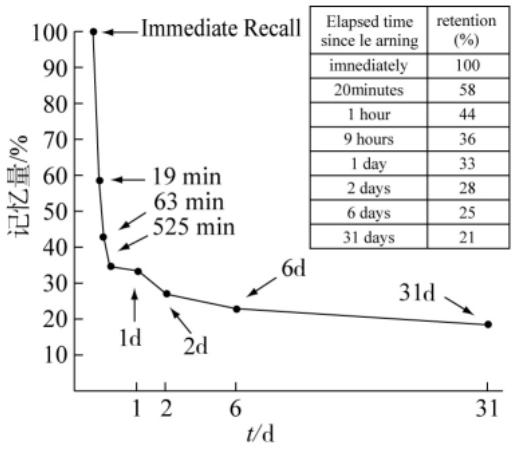
第五章：总结与展望。总结本文的研究内容，指出了文中的不足，并展望了下一步的研究工作。

2 相关理论基础

2.1 艾宾浩斯遗忘曲线

人们会记住他们经历过的事情、学过的知识和思考过的问题，这些记忆会经历从“识记”开始，最终到“回忆”的过程，这其中包括识记、遗忘、再回忆。在信息处理中，记忆是一个编码、存储以及检索输入信息的过程，比如第一次学习并背诵英语其实是将信息输入并编码的过程。人具有非凡的记忆力，每个人的记忆库能够存储 10 的 15 次方比特的信息量，但是其中可以被探索挖掘的信息大概只有 10%，所以还有更多的记忆存储空间可待开发，出现这一现象的原因是有些人只注重当时记忆的效果，却忽略了记忆中最重要的问题——记忆的牢固程度，这就涉及到了心理学中经常提到的记忆遗忘规律[错误!未找到引用源。]。

德国心理学家艾宾浩斯（H.Ebbinghaus）描述了揭示遗忘规律的遗忘曲线，他认为“保持和遗忘是时间的函数”，并通过实验将记忆的变化趋势描绘成反映遗忘进程的曲线，这便是著名的艾宾浩斯记忆遗忘曲线，如图 2.1[9]，遗忘是在学习之后立即出现的，而且遗忘的速度由最初的急速下降到逐渐缓慢。图 2.1 中纵轴表示学习之后记忆保持百分比，横轴表示学习之后时间的变化。



Elapsed time since le arning	retention (%)
imnediately	100
20minutes	58
1 hour	44
9 hours	36
1 day	33
2 days	28
6 days	25
31 days	21

图2.1 艾宾浩斯遗忘曲线

这是典型的遗忘曲线图，随着时间的推移，信息的记忆量下降速度越来越快，这就是自然遗忘的规律。艾氏曲线的提出对于知识的巩固与加深具有非常重要的作用，比如学校教师在指导学生如何复习课程时，使用了艾氏曲线会提高学生们

的复习效率，因此，艾对于教育工作者们来说，艾式曲线为他们提供了极为有效的帮助，人们遵循艾氏曲线，可以快速而牢固地记住所学习的知识。有人曾经做过一个实验，有两组学生学习同一段功课，A 组在学习后不久再复习一次，B 组在学习之后并没有复习，24 小时后，A 组的记忆量大概是 98%，B 组的记忆量大概是 56%，一星期以后，A 组的记忆量保持在 83%而 B 组保持在了 33%，这一实验结果表明，B 组的遗忘程度的平均值高于 A 组[错误!未找到引用源。]。依据艾氏遗忘曲线的观察结果，对记忆的整个过程和需要记忆的事物数量进行定量分析。该记忆的存储量会随着时间推移而不断的变化，从一开始的快，中间到逐渐变慢，最后逐渐趋于一个稳定的值。正如曲线所反映的那样，通过学习获得的记忆，随着时间的不断流逝，一部分被遗忘，而一部分则被保留在大脑中。

人们的兴趣亦是如此，随着时间的推移逐渐衰减，速度先快后慢[错误!未找到引用源。]。将艾宾浩斯遗忘曲线应用到推荐系统中，对距离用户久远的项目进行适当地降低权重，对近期的项目适当地增加权重，这样为用户推荐的项目才能更符合用户的兴趣偏好[错误!未找到引用源。]。

2.2 时间衰减函数

通过阅读大量应用时间信息展开推荐算法的相关文献，经过分析能够将与时间信息相关的处理方法进行类别划分，总结如下：

（1）采用时间衰减函数的形式。时间衰减函数能够通过指数函数、幂函数以及对数函数来进行表示[错误!未找到引用源。]，其中自变量定义为最近一次的行为时间与最初的行为时间之间的间隔，函数值是根据用户的行为时间，将其对物品的喜好程度赋予权重，自变量与时间权重之间呈反比，也就是说降低历史行为数据在目前时刻的权重。

（2）采用滑动时间窗口方法。文献**错误!未找到引用源。**中提出，以动态数据流环境为基础，通过时间窗口挑选合适的建模数据，从实质上来说也就是以邻近的协同过滤算法为基础，以数据流的时间为依据对时间窗口进行划分，对相似度展开计算的过程中仅利用目前时间内窗口所展示的数据评分。滑动窗口方式通常情况下仅利用目前时间窗口内所显示的数据，并将其他时间数据进行清除，常规的处理方式仅对最近的时间段中产生的评分数据有用。然而这种处理方式也存

在一定弊端，会使得矩阵稀疏程度更高，从而影响预测算法的精确度。

（3）采用增量计算的方法。推荐系统具有一定的动态性，新的评分数据会源源不断地涌入。如果新的数据进入系统，则推荐系统内已有的全部评分数据都必须与 0-1 之间的某一常量相乘，从而实现“折旧”。评分数据间隔时间越久，折旧次数也随之增加，因此数据会逐渐变小，从而使得有效信息的数量降低，对用户兴趣产生的干扰也就越小。

（4）将用户/项目的变化融入到模型中。在以模型为基础的协同过滤算法中，针对用户状态展开建模过程时，建模函数通常将时间作为自变量。比如，Koren 提出的 time SVD++模型中，用户偏差以及兴趣偏好都通过时间的函数进行表示。

本文在对用户进行推荐时，考虑到目标用户兴趣会随时间的变化而改变，引入了如式（2-1）所示的时间函数 $f(t)$[68]。

$$f(t)=m\cdot(\frac{T_{cur}-T_0}{T_{\max}-T_0})+1-m \tag{2-1}$$

其中，T_{cur} 表示目标用户对项目产生评分的实际时间，T_0 表示推荐系统刚开始运用的时间，即系统中首位用户的行为时间，这样设置的主要目的是将此时间点作为起始节点，$T_{\max}$ 表示推荐系统中目标用户最近一次的行为时间。m 表示 $f(t)$ 变化的幅度，取值范围是（0，1），m 的值越小，表示用户的兴趣变化速度越慢，反之越块。

2.3 LightGBM 算法

2.3.1 集成学习的思想

集成学习是一种机器学习的方法，它将多个单个学习器进行组合来完成学习任务。以分类任务为例，在对新数据进行分类时，训练多个分类器，然后根据不同的组合方式将这些分类器的结果进行组合，最终得到分类结果。对于一个复杂的任务来说，多个专家的综合判断比单个专家的判断要好得多，同时，结合多个个体学习者一起决策，也可以提高分类器的泛化能力。

目前来讲，以根据个体学习器不同的方式进行组合这个角度划分集成学习，可以分为三大类：

（1）Bagging：通过有放回的从原始数据集中随机选取样本数据来训练多个分类器。该方法通过减少单个学习器的方差来提高泛化能力，因此 bagging 的性能取决于单个学习器的稳定性。如果单个的学习器不稳定，则 bagging 可以减少由随机选取训练数据而造成的误差，但是倘若单个学习器稳定，即对数据的变化不敏感，那么 bagging 方法就不能提高甚至会降低性能。这种方法的典型代表就是随机森林算法。

（2）Boosting：此算法的优化是一个迭代过程。分类器通过改变数据样本的分布，对难以划分的数据进行收集，加强被容易错误划分的数据样本的学习，增加被错误划分的数据样本权重。这样一来，错误划分的数据在下一次迭代中能够发挥更大的作用，即惩罚错误划分的数据。此方法的典型例子有 AdaBoost 算法、GBDT 算法等。

（3）Stacking：即模型融合，包括模型训练和预测过程两个阶段。在模型训练阶段，训练多个模型，这些模型可以属于不同类型，也可以是同一类型，但参数设置不同，从而使模型之间存在一定的差异。然后，把这些不同模型的预测结果加权融合，这将成为预测阶段的训练数据，继而开始新一轮的模型构建与最终结果预测。

LightGBM 算法属于上述三大类型中的 Boosting 类型。

2.3.2 梯度提升决策树算法

对于集成学习领域来说，也需要以调整样本分布的思路来优化模型的准确性。AdaBoost 算法主要对已具备的模型预测错误样本权重进行提升，使上一阶段的学习器在后续的训练中对于错误的训练样本较为关注，从而将已有模型的缺陷进行补充。梯度提升算法的原理则是以模型迭代的所有步骤中建立起一个可以按照梯度最陡的方向来减小损失量的学习器，从而对已有模型的缺陷进行补充。同时还添加正则项等方式来避免训练数据中产生的噪音干扰，由此使得模型获得更为优良的健壮性。现阶段梯度的提升方式较多，针对可微损失函数展开调节能够对于多种学习任务进行处理，因此具有较为理想的应用前景。

GBM（Gradient Boosting Machine）是以梯度提升算法为基础的学习器。从理论层面上来看，GBM 能够利用多种类型的学习算法来充当个体学习器。然而在实际的应用中，会使用频率最高的个体学习器作为决策树，进一步究其原因，是由于决策树算法具备诸多优良的特点。首先，决策树能够被看作为 if-then 规则的集合，其算法原理较为简单，具有较高的可解释性，除此之外，决策树算法与其他算法进行比较，不需要那么多特征工程，比如不需要进行特征标准化，由此能够更高效地对字段缺失的数据进行快速处理，同时也无需考虑关心特征之间是否具有依赖性等。决策树可以将若干个特征进行自动组合，还能够轻松处理各种特征之间存在的相互关系，所以不需要担心中心数据范围内出现异常值等问题。

但是值得强调的是，独立运用决策树算法出现过拟合的可能性会很大。然而在集成学习算法中，正好有大量方式能够有效降低决策树的复杂性，从而避免单个决策树具有较高的拟合能力，比如对树的最大深度进行调整、对叶子节点的最低样本数量进行限制等，随后再利用梯度提升法对决策树进行大面积集成，从而避免发生过拟合的现象。因此能够说明，梯度提升法以及决策树学习法彼此之间能够相互弥补缺陷，将长处结合在一起，从而优化模型的稳定性以及准确性。

梯度提升决策树（GBDT）本质上属于一种迭代决策树算法，这一算法的主要原理为通过利用最速下降法，将损失函数的负梯度以当前的值直接看作残差的近似值，随后再通过残差近似值进行拟合，由此获得一个回归树。此算法在决策的同时会生成其他独立的决策树，随后将全部树的运行结果集合起来展开累加运算，从而获得最后的结果。GBDT 算法在进行训练的过程中，首先要对样本展开多次遍历。如果要缩减训练所需时长，则要求训练数据完全加载转移至内存里，由此一来单次输入的样本数量将会产生局限，不可能高于内存容量。若是将样本载入外存储器，则需要选择决策树算法。当 I/O 较为频繁的情况下，速度则会随之有所降低。LightGBM 便能够合理地解决以上问题。

GBDT 是通过不断的迭代来提升学习器的性能。在 GBDT 的迭代中，倘若用 $F_{t-1}(x)$ 表示上一次迭代完成后构建的学习器，用 $L(y,F_{t-1}(x))$ 表示损失函数，那么本轮轮训练的目的是找到一个使损失函数达到最小的弱学习器 $h_1(x)$，本轮的损失函数如式（2-2）：

$$h_1(x) = \arg\min_{h \in H} \sum L(y, F_{t-1}(x) + h_1(x)) \tag{2-2}$$

通过计算损失函数的负梯度，获得这一轮损失函数的近似值，损失函数的近似值的表示如式（2-3）所示：

$$r_{ti} = -\frac{\partial L(y, F_{t-1}(x_i))}{\partial F_{t-1}(x_i)} \tag{2-3}$$

通常使用平方差来近似的拟合 $h_1(x)$，如式（2-4）所示：

$$h_1(x) = \arg\min_{h \in H} \sum (r_{ti} - h(x))^2 \tag{2-4}$$

最终得到本轮的强学习器，如式（2-5）所示：

$$F_1(x) = h_1(x) + F_{t-1}(x) \tag{2-5}$$

2.3.3 LightGBM 算法的原理

LightGBM 为微软亚洲研究院于 2017 年所研究开发得到的一个以决策树算法为基础的提升框架，具有开源、高速等诸多优势，广泛应用于排序、分类等各种类型的机器学习中，除此之外还能够接受高效的并行训练。LightGBM 被开发出来的原因是以攻克 GBDT 在大批量的数据中产生的各种问题，从而使 GBDT 能够在实践当中得到更好的应用。GBDT 由于具备高效且精确性强等优势逐渐发展为一种应用范围极其广泛的算法，在多个领域都得到了良好的应用。然而目前数据类型逐渐增加，复杂程度日益升高，对大数据进行处理的过程中会产生极大的开销，无法在精度以及效率二者间维持良好的平衡状态[69]，LightGBM 这一模型中的决策树子模型则是通过利用按叶子分裂的方式来完成节点分裂的，所以其计算代价较低，由此一来能够对树的深度以及所有叶子节点的最低数据量进行调解，进一步防止过拟合的问题出现。LightGBM 通过利用以 histogram 的决策树为基础的算法，对特征值进行分解，得到多个小“桶”，进一步在此类“桶”上搜索分裂，由此能够有效缩减储存成本以及计算成本。同时，对于类别特征的处理，也能够促进 LightGBM 在一定的数据下获得较为理想的改进。LightGBM 不仅能够缩减计算开销，同时还能够有效优化模型的计算效率，同时在维持高效计算的同时还能够保证一定的准确率[70]。

2.3.4 LightGBM 对 XGBoost 的改进优化

与传统的 GBDT 进行比较来说，XGBoost 在许多方面的研究都作出了大幅度的优化，例如以树模型的复杂度作为正则项直接添加到目标函数中进行优化、在进行迭代优化的过程中通过目标函数进行的二近似求解等等。然而对于实际应用来说，如果我们所面临的数据维度较高，除此之外数据样本还非常庞大，那 XGBoost 的训练速度以及可扩展性方面的缺陷就暴露出来了，具体体现在如下几方面：

（1）每次迭代都需遍历所有的数据，过于消耗时间与内存；

（2）通过贪心策略对最佳分裂节点展开计算的过程中必须要遍历所有叶子节点，同时对其特征取值展开排序，排序完毕后进一步将信息转移至内存中，随后再通过计算得出后续的信息增益，整个过程所需时间较长同时对内存消耗较为明显；

（3）XGBoost 能够生成的决策树的级别为 level-wise，也就是提前预设树的深度，随后所有树都仍会生长至这个深度值，所以存在部分树在经过某一次分裂过程后效果并没有显著提升也必须要持续划分树枝，由此一来导致模型在进行迭代计算的过程中不得不做大量的无用功。以弥补 XGBoost 的不足之处为目的，同时在保证准确率的基础上推进梯度优化决策树模型的训练速度，微软分布式及其学习工具包团队于 2017 年在 GitHub 上开源了性能超群的 LightGBM，以 XGBoost 为基础作出了多种改进，具体总结如下：

① 以梯度的单边采样（GOSS：Gradient-based One-Side Sampling）为基础。以信息增益定义为依据，所有具备大梯度的实例都能够对信息增益提供更大的作用，梯度较小的样本进行后续的学习时，对于优化最终结果的精度来说没有明显的作用。所以，以维持信息增益估计的准确性为目的，针对实例展开采样过程中需要将梯度较大的实例进行保留，同时随机抽取部分梯度较小的样本，利用 GOSS 算法，能够有效保证学习精度的同时还能够提升学习的速度。

② 互补特征压缩（EFB：Exclusive Feature Bundling），是一种可以减少高维数据的特征数目并且使损失降到最小的一种算法[71]。高维的数据通常是十分稀疏的，并且许多特征之间是互斥的，因此可以将这些特征合并起来。

③ 直方图算法（Histogram-based Algorithm）。LightGBM 通过直方图算法

对互斥特征展开合并过程，尽管以 histogram 为基础的决策树算法也不是只有 LightGBM 才具有的，但是仍然具有与众不同的特点，它首先将连续的浮点特征值进行离散化，得到整数，与此同时构造直方图[72]。在遍历数据的过程中，将经过离散化的值进一步视为索引在直方图中不断积累的统计量，在遍历完一次数据后，直方图已经得到了足够的统计量，随后以直方图的离散值为依据继续遍历得到最佳的分割点。

④ 带深度限制的叶子生长策略（Leaf-wise）。XGBoost 算法在迭代这一步骤中选择 Leaf-wise 的叶子生长策略，也就是在同一时间分裂同一层的叶子，其优点为多线程优化的效果较为良好，同时对于模型的复杂程度来说，也较容易控制。然而 Leaf-wise 属于一种效率较低的算法，这是由于其无法区分叶子之间的区别，对同一层叶子一视同仁，许多叶子具有较低的分裂增殖，因此无需继续搜索、分裂，也就导致无形中的开销会增多。所以，LightGBM 选择一种效率更高的策略，每次都从全体目标叶子中搜寻分裂增益最大的叶子，随后进行分裂，不断循环这一过程。与 Leaf-wise 进行对比，如果分裂次数完全一致，那么 Leaf-wise 能够有效减少误差，从而提升精度，然而缺点也比较明显，可能会出现较深的决策树，造成过拟合的后果。所以 LightGBM 针对 Leaf-wise 专门添加了对最大深度的限制，从而保证较高的效率，还能够避免过拟合现象出现。

除此之外，在类别特征方面，LightGBM 具有分割最优、并行计算通信成本低等优势。因此，LightGBM 不仅能够缩减计算开销，能处理庞大的数据集，同时在维持高效计算的同时还能够保证一定的准确率，基于这些优点，本文决定使用该算法进行音乐评分预测，从而对用户进行偏好推荐。

2.3.5 LightGBM 的参数设置

LightGBM 算法主要通过表 2.1 中参数实现算法优化：

表 2.1 LightGBM 参数设置

参数名称	参数含义
leaves	每棵数的叶子数量
learning rate	学习率

续表 2.1 LightGBM 参数设置

max_depth	最大学习深度（控制过拟合现象）
min data	叶子数据的最小数（控制过拟合现象）
feature fraction	选择特征占总特征数的比例，取值为（0，1）
bagging fraction	选择数据占总数据量的比例，取值为（0，1）

2.4 推荐算法

目前推荐系统已被应用于电子商务、电影、音乐、图书检索等各领域，推荐系统中最常用的推荐算法如图 2.2 所示。

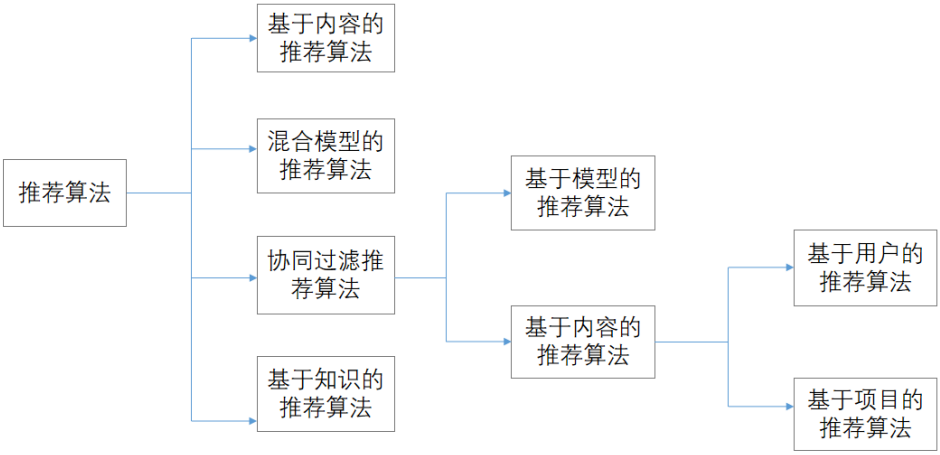


图2.2 常见推荐算法分类

2.4.1 基于内容的推荐

基于内容的推荐算法是信息过滤技术中的应用方式之一[72]，起初是在信息检索领域应用的**错误!未找到引用源。**。该算法侧重于分析项目的属性，比如，用户在某影视网站的浏览记录或者观看过的电影，分析这些记录或者影片的各种属性，这些属性包括电影的类型、内容以及演员等，例如用户观看了“雷神”此类型的电影，分析这部电影的内容信息，向用户推荐“复仇者联盟”这个电影，他可能会喜欢，于是提高了用户观看影视的观看率。因此，基于内容的推荐算法需要为用户推荐的项目具有多样性，如果为用户推荐的物品比较单一，那么整个推荐系统的质量

就会降低；如果能给用户更准确地推荐其它类别的商品，推荐系统的质量就会提高。该算法原理如图 2.3 所示。用户 A 喜欢物品 a，用户 C 喜欢物品 c，而物品 a 和物品 c 是相似项目，所以我们认为用户 C 也会喜欢物品 a，所以系统将物品 a 推荐给了目标用户 C。

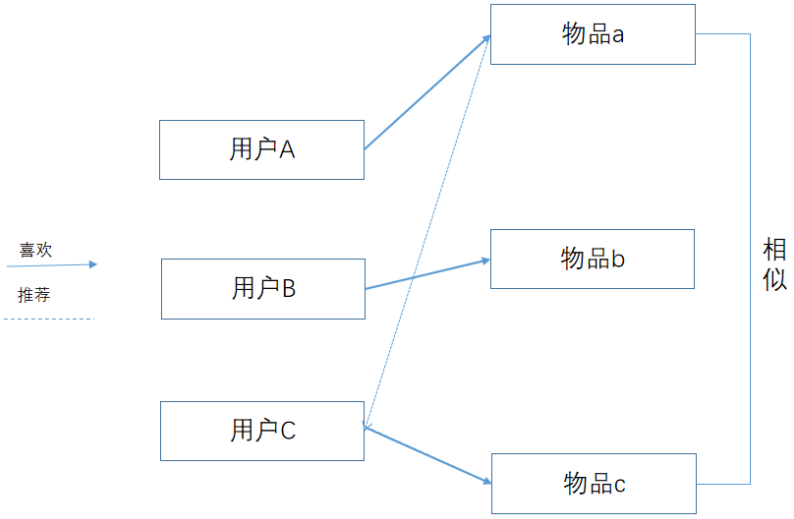


图 2.3　基于内容的推荐算法示例

2.4.2 协同过滤的推荐

协同过滤推荐技术。这一方法无需对项目内容特征进行全面考虑，其核心为分析“近邻”，例如生活中常见的同类集中现象。因此，若能够在已经了解偏好的用户群体中，寻找到与未知用户 A 具有相似特征的用户，将具有相似特征的用户所产生的购买、浏览记录作为依据，对未知用户 A 进行推荐，那么 A 可能对这些推荐产生兴趣，即为协同过滤的核心思路，重点在于搜寻与未知样本具有相似特征的样本，且以上举例内容，反过来针对商品也具有同样的适用性。协同过滤算法分类两类[74]，其结构如图 2.4 所示：

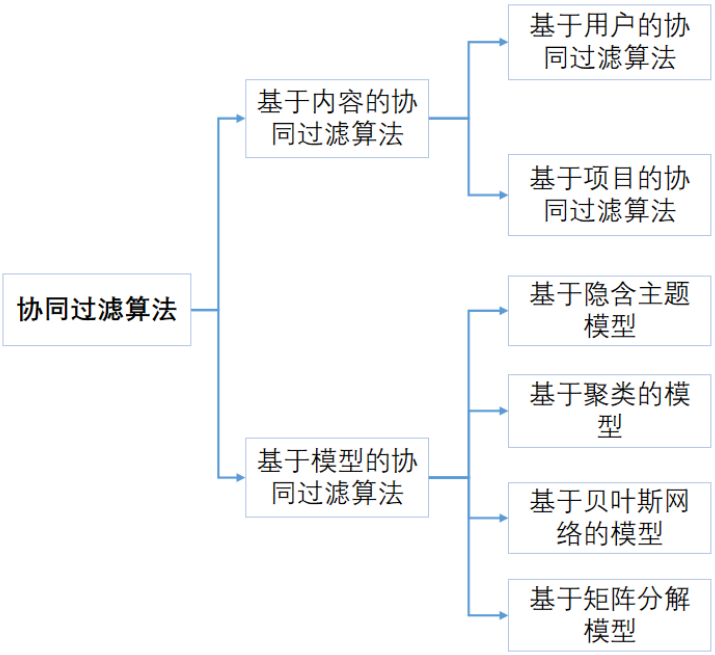


图 2.4 协同过滤算法的分类

1. 基于用户的协同过滤算法

UBCF（User-based Collaborative Filtering）推荐算法的步骤如下：（1）挖掘用户集合，其中用户的兴趣与目标用户是相似的。（2）分析上一步中用户集合感兴趣的物品，把这些物品与目标用户没有交互关系的物品推给该用户。第一步中是根据用户之间的兴趣相似度来挖掘的，基于用户的协同过滤算法就是根据用户的行为相似度来计算的。其推荐过程如图 2.5 所示。用户 A 与用户 C 都喜欢同一类型的物品（物品 a 与物品 c），因此，推荐系统会认为用户 A 与用户 C 是相似用户；而用户 C 还喜欢物品 d，所以该算法会预测用户 A 也喜欢物品 d，进而把物品 d 作为用户 A 感兴趣的项目推荐给用户 A。

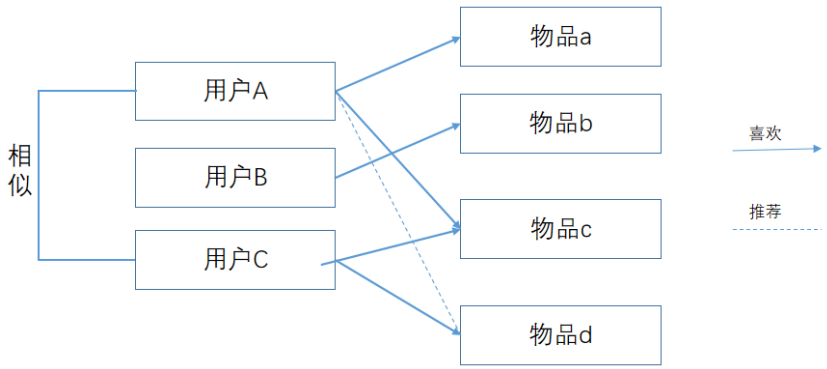


图 2.5 基于用户的协同过滤推荐示例

可以通过 Jaccard 计算用户 u 与 v 的兴趣相似度，如公式（2-6）：

$$w_{uv} = \frac{|N(u) \cap N(v)|}{|N(u) \cup N(v)|} \tag{2-6}$$

或者通过余弦相似度计算，如公式（2-7）：

$$w_{uv} = \frac{|N(u) \cap N(v)|}{\sqrt{|N(u)||N(v)|}} \tag{2-7}$$

其中，$|N(u) \cap N(v)|$是用户 u 与用户 v 感兴趣的物品集合的数量。

2. 基于物品的协同过滤算法

亚马逊公司于 2003 年提出了以物品为基础的协同过滤算法。该算法主要有两个步骤：（1）对目标物品和用户有过行为的物品间存在的相似度进行计算；（2）以计算所获取的物品与物品之间的相似度矩阵为依据展开后续的评分预测，同时对用户容易产生兴趣的物品纳入推荐范围。这一算法的基本原理为，如果两个物品间存在较高的相似度，那么对其中一个物品产生兴趣的用户很可能对另一个物品也会产生兴趣。网络的电商平台中产品通常较为稳定，因此物品间的相似度能够在线下预先进行计算，在推荐的过程中只需要分析用户已经完成评分操作的物品与其他物品存在的相似度，从而大幅度缩减了计算成本，还能够缩短计算所需时间。

IBCF（Item-based Collaborative Filtering）推荐算法，针对一些确定目标的项目，利用对项目的相似性进行计算，从而找到与上一个项目具有最高相似度的 K 个项目，随后将此类项目进一步推荐给对目标项目产生兴趣的用户[75]。推荐过程如图 2.6 所示。用户 A、用户 B 与用户 C 对物品 a 感兴趣，而用户 A 与用户 B 对物品 c 感兴趣，所以用户 A 与用户 C 被认为是相似用户，而用户 C 喜欢物品 a，所以推荐系统把物品 c 作为目标用户 C 感兴趣的物品进行推荐。

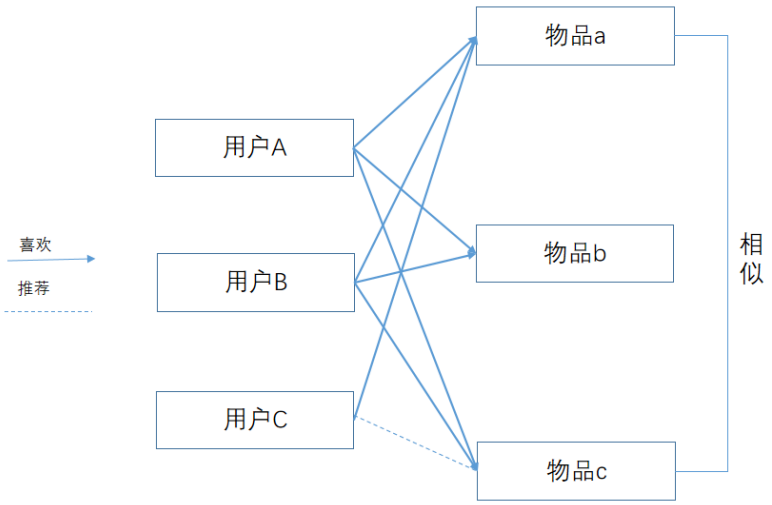


图 2.6 基于项目的协同过滤示例

通过公式（2-8）计算物品 i 和 j 的余弦相似度：

$$w_{ij} = \frac{|N(i) \cap N(j)|}{\sqrt{|N(i)||N(j)|}} \tag{2-8}$$

其中，$N(i)$是对物品 i 感兴趣的用户个数，$|N(i) \cap N(j)|$是对物品 i 与物品 j 的都感兴趣的用户数。IBCF 算法是通过计算物品的相似度，基于公式（2-9）计算用户 u 对物品 i 的喜好程度。

$$P(u,i) = \sum_{j \in S(i,K) \cap N(u)} W_{ij} r_{uj} \tag{2-9}$$

3. 基于内存的协同过滤算法

基于内存的协同过滤算法是通过用户（项目）的邻域来计算用户（项目）之间的相似度。该算法主要有两步：第一步，根据用户（项目）的历史数据，计算他们之间的相似度来挖掘相似的用户（项目）；第二步，通过这些相似用户（项目）的评分来获取目标用户的未知评分。协同过滤推荐算法中主要使用的相似度度量方法有以下几种：

（1） 余弦相似度，其原理简单，如式（2-10）所示：

$$sim(u,v) = \cos(a*b) = \frac{a*b}{|a|*|b|} \tag{2-10}$$

（2） 修正余弦相似度

修正的余弦相似度如式（2-11），修正的余弦相似度可以降低个别特殊用户

的评分影响。

$$\mathrm{si}\,m(u,v)=\frac{\sum_{i\in I_{uv}}(r_{ui}-\bar{r}_u)*(\mathrm{r}_{vi}-\bar{r}_v)}{\sqrt{\sum_{i\in I_u}(r_{ui}-\bar{r}_u)^2}\sqrt{\sum_{i\in I_v}(r_{vi}-\bar{r}_v)^2}} \tag{2-11}$$

（3） Pearson 相似度

Pearson 相似度如式（2-12），通过各数据集合之间的相似性来挖掘兴趣相似的用户。

$$\mathrm{si}\,m(u,v)=\frac{\sum_{i\in I_{uv}}(r_{ui}-\bar{r}_u)*(\mathrm{r}_{vi}-\bar{r}_v)}{\sqrt{\sum_{i\in I_{uv}}(r_{ui}-\bar{r}_u)^2}\sqrt{\sum_{i\in I_{uv}}(r_{vi}-\bar{r}_v)^2}} \tag{2-12}$$

（4） Jaccard 相似度

Jaccard 相似度与 Pearson 相似度类似，如式（2-13）。

$$sim(u,v)=\frac{|I_u\bigcap I_{\mathrm{v}}|}{|I_u\bigcup I_v|} \tag{2-13}$$

其中，$sim(u,v)$是 u 与 v 之间的相似度，I_u和I_v是用户评分的项目集合，r_{ui}和r_{vi}是用户 u 与 v 对项目 i 的评分，$\bar{r}_u$和$\bar{r}_v$分别是用户 u 与 v 都评分过的的项目集合$|I_u\bigcap I_{\mathrm{v}}|$的平均值。若$sim(u,v)$的值越大，代表两个用户 u 与 v 越相似。

通过以上相似度计算公式完成相似度计算后，运用式（2-14）预测用户 u 对项目 i 的评分。

$$P_{ui}=\bar{r}_u+\frac{\sum_{(v\in N_u)\wedge(r_{vi}!=0)}sim(u,v)*(r_{vi}-\bar{r}_v)}{\sum_{(v\in N_u)\wedge(r_{vi}!=0)}sim(u,v)} \tag{2-14}$$

其中，p_{ui}表示用户 u 对项目 i 的预测评分。

在推荐系统中，由于项目和用户的数量越来越大，基于内存的推荐算法计算复杂度也越来越大，这影响了推荐系统的性能。为了解决这个问题，学术界提出了一些基于机器学习的推荐算法，比如贝叶斯网络[76]，线性回归[77]，矩阵分解[78]等，这种融合了机器学习技术的模型就叫基于模型的协同过滤算法。

贝叶斯网络是一种有向无环图。在贝叶斯网络中，若节点变量 a 与 b 直接相连，说明 a 会影响 b 的可靠性，它们之间的箭头方向表示因果关系，两个变量之间会产生概率值。简单的贝叶斯网络图与概率计算公式如图 2.7 所示。

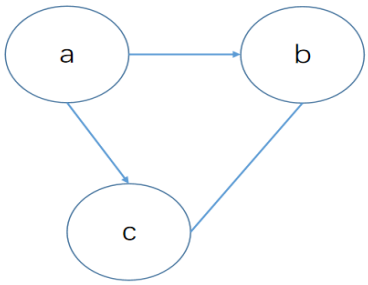


图 2.7　贝叶斯网络及概率计算公式

矩阵分解模型，主要涵盖了 BiasSVD，SVD++，TimeSVD 等模型。SVD++是以用户隐性评分为基础对用户展开建模并进一步法分析。TimeSVD 为一种以用户兴趣与时间相关的假设为基础构建的模型。BiasSVD 为一种针对较为极端的用户行为进行考虑的用户行为而推出的模型。

2.4.3 混合推荐

以上几种推荐算法各有优缺点，所以许多推荐系统会采用混合算法进行推荐。混合推荐算法通过融合多个推荐算法，利用每个算法的优点来达到比单一的推荐算法效果更好的目的。混合推荐方法有加权组合、动态组合和混合组合等方式。加权组合是对多种算法的结果予以不同的权重，最终按照加权之后的结果进行推荐，推荐模型如图 2.8 所示。动态组合根据用户对项目的评价选择不同的算法组合，推荐模型如图 2.9 所示。混合组合与动态组合完全不同，其以各种推荐算法产生的最终结果进行混合，随后生成一个全新的推荐列表，推荐模型如图 2.10 所示。

加权组合模型是以多种各不相同的推荐算法进行整合，形成一个具有独立性的推荐模块。推荐模型如下：

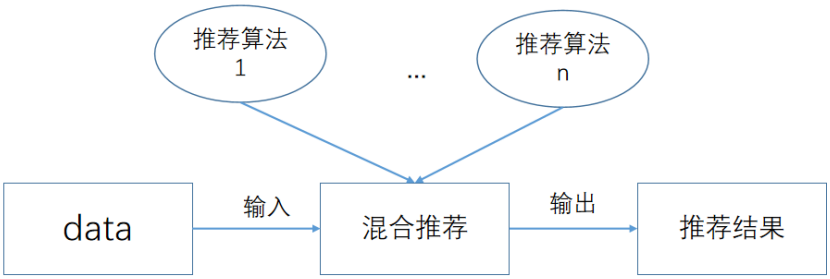


图 2.8 加权组合推荐模型

动态组合模型与流水线的结构相类似，其中包括 n 个推荐单元，每个单元都能够输入上一个单元的结果，除此之外这一单元的结果能够供下一单元进行输入：

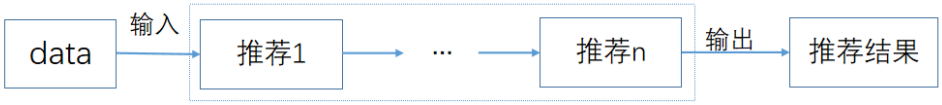


图 2.9 动态组合推荐模型

混合组合模型以各种算法最终获得的结果为要素展开混合处理过程，随后生成一个信度列表。通常情况下来说，此模型来自于两个以及两个以上具有独立性的推荐单元所产生的最终结果，根据一定的原则进行混合，从而生成新的推荐列表。推荐模型如图 2.10 所示。

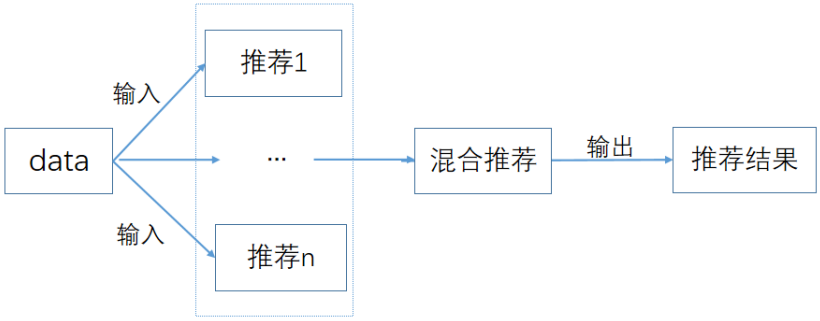


图 2.10 混合组合推荐模型

2.5 推荐算法总结

对以上推荐算法进行大致介绍后能够发现，各种算法都具有其优点与缺点，并且适用情况也各不相同。基于内容的推荐算法主要侧重于项目自身的属性，不过部分属性在获取的过程中存在一定的难度，但其产生的推荐结果是比较清晰直

观的，但是在对非结构化数据进行处理时，往往会产生比较离谱的结果。协同过滤推荐算法对于各种不规则的元素可以进行跨类别推荐，但其容易产生冷启动以及数据稀疏等问题。混合推荐算法能够将各种推荐算法进行有效结合，通过利用各自的优势，获得优于单一算法所产生的推荐结果。同时，除了以上所提及的各种普遍算法以外，还有以关联规则为基础的推荐算法、以效用为基础的推荐算法以及以知识为基础的推荐算法等等。

2.6 本章小结

本章介绍了与本文研究相关的理论基础，包括艾宾浩斯遗忘曲、时间衰减函数以及 LightGBM 算法与协同过滤算法的原理与应用。此外阐述了推荐系统中几种推荐算法存在的问题，只有充分了解推荐算法才能针对其做进一步的研究，也为本文后续内容的介绍打下了良好基础。最后我们总结了目前流行的几种推荐算法的优缺点以及如何克服这些缺点。

3 引入时间衰减函数的协同过滤音乐推荐算法

3.1 问题解决和思路描述

传统的协同过滤推荐算法侧重于挖掘用户的兴趣与项目之间的关系，在个性化推荐中也具有以下优点：基于用户行为和项目相似性，不需要先验知识，在用户行为比较丰富的时候，推荐效果不错，但是忽略了用户兴趣与时间变化之间的关系，如果同等对待不同时间段的项目评分，会降低对目标用户推荐的准确率。因此，为了解决用户的偏好变化问题以及评分稀疏问题从而有效的改进音乐推荐系统的质量，提出了一种结合用户评分与融合时间衰减函数获得音乐综合相似度的协同过滤音乐推荐方法（TDF-CF），该算法在传统协同过滤基础上结合了用户的遗忘规律与音乐综合相似度，从而进行预测评分。通过计算用户评分相似度与歌曲间的相似度，然后融合时间衰减函数得到音乐综合相似度，计算歌曲的预测评分。如果预测值大于等于阈值，则推荐歌曲；如果预测值小于阈值，则不推荐。该方法最大的优势为：考虑了时间因素对目标用户当前偏好的影响，对实验结果进行分析，表明改进的协同过滤算法比传统的协同过滤算法推荐效果好。下面的小节将进行详细的描述。

3.2 时间衰减函数

用户行为习惯是具有周期重复性的，但是用户的兴趣会随着时间的变化而衰减，进而影响用户的行为，这种变化符合人类的遗忘现象，因此将时间衰减函数融入到音乐推荐算法中，可以更好根据用户兴趣预测目标用户的偏好，提高推荐的准确率。以艾宾浩斯实验数据为模拟数据，观察遗忘曲线的数据分布，拟合指数函数和幂函数，接下来介绍指数型衰减函数和幂函数型衰减函数。

3.2.1 指数函数型衰减函数

艾宾浩斯的记忆与遗忘关系的数据如表 3.1 所示，输入实验数据，运用 Matlab 拟合艾氏曲线，选取指数函数与幂函数来逼近艾宾浩斯遗忘曲线，如图 3.1 所示。

表 3.1 艾宾浩斯实验数据

时间	记忆保留比率	记忆遗忘比例
记忆开始	100%	0
20 分钟后	58.2%	41.8%
一小时后	44.2%	56.6%
8-9 小时后	35.8%	64.2%
1 天后	33.7%	66.3%
2 天后	27.8%	72.2%
6 天后	25.4%	74.6%
31 天后	21.1%	78.9%

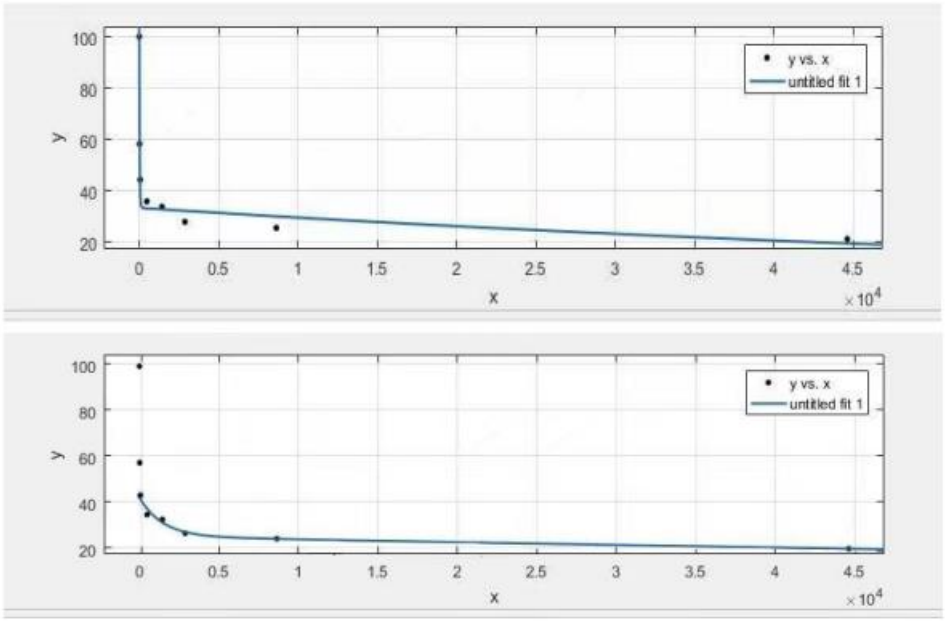


图 3.1 Matlab 曲线拟合图

选取指数函数对艾宾浩斯遗忘曲线进行函数拟合，公式如下：

$$y = 68.9e^{-0.045x} + 33.23e^{-0.123*10^{-4}x} \quad (3\text{-}1)$$

当上式中的 x 体现用户兴趣的时间时，可以将用户最近一次的行为时间与 i 时刻的行为时间作差来计算，根据时间数据除以相应的时间颗粒度（比如，若按分钟计算，时间颗粒度取 60），这样距离时间 i 越远，时间权重就越小，对用户的贡献也就越低。因此，可以得到用户随时间变化的衰减函数。

指数型时间衰减函数如式（3-2）所示：

$$y = 68.9e^{-0.045(\frac{T_{\max}-T_{ui}}{t})} + 33.23e^{-0.123*10^{-4}(\frac{T_{\max}-T_{ui}}{t})} \quad (3-2)$$

其中，$T_{\max}$ 是用户 u 最近一次访问项目的时间，T_{ui} 是用户 u 在时刻 i 时访问了项目的时间，t 是时间颗粒度。

3.2.2 幂函数型衰减函数

选取幂函数对艾宾浩斯遗忘曲线进行函数拟合，公式如下：

$$y = 84.21x^{-0.2383} + 15.66 \quad (3-3)$$

当上式中的 x 体现用户兴趣的时间时，可以得到用户随时间变化的衰减函数，幂函数型衰减函数如式（3-4）所示：

$$y = 84.21(\frac{T_{\max}-T_{ui}}{t})^{-0.02383} + 15.66 \quad (3-4)$$

其中，$T_{\max}$ 是用户 u 最近一次访问项目的时间，T_{ui} 是用户 u 在时刻 i 时访问了项目的时间，t 是时间颗粒度。

3.3 综合相似度

根据用户的收听记录，结合内容特征与协同过滤的特点，计算歌曲之间的评分相似度，但是由于数据稀疏性，获得的歌曲相似度矩阵也存在稀疏性，所以若只通过评分相似度对用户进行推荐，会降低推荐结果的准确性，而通过音乐多样性与语义性的特点，结合音乐的信息内容，可以为音乐分类带来高效、多元化的效果，这些信息内容客观地描述了音乐的多方面特征，具有较高的语义解释性。根据音乐的信息内容特征，计算相似度，使得推荐的音乐更具有真实性。因此将评分相似度与歌曲相似度相融合，得到综合相似度，这可以解决矩阵稀疏性的问题，再结合时间效应，对综合相似度进行衰减修正，能更好地预测用户偏好，所以提出了一种引入时间效应的音乐推荐方法，来提高对目标用户的推荐准确率。

3.3.1 获取评分

通过使用 Last.fm 数据库获得的信息来推荐歌曲，由于数据集中所包含的内容通常是对某些数据的统计值，不会包含用户相关偏好的任何有价值信息，因此，为了更好的预估用户评分，往往借助数据集中的唯一可反馈客户有用信息的播放次数数据进行分析。播放次数可解释为某一用户对某一歌曲、某一歌手的收听次数，这一信息可间接的预测出用户的音乐偏好。播放次数的多少还可作为研究某一歌手当前的社会性接受程度，某一歌手的普遍播放次数都较高，则说明该歌手的受欢迎程度较高，用户通过收听某一歌手也可表示了该用户的喜好程度。文献**错误!未找到引用源。**中所提及的方法适用于具有明确幂律分布的播放频率。

对歌手 i 与用户 j 的播放频率定义如下：

$$Freq_{i,j}=\frac{p_{i,j}}{\sum_{i'} p_{i',j}} \tag{3-5}$$

其中，$p_{i,j}$ 是用户 j 播放歌手 i 的次数。

用户 j 对第 k 位歌手的评分 $r_{k,j}$ 可以用频率线性函数计算，如式（3-6）：

$$r_{k,j}=4*(1-\sum_{k'=1}^{k-1} Freq_{k'}(j)) \tag{3-6}$$

其中，$Freq_{k'}(j)$ 表示用户 j 收听过的次数最多歌手列表中的第 k 位歌手的播放频率。

3.3.2 评分相似度

协同过滤推荐算法通过计算评分相似度，运用评分矩阵对偏好相似的用户或者项目进行推荐。首先通过公式（3-6）计算评分，获得用户歌曲评分矩阵用户歌曲评分矩阵 R，如式（3-7），然后根据相似歌曲之间的关系，寻找相似度高的其他歌曲，将其推荐给目标用户。计算歌曲 i_x 与 i_y 的评分相似度，即对不同用户收听的歌曲评分进行比较，评分相近的歌曲相似度高，所以可以利用评分计算相似度。

$$R=\begin{bmatrix} \mathrm{r}_{11} & \cdots & \mathrm{r}_{1j} & \cdots & r_{1n} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ r_{i1} & \cdots & r_{ij} & \cdots & r_{an} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ r_{m1} & \cdots & r_{mj} & \cdots & r_{mn} \end{bmatrix} \tag{3-7}$$

其中，$\mathrm{r}_{i,j}$ 表示第 i 个用户对第 j 个项目的评分值。

根据修正的余弦相似度公式计算评分相似度，如式（3-8）所示。

$$sim_1(i_x,i_y)=\frac{\sum_{u\in U_{i_x,i_y}}(r_{u,i_x}-\overline{r}_{i_x})\times(r_{u,i_y}-\overline{r}_{i_y})}{\sqrt{\sum_{u\in U_{i_x}}(r_{u,i_x}-\overline{r}_{i_x})^2}\sqrt{\sum_{u\in U_{i_y}}(r_{u,i_y}-\overline{r}_{i_y})^2}} \tag{3-8}$$

其中，$sim(i_x,i_y)$ 是歌曲 i_x 与歌曲 i_y 间的评分相似度，U_{i_x,i_y} 是对歌曲 i_x 与 i_y 表现出偏好的所有用户集合，r_{u,i_x} 是用户 u 对歌曲 i_x 的评分，$\overline{r}_{i_x}$ 与 $\overline{r}_{i_y}$ 是对歌曲 i_x 与 i_y 评分的平均值。

3.3.3 歌曲相似度

用户有多种选择收听某一段音乐的原因，根据其偏好进行推荐的效果会更好，比如，对喜欢收听流行音乐的用户推荐流行音乐时，会更符合用户的喜好；对喜欢某位歌手的用户推荐这位歌手的音乐时，也会更符合该用户的偏好，所以通过挖掘音乐的各种特征可以提高音乐推荐的准确率。对于歌曲来说，它具有许多丰富的信息特征，比如歌曲的发布时间、歌手、流派等，这些数据特征都是用户在选择收听歌曲时的原因，通过分析这些数据特征给能够得到歌曲的信息相似度。

各大音乐平台拥有庞大的音乐数据库，面对海量的音乐数据，用户很难找到自己感兴趣的音乐，所以为了方便用户可以快速地查找自己感兴趣的音乐，许多音乐平台会对音乐进行标签分类，比如流派标签、有流行、爵士、摇滚等，用户可以通过标签，直接在音乐分类中找到自己喜欢的音乐，为用户提供更便捷的查找渠道。对于歌曲的分类，就是根据歌曲的标签信息对其进行分类，计算歌曲相似度，可以把相似度高的歌曲分为同一类型。

随着推荐技术的不断发展，歌曲的信息特征成为推荐算法中十分重要的研究内容，把歌曲的流派、发布时间、歌手等信息相融合以后，计算歌曲相似度，可以降低相似度矩阵的稀疏性。本文是以流派、发布时间、歌手这三种歌曲信息标签来计算歌曲相似度。

（1）流派标签。流派标签作为标签属性，有流行、爵士、摇滚等类型，各标签之间的相似度计算如式（3-9）所示。

$$SS_{1(i_x,i_y)}=\frac{alo}{com} \tag{3-9}$$

其中，$SS_{1(i_x,i_y)}$是歌曲i_x与歌曲i_y之间的流派相似度，alo 是两首歌曲的共有标签数，com 是该属性的所有标签数。

（2）发布时间。分段处理歌曲发布的时间，比如以 5 年划分一段，2010-2015 年标记为 1，2015-2020 年标记为 2，划分后利用公式（3-10）计算两首歌曲的发布时间相似度。

$$SS_{2(i_x,i_y)}=e^{-|T\mathrm{ime}_{ix}-Time_{iy}|} \tag{3-10}$$

其中，$SS_{2(i_x,i_y)}$是歌曲i_x与i_y间的发布时间相似度，$T\mathrm{ime}_{ix}$与$T\mathrm{ime}_{iy}$分别表示两首歌曲的发布时间所在的分段值。

（3）歌手。把歌手标签看作二元属性，比如歌曲i_x与i_y的歌手相同就标记为 1，不同就标记为 0，用$SS_{3(i_x,i_y)}$表示两首歌曲间的歌手相似度。

这些标签信息都比较直观，根据用户的收听记录就可以获得这些信息，而且这三种标签具有相关性，比如歌手 A 在某一时间段具有较大的影响力，则他的歌曲所属流派在这一时间段也会流行。把流派、发布时间、歌手这三种标签信息进行加权融合，可以计算歌曲信息相似度，如式（3-11）所示。

$$\begin{aligned} &sim_2(i_x,i_y)=f_1SS_{1(i_x,i_y)}+f_2SS_{2(i_x,i_y)}+f_3SS_{3(i_x,i_y)} \\ &\sum_{g=1}^{3}f_g=1 \end{aligned} \tag{3-11}$$

其中，$\mathrm{sim}_2(i_x,i_y)$表示歌曲的信息相似度，f_1、f_2、f_3分别表示流派、发布时间、歌手三种标签的权重。

3.3.4 歌曲综合相似度

歌曲的评分相似度与歌曲的信息相似度是为用户进行推荐时的依据，表示不同歌曲间的偏好相关性。将两者融合，能够得到歌曲相似矩阵，从而降低了歌曲综合相似度矩阵的稀疏性。

对歌曲的评分相似度 $\mathrm{sim}_1(i_x,i_y)$ 与歌曲的信息相似度 $\mathrm{sim}_2(i_x,i_y)$ 进行加权求和，得到歌曲综合相似度 $sim_3(i_x,i_y)$，如式（3-12）所示。

$$sim_3(i_x,i_y)=\alpha sim_1(i_x,i_y)+\beta sim_2(i_x,i_y) \tag{3-12}$$

其中，参数 α 、β 分别表示歌曲评分相似度和歌曲相似度所占的权重。

3.3.5 融入时间衰减函数的综合相似度

在计算综合相似度时融入时间衰减函数，可以客观地反映用户的兴趣变化，从而提高对用户推荐的准确性。根据用户的收听记录，首先计算本条记录中的待测音乐和历史记录中的歌曲综合相似度，然后融入时间衰减函数得到音乐综合相似度。音乐综合相似度是指，历史记录音乐和待测音乐的综合相似度与时间衰减函数（两种时间衰减）的乘积的平均值，如式（3-13）所示。

$$sim(i_x,i_y)=\frac{1}{n}\cdot y\cdot sim_3(i_x,i_y) \tag{3-13}$$

其中，n 是历史收听音乐的数量，y 是时间衰减函数，$sim_3(i_x,i_y)$ 是歌曲综合相似度。

3.3.6 评分预测

根据训练数据集中用户收听歌曲的历史记录，依据公式（3-6）获得评分数据，利用修正的余弦相似度方法，计算歌曲的评分相似度。

根据公式（3-13）计算音乐综合相似度之后，预测目标用户对歌曲的评分，比如，用户 u 对歌曲 i_x 的预测评分 r_{u,i_x} 的计算公式如式（3-14）所示：

$$\mathrm{r}_{u,i_x}=\overline{r}_{ix}+\frac{\sum_{i_y\in I(i_x)}sim_3(i_x,i_y)(\mathrm{r}_{u,i_y}-\overline{\mathrm{r}}_{i_y})}{\sum_{i_y\in I(i_x)}sim_3(i_x,i_y)} \tag{3-14}$$

其中，$\overline{r}_{ix}$表示歌曲i_x的平均得分，$I(i_x)$表示用户 u 评过分的歌曲集合，$sim_3(i_x,i_y)$表示歌曲i_x和歌曲i_y的相似度。

得到用户对歌曲的评分预测值后，将预测值大于等于阈值的歌曲进行推荐，小于阈值的歌曲不予推荐。遍历完测试集所有记录，最终结果为推荐的，预测正确，不推荐的，预测错误，可计算预测的准确度。

3.4 评价指标

在推荐系统中，为用户提供个性化推荐服务时，一般采用推荐列表的形式进行推荐，本文实验中采用了两种指标：平均准确率 MAP 和均方根误差 RMSE。MAP 是对推荐列表中每首歌的平均准确率计算其平均值，反映了推荐算法在所有音乐数据上推荐性能的单值指标，如公式（3-15）所示。

$$MAP=\frac{|L_{\mathrm{i}}\bigcap T_{\mathrm{i}}|}{|T_{\mathrm{i}}|} \tag{3-15}$$

其中，L_{i}表示对用户u_i推荐的歌曲列表，T_i表示在测试集中听过该歌曲的用户u_i。

RMSE 是协同过滤算法最常用的评估方法，用来度量预测精度，它是预测值与真实值之间差异的标准差，可以评估真实值与预测值之间的偏差，如公式（3-16）。

$$RMSE=\sqrt{\frac{1}{|\mathrm{c}|}\sum_{(i,j)\in c}(\hat{r}_{ij}-r_{ij})^2} \tag{3-16}$$

其中，c 是用户 j 已评分的项目数量，$\hat{r}_{ij}$是用户对歌曲的评分，r_{ij}是用户对歌曲的预测评分。

3.5 实验设置

3.5.1 实验环境

本文的实验环境如表 3.2 所示。

表 3.2 实验环境

实验环境	
操作系统	Windows 7 64 位操作系统
CPU	Intel(R) Core(TM) i7-9750H CPU @2.6GHz 2.59GHz
RAM	8.0GB
编程语言	Python
软件环境为	spyder3.6，Matlab

3.5.2 实验数据

实验所中用的数据集是国际上公开的音乐数据集“Last.fm Dataset-1k”，它是具有上下文信息的隐性反馈数据集的代表。该数据包含 584897 首歌曲，522366 个独特的标签，8598630 个轨道-标签对，56506688 个轨道-类似轨道对。Last.fm 数据集分别分布在用户信息与歌曲信息两个文件中，其中用户信息包括用户的国家、年龄、性别、注册时间等信息；歌曲信息包括用户所有的音乐播放记录、播放时间、音乐的名称以及作曲家的名字等信息。

在数据预处理时，先对时间进行特征处理，将常规时间转换为时间戳（timestamp）的形式，方便在实验过程中使用，其次，数据集中的一些属性（比如 user_id、artist_id 等）以字符串的形式储存，为了降低在实验过程中这些特征对内存的占用，以及为了避免预处理前后，这些特征的转化不影响实验结果，本文对数据进行 labelencode 编码处理，将所需特征转换为连续的数值型变量。根据标识对对数据进行整合后进行实验，实验时选择 70%的数据集用于训练，30%的数据集用于测试。

3.5.3 对比方法与参数描述

1. 对比方法

通过对比实验来验证提出的方法的有效性，对比实验：基于两种时间衰减函数的协同过滤推荐方法（TDF-CF）和传统的协同过滤推荐方法（CF）。

CF（Collaborative Filtering），即传统的协同过滤方法，利用用户的显示评分来计算相似度。

TDF-CF（Collaborative Filtering based on time decay function），即基于时间衰减函数的的协同过滤方法，在计算相似度时考虑了时间对用户偏好的影响。

实验步骤：

输入：输入训练集 S，它是一个$m \times n$的用户评分矩阵，其中有 m 个用户 U={u1，u2，u3.....um}，n 首歌曲 M={m1，m2，m3.....mn}，r_{ui}为用户 u 对歌曲 i 的评分。

输出：RMSE 值

Step1：利用公式（3-6）计算用户评分，构建用户评分矩阵 S。

Step2：利用公式（3-9）、（3-11）、（3-12）分别计算评分相似度、歌曲相似度与歌曲综合相似度。

Step3：利用公式（3-13）进行时间衰减，得到音乐综合相似度（对比实验中没有此步）。

Step4：利用公式（3-14）预测评分，计算 RMSE 的值。

2. 参数设置

经过多次实验，权重参数如表 3.3 所示。计算歌曲信息相似度时，歌曲流派、发布时间、歌手三者之间存在一定的联系，比如某一歌手在某一时间发布他的专辑，所以将他的歌曲的这三类标签信息所占的权重均设置为 1/3。计算歌曲相似度时，歌曲的评分相似度与歌曲的信息相似度的权重设置如表 3.3。

表 3.3 相似度权重参数

α	β
0.3	0.7

续表 3.3 相似度权重参数

0.4	0.6
0.5	0.5
0.7	0.3
0.6	0.4

经过多次实验后，当 α=0.6， β=0.4 时实验的效果最好，这说明用户对歌曲的评分更能表达用户的兴趣偏好，同时歌曲的标签信息也会影响用户的偏好选择。

3.5.4 实验结果与分析

依据本章 4.3.5 节提出的融合时间因素的音乐推荐方法，通过用户听歌的时间与基于项目的协同过滤推荐方法相结合，首先计算音乐综合相似度，然后预测评分。取不同阈值，对比两种时间衰减系数影响下推荐准确度，并用 RMSE 进行模型评估。

推荐算法的所有参数调整结束后，对不同算法的性能进行比较。基于 Last.fm 数据集，根据上一节 3.4 的评价指标，将融合了两种时间衰减函数的推荐算法 TDF-CF 与基于项目的协同过滤歌曲推荐算法在平均推荐准确率 MAP 和召回率 RMSE 上进行对比。对比结果如图 3.2 和表 3.4 所示。

表 3.4 不同阈值的推荐准确率

阈值	融入幂函数的 CF	融入指数函数的 CF	CF
0.3	0.958	0.958	0.824
0.35	0.958	0.654	0.537
0.4	0.958	0.38	0.281
0.45	0.524	0.223	0.075
0.5	0.115	0.076	0.024

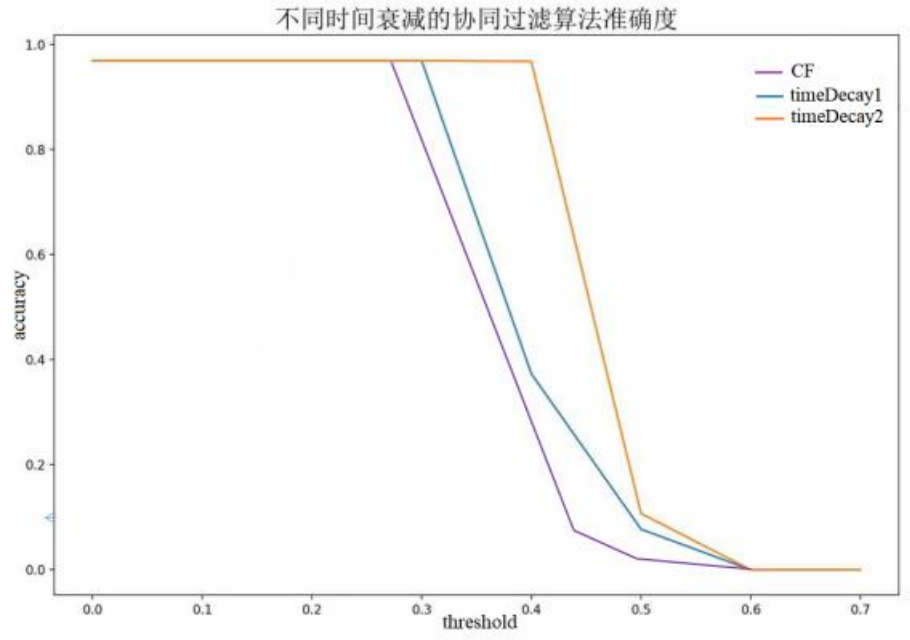


图 3.2 TD-IBCF 与 IBCF 的准确度对比

从图 3.2 中可以看到，将指数型时间衰减函数引入到协同过滤模型中时，用户在阈值 0.35-0.45 之间的音乐推荐结果最佳，将幂函数型时间衰减函数引入到协同过滤模型中时，用户在阈值 0.42-0.5 之间的音乐推荐结果最佳且是有意义的。

表 3.5 不同方法的 RMSE 值

迭代次数	融入幂函数的 CF	融入指数函数的 CF	CF
1	0.232031	0.257731	0.267031
5	0.152062	0.167862	0.182062
10	0.0819173	0.1069173	0.1169173
15	0.0497363	0.0729008	0.0799003
20	0.0240519	0.0408384	0.0456519

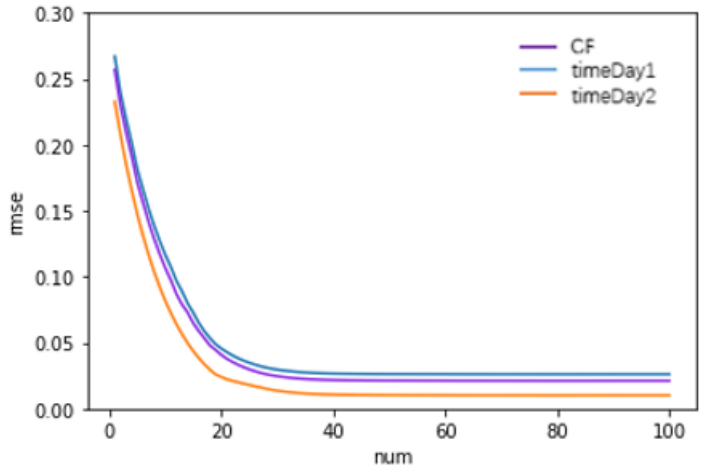


图3.3 融入时间衰减函数后的RMSE值对比

通过实验比较两个算法的 RMSE，如表 3.5 和图 3.3 所示，引入幂函数型时间衰减函数的协同过滤推荐算法比引入指数函数型时间衰减函数的协同过滤推荐算法的 RMSE 值更小，所以幂函数型的遗忘曲线对于推荐更有优势，而融合时间效应的协同过滤推荐算法的 RMSE 值比传统的协同过滤推荐算法小，可见，引入了时间衰减函数的协同过滤推荐算法（TDF-CF）的推荐效果优于传统的协同过滤推荐算法（CF）。综上所述，融入了幂函数的 TDF-CF 算法的推荐结果更符合目标用户的偏好。

3.6 本章小结

本章通过实验验证了本文提出的改进算法的有效性。首先，介绍了目前在协同过滤算法研究领域内的现状，其次根据余弦相似度公式，融合评分相似度与歌曲相似度，并进行时间衰减，得到音乐综合相似度，然后进行评分预测。通过多组实验分析，确定算法中各个参数的数值，最后，设计对比试验，将本文提出的 TDF-CF 算法和传统的协同过滤推荐算法在评价指标 MAP 与 RMSE 上进行比较。实验表明，提出的 TDF-CF 音乐推荐方法的准确度比传统的协同过滤推荐方法高，推荐效果也更好。因此，结合了评分机制与时间效应的音乐推荐方法，不仅提高了推荐的准确度，还解决了用户偏好随时间变化的问题，相较于其他传统的推荐算法，改进后的算法推荐准确率更高。

4 引入时间衰减函数的 LightGBM 音乐推荐算法

4.1 问题描述和解决思路

推荐系统中，要提高给用户推荐信息的准确性就需要先对用户的偏好进行有效预测。用户的偏好即用户对某一项目或某一类型的商品的喜好程度，推荐系统通过对用户的偏好进行合理分析，预测用户可能感兴趣的商品或者项目类型，再结合该目标用户在某些网页中的浏览记录、购买记录及项目相关类型信息，就能更准确的获得目标用户的偏好。在传统音乐类的推荐系统中，仅仅考虑了用户对于已收听音乐和收藏的音乐进行偏好分析，却忽略了由于时间推移而导致用户的喜好发生的变化，使得用户对于所推荐音乐的满意程度较低。推荐系统中的时间信息是一个很重要的判断依据，对用户偏好的研究往往需要结合时间变化对其产生的影响来进行合理预测，所以可以将反映人们遗忘规律的曲线引入到音乐推荐算法中，用时间衰减函数来模拟用户的兴趣的变化趋势，而 LightGBM 算法是近几年在机器学习领域比较前沿的算法，它的提出不但降低了计算开销，提升了模型的计算效率，并且在维持较高计算效率的情形下还能得到较高的准确率。因此，本文基于艾宾浩斯遗忘曲线理论，提出基于两种时间衰减函数的 LightGBM 音乐推荐模型（TDF-LGBM），融合音乐数据特征与用户行为特征，针对用户兴趣对项目进行更准确地推荐，从而提高音乐推荐的准确度。

4.3 引入时间效应的 LightGBM 的音乐推荐算法

在上一章 3.2 节中描述了时间衰减函数，考虑到时间对用户兴趣的影响，将其融入到用户评分中，对评分进行修正，然后在 LightGBM 模型中进行训练和测试，并对模型进行评估。下面将对引入了时间效应的 LightGBM 音乐推荐算法进行描述。

4.3.1 融入时间衰减函数的评分修正

根据 3.3.1 节的公式（3-6），得到用户评分后，融合时间衰减函数，根据公式（4-1）对评分数据进行衰减修正。

$$r_{k',j} = y \cdot r_{k,j} \tag{4-1}$$

其中，y 是时间衰减函数，$r_{k,j}$ 是用户评分。

通过时间衰减函数来模拟人类大脑对事物的遗忘过程，进而修正用户对某一项目的评分，时间衰减函数呈现出的项目评分值随着时间的变化而变化，经过一段时间后，人类对这一项目逐渐淡忘，致使用户在对这一项目的评分会衰减，到某一特定程度后不再随时间变化而改变，这个时候所表现出来的状态，就是用户评分处于最终的遗忘状态。

4.3.2 基于 LightGBM 的歌曲评分预测

LightGBM模型的训练是通过构建的LightGBM模型对选择的训练样本进行训练，不断调整各种参数，逐渐减小模型的误差。训练前，需要预先设定模型训练参数，比如学习率、迭代次数、可接受误差等，直至模型误差在期望误差范围内或者满足预先设定的条件即可完成模型的训练，否则需要不断地调整模型参数，直到模型训练成功。模型训练完成后，输入测试集的数据生成预测结果，计算预测结果与实际结果之间的误差。如果误差符合预期要求，则可以将模型用于音乐推荐，如果不符合，那么需要找出问题原因，不断地优化模型结构和参数，重复进行模型训练与测试，直到模型表现符合预期要求为止。

在使用LightGBM模型进行预测时，计算量比较大，所以在模型训练及预测时，需要使用相关软件进行实验，本文的实验采用python语言构建与训练模型，python语言包含丰富的程序包，Lightgbm程序包是专门为LightGBM模型训练及预测而设计的，本文通过Lightgbm程序包中对Scikit-learn支持的API进行模型训练和测试。参数的设置对模型的训练与评估有很大的影响，训练过程中通过不断地调参得到最终的LightGBM模型，LightGBM模型部分参数的设置如表4.1。

表4.1 部分参数设置

参数名	参数含义	参数值
num_leaves	基学习器叶子节点最大数	31
num_trees	提升树数量	800

续表4.1 部分参数设置

eval_metric	损失函数惩罚项	L2
learn_rate	学习率	0.05

基于LightGBM的音乐推荐算法流程如图4.1。根据公式（4-1），对修正过的评分数据与用户信息数据、歌曲信息数据进行整合，作为完整的数据集应用到LightGBM模型中，预测每一位用户听歌的评分 $\hat{r}_{ij}$ 。

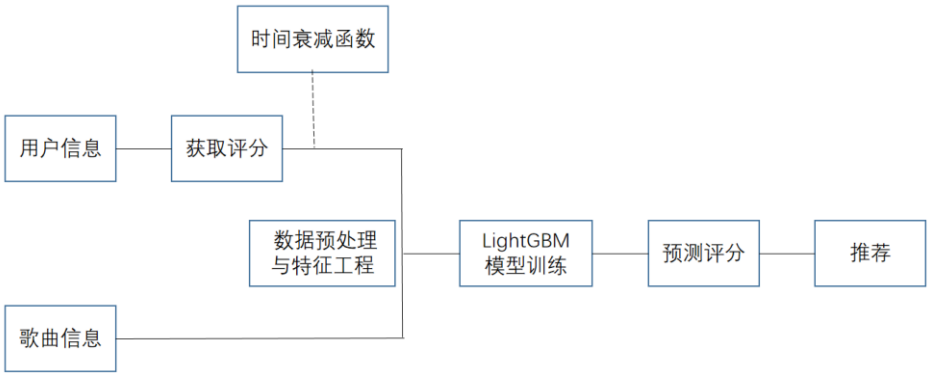


图4.1 LightGBM音乐推荐算法流程图

4.3 实验设置

4.3.1 实验数据

本实验采用 3.5.2 节中描述的数据集进行实验分析，将预处理后的原始文件的数据整合到一个文件中，实验的数据集包含了 12293 个用户、320 位歌手以及相关的 18698 条记录。实验时，选择 70%的数据集用于训练，30%的数据集用于测试。在训练集上建立用户兴趣模型，并在测试数据集上验证方法的性能。

4.3.2 实验结果与分析

为了对改进的 LightGBM 方法的性能进行评估，在 Last.fm 数据集进行了三个方面的实验：

（1）在 LightGBM 算法中引入指数型衰减函数和幂函数型时间衰减函数的

对比分析；

（2）将引入时间效应的TDF-LGBM算法与未引入时间效应的LightGBM算法进行比较；

（3）将引入时间效应的 TDF-LGBM 算法与第三章中引入时间效应的TDF-CF 推荐算法进行对比分析。

使用 LightGBM 完成用户评分模型训练，并在测试集上进行验证，对于测试集的记录，模型预测的用户评分大于等于阈值，则进行推荐，如果小于阈值，则不推荐。实验结果如表 4.2 和图 4.2 所示。

表 4.2 不同阈值的推荐准确率

阈值	融入幂函数的 LGBM	融入指数函数的 LGBM	LGBM
0.3	1	1	0.792
0.35	1	0.5799	0.512
0.4	1	0.51	0.182
0.45	0.582	0.182	0.093
0.5	0.0358	0.026	0.031

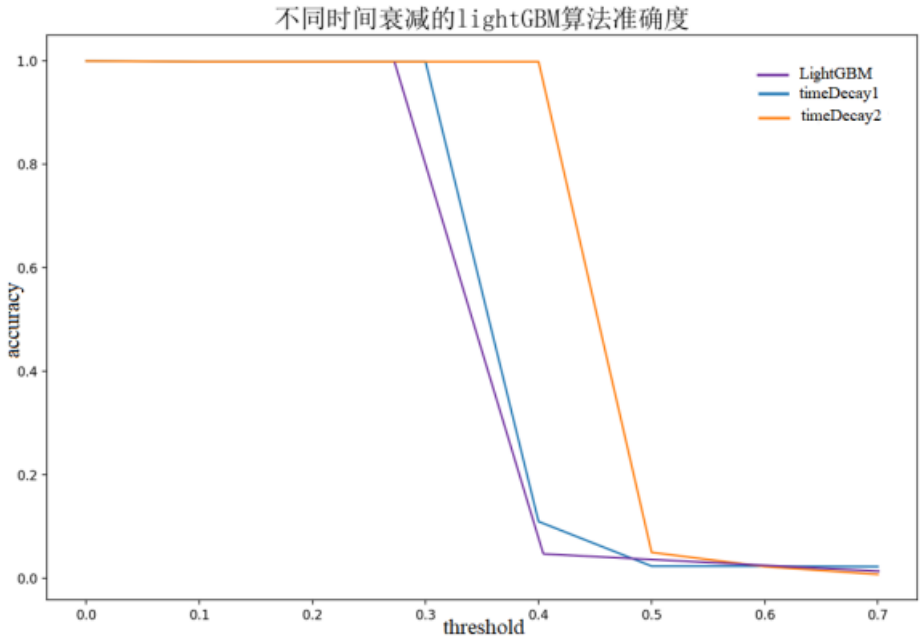


图 4.2 融入时间衰减函数 LightGBM 算法准确度对比

从表 4.3 和图 4.3 中可以看到，将指数型时间衰减函数引入到模型中时，用

户在阈值 0.3-0.35 之间的音乐推荐结果最佳，将幂函数型时间衰减函数引入到模型中时，用户在阈值 0.4-0.45 之间的音乐推荐结果最佳且是有意义的。

表 4.3 不同方法的 RMSE 值

迭代次数	融入幂函数的 LGBM	融入指数函数的 LGBM	LGBM
1	0.20009	0.23907	0.261731
5	0.131511	0.141511	0.162862
10	0.0788954	0.0888954	0.0969173
15	0.049596	0.0582	0.0587228
20	0.029509	0.0356519	0.0376519

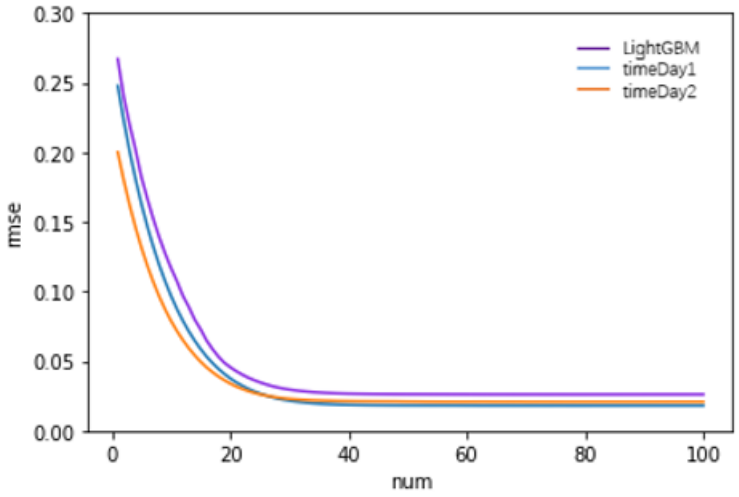


图4.3 融入时间衰减函数RMSE值对比

实验中使用均方根误差对模型进行评估，经过实验对比，引入幂函数型时间衰减函数的 LightGBM 音乐推荐方法比引入指数函数型时间衰减函数的 LightGBM 音乐推荐方法 RMSE 值低（如表 4.3 和图 4.3），所以幂函数型的遗忘曲线对于推荐更有优势。引入合时间效应的 LightGBM 推荐方法的 RMSE 值比未引入时间效应的 LightGBM 推荐方法低，所以引入了时间衰减函数的 LightGBM 推荐方法（TDF-LGBM）的推荐效果优于未引入时间效应的 LightGBM 音乐推荐方法。综上所述，引入幂函数型时间衰减函数的 LightGBM 音乐推荐模型更能够向用户推荐其感兴趣的音乐。

4.4 两种模型实验结果对比

基于公开音乐数据集 Last.fm，对第三章的实验结果与本章实验结果进行对比分析，在同一阈值下，第三章提出的融入时间衰减函数的协同过滤（TDF-CF）音乐推荐模型的准确率比 TDF-LGBM 音乐推荐模型的准确率比低，对比两个模型的 RMSE 值，TDF-LGBM 音乐推荐模型的 RMSE 值更低，表 4.4 是将引入幂函数型时间衰减函数的两种模型的 RMSE 值进行比较，可见，引入了时间效应的 TDF-LGBM 音乐推荐模型的推荐效果比引入时间衰减函数的协同过滤推荐模型（TDF-CF）的效果更好。因此，在音乐推荐过程中，引入了时间效应的 TDF-LGBM 音乐推荐模型的推荐可以更好的根据用户的偏好进行推荐，从而提高推荐的准确率。

表4.4 两种模型RMSE值对比

迭代次数	TDF-LGBM	TDF-CF
1	0.20009	0.232031
5	0.131511	0.152062
10	0.0788954	0.0819173
15	0.049596	0.0497363
20	0.029509	0.0260519

4.5 本章小结

艾氏曲线认为人类的记忆与遗忘是具有一定规律性的，本文对推荐系统的研究以遗忘曲线为基础，结合两种不同的时间衰减函数，将艾氏曲线的思想广泛应用到推荐系统的开发研究中。人类的自然认知以及人类的大脑记忆都在一定程度上符合记忆的遗忘规律，随着时间的流逝，用户的偏好也潜移默化地发生改变，在本文的研究讨论中，基于用户偏好随时间变化的特征，使用时间衰减函数对用户的评分进行时间上的衰减，提高了推荐系统的推荐准确性。

通过仿真实验验证本文提出的模型的有效性。首先对 Last.fm 实验数据集进行预处理操作；其次建立模型，通过多组实验分析，确定算法中各个参数的数值。然后设计对比试验，将本文提出的 TDF-LGBM 推荐算法和未引入时间效应的

LightGBM 算法在推荐准确率上进行对比，并用 RMSE 进行模型的评估；最后将实验结果与第三章中结合时间效应地协同过滤方法（TDF-CF）进行对比。实验结果表明，融合了时间效应的 TDF-LGBM 推荐算法优于融合了时间效应的 TDF-CF 推荐算法，提高了对目标用户的推荐准确率。

5 总结与展望

5.1 总结

随着音乐的发展，顺势而生的各类音乐流媒体的数量不断扩大，众多在线音乐平台逐渐出现在人们的视线范围中，例如：Pandora、网易云等音乐播放器，要想吸引更多用户并留住用户，就需要对不同用户的喜好进行后台分析，挖掘并推荐符合用户偏好的音乐是不同音乐平台争相研究的热门话题。在大数据信息快速发展的时代，传统的音乐搜索引擎工具已远远满足不了用户对音乐偏好的定位，音乐平台要想得以长久发展，就必须突破传统搜索引擎的弊端，通过用户的搜索记录、音乐播放次数等进行分析，发掘用户潜在的音乐偏好特征，建立用户的特有音乐偏好模型，在用户收听音乐时为其提供个性化的推荐服务。艾氏曲线认为人类的记忆遗忘现象是具有一定规律性的，所以本文对推荐系统的研究以遗忘曲线为基础，结合两种不同的时间衰减函数，将艾氏曲线的思想广泛应用到推荐系统的开发研究中，使推荐系统的认知度得以提升，在一定程度上可以提高推荐系统的性能，也更符合自然发展规律，这种采用人们自然认识规则的方法为今后人们在设计个性化推荐系统时奠定了基础，以便能更有效的提供个性化的推荐服务给系统用户。

本文对基于用户偏好的个性化音乐推荐进行了深入研究，主要完成了以下工作：

（1）详细阐述了音乐推荐系统的研究现状，通过对各研究领域所使用的推荐方法进行分析，总结其优缺点，基于音乐自身内容的本质特征具有难以提取性，所以在实际音乐推荐方法中，根据音乐内容进行推荐的方式还有待进一步的研究。此外，目前市场上的音乐平台所使用的推荐方法在对用户的音乐偏好上还需要不断完善，现有的推荐方法对用户的不同偏好缺乏更深层次的研究，没有建立合适的推荐模型来合理综合地利用这些多方面的偏好特征。

（2）提出了引入时间衰减函数的协同过滤音乐推荐方法。通过分析用户的行为历史记录，获取用户评分，将用户对于音乐的收听频度转化为用户对于音乐的评分值，在数据集 last.fm 上结合时间效应，计算音乐综合相似度，从而为用

户做出推荐，实验结果表明，引入时间衰减函数的协同过滤音乐推荐方法（TDF-CF）在推荐准确性上优于传统协同过滤音乐推荐方法（CF），一定程度上提高了音乐推荐的准确性。

（3）提出了基于时间衰减函数的 LightGBM 音乐推荐算法。将用户对歌曲的评分进行时间上的衰减修正，以此来反映用户的偏好随着时间的变化而变化。在数据集 last.fm 上进行实验，实验结果表明，融合时间衰减函数的 LightGBM 音乐推荐方法（TDF-LGBM）在推荐准确性上优于未融合时间衰减函数的 LightGBM 音乐推荐方法。最后对比分析本文提出的两种改进的音乐推荐方法，得出结论，即结合时间效应的 TDF-LGBM 音乐推荐方法的推荐结果优于结合时间效应的 TDF-CF 音乐推荐方法。因此，结合时间效应的 TDF-LGBM 音乐推荐方法可以为用户提供更满意的个性化音乐推荐。

5.2 展望

本文结合时间效应与用户兴趣变化之间的关系，提出了引入时间衰减函数的 LightGBM 音乐推荐方法（TDF-LGBM）和引入时间衰减函数的协同过滤音乐推荐方法（TDF-CF），虽然在推荐效果上有了一定的提升，但仍存在一些地方需要进一步改善：

（1）本文提出的两种推荐方法，虽然能在一定程度上提高了推荐准确率，但是在较大音乐平台中的推荐系统中，所面临的难题是对新用户的信息进行收集，要想平台得以长久发展，需要挖掘新用户在平台中的主动行为并对其偏好进行预测，为用户制定个性化的合理推荐。

（2）本文研究的个性化音乐推荐的基础是对时间衰减函数的理论研究，对用户的音乐偏好进行预测时，利用了现有歌曲和歌手的属性标签，但没有对这些标签的上下文环境和类型进行具体分析，因此，在以后更深入的研究分析中，可以将这些不同的标签进行分类，从而更准确的对用户的音乐偏好进行更高效地定位。

（3）本文使用的数据集的评分是通过播放次数与频率之间的关系来计算，若能获取真实的用户评分音乐数据，改进的推荐算法的推荐效果会更好。

（4）人们对音乐的追求程度不断提升，音乐的数量也在持续增加，各大音乐平台获取用户信息后，对于音乐推荐系统的研究还可以从深度学习训练音乐特

征方面进行考虑。此外，随着大数据平台的不断普及，未来可充分利用 hadoop 等大数据平台对用户的信息进行提取和分析，以便更高效地获得用户的音乐偏好，为用户更好地提供个性化推荐服务。

参考文献

[1] Endres I, Hoiem D. Category-Independent Object Proposals with Diverse Ranking[J]. IEEE Transactions on Pattern Analysis & Machine Intelligence, 2014, 36(2): 222-234.

[2] 李丽, 戚桂杰. 从雅虎的分类目录分析信息构建的发展[J]. 情报理论与实践, 2006, 29(2): 164-167.

[3] Su A J, Kuzmanovic A, et al. How to Improve Your Search Engine Ranking: Myths and Reality[J]. Acm Transactions on the Web, 2014, 8(2): 1-25.

[4] Cambazoglu B B, Altingovde I S, Ozcan R, et al. Cache-Based Query Processing for Search Engines[J]. ACM Transactions on the Web, 2012, 6(4): 1-24.

[5] 时念云, 李月. 基于情境感知的个性化推荐算法[J]. 计算机系统应用, 2017, 26(9): 135-139.

[6] 冷亚军, 陆青, 梁昌勇. 协同过滤推荐技术综述[J]. 模式识别与人工智能, 2014, 27(8): 720-734.

[7] 高凤丽, 孙连山. 个性化推荐系统概述[J]. 技术与市场, 2015, 22(2): 78-79.

[8] 顾丽敏. 个性化推荐系统研究[J]. 无线互联科技, 2013(8): 53-53.

[9] HERMANN E. Memory: A contribution to experimental psychology[EB/OL]. [2011-12-09].

[10] 刘建国, 周涛, 汪秉宏. 个性化推荐系统的研究进展[J]. 自然科学进展, 2009, 19 (1): 1-15.

[11] Nitanda A, Suzuki T. Functional Gradient Boosting based on Residual Network Perception. 2018.

[12] Zeng L, Lin L. An Interactive Vocabulary Learning System Based on Word Frequency Lists and Ebbinghaus'Curve of Forgetting[C]. Digital Media and Digital Content Management (DMDCM), 2011. Workshop on. IEEE, 2011: 313-317.

[13] 朱天宇, 黄振亚, 陈恩红. 基于认知诊断的个性化试题推荐方法[J]. 计算机学报, 2017(1): 176-191.

[14] Horowitz D, Kamvar S D. The anatomy of a large-scale social search engine[C]. Proceedings of the 19th International Conference on World Wide Web. ACM, 2010: 431-440.

[15] Chang C C, Lin C J. LIBSVM: A library for support vector machines[M]. ACM Transactions on Intelligent Systems and Technology, Vol. 2, No. 3, 2011, PP. 1-27.

[16] 刘威. 面向普适计算的上下文建模和推理技术研究[D]. 南京邮电大学, 2011.

[17] Raymond J. Mooney. Content-based book recommending using learning for text categorization[C]. Submission to Fourth ACM Conference on Digital Libraries, 2000.

[18] Shen J, Shepherd J, Ngu A H H. Towards Effective Content-Based Music Retrieval With Multiple Acoustic Feature Combination[J]. IEEE Transactions on Multimedia, 2006, 8(6): 1179-1189.

[19] Segal A, Katzir Z, Shapira B, et al. EduRank: A Collaborative Filtering. Approach to Personalization in E-learning[C]. Educational Data Mining. 2014.

[20] 曹毅. 基于内容和协同过滤的混合模式推荐技术研究[D]. 中南大学, 2007.

[21] Burke R. Hybrid Recommender Systems: Survey and Experiments [C]. Interaction. 2002: 331-370.

[22] Hu Y, Koren Y, Volinsky C. Collaborative Filtering for Implicit Feedback Datasets[C].Eighth IEEE International Conference on Data Mining. IEEE,2009:263-272.

[23] Pan R, Zhou Y, Cao B, et al. One-Class Collaborative Filtering[C]. Eighth IEEE International Conference on Data Mining. IEEE Computer Society, 2008: 502-511.

[24] Koren Y, Bell R, Volinsky C. Matrix Factorization Techniques for Recommender Systems[J]. Computer, 2009, 42(8): 30-37.

[25] Nabizadeh A H, Yu Y. Predicting User Preference Based on Matrix Factorization by Exploiting Music Attributes[C]. International Conference on Computer Science & Software Engineering. ACM, 2016: 61-66.

[26] Su J H, Chang W Y, Tseng V S. Personalized Music Recommendation by Mining Social Media Tags [J]. Procedia Computer Science, 2013, 22: 303-312.

[27] Oord A V D, Dieleman S, Schrauwen B. Deep content-based music recommendation[J]. Advances in Neural Information Processing Systems, 2013, 26: 2643-2651.

[28] FOUSS F, PIROTTE A, RENDERS M. et al. Random-walk computation of similarities between nodes of a graph with application to collaborative recommendation[J]. Knowledge and Data Engineering, 2007, 19(3): 355-369.

[29] Karen H, Tso K, Schmidt-Thieme L. Empirical Analysis of Attribute-Aware

Recommendation Algorithms with Variable Synthetic Data[J]. Studies in Classification Data Analysis & Knowledge Organization, 2006: 271-278.

[30] Bala A, Kaur T. Local texton XOR patterns: A new feature descriptor for content-based image retrieval[J]. Engineering Science & Technology An International Journal, 2016, 19(1): 101-112.

[31] Lee S, Masoud M, Balaji J, et al. A survey of tag-based information retrieval[J]. International Journal of Multimedia Information Retrieval, 2016: 1-15.

[32] Li S, Li J, Pan R. Tag-weighted topic model for mining semi-structured documents[C]. International Joint Conference on Artificial Intelligence. 2013:2855-2861.

[33] Wang. Image Tag Recommendation Algorithm Using Tensor Factorization[J]. Journal of Multimedia, 2014, 9(3): 416-422.

[34] Nanopoulos A, Karydis I. Know Thy Neighbor: Combining audio features and social tags for effective music similarity[C]. IEEE International Conference on Acoustics, Speech and Signal Processing. IEEE, 2011: 165-168.

[35] Hariri N, Mobasher B, Burke R. Using social tags to infer context in hybrid music recommendation[C]. Twelfth International Workshop on Web Information and Data Management. ACM, 2012: 41-48.

[36] Hariri N, Mobasher B, Burke R. Context-aware music recommendation based on latenttopic sequential patterns[C]. ACM Conference on Recommender Systems. ACM, 2012: 131-138.

[37] Dias R, Fonseca M J. Improving Music Recommendation in Session-Base Collaborative Filtering by Using Temporal Context[C]. Proceedings of the 2013 IEEE 25th International Conference on Tools with Artificial Intelligence, 2013: 783-788.

[38] Liu J Y, Yang Y H. Inferring personal traits from music listening history[C]. International ACM Workshop on Music Information Retrieval with User-Centered and Multimodal Strategies. ACM, 2012: 31-36.

[39] Dolgikh D. Graph-based music recommendation approach using social network analysis and community detection method[C]. International Conference on Computer Systems and Technologies. ACM, 2015: 221-227.

[40] Aaron, Sander, Dieleman, Schrauwen, Benjamin. Deep content-based music

recommendation[J]. Advances in Neural Information Processing Systems, 2013(5): 29-35.

[41] Shen J, Pang H H, Wang M, et al. Modeling concept dynamics for large scale music search.[J]. Research Collection School of Information Systems, 2012, 29(4): 455-464.

[42] M. Gorgoglione, U. Panniello, A. Tuzhifin. The Effect of Context-aware Recommendations on Customer Purchasing Behavior and Trust[C]. Proceedings of the fifth ACM conference on recommender systems. 2011: 85-92.

[43] Hariri N, Mobasher B, Burke R, Context-aware music recommendation based on latenttopic sequential patterns[C]. RecSys'12 Proceedings of the sixth ACM conference on Recommender systems, Dublin, Ireland, 2012, PP. 131-138.

[44] X. Wang, D. Rosenblum, Y. Wang. Context-aware mobile music recommendation for daily activities[C]. MM'12 Proceedings of the 20th CM international conference on Multimedia, Nara, Japan, 2012, PP. 99-108.

[45] K. Kapoor, V. Komal, et al. I like to explore sometimes[C]. RecSys'15 Proceedings of the 9th CM Conference on Recommender Systems, Vienna, Austria, 2015, PP. 19-26.

[46] V. Saúl, P. Castells. Rank and relevance in novelty and diversity metrics for recommender systems[C]. RecSys '11 Proceedings of the fifth ACM conference on Recommender systems, Chicago, Illinois, 2011, PP. 109-116.

[47] [1] Nanou T, Lekakos G, Fouskas K. The effects of recommendations' presentation on persuasion and satisfaction in a movie recommender system[J]. Multimedia Systems, 2010, 16(4): 219-230.

[48] Y. Zhang, D. Quercia, T. Jambor. Auralist: Introducing serendipity into music recommendation[C]. WSDM'12 Proceedings of the fifth CM international conference on Web search and data mining, Seattle, Washington, 2012, PP. 13-22.

[49] M. Schedl, D. Hauger. Tailoring Music Recommendations to Users by Considering Diversity, Mainstreaminess, and Novelty[C]. SIGIR'15 Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval. Santiago, Chile, 2015, PP. 947-950.

[50] Campos P, F Diez, A Bellogin. Temporal Rating Rabits: A Valuable Tool for Rating Discrimination[C]. Proceedings of the 2nd Challenge on Context-Aware Movie

Recommendation. 2011: 29-35.

[51] CHEN W, HSU W, LEE M L. Modeling user's receptiveness over time for recommendation[C]. Proceeding of the 36th International ACM SIGIR Conference on Research and Development in Information Retrieval. New York. ACM. 2013: 373-382.

[52] 汪静，印鉴，郑利荣等. 基于共同评分和相似性权重的协同过滤推荐算法[J]. 计算机科学, 2010, 37(2): 99-104.

[53] L. Baltrunas, X. Amatriain. Towards Time-dependant Recommendation Based on Implicit Feedback[C]. Workshop on context-aware recommender systems (CARS'09). 2009: 1-5.

[54] U. Panniello, A. Tuzhilin, M. Gorgoglione, et al. Experimental Comparison of Prevs. Post-filtering Approaches in Context-aware Recommender Systerns[C]. Proceedings of the third ACM conference on recommender systems. 2009: 265-268.

[55] 朱思丞，黄瑛，孙志. 推荐算法时间动态特性研究进展[J]. 工业控制计算机，2015, 28(8): 99-100.

[56] 郭晶晶，马建峰. 面向虚拟社区物联网的信任推荐算法[J]. 西安电子科技大学学报(自然科学版), 015, 42(2): 59-65.

[57] 孙光福，吴乐，刘淇，等. 基于时序行为的协同过滤推荐算法[J]. 软件学报，2013, 24(11): 271-2733.

[58] Ding Y, Li X Time weight collaborative filtering[C]. CIKM'05 Proceedings of the 14th ACM international conference on Information and knowledge management, 2005: 485-492.

[59] Liu NN, Zhao M, Xiang E, et al. Online evolutionary collaborative filtering[C]. RecSys'10 Proceedings of the fourth ACM conference on Recommender systems. 2010: 95-102.

[60] Li L, Qin S, Guo F. A Listwise Collaborative Filtering based on Temporal [C]. ICCIS 2017, 2017: 340-344.

[61] Hariri N Mobasher B, Burke R. Context-aware music recommendation based on latent topic sequential pattems[C]. ACM conference on Recommender Systems. ACM, 2012: 131-138.

[62] 乐国安. 心理学教授谈记忆魔法——艾宾浩斯遗忘曲线[J]. 中学生英语(高中版), 2008, 000(0Z6):49-50.

[63] HERMANN E．Memory: a contribution to experimental psychology[EB/OL]. [2011-12 -09]. http://psy.ed.asu.edu /~ classics/Ebbinghaus/index.html．

[64] NITANDA A, SUZUKI T. Functional gradient boosting based on residual network perception[C]. International Conferenee on Machine Learning (ICML), 2018.

[65] KADIYALA A, KUMAR A. Applications of python to evaluate performance of decision tree-based boosting algorithms[J]. Environmental Progress Sustainable Energy, 2018, 37 (2): 618-623.

[66] Rongfei J, Maozhong J, Chao L. Using Temporal Information to Improve Predictive Accuracy of Collaborative Filtering Algorithms[C]. APWEB'10 Proceedings of the 2010 12th International Asia-Pacific Web Conference, 2010: 301-306.

[67] Nasraoui O, Cerwinske J, Rojas C, et al. Performance of Recommendation Systems in Dynamic Streaming Environments(C). Siam International Conference on Data Mining, 2009: 569-574.

[68] Ray S, Sharma A. A Collaborative Filtering Based Approach for Recommending Elective Courses[J]. Information Intelligence, Systems, Technology and Management, 2011: 330-339.

[69] Guolin Ke, Qi Meng, Thomas Finely, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye, Tie-Yan Liu. LightGBM: A Highly Efficient Gradient Boosting Decision Tree[J]. Advances in Neural Information Processing Systems 30(NIP 2017). December 2017.

[70] 马晓君，沙靖岚，牛雪琪. 基于 LighIGBM 算法的 P2P 项目信用评级模型的设计及应用数量经济技术经济研究, 2018, 35(5): 145-161.

[71] 王华勇，杨超，唐华. 基于 LightGBM 改进的 GBDT 短期负荷预测研究[J]. 自动化仪表, 2018(9): 76-78.

[72] 张丹峰. 基于 LightGBM，XGBoost，ERT 混合模型的风机叶片结冰预测研究[D]. 上海师范大学, 2018.

[73] Belkin N J, Croft W B. Information filtering and information retrieval: two sides of the same coin[J]. Communications of the ACM, 1992, 35(12): 29-38.

[74] 郭艳红. 推荐系统的协同过滤算法与应用研究[D]. 大连理工大学, 2008.

[75] Ana Belén Barragáns-Martínez, Costa-Montenegro E, Burguillo J C, et al. A hybrid content-based and item-based collaborative filtering approach to recommend TV

programs enhanced with singular value decomposition[J]. Information Sciences, 2010, 180(22): 4290-4311.

[76] 王丹, 贺钱琛. 基于贝叶斯信念网络的协同过滤算法的研究[J]. 现代工业经济和信息化, 2016, 6(8): 81-82.

[77] Bundasak S, Chinnasarn K. Emenu recommender system using collaborative filtering and Slope One Predictor[C]. The 2013 10th International Joint Conference on Computer Science and Software Engineering (JCSSE). IEEE, 2013: 37-42.

[78] 杨亚东, 熊庆国. 基于动态标签偏好信任概率矩阵分解模型的推荐算法[J]. 计算机工程, 2017, 43(10): 160-166.

[79] Lin C Y, Wang L C, Tsai K H. Hybrid Real-Time Matrix Factorization for Implicit Feedback Recommendation Systems[J]. IEEE Access, 2018, PP(99): 1-1.

致 谢

时光荏苒，三年的求学生涯即将结束，不知不觉三年的研究生时光已经悄然而逝，在这三年里，有过欢喜，有过悲伤，但终究有了满满的收获。兰州财经大学赋予我的每一帧回忆都充满无限感激，在这里我认识了博学多才的老师，活泼上进的同学，在这里我努力研读所学课业并积极的拓宽自身视野扩大知识面。研究生的求学生涯是我人生中如明珠一般的宝贵财富，值得我用一生去回味与感激，我能顺利毕业离不开老师和同学们的指导和帮助，在这里真心的感谢他们每一个人！

感谢我的导师米红娟教授，在学业上，老师对我一直是严格要求，让我有了扎实的专业理论知识和丰富的实践经验；生活中，米老师对我仿佛亲人般的关怀，让我身在学校却感受到了家的温暖。从老师身上，我学到了严于律己、宽以待人的处世态度，学到了谦虚谨慎、严谨细致的工作作风，这些都将伴随着我的一生，它们将是我一辈子取之不尽、用之不竭的宝贵财富。我一定会一直谨记米老师的教诲，在今后的成长道路上踏实奋进，再次感谢米老师三年来对我的关怀和帮助，祝米老师今后身体健康，永远年轻！

感谢学院领导和学院所有传道授业解惑之恩师，承蒙教诲，这三年学习生涯不断提高专业知识和技能，祝各位老师万事如意，在学术研究上更上一层楼！

感谢我的父母，焉得艾草，言树之心，养育之恩，无法回报，谢谢你们让我的身后永远有一个温馨的港湾，祝愿我的父母永远身体健康，平安喜乐！

感谢我的同学、室友以及师姐们，在生活中我们互相陪伴，在学习中我们共同进步，共同度过充实而美好的三年研究生生涯，愿她们未来光明且如遂！

三年即逝，转眼将各奔东西。祝愿你们都能实现理想，平安快乐，万事顺遂。最后，致自己，道阻且长，不忘初心！