



南京大學

研究生畢業論文 (申請碩士學位)

論文題目 一種基於多维時間序列分析的音乐推荐系統研究與實現

作者姓名 王守濤

學科、專業方向 計算機技術

指導教師 徐鋒 教授

研究方向 可信計算

2014 年 5 月

学 号 : MF1133033

论文答辩日期 : 2014 年 6 月 1 日

指 导 教 师 : (签字)

Research and Implementation of A Music Recommender System Based on Multidimensional Time Series Analysis

by

Shoutao Wang

Directed by

Professor Xu Feng

Computer Technology

May 2014

*Submitted in partial fulfilment of the requirements
for the degree of Master in Computer Technology*

南京大学研究生毕业论文中文摘要首页用纸

毕业论文题目：一种基于多维时间序列分析的音乐推荐系统研究与实现

计算机技术 专业 2014 级硕士生姓名：王守涛

指导教师（姓名、职称）：徐锋 教授

摘 要

近些年来，互联网相关技术得到了快速发展，而信息过载 [34]和选择悖论 [29]的问题也愈发显著。为了提升信息的利用率，推荐系统应运而生并已广泛应用于电子商务、音乐电台、个性化阅读等领域。音乐作为一种强力的沟通和自我表达工具已经成为人们日常生活中一种非常重要的消费品,而相关研究也表明人们平时的听歌行为要远远多于读书、看电视、看电影等行为 [24]。为了解决人们在音乐消费过程中遇到的信息过载和选择悖论问题，一些个性化音乐推荐系统被开发了出来，比如Last.fm [15]、Pandora [21]、豆瓣音乐 [6] 等。然而，大多数音乐推荐系统没有充分考虑到用户听歌行为与其所处上下文环境之间的密切关系，导致做出的推荐不符合用户的即时需求，而一些考虑用户上下文的工作对用户行为历史的分析也不够细致和全面。此外，用户的未来行为受到其长期行为、即时行为 and 中期行为的共同影响，但目前未见工作将这几方面的影响综合起来加以考察。因此，本文给出了一种基于多维时间序列分析的个性化音乐推荐系统以全面细致地对用户行为进行分析。本文的主要贡献和工作如下所示：

1. 提出了一种基于多维时间序列分析的音乐推荐方法。该方法首先使用主题模型将歌曲表示成由若干隐含主题构成的概率分布，进而将用户行为建模为多维时间序列并对该序列进行分析和预测，从而实现了用户对用户行为的定量分析。

2. 从更细的粒度对用户听歌行为进行分析和预测，认为用户的听歌行为在每个隐含主题上都对应着一个时间序列，进而通过对这些时间序列的分析全面考察每一个隐含主题的贡献和作用。

3. 给出了一种基于用户三期行为的综合音乐推荐方法，综合考虑了用户长期行为、中期行为以及即时行为对用户未来行为的影响和作用。

4. 实现了一个原型系统，验证了本文所述方法的可行性。其中，本文利用分布式实时计算框架Storm实现了所述方法的并行化，提升了系统的执行效率。

关键词： 音乐推荐系统; 主题模型; 多维时间序列; 综合推荐

南京大学研究生毕业论文英文摘要首页用纸

THESIS: Research and Implementation of A Music Recommender System Based on Multidimensional Time Series Analysis

SPECIALIZATION: Computer Technology

POSTGRADUATE: Shoutao Wang

MENTOR: Professor Xu Feng

Abstract

In recent years, internet-related technologies have grown rapidly. However, information overload problem and paradox of choice become much more obviously. In order to enhance the utilization of information, recommender systems appear and have been applied to many fields, such as e-commerce, music, and and reading. Being a powerful communication and self-express tool, music has been an important consumer goods in people's daily life and previous research has also indicated that participants listened to music more often than other activities, such as watching television, reading books and watching movies. To alleviating the information overload problem and paradox of choice in music field, some music recommender systems have been developed, such as Lastfm, Douban, Pandora. However, most of recent music recommender systems do not take the close relationship between user's behavior and his/her current context into consideration, leading to bad recommendations that do not match user's current requirement. Although there are some researchers have done some related works, they are not meticulous and comprehensive enough. Besides, user's future behaviors are affected by his/her long-term behaviors, immediately behaviors and middle-term behaviors at the same time and there are few work giving a composite idea on them. In order to get better recommendations, we propose a music recommender system based on multidimensional time series analysis that can analyze the user's behavior more meticulous and comprehensive. Our main contribution and work are as follows:

1. A method for music recommendation based on multidimensional time series analysis was proposed. The method firstly using topic models to model a song as a probability distribution that is constituted by some latent topics and then it model the

user's behavior as a multidimensional time series. By analyzing the time series, the method can forecast the user's future behavior in a quantitative way.

2. User's behavior was analyzed in finer granularity. With the basic idea of thinking every latent topic corresponds to a time series, we analyze every time series and forecast the future values leading to comprehensive analyzing the influence of all latent topics.

3. A composite music recommender method was proposed. The method takes into consideration user's long-term behavior, immediate behavior and middle-term behavior at the same time.

4. A prototype that is used to verify the feasibility of our recommender methods was implemented. Specially, we paralleled the method with an distributed realtime computation framework-Storm, making the system running faster.

Keywords: Music Recommender System, Topic Model, Multidimensional Time Series, Composite Recommendation

目 录

目录	iv
第一章 背景介绍	1
第二章 相关工作	5
2.1 音乐推荐算法	5
2.1.1 基于用户及时行为的推荐	5
2.1.2 基于用户长期行为的推荐	6
2.1.3 基于用户中期行为的推荐	7
2.2 相似度度量	8
2.2.1 余弦相似度	8
2.2.2 KL距离	8
2.2.3 Hellinger距离	9
2.3 评测指标	9
2.3.1 用户满意度	9
2.3.2 预测准确度	10
2.3.3 其他评测指标	10
2.4 分布式实时计算系统	11
2.4.1 基本组成	11
2.4.2 分布式过程调用	13
2.5 文本建模	13
2.5.1 向量空间模型	14
2.5.2 隐含狄利克雷模型	15
2.6 时间序列预测	16
2.6.1 简单平均法	17
2.6.2 指数平滑法	17
2.6.3 差分整合移动平均自回归模型	17

第三章	一种基于多维时间序列分析的音乐推荐方法	19
3.1	问题描述	19
3.2	方法框架	21
3.3	主题模型建模	23
3.4	多维时间序列构造与预测	24
3.5	相似度计算	25
3.6	推荐列表生成	25
3.7	实验设计和结果	26
3.7.1	数据收集	26
3.7.2	评测标准	26
3.7.3	实验设置	26
3.7.4	结果分析	26
第四章	一种基于用户三期行为的综合音乐推荐方法	27
4.1	问题分析	27
4.2	常用综合策略	27
4.2.1	整体式综合	27
4.2.2	并行式综合	27
4.2.3	流水线式综合	27
4.3	方法详述	27
4.3.1	方法框架	27
4.3.2	方法选取	27
4.3.3	方法综合	27
4.4	实验结果	27
4.5	分析讨论	27
第五章	系统实现	28
5.1	系统架构	28
5.2	离线处理模块	28
5.2.1	爬虫系统	28
5.2.2	结巴分词	28
5.2.3	Gensim软件包	28

5.3	在线处理模块	28
5.3.1	序列生成	28
5.3.2	推荐引擎	28
5.4	系统效果	28
第六章	总结与展望	29
6.1	工作总结	29
6.2	工作展望	29
附录 A	数据集使用说明	30
A.1	权利声明	30
A.2	数据特点	30
A.3	组织形式	30
A.4	字段解析	31
A.4.1	scale	31
A.4.2	playlist	31
A.4.3	toptag	32
A.4.4	其他字段	32
A.5	应用场景	32
	参考文献	33
	简历与科研成果	36
	致谢	37

表 格

1.1	信息过载问题主流解决方案	1
3.1	歌曲Squares的显著主题及隶属度	24
A.1	数据集的统计信息	32

插图

1.1	Amazon电商推荐及PPTV视频推荐	2
1.2	豆瓣及虾米音乐电台示意图	3
2.1	典型Twitter Storm示意图	12
2.2	DRPC工作流程示意图	14
3.1	下一首歌预测问题的说明图	19
3.2	基于多维时序分析的音乐推荐方法的工作流程示意图	22
3.3	Last.fm中歌曲Squares的标签云	23

第一章 背景介绍

近些年来，互联网相关技术得到了快速发展，信息呈指数级增长，这造成了严重的信息过载(Information Overload)问题 [34]。特别是随着大数据时代的来临，信息的增长速度已经远远超过人们可以理解的速度，而海量信息中更是包含着很多对人们来说无用的冗余信息以及会干扰人们抉择的错误信息。面对海量的信息和众多的选择，人们往往无所适从而陷入到选择悖论(Paradox of Choice) [29]之中，或无法做出合理的选择，或需要消耗很大的精力才能做出正确选择。为了解决信息过载问题，减轻人们在做出抉择时所承受的沉重负担，信息分类、搜索引擎和推荐系统等技术应运而生，如表1.1所示。

表 1.1: 信息过载问题主流解决方案

名称	特点	案例
信息分类	分门别类地组织信息	Yahoo!、58同城
搜索引擎	根据关键字进行查询	Google、百度
推荐系统	分析用户行为历史，主动推荐	Amazon、淘宝

信息分类技术是通过将互联网上的各种信息分门别类地组织起来来提高人们查询效率的，比如早期的Yahoo!和国内的58同城等信息分类网站。然而，信息分类技术往往依赖于人工，可扩展性差。随着互联网上信息的增加，这显然不合时宜。搜索引擎技术通过建立索引的方式将互联网上的网页组织起来，然后根据用户输入的关键字进行查询，比如Google 和百度。搜索引擎相对于信息分类技术来说已经有了很大的进步，能够适应互联网的快速发展。然而，人们很多时候要么不愿意费时费力地去输入关键字，要么无法准确地用关键字去描述自己的想法，比如“今天看什么电影好呢?”、“去哪儿吃饭呢?”等，此时搜索引擎就无能为力了。为了解放人们的双手以及深度挖掘人们的内在偏好和需求，推荐系统技术应运而生。

推荐系统本质上是一种信息过滤系统，其通过对用户行为历史的分析挖掘出用户的行为偏好，进而帮助用户将海量信息中的无用信息过滤掉并将符合用户偏好的信息推荐给用户 [1]。目前，推荐系统已经在各个领域得到了广泛的应用，比如电子商务领域的亚马逊和淘宝，在线视频领域的Netflix和优酷，在线音乐领域的Lastfm [15] 和豆瓣电台 [6]以及个性化阅读领域Flipboard和无觅阅读等。图1.1展示了电子商务领域Amazon的商品推荐系统界面和视频领域PPTV的

在线视频推荐系统界面。



图 1.1: Amazon电商推荐及PPTV视频推荐

文献 [24]表明，人们在日常生活中的听歌行为要远远多于读书、看电影等行为，这说明音乐已经成为人们生活中不可缺少的一部分。音乐推荐系统正是推荐系统在音乐领域的应用，其旨在将人们从海量的音乐中解脱出来，通过分析用户的收听习惯以及歌曲本身的特征来为用户推荐符合用户需求的歌曲 [30]。与传统的书籍、电影等物品相比，音乐本身以及人们消费音乐的方式具有如下一些特点：

1. 歌曲往往时间比较短，而且大多是免费的，因此人们消费音乐作品的代价比较低，可以一次性收听一组歌曲。
2. 歌曲本身没有统一的划分标准，流派、心情、场合、年代等都可以作为歌曲的划分标准。比如，按照流派可以将歌曲分为流行、经典、摇滚等类别，按照心情可以分为欢快、伤感、忧伤等类别。
3. 即使是按照一个统一的标准对歌曲进行划分，一首歌曲往往并不是确定地

属于某一个类别，而是以不同的程度分属于多个类别。比如，按照流派进行划分，张雨生的经典歌曲“大海”既属于经典也属于流行。

4. 用户在同一个平台下收听歌曲时往往是按照次序一首接着一首收听的，这说明用户对音乐的消费与次序有很强的关联。最后，用户在不同的上下文环境下对音乐作品的喜好也有所不同。比如，用户在睡觉前可能倾向于收听舒缓的歌曲而在运动时倾向于收听快节奏歌曲。

为了满足人们对音乐的个性化需求，一些音乐推荐系统被设计和实现出来，如国外的Lastfm [15]、Pandora [21]以及国内的豆瓣电台 [6]、虾米音乐 [33]，图1.2展示了豆瓣电台和虾米音乐的推荐界面。这些音乐推荐系统首先建立自己的曲库，然后分析歌曲的特征和用户的听歌习惯继而为用户做出推荐。其中，Pandora [21] 是当今最流行的音乐推荐系统之一，其通过Music Genome Project将400种属性分配给每一首歌曲，进而按照歌曲的相似程度为用户做出推荐。从可见的资料看，Last.fm [15]、豆瓣 [6] 主要是根据“相似的人往往具有相似的行为”这样的假设为用户做出推荐。虾米音乐 [33] 还给出了推荐的同时还给出了做出推荐的原因，这在一定程度上提升了用户对推荐结果的接收程度。



图 1.2: 豆瓣及虾米音乐电台示意图

在学术界，学者们也给出了一些有效的音乐推荐算法也被提了出来。按照它们参考用户行为序列长度的长短，我们将这些算法分为全面考察用户所有收听行为的基于用户长期行为的音乐推荐 [25] [5]、仅考察用户当前收听行为的基于用户即时行为的音乐推荐 [7] [12] 以及考察用户在当前会话期内行为的基于用户中期行为的音乐推荐 [9] [19] 三类。其中，基于用户中期行为的音乐推荐由于考虑了上下文环境对用户行为的影响，因此得到了越来越多的关注。然而，以 [9] [19] 为代表的基于用户中期行为的音乐推荐算法还存在如下问题：

1. 只能够定性地分析出目标歌曲可能隶属的类别，但无法定量地给出属于该类别的程度，对用户行为的描述不够细致。
2. 只考虑了若干显著类别的影响，忽视了其他类别对用户行为的贡献和作用，对用户行为的描述不够全面。
3. 在一定程度上依赖于其他用户的行为，但很多时候其他用户的行为很难获取，导致对当前用户行为的预测效果比较差。

为了提高基于中期行为的音乐推荐算法的效果，本文首先给出了一致基于多维时间序列分析的个性化音乐推荐方法，该方法将用户行为建模为一个多维时间序列，从而实现对用户行为细致、全面地分析。

然而，用户的未来行为除了受到用户中期行为的影响之外，还受到用户长期行为和即时行为的作用，但是目前未见工作将这三方面的影响和作用综合起来进行考虑。因此，本文进一步给出了一种基于用户三期行为的综合音乐推荐方法，全面地考察用户长期行为、即时行为 and 中期行为的贡献和作用。在给出基于多维时间序列分析的音乐推荐方法和基于用户三期行为的综合推荐方法之后，本文通过实验验证了所给方法能够取得比参考方法更高的预测命中率以及更低的推荐误差。最后，本文基于上述方法实现了一个原型系统并且在该系统实现了本文方法的并行化，验证了所给方法的可行性。

本文结构按照如下方式进行组织：第2章介绍与本文工作相关的一些工作，包括常用的音乐推荐算法、常用的物品相似度计算方法、推荐系统的评测指标以及文本处理、时间序列分析的相关理论等；第3章描述了本文所提的一种基于多维时间序列分析的个性化音乐推荐方法，包括问题的定义、方法的执行流程等；第4章描述了本文在基于多维时间序列分析的音乐推荐方法的基础上进一步给出的基于用户三期行为的综合推荐方法；第5章给出了本文所实现的原型系统及相关实现技术；第6章对全文进行总结并给出对未来工作的展望。

第二章 相关工作

本章我们将介绍与本文工作相关的一些工作，首先我们介绍音乐推荐系统领域常用的一些推荐算法、物品相似度的常用计算方法以及一些常用的算法评估标准，然后我们介绍文本建模、时间序列分析以及实时流处理框架的相关工作，这些方法和技术将在本文后续工作中得到应用。

2.1 音乐推荐算法

目前，音乐推荐系统领域的研究工作已经取得了长足的进步，出现了很多类型的音乐推荐算法。要想为用户推荐合适的歌曲，很多音乐推荐方法都要对用户的听歌行为进行分析。按照算法参照用户行为序列长度的长短，我们将这些音乐推荐算法分为基于用户即时行为的音乐推荐、基于用户长期行为的音乐推荐和基于用户中期行为的音乐推荐三类。本节将对这些推荐算法进行简单的介绍。

2.1.1 基于用户及时行为的推荐

基于用户即时行为的音乐推荐是最为朴素的一种音乐推荐方法，此类音乐推荐算法认为用户的状态在短期内保持稳定，而用户可能收听的下一首歌曲与用户当前收听的歌曲具有类似的特征。假如用户正在收听老狼的《同桌的你》，此类算法首先通过一定的方式抽取歌曲对应的特征即构建歌曲特征向量，并在此基础上将与《同桌的你》具有相似特征的歌曲推荐给用户，比如老狼的另一首经典校园民谣《睡在我上铺的兄弟》，豆瓣电台的“听相似歌曲”实现的正是这样一个功能。因此，以何种方式抽取歌曲特征就成了此类音乐推荐算法工作的关键。音乐作品作为人类情感表达的强有力工具除了具有基本的声学特征之外，还具有流派、心情、场合等特征以及歌曲对应的文本特征，而从不同的角度刻画往往可以得到不同的结果。比如，按照流派划分歌曲可以有流行歌曲、经典歌曲；按照心情划分可以有寂寞、开心、欢快；按照场合划分可以有学习、休息；按照年代划分可以有70后、80后等。文献 [7] 充分利用歌曲的编辑属性，将与当前歌曲具有相同或相似曲作者、歌曲名称以及歌词的歌曲推荐给用户。文献 [3] [4] [17] 使用歌曲的声学特征为用户做出推荐，将与当前歌曲具有相似节奏、韵律、音色的歌曲推荐给用户。文献 [13] 使用心情特征对歌曲进行刻画并将与当前歌曲具有近似心情属性的歌曲推荐给用户。上述工作都是从

单一方面对歌曲进行刻画并进而为用户做出推荐，但这种推荐往往是片面的，比如文献 [3] [4] 对于对声学特征不敏感但对于情感特征敏感的用户就不适用。文献 [12] 对歌曲对应的文本文档进行分析，使用语义特征对歌曲进行刻画，取得了不错的效果。

基于用户即时行为的音乐推荐是一类比较朴素的推荐算法，其研究核心在于从何角度对歌曲进行刻画和描述并在抽取歌曲特征的基础上为用户推荐相似的歌曲。此类算法尽管也能够取得不错的推荐效果，但其问题也是显而易见的，即对用户行为特征的刻画过于简单，仅仅考察了用户当前行为而忽视了之前的行为。

2.1.2 基于用户长期行为的推荐

与基于用户即时行为的音乐推荐不同的是，基于用户长期行为的音乐推荐对用户所收听的所有歌曲都进行了考察和分析，常见的基于协同过滤的音乐推荐和基于全局特征的推荐都属于此类。

协同过滤推荐(Collabrative Filtering, CF)是当前最为流行的一类推荐算法，其挖掘用户所处的社会化环境，利用群体智能为用户做出推荐。协同过滤推荐基于这样一个假设，即如果两个用户在过去有相同的行为，那么系统认为他们在未来也会有类似的行为 [28] [16] [25]。当需要预测一个用户是否喜欢一个物品时，协同过滤推荐算法首先找到与当前用户喜好类似的用户，进而综合这些相似用户的喜好为用户推荐新的物品。假如用户A收听了《同桌的你》、《睡在我上铺的兄弟》、《朋友》，用户B收听了《同桌的你》、《朋友》、《白桦林》，用户C收听了《一无所有》、《北京北京》、《太阳》，那么此类算法认为用户A和用户B属于相似用户，进而可以将用户B收听但用户A收听的歌曲《白桦林》推荐给A。

基于全局特征的音乐推荐是基于用户长期行为推荐算法中比较朴素的一类推荐方法，此类算法认为用户的长期行为能够反映用户对音乐的偏好。因此，在抽取歌曲特征的基础上，此类算法将用户所收听的所有歌曲的平均特征作为用户的特征并推荐与此平均特征类似的歌曲给用户。文献 [5] 使用声学特征对歌曲进行刻画，然后计算用户所收听的所有歌曲的平均特征并以此作为用户特征。

由于基于用户长期行为的音乐推荐对用户的分析比较全面，因此此类算法往往能够得到相较于基于用户即时行为的音乐推荐能够得到比较不错的推荐效果。但是，此类算法对用户所处上下文的考虑不足，无法满足用户的即时需求。

2.1.3 基于用户中期行为的推荐

基于用户中期行为的音乐推荐是这样一类推荐算法，它们认为用户在当前会话期内所收听的歌曲序列能够在一定程度上反映用户所处的上下文环境。因此，它们通过对用户在当前会话期内所收听的歌曲序列进行分析来完成对用户特征的抽取和歌曲的推荐。文献 [22] 中给出了“当前会话期”的定义，即指用户正在收听歌曲的这一段连续时间。文献 [22] 使用最近收听的几首歌曲去匹配用户的收听历史从而找到类似的收听行为并依据类似行为作出推荐。尽管能够取得一定的效果，但用户所收听的历史不一，对于历史行为少的用户，文献 [22] 便无能为力。此外，文献 [22] 没有对歌曲空间进行降维，而是直接使用歌曲本身去匹配，这更降低了算法的可行性。文献 [9] [19] 是基于用户中期行为的两类典型代表，这两类算法首先使用文本分析的方法将歌曲表示成由若干隐含主题构成的概率分布并使用若干显著主题表征歌曲。这样，它们便将用户在当前会话期内的收听行为建模为主题序列，通过对该主题序列的分析即可预测目标歌曲可能隶属的主题，进一步地可以讲目标主题中的显著歌曲推荐给用户。其中，文献 [9] 使用以 PrefixSpan 为代表的模式挖掘算法对主题序列进行挖掘，而文献 [19] 是使用马尔可夫模型的方法对主题序列进行分析。文献 [9] [19] 能够取得一定地推荐效果，但是他们只是定性地预测目标歌曲可能隶属于哪个隐含主题，无法定量的给出具体的隶属度。此外，它们只是考察若干显著主题的作用和贡献而忽视了其他隐含主题的影响。再者，它们需要依赖其他用户的行为当前用户做出推荐。正是由于存在这些问题，使得文献 [9] [19] 的推荐效果收到很大的限制。

由于基于用户行为的音乐推荐考察了用户所处的上下文环境，因此目前越来越多的工作开始关注此类工作。为了解决文献 [9] [19] 中存在的问题，本文给出一种基于多维时间序列分析的音乐推荐方法，使用主题模型将歌曲表示成一个由若干隐含主题构成的概率分布，并在此基础上将用户在当前会话期的行为表示成一个多维时间序列。通过对该多维时间序列的分析，预测得出用户可能收听的下一首歌曲的特征并从曲库中选择类似的歌曲推荐给用户。

用户的未来行为除了受到用户中期行为的作用之外还受到用户即时行为和长期行为的影响，但目前没有看到比较直观地工作将这三者进行综合考虑。为此，本文在给出一种基于多维时间序列分析的音乐推荐方法的基础上，进一步给出一种基于用户三期行为的综合推荐方法，综合考虑用户三期行为的贡献和作用。

2.2 相似度度量

由前文的介绍可知，几乎所有的音乐推荐方法都会涉及到用户之间相似度的计算或者歌曲之间相似度的计算，而本文工作主要涉及歌曲之间相似度的计算，因此本节将简单介绍几种歌曲相似度的计算方法。

2.2.1 余弦相似度

余弦相似度是推荐系统中计算用户与用户或者物品与物品之间相似度的一种常用方法，其通过测量两个向量内积空间的夹角的余弦值来度量它们之间的相似性。0度角对应的余弦值是1，而其他任何角度对应的余弦值都不大于1，并且其最小值是-1。如果两个向量的指向越接近，那么它们内积空间夹角的余弦值越接近于1，即二者越相似。相反，如果两个向量的指向越相离，那么它们内积空间夹角的余弦值越接近于-1，即二者越不相似。设向量 \vec{X}_a 和向量 \vec{X}_b 分别表示用户 a 和用户 b 的偏好向量，其中 $\vec{X}_a(i)$ 表示用户 a 对编号为 i 的物品的偏好值或评分， $\vec{X}_b(i)$ 表示用户 b 对编号为 i 的物品的偏好值或评分。那么，用户 a 和用户 b 的相似度可以按照公式2.1进行计算，其中 K 表示物品集合中物品的数目。

$$\text{sim}(a, b) = \cos(\vec{X}_a, \vec{X}_b) = \frac{\sum_{i=1}^K \vec{X}_a(i) \vec{X}_b(i)}{\sqrt{\sum_{i=1}^K \vec{X}_b(i)^2} \sqrt{\sum_{i=1}^K \vec{X}_a(i)^2}}$$

2.2.2 KL距离

KL距离也称相对熵或KL散度，是两个概率分布 P 和 Q 差别的非对称性的度量，用来度量使用基于 Q 的编码来编码来自 P 的样本平均所需的额外的比特个数 [14]。如果能够把物品表示成一个概率分布，那么显然可以用KL距离来衡量两个物品之间的相似度。设 $P = (p_1, \dots, p_i, \dots, p_K)$ 表示物品 p 对应的概率分布，物品 p 对应的概率分布用 $Q = (q_1, \dots, q_i, \dots, q_K)$ 表示，其中 K 表示随机变量的取值数目。那么，物品 p 和 q 之间的KL距离可以按照公式2.2进行计算，其取值非负。KL散度仅当概率 P 和 Q 各自总和均为1，且对于任何 i 皆满足 $p_i > 0$ 及 $q_i > 0$ 时才有定义，若式中出现的情况，其值按0处理。

$$\text{dis}_{KL}(p, q) = \sum_{i=1}^K \ln \frac{p_i}{q_i}$$

由于KL距离是非对称的，即

$$\text{dis}_{KL}(p, q) \neq \text{dis}_{KL}(q, p)$$

而物品之间的相似度应该满足对称性，即

$$\text{sim}(p, q) = \text{sim}(q, p)$$

因此，不能直接使用KL距离来计算物品之间的相似度，但可以使用 p 到 q 的KL距离与 q 到 p 的KL距离的平均值来作为二者的最终距离，这样就可以满足对称性的

要求，如公式2.5 所示。进一步地，可以使用如公式2.6所示的方法计算二者之间的相似度。

$$dis_{avgKL}(p, q) = \frac{dis_{KL}(p, q) + dis_{KL}(q, p)}{2}$$

$$sim_{KL}(p, q) = \frac{1}{1 + dis_{avgKL}(p, q)}$$

2.2.3 Hellinger距离

Hellinger距离也是一种度量两个概率分布之间相似度的方法。与KL距离不同的是，其天然满足对称性，因此可以直接用来计算物品之间的相似度 [20]。物品p和物品q之间的Hellinger 距离可以按照公式2.7计算，进而按公式2.8可以计算二者之间的相似度。

$$dis_{hellinger}(p, q) = \frac{1}{\sqrt{2}} \sqrt{\sum_{i=1}^K \sqrt{p_i} - \sqrt{q_i}}^2$$

$$sim_{hellinger}(p, q) = \frac{1}{1 + dis_{hellinger}(p, q)}$$

本文所提的基于多维时间序列分析的音乐推荐方法及在此基础上给出的基于用户三期行为的综合推荐方法将使用主题模型建模的方法将每一首歌曲表征为由若干隐含主题构成的概率分布。因此，KL 距离和Hellinger 距离将被应用于本文计算两首歌曲之间的相似度的工作中。

2.3 评测指标

推荐系统所研究的问题主要包括评分预测问题、Top-N推荐问题、冷启动问题、可解释性问题以及用户交互问题等 [1]。其中，评分预测问题和Top-N推荐问题是得到最广泛研究且最为重要的内容。所谓评分预测问题即根据用户已经产生的评分记录来预测其对尚未评分物品的可能打分，而Top-N推荐是指为用户生成一个包含N个符合其偏好的物品列表。围绕着这两类问题，研究人员给出了众多推荐算法，本文将简单介绍评测推荐算法优劣的一些指标。

2.3.1 用户满意度

用户作为推荐系统的重要参与者也是推荐系统最终的服务对象，其满意度是评测一个推荐系统优劣的最重要指标。一般来说，用户满意度可以通过对一些用户行为的统计得到。比如，在电子商务网站中可以通过用户的实际购买情况来评判，或者通过设置“满意”/“不满意”按钮进行显示的统计。更一般的情况是，可以使用点击率、用户停留时间和转化率等指标度量用户的满意度。对于本文所研究的音乐推荐系统，可以统计用户在系统上的“喜欢”/“不喜欢”或者“收听”/“放弃”等行为来评估不同算法的推荐效果。显然，用户满

意度作为真实场景中的评测指标往往只能通过用户调查或者在线实验的方法获得而无法实现离线计算，这增加了研究人员评估算法优劣的难度。因此，在实际的研究工作中很少通过这个指标来评估不同的推荐算法。

2.3.2 预测准确度

预测准确度是度量一个推荐系统或者推荐算法预测用户行为能力的重要指标。从推荐系统诞生的那一天起，几乎99%与推荐系统相关的论文都在讨论这个指标，这主要是因为该项指标可以通过离线实验计算，方便了很多学术界的研究人员研究推荐算法。对于评分预测问题，一般使用均方根误差(Root Mean Square Error, RMSE)和平均绝对误差(Mean Absolute Error, MAE)来表征算法的预测准确度 [10]。RMSE可由公式2.9计算得到。

$$RMSE = \sqrt{\frac{\sum_{u,i \in T} (r_{ui} - \hat{r}_{ui})^2}{|T|}}$$

其中， T 代表测试集， u 和 i 表示测试集中的用户和物品，是用户 u 对物品 i 的实际评分，而是推荐算法给出的预测评分。

MAE采用绝对值计算预测误差，如公式2.10所示。

$$MAE = \frac{\sum_{u,i \in T} |r_{ui} - \hat{r}_{ui}|}{|T|}$$

显然，预测误差越小，那么预测准确度越高，那么推荐算法的效果越好。也就是说，一个优秀的推荐算法在RMSE和MAE这两项指标上的取值往往比较低。

Top-N推荐一般通过准确率(precision)和召回率(recall)来度量算法的优劣。设 $R(u)$ 是根据用户在训练集上的行为给用户作出的推荐列表，而 $T(u)$ 是用户在测试集上的行为列表 [23]。那么，推荐结果的准确度和召回率如公式2.11和2.12所示。

$$Precision = \frac{\sum_{u \in U} |R(u) \cap T(u)|}{\sum_{u \in U} |R(u)|}$$

$$Recall = \frac{\sum_{u \in U} |R(u) \cap T(u)|}{\sum_{u \in U} |T(u)|}$$

2.3.3 其他评测指标

除了预测准确度这一重要指标之外，推荐系统的评测还有诸如覆盖率、多样性等指标，这些指标从不同的角度看待推荐的有效性。其中，覆盖率(Coverage)描述一个推荐系统对物品长尾的发掘能力，可以简单地定义为推荐系统能够推荐出来的物品占总物品集合的比例。覆盖率越大，说明系统所能推荐的物品越广泛，也说明系统挖掘物品长尾的能力越大。多样性(Diversity)用来

描述推荐列表中物品两两之间的不相似性，列表中物品两两之间的相似性越小表示推荐的多样性越大。

考虑到用户满意度无法进行离线实验以及覆盖率、多样性的对推荐算法的表征能力不佳，本文后面主要使用预测准确度来评估本文所提的推荐方法及框架，即评估本文所提推荐方法和框架是否能够得到比其他算法更高的准确度和召回率以及更低的预测误差。

2.4 分布式实时计算系统

尽管目前单机的处理能力已经得到极大的提升，但其在应对大数据时代产生的海量数据时仍然非常吃力。为了解决大数据时代海量数据的处理和分析的问题，Google提出了一种分布式的计算模型，即MapReduce，该模型使得由一般能力机器组成的集群可以完成大规模或者超大规模的计算工作。在Google工作的启发下，Apache于2005年开发了目前得到广泛应用的分布式应计算框架Hadoop。Hadoop [8]对于批处理的工作以及离线的大量数据分析比较有优势，但其对一些在实时性方面要求比较高的计算任务的处理能力有所欠缺。为了弥补Hadoop的这一缺憾，以Storm [31]为代表的一些分布式实时计算系统被开发了出来，这些系统及框架在实时数据流分析方面能够取得比Hadoop更好的效率和效果。这里，Storm是由Twitter开发的一款开源的分布式实时计算框架，其主要适用于流数据处理和分布式远程过程调用两种场景。对于流数据处理场景，Storm可以用来处理源源不断流进来的消息，处理之后将结果写入到某个存储中去。此外，Storm的处理组件是分布式的且处理延迟极低，使得其在分布式远程过程调用的场景中也能够得到比较充分的应用。本文所要解决的问题恰恰是一个实时数据流的分析问题，因此本文后面将会选用Storm进行数据的分析和处理，本节将对其基本组成及其在分布式远程过程调用中的应用进行介绍。

2.4.1 基本组成

一个传统的Storm集群往往是由一个主控节点(Master Node)和多个工作节点(Work Nodes)组成。其中，主控节点上运行着一个名为“Nimbus”的守护进程，用于分配代码、布置任务及故障检测，而每个工作节点都运行一个名为“Supervisor”的守护进程，用于监听工作、开始及终止工作进程。Nimbus和Supervisor都能快速失败，而且是无状态的，这样一来它们就变得十分健壮，而两者的协调工作是由Apache ZooKeeper来完成的。

在Storm中，一个实时应用的计算任务被打包成一个Topology任务发布，且Topology任务一旦提交后永远不会结束，除非用户显式地去停止任务。这里，计算任务Topology是由多个Spout和Bolt计算组件构成，而这些计算组件之间是通过数据流连接起来的。其中，Spout是Storm中的消息源，用于为Topology生产消息（数据），一般是从外部数据源（如Message Queue、RDBMS、NoSQL、Realtime Log）不间断地读取数据并发送给Topology；Bolt是Storm中的消息处理者，用于为Topology进行消息的处理，Bolt可以执行过滤，聚合，查询数据库等操作且可以一级一级的进行处理。Topology中每一个计算组件（Spout和Bolt）都有一个并行执行度，在创建Topology时可以进行指定，Storm会在集群内分配对应并行度个数的线程来同时执行这一组件。图2.5是Twitter Storm官方给出的一个典型的Topology示意图，其中水龙头表示用以生产数据的Spout组件，闪电表示用户处理数据Bolt组件，消息或数据由Spout组件产生后便在不同的Bolt组件中进行流动并被处理。

除此之外，Storm还有Stream、Stream Grouping、Task、Worker等关键概念。其中，Stream表示被处理的数据；Stream Grouping表示Bolt接收什么样的数据作为输入数据，即上游组件以何种方式将消息或数据分组并传给下游组件，一般有广播(All Grouping)、随机分组(Shuffle Grouping)、按字段分组(Field Grouping)、直接分组(Direct Grouping)以及不分组(No Grouping)等分组方式；Task表示运行于Spout和Bolt中的线程，而Worker表示运行这些线程的进程。在具体实现时，用户只需要在Spout组件中实现数据读取及分割的逻辑，在Bolt组件中实现数据的处理逻辑，同时在各组件中指定数据流动的分组方式即可轻松地完成并发布一个实时计算的任务。

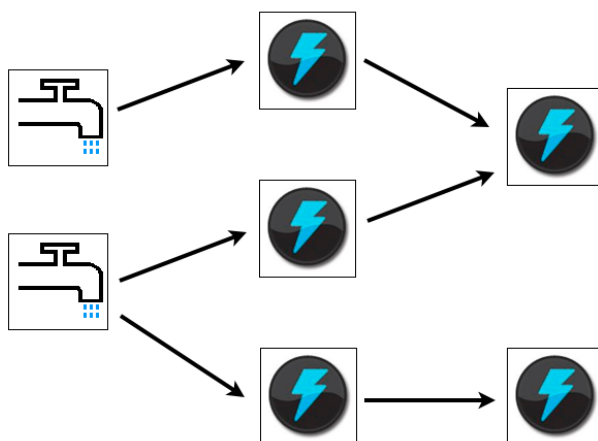


图 2.1: 典型Twitter Storm示意图

2.4.2 分布式过程调用

分布式远程过程调用（Distributed Remote Procedure Call, DRPC）是联系客户端与Storm集群的一种机制，Storm中引入这一机制的主要目的是利用Storm的实时计算能力来并行化CPU密集型计算。Storm集群上运行的拓扑接收调用函数的参数信息作为输入流，并将计算结果作为输出流发射出去。其中，DRPC通过是DRPC Server实现的，其整体工作过程如下：

1. 接收到一个RPC调用请求；
2. 发送请求到Storm上的拓扑；
3. 从Storm上接收计算结果；
4. 将计算结果返回给客户端。

图2.6更为细致地描述了DRPC的工作流程，大致可以分为如下五个步骤：

1. Client向DRPC Server发送被调用执行的DRPC函数名称及参数；
2. Storm上的Topology通过DRPCSpout实现这一函数，从DRPC Server接收到函数调用流；
3. DRPC Server会为每次函数调用生成唯一的id；
4. Storm上运行的Topology开始计算结果，最后通过一个ReturnResults的Bolt连接到DRPC Server，发送指定id的计算结果；
5. DRPC Server通过使用之前为每个函数调用生成的id，将结果关联到对应的发起调用的Client，将计算结果返回给Client。

2.5 文本建模

为了对歌曲进行全面且完整的描述和刻画，本文将使用文本分析的方法对表征歌曲的文本信息进行分析，本节简单介绍一些基本的文本建模方法。

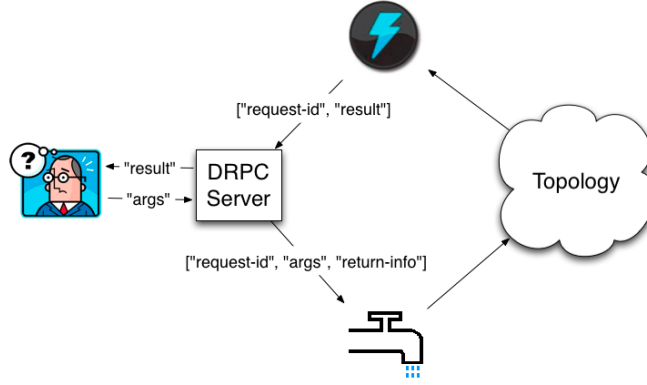


图 2.2: DRPC工作流程示意图

2.5.1 向量空间模型

计算机不具备人脑的结构，无法理解自然语言，所以需要首先将无结构的自然语言文本转化为计算机可计算的特征文本。为此，Salton等人在20世纪70年代提出了向量空间模型(Vector Space Model, VSM) [27]。向量空间模型首先将每一个文档看做一个词袋(Bag of Words)，即认为一篇文档是由一组词构成的一个集合且词与词之间没有顺序以及先后的关系。其次，向量空间模型将文档表示成一个向量，向量的每一维表示一个词项，而向量每一维的取值表示该词项在文档中的权重。对于文档集合 D 中编号为 j 的文档 d_j ，可以将之表示成一个 t 维的向量 $\vec{d}_j = (\omega_{1,j}, \dots, \omega_{t,j})$ ，其中 t 表示词项的数目， $\omega_{i,j} (1 \leq i \leq t)$ 表示第 i 个词项在文档 d_j 中的权重。在对文本进行建模的过程中，词的选取及权重的计算有以下几种典型方式：

1. 布尔模型。这是最为简单直观的一种计算词项权重的方法，即将词项在文档中是否出现作为其权重，如果词项在文档中出现那么将其权重记为1，否则记为0。虽然这种方法比较简单，但是它没有体现词语在文档中出现的频率。一般来讲，词语在文档中出现的越多，说明它对该篇文档的重要性越大（“的”、“得”、“地”、“是”等停用词除外）。
2. 词频模型。与布尔模型不同的是，词频(Term Frequency, TF)模型统计词项在文当中出现的次数，然后得到词项的频率，并将之作为词项的权重。词项 t_i 相对于文档 d_j 的词频表示成：

$$tf_{i,j} = \frac{n_{i,j}}{\sum_k n_{k,j}}$$

这里, $n_{i,j}$ 表示该词项在该文档中出现的次数。这突出了词频对词项重要性

的影响，能够取得比布尔模型较好的效果。但是，词语的重要性不仅随着它在文档中出现的次数成正比增加，而且可能会随着它在语料库中出现的频率成反比下降。也就是说，一个词语在整个语料库中出现得越频繁，则它对于文档的重要性越低，对文档的区分度量越差。

3. 词频-逆向文本频率模型。词频-逆向文本频率(Term Frequency - Inverse Document Frequency, TF-IDF)是对上述TF模型的补充，词项的重要性随着其在特定文档中出现次数的增加而增强，但同时随着其在全体文本中出现次数的增加而减弱，即该模型认为对区别文档最有意义的词语应该是那些在文档中出现频率高、而在整个语料库中的其他文档中出现频率少的词语。词项 t_i 相对于文档 d_j 的TF-IDF取值表示为：

$$tfidf_{i,j} = tf_{i,j} \times idf_i$$

这里， idf_i 为逆向文本频率，定义为：

$$idf_i = \log \frac{|D|}{|\{j : t_i \in d_j\}|}$$

其中， $|D|$ 表示文档集合中的文件总数， $|\{j : t_i \in d_j\}|$ 表示文档集合中包含词项 t_i 的文件数目。TF-IDF结构简单，容易理解，被广泛应用。但是，其无法准确捕捉文档内部与文档间的统计特征，也不能解决同义词和多义词的问题，因此精确度不是很高

2.5.2 隐含狄利克雷模型

为了解决同义词和多义词的问题，Blei等人于2003年提出了隐含狄利克雷分配模型(Latent Dirichlet Allocation, LDA) [2] [11]。LDA也是一种典型的词袋模型，其认为一篇文档由多个主题构成，而文档中每一个词都是由对应的主题生成而来。进一步地，LDA被看做是一种主题模型，它将文档看做是一组隐含主题的概率分布。其中，主题表示一个概念、一个方面，表现为一系列相关的单词，是这些单词的条件概率。形象来说，主题就是一个桶，里面装了出现概率较高的单词而这些单词与这个主题有很强的相关性。这样，LDA模型便通过隐含主题将文本与词项联系起来，从而达到降维的目的。LDA是一种生成模型，一篇文档按照如下所示的规则生成：

1. 假设有两种类型的桶，一种是文档-主题桶，桶里的每一个球代表一个主题；另一种桶是主题-词汇桶，桶中的每一个球代表一个词汇。
2. 文档的生成过程就是不断从桶中取球的过程，每一次先从文档-主题桶中取出球，得到该球代表的主题编号 z 。

3. 从编号为 z 的主题-词汇桶中取球，得到一个词汇。
4. 不断重复2,3两步，即可生成一篇文档。

在LDA模型中，记文档-主题的概率分布为多项式分布 $\vec{\theta}$ ，主题-词汇的概率分布为多项式分布 $\vec{\varphi}$ 。LDA模型认为 $\vec{\theta}$ 和 $\vec{\varphi}$ 是模型中的参数，而考虑到和都是多项式分布，模型选择狄利克雷(Dirichlet)分布作为其先验分布。在确定了这些分布之后，LDA下面需要做的就是估计这些分布的参数，如下所示的吉布斯取样(Gibbs Sampling)是目前比较流行的采样方法：

Algorithm 1 LDA模型的Gibbs采样算法

- 1: 首先对所有文档中的所有词遍历一遍，为其都随机分配一个主题，即 $z_{m,n} = k \sim Mult(1/K)$ ，其中 m 表示第 m 篇文档， n 表示文档中的第 n 个词， k 表示主题， K 表示主题的总数，之后将 $n_{m,k}^k$ 、 n_m 、 $n_{k,t}^t$ 、 n_k 都加1，它们分别表示在第 m 篇文档中主题 k 出现的次数、第 m 篇文档中主题数量的和、主题 k 对应的词 t 的次数， k 主题对应的总词数。
- 2: 对第1篇文档中的所有词进行遍历，假如当前文档中的词 t 对应主题为 k ，则将 $n_{m,k}^k$ 、 n_m 、 $n_{k,t}^t$ 、 n_k 都减1，即先拿出当前词，之后根据LDA中Topic Sample的概率分布取样出新的主题，再将 $n_{m,k}^k$ 、 n_m 、 $n_{k,t}^t$ 、 n_k 都加1。其中， α 和 β 为对应的Dirichlet分布的参数， V 为词汇总数。

$$p(z_i = k | z_{-i}, w) \propto \frac{(n_{k,-i}^t + \beta_t)(n_{m,-i}^k + \alpha_k)}{\sum_{t=1}^V (n_{k,-i}^t + \beta_t)}$$

- 3: 重复步骤2直至遍历所有文档。
- 4: 输出LDA模型中的参数 $\vec{\theta}$ 和 $\vec{\varphi}$ 。

$$\theta_{m,k} = \frac{n_{m,k}^k + \alpha_k}{n_m + \alpha_k}$$

$$\varphi_{k,t} = \frac{n_{k,t}^t + \beta_t}{n_k + \beta_t}$$

在本文后续实验中，我们将分别以TF-IDF为代表的向量空间模型和以LDA为代表的主题模型对歌曲对应的文档进行建模，发现LDA能够获得较高的推荐准确率，因此我们将采用LDA作为我们主要的文本建模方法。

2.6 时间序列预测

歌曲时间短、消费代价低的特点决定了其能够较容易的形成序列，且是这种序列有严格的时间顺序，本文工作将通过对用户在会话期内所收听歌曲形成的时间序列的分析来预测用户的接下来的行为。因此，本节将简单介绍一些常用的时间序列预测模型。

2.6.1 简单平均法

简单平均法是以观察期内时间序列的各期数据（观察变量）的平均数作为下期预测值，按照采用的平均方法又可以分为算术平均法、加权平均法和几何平均法三类。其中，算术平均法以观察变量的算术平均数作为下期预测值，加权平均法以观察变量的加权算术平均数作为下期的预测值，而几何平均法是以观察变量的几何平均数作为下期的预测值。简单平均法比较简单、直观，但其预测误差一般偏高。

2.6.2 指数平滑法

指数平滑法是由移动平均法改进而来的，是一种特殊的加权移动平均法。这种方法既有移动平均法的长处，又可以减少历史数据的数量。首先，它把过去的的数据全部加以利用。其次，它利用平滑系数加以区分，使得近期数据比远期数据对预测值影响更大。它特别适合用于观察值有长期趋势和季节变动，必须经常预测的情况。按照平滑的次数可以分为一次指数平滑法和多次指数平滑法。其中，一次平滑法是计算时间序列的一次指数平滑值，以当前观察期的一次指数平滑值为基础，确定下期预测值。与二次移动平均法类似，二次指数平滑法就是对时间序列的一次指数平滑值再次进行指数平滑。

2.6.3 差分整合移动平均自回归模型

差分整合移动平均自回归模型(Autoregressive Integrated Moving Average model, ARIMA)又称为Box-Jenkins方法，是一种由Box和Jenkins于1970年提出的一种时间序列预测方法，目前已经得到广泛的应用 [18]。在模型ARIMA(p,d,q)中，“AR”表示自回归模型，p为自回归阶数，“MA”表示滑动平均模型，q表示滑动平均的阶数，d表示将时间序列转化为平稳时间序列所作的差分次数，即差分阶数。显然，ARIMA首先需要进行d次差分从而将非平稳序列转化为平稳序列，然后利用自回归模型和滑动平均模型对转换后的平稳序列进行预测。进一步的，模型ARIMA(p,d,q)可以表示成如下三个式子。

$$\Phi(B)(1-B)^d y_t = \delta + \Theta(B)\varepsilon_t$$

$$\Phi(B) = 1 - \sum_{i=1}^p \varphi_i B^i$$

$$\Theta(B) = 1 + \sum_{i=1}^q \theta_i B^i$$

其中,为时间序列Y的第t个取值,B是滞后算子, 和为模型参数,它们可以通过最小二乘等方法获得。

ARIMA的执行流程主要分为模型识别(Model Identification)、参数估计(Parameter Estimation)和诊断检测(Diagnostic Checking)三个阶段。其中,模型识别阶段主要完成检测序列是否平稳的工作。如果序列不平稳,那么则通过差分的方法将序列转化为平稳序列并给出差分阶数 d 。在此基础上,识别序列适用的可能模型,如自回归模型或滑动平均模型或者二者的混合。而参数估计阶段主要完成模型参数的估计工作,即通过最小二乘法估计参数和。诊断检测阶段用以检测所给出的模型及参数是否符合条件,如果符合则选用此模型进行预测,否则重新识别模型。

ARIMA模型将应用在本文后续的工作中,已完成对用户行为序列的分析和预测。后续实验结果显示,使用ARIMA模型进行时间序列预测,能够得到较好的效果。

第三章 一种基于多维时间序列分析的音乐推荐方法

本章将给出一种基于多维时间序列分析的音乐推荐方法，该方法实现了对用户听歌行为定量且全面地分析和预测，同时减少了对其他用户行为的依赖，使得推荐的结果更加符合用户偏好。首先，该方法使用用户给歌曲所打的标签构造歌曲对应的文档，进而通过主题模型的方法对文档集合进行建模并最终把每一首歌曲都表示成由若干隐含主题构成的一个概率分布。在此基础上，该方法将用户在当前会话期内的听歌行为建模为一个多维时间序列，并通过对该多维时间序列分析来预测用户行为，而这样做的原因是该方法认为用户在当前会话期内所收听的歌曲序列能够在一定程度上表征用户所处上下文环境。

我们将在本章中对该方法进行详细的阐述。首先，我们将介绍本文工作所研究问题的相关描述和定义。然后，我们给出该方法的工作流程，包括隐含主题的抽取、多维时间序列的构造及分析预测、最终推荐结果的生成等内容。最后，我们介绍为验证该方法有效性而设计的实验及实验结果。

3.1 问题描述

音乐推荐的目的在于通过对音乐本身属性和用户行为习惯的分析，帮助用户过滤掉不必要的信息，并最终为用户推荐符合其喜好的音乐作品。换句话说来说，就是解决如何在已知歌曲特征以及用户之前所收听歌曲的情况下准确地预测用户可能收听的下一首歌曲的问题 [22]，如图3.1所示。

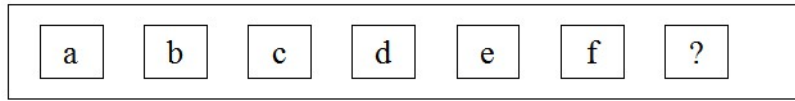


图 3.1: 下一首歌预测问题的说明图

为了更好地描述本文所要研究的问题，我们首先定义用户集和歌曲集，如下所示：

用户集 U ：所有用户的集合，如下式所示。其中， v 表示用户的数目，即 $v = |U|$ 。

$$U = \{u_1, u_2, \dots, u_v\}$$

歌曲集 S : 所有可以推荐给用户的物品(这里就是歌曲, 本文不加区分地使用“歌曲”和“物品”)的集合, 如公式3.2所示。其中, m 表示曲库中歌曲的数目, 即 $m = |S|$ 。为方便起见, 我们认为用户收听的歌曲一定在“可推荐歌曲”中。

$$S = \{s_1, s_2, \dots, s_m\}$$

对于用户集 U 中给定的一个用户 u , 本文所要研究的目标就是为该用户推荐其可能喜欢的下一首歌曲。为了衡量用户对歌曲的喜欢程度, 我们定义如下所示的效用函数:

效用函数 $u(c, s)$: 表征歌曲 s 对用户 u 的推荐度(如歌曲 s 符合用户 u 喜好的程度、歌曲 s 与系统所预测歌曲的相似度等)。效用函数反映了用户对某首歌曲的喜爱程度, 其值越大表明喜欢程度越大。

如前所述, 本文所给的方法是建立在隐含主题分类和用户在一定会话周期内听歌序列的基础上, 我们下面进一步地定义歌曲对应的隐含主题集合以及用户所收听歌曲对应的序列。

主题集 T : 由所有隐含主题组成的集合, 如式公式3.3所示, 其中 K 为隐含主题的数目。

$$T = \{t_1, t_2, \dots, t_K\}$$

事件 $e(u, \tau, s)$: 表示用户 u 在时刻 τ 收听了歌曲 s 。显然, 歌曲 s 可由用户 u 和时刻 τ 唯一决定, 因此 $e(u, \tau, s)$ 可简化为 $e(u, \tau)$ 。

序列 $Q(u)$: 用户 u 在给定平台下的所有听歌事件按照时间顺序排列开来得到的序列, 公式3.4所示。

$$Q(u) = \langle e(u, \tau_1), e(u, \tau_2), \dots, e(u, \tau_\pi), e(u, \tau_{\pi+1}), \dots, e(u, \tau_{\pi+n}) \rangle$$

其中, $\pi + n$ 为用户 u 所收听或喜欢的歌曲数目, τ 为事件发生的时间, 且 $\tau_1 < \tau_2 < \dots < \tau_\pi < \tau_{\pi+1} < \dots < \tau_{\pi+n}$, 即用户收听这 $\pi + n$ 首歌曲是按照一定的次序一首接一首收听的。另外, $Q(u)$ 中最后 n 个事件是用户在当前会话期内产生的, 即从事件 $e(u, \tau_{\pi+1})$ 到 $e(u, \tau_{\pi+n})$ 对应的 n 首歌曲是用户当前连续收听的, 而序列中的最后一个事件是正在发生的。也就是说, 这 n 个听歌事件两两之间没有明显的时间间隔, 而事件 $e(u, \tau_{\pi+1})$ 和事件 $e(u, \tau_\pi)$ 之间是由明显的时间间隔的, 如下所示。

$$|\tau_{\pi+i} - \tau_{\pi+i-1}| \leq \varepsilon (1 < i \leq n)$$

$$|\tau_{\pi+1} - \tau_\pi| > \varepsilon$$

其中 ε 是给定的时间间隔, 用以分割不同的听歌会话, 在后文的实验中我们将之设为8分钟, 而在系统实现中将之设为2小时。为了方便, 我们只考察在给

定平台下确实有听歌行为的用户，即用户在给定平台下确实有听歌事件且在当前正在该平台下收听歌曲，即当前会话期内的收听事件也是存在。而对于在给定平台下没有任何进行任何听歌行为以及当前不在该平台下听歌的用户，为他们进行推荐比较困难，这是整个推荐系统领域需要重点研究的冷启动问题，本文对此不做深究。

目标歌曲: 符合用户喜好或者用户接下来可能收听的歌曲。

由之前的分析可知，本文所提方法的本质输入是歌曲对应的隐含主题 T 和用户 u 当前会话下的收听序列 $Q(u)[\pi + 1 : \pi + n]$ ，而最终目标是为用户推荐其最可能收听的下一首歌曲。也就是说，在已知主题集 T 和用户 u 当前会话期收听序列 $Q(u)[\pi + 1 : \pi + n]$ 的情况下从歌曲集 S 中找出那些对用户的效用函数取值最大的目标歌曲 $N(u)$ 并推荐给用户，如公式3.6所示。

$$N(u) = \arg \min_{s \in S} u(u, s)$$

为了解决预测用户可能收听的下一首歌曲的问题，有两个难点需要解决：

1. 如何全面完整地歌曲进行刻画。
2. 如何对用户的行为序列进行分析以预测用户行为。

3.2 方法框架

针对上文所定义的音乐推荐问题，本文提出一种基于多维时间序列分析的音乐推荐方法，该方法按如图3.2所示的流程工作。其中，虚线箭头表示离线处理模块，主要用来对歌曲进行刻画，即解决难点1；实线箭头表示在线处理模块，主要用来对用户的行为序列进行分析进而预测用户未来的可能行为，即解决难点2。

在离线模块中，我们首先定期从音乐网站上抓取以用户标签为主的歌曲文本信息，然后将这些标签信息按照一定的方式组合成文档，那么这些文档构成的文档集合将和歌曲集 S 一一对应。关于文档的构造方式，本文采用的方法是被标记到目标歌曲的标签按照其被标记的次数在目标歌曲对应的文档中重复若干次。假设目标歌曲 s 所具有的标签信息可以表示为一个标签集合，比如 $TagSet(s) = \langle "rock", 5 \rangle, \langle "pop", 3 \rangle, \langle "male", 2 \rangle$ ，那么其对应的文档可以表示为 $Doc(s) = "rock rock rock rock rock pop pop pop male male"$ 。其中，标签“rock”在 $Doc(s)$ 中出现5次，标签“pop”在 $Doc(s)$ 中出现3次，标签“male”在文档 $Doc(s)$ 中出现2次。由于主题模型是将文档看做一个词袋模型，因此“rock”、“pop”、“male”在文档中出现的顺序不影响建模结果。在获得歌

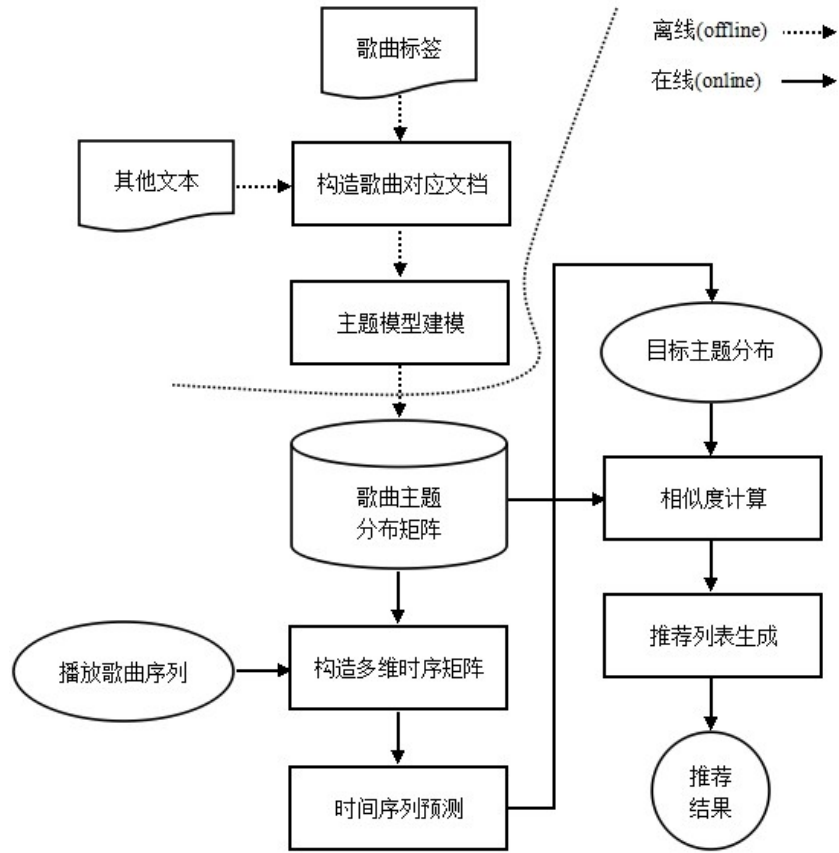


图 3.2: 基于多维时序分析的音乐推荐方法的工作流程示意图

曲集对于的文档集之后，我们对该文档集进行主题模型建模，从而抽取出歌曲中包含的隐含主题。这样，每一首歌曲可以用一个主题权重向量表示，向量中的每一维取值表征该主题对歌曲内容的贡献程度。

在在线处理模块中，我们首先获取用户在当前会话期内所听的歌曲，然后将这些歌曲按照时间先后顺序排开得到用户的当前会话序列。在此基础上并结合离线模块的处理结果，我们将用户的当前会话序列建模为一个多维时间序列。进一步地，我们使用一些经典有效的时序预测方法对该多维时间序列的每一个分量进行分析并预测单变量时间序列的下期取值。然后，我们将这些单变量时间序列的预测值组合起来构成目标歌曲的主题向量。得到目标概率分布之后，我们计算曲库中所有歌曲与目标歌曲之间的相似度并按照相似度由大到小排列。最后，我们选取排位靠前的 N 首歌曲作为推荐列表推荐给用户。在下面的章节中,我们将对方法工作流程中的主要步骤进行一一介绍，包括如何进行主题模型建模、为何选择多维时间序列作为用户行为序列的模型以及相似度

计算、最终推荐列表生成等。

3.3 主题模型建模

如前所述,为了更好地刻画歌曲的特征,本文在将歌曲映射成一个个文档的基础上通过主题模型建模来对歌曲进行描述和刻画。隐含狄利克雷分配(Latent Dirichlet Allocation,LDA)模型 [2]是当前最具代表性也是最流行的概率主题模型,已经在文本挖掘、信息处理、多文档摘要等领域得到了广泛的应用 [26] [32],其能够将文档映射到若干隐含主题的分布,而这个分布可以用一个主题权重向量表示,向量的每一维表征了该维主题对文档内容的贡献程度。通过LDA 主题模型建模,我们不但可以抽象出文档包含的隐含主题集合 T ,而且能够以量化的方式表达不同文档之间的距离和相似度。本节详细介绍利用LDA主题模型对歌曲建模的过程。

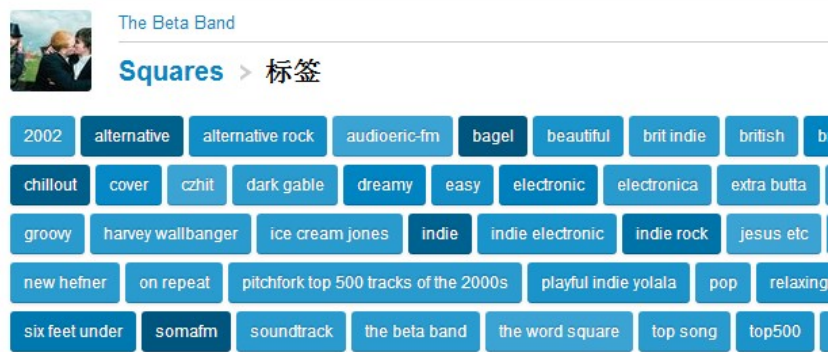


图 3.3: Last.fm中歌曲Squares的标签云

首先,我们从Lastfm(<http://last.fm>)、豆瓣(www.douban.com)等音乐网站上抓取用户对歌曲所标注的标签,这些标签对歌曲内容的描述比较全面,既包含了歌曲的名称、曲作者信息、专辑信息、发行年代等基本信息,还包含了歌曲所表达主题、歌曲类型、用户心情、适合场合等扩展信息。以The Beta Band的Squares为例,用户为其标注的标签有表征年代的“2002”/“2000s”、表征类型的“indie rock”/“folk pop”、表征用户感受的“beautiful”/“want”等,如图3所示。在得到歌曲对应的标签信息后,我们将这些标签按照一定的方式进行组织,从而生成歌曲对应的文本文档。为了减少噪音,我们只利用那些被多数人使用的标签来完成歌曲对应文档的构造,具体来说我们只考虑被标记次数大于10的标签。这样,每一首歌曲 s 都将对应一篇文档 d ,而歌曲集 S 将对应到一个文档集 D 。最后,我们对文档集 D 进行LDA主题模型建模,从而得到包含 K 个

用来全面刻画歌曲特征的隐含主题集合 T 。同时，对于歌曲集 S 中的任意歌曲 s ，我们能够得到其对应的隐含主题权重的向量，如式公式3.7所示。

其中， θ_{si} 表示歌曲 s 完全不属于该隐含主题代表的类别，即该隐含主题对歌曲的内容完全没有贡献； θ_{si} 表示歌曲 s 完全属于该隐含主题代表的类别； θ_{si} 表示歌曲 s 在一定程度上隶属于该隐含主题代表的类别。

表3.1展现了歌曲Squares的若干显著主题及其隶属于这些主题的概率。由表1可以看出，歌曲Squares隶属于主题508的概率为0.215，这说明主题508对该歌曲贡献度为0.215。类似的，主题231对歌曲的贡献度为0.072，这说明主题75对该歌曲的贡献度为0.072。因此，我们认为歌曲能够以更大的概率划分到主题508中。如前所述，我们不去也无法描述这些主题具体是什么含义，而我们正是利用这一点来回避确定性分类带来的弊端。

表 3.1: 歌曲Squares的显著主题及隶属度

主题id	508	313	296	106	231
隶属度	0.215	0.186	0.134	0.084	0.072

3.4 多维时间序列构造与预测

传统的基于“最相似”假设的推荐方法只考虑用户当前的收听习惯，较少考虑用户行为所形成的序列性趋势，而本文所提出的基于多维时间序列分析的音乐推荐方法则对此进行了考虑。在通过LDA主题模型对歌曲进行建模而得到歌曲对应的主题权重向量后，我们需要做的就是获取用户对应的在一定会话周期内收听的歌曲构成的序列并构造其对应的时间序列，然后使用相应的时间序列分析方法进行预测。

设用户 c 当前会话周期内收听的歌曲序列长度为 n ，为了预测其可能收听的下一首歌曲，我们需要获取该歌曲对应的主题权重向量，如式3.8所示。

其中， θ_{ij} 表示第 j 首歌曲隶属于第 i 个隐含主题的概率。换句话说，我们需要准确地预测对应的主题权重向量中每一维的值，从而能够在歌曲集 S 中找到最匹配的歌曲推荐给用户。由前文可知，用户 c 的听歌事件按照事件发生的先后顺序构成了一个与时间相关的序列 $Q(c)$ ，且每一个听歌事件中对应的歌曲可以表示成如式3.7所示的 K 维隐含主题权重向量，那么我们将 $Q(c)$ 按照式3.7展开，可以得到如式3.9所示的矩阵 $M(c)$ 。

其中， $M(c)$ 的行向量表示用户 c 对应的事件序列 $Q(c)$ 在各个隐含主题上的展开，而矩阵的列向量即为用户 c 收听事件序列中每一个事件对应的歌曲的隐含主

题权重向量。考察如式3.10所示的 $M(c)$ 的第 i 个行向量对应的序列。

其中， x_{it} 的含义如前所述。显然， x_{it} 是一个随着时间而变化的实数变量，因此可以将 x_{it} 看作一个以为时间变量的长度为 n 的时间序列。进一步地，我们可以利用时间序列分析和预测的方法对该时间序列进行分析和预测，从而可以估算第 i 个隐含主题对用户可能收听的下一首歌曲的贡献程度，即 $N(c)$ 对应的主题权重向量中第 i 维元素值的估计值。

我们可以将 $M(c)$ 看做是一个多维的时间序列，其每一个行向量对应一个维度，共计 K 个维度。然后，我们对这 K 个维度的时间序列逐一利用ARIMA模型进行分析并预测该维度的估计值。最终我们可以获得如式3.11所示各个维度的估计值。显然，该向量可以看作是用户下面可能收听的歌曲在 K 个主题上的近似分布。为简便起见，我们将该分布对应的歌曲记为。

3.5 相似度计算

通过LDA主题建模我们得到了如式3.7所示歌曲集 S 中歌曲对应的主题权重向量，考虑到向量中每一维度的值表征歌曲隶属于某一隐含主题的概率，我们将这些主题权重向量看作是离散的概率分布。进一步地，通过对多维时间序列的分析和预测，我们得到了如式3.11所示用户 c 可能收听的第 $(n+1)$ 首歌曲对应的主题权重向量的估计向量，即一个估计概率分布。这样，我们便可以计算歌曲集中的歌曲对应的概率分布与这个估计概率分布的距离。显然，如果 s 与距离足够小，那么我们就可以将歌曲 s 推荐给用户 c 。因此，我们可以用 s 与 c 的距离的倒数表示歌曲集中任一歌曲 s 对用户 c 的推荐度，即上文定义的效用函数 $u(c,s)$ ，如式3.12所示。

因为我们是通过主题模型建模将歌曲表示成离散的概率分布，所以我们可以使用KL距离(Kullback-Leibler Divergence)[19]以及Hellinger距离[20]等来度量两首歌曲之间的距离。考虑到KL距离不具有对称性，本文采用Hellinger距离来度量歌曲之间的距离，如式3.13所示。

其中， p_k 为歌曲和对应的主题概率分布之间的Hellinger距离， K 为隐含主题数目， p_{ik} 为歌曲在第 k 个隐含主题上的概率。显然，当两首歌曲越相似，那么其对应的距离越小。相反的，若两首歌曲越不相似，其主题概率分布对应的距离越大。

3.6 推荐列表生成

最后，我们根据效用函数计算 S 中所有歌曲的效用值，然后根据效用值对

歌曲排序，并将排名最高的 N 首推荐给用户，这样我们就为用户 c 推荐了一个长度为 N 的歌曲列表供用户选择。

3.7 实验设计和结果

3.7.1 数据收集

3.7.2 评测标准

3.7.3 实验设置

3.7.4 结果分析

第四章 一种基于用户三期行为的综合音乐推荐方法

4.1 问题分析

4.2 常用综合策略

4.2.1 整体式综合

4.2.2 并行式综合

4.2.3 流水线式综合

4.3 方法详述

4.3.1 方法框架

4.3.2 方法选取

4.3.3 方法综合

4.4 实验结果

4.5 分析讨论

第五章 系统实现

5.1 系统架构

5.2 离线处理模块

5.2.1 爬虫系统

5.2.2 结巴分词

5.2.3 Gensim软件包

5.3 在线处理模块

5.3.1 序列生成

5.3.2 推荐引擎

5.4 系统效果

第六章 总结与展望

6.1 工作总结

6.2 工作展望

附录 A 数据集使用说明

A.1 权利声明

本数据集抓取自Lastfm，所有数据归Lastfm所有，禁止商业用途。如果您想使用该数据集进行科研活动，请务必给出对Last.fm及本文的引用信息。

A.2 数据特点

1. 包含完整的用户、歌曲、曲作者的基本信息。
2. 包含丰富的用户行为记录，可用于构造用户行为序列。
3. 包含歌曲、曲作者的显著标签信息，可用于从文本的角度描述歌曲和作者。
4. 提供了标签的基本信息。
5. 数据被随机分组，可直接用来实验。

A.3 组织形式

本数据集使用Mysql进行管理，对应的数据库名为lastfm，您可以非常容易地将其导入并使用。数据集包含有5个基本的数据表：

1. 记录表record用于记录用户的收听行为，如某用户在某时间段收听了某歌曲。record由记录标识符(rid:int)、用户标识符(uid:vachar)、歌曲mbid(mbid:vachar)、记录发生的unix时间戳(uts:vachar)、记录发生的日期时间(datetime:vachar)、记录所属分组(scale:int)等字段构成。
2. 用户表user用于记录用户的基本信息，其由用户标识符(uid:vachar)、用户名(username:vachar)、用户国籍(country:vachar)、用户年龄(age:vachar)、用户性别(gender:vachar)、用户注册时的unix时间戳(registeredTime:vachar)、用户注册日期时间(registeredText:vachar)、播放序列(playlist:text)、用户所属分组(scale:int)等字段构成。

3. 歌曲表song用于记录歌曲的基本信息，其由歌曲标识符(sid:vchar)、歌曲对应mbid (mbid:vchar)、歌曲名称(name:text)、歌曲时长(duration:vchar)、曲作者标识符(aid:vchar)、曲作者名称(aname:vchar)、专辑名(album:text)、听众数目(listeners:vchar)、播放次数(playcount:vchar)、描述歌曲的显著标签(toptag:text)等字段构成。
4. 曲作者表artist用于记录曲作者的基本信息，其由曲作者标识符(mbid:vchar)、曲作者名称(name:text)、曲作者图片的链接(img:text)和描述曲作者的显著标签(toptag:text) 构成。
5. 标签表tag用于记录标签的基本信息，其由标签标识符(id:vchar)、标签名称 (name:text)、标签被创建的次数(reach:vchar)、标签被使用的次数(taggings:text)等字段构成。

A.4 字段解析

A.4.1 scale

记录表record和用户user中的scale字段用以表征记录和用户所处的分组编号。为了方便，本数据集将用户记录和用户分为Unused、Small、Whole和Session四类。

其中，Small、Whole和Session被scale字段分割成40组，其中第0组到第9组属于Small数据集，第0组到第29组属于Whole数据集，第30组到第39组属于Session数据集。Unused数据的scale设为-1。显然，Small数据集是Whole数据集的一部分，它们的特点是每一个用户所收听的歌曲都在一个会话期内，即不存在长时中断。从Whole数据集中划分出Small数据集的主要目的是方便机器性能不佳的用户使用，对于Small数据集，用户可以使用10组中的9组作训练集而余下的一组作测试集。Session数据集与Whole数据集的主要区别是每一个用户所收听的歌曲至少在两个会话期内。类似的，用户可以用其中9组作训练集而余下的一组作测试集。下表给出了Small、Whole和Session三类数据集的基本统计信息。

A.4.2 playlist

数据表user中的playlist字段用以表征用户按序收听的歌曲构成的序列，数据如“sid1:ratio1==,sid2:ratio2==, ...,sidn:ration”所示。其中，sid表示被听歌曲的标识符(注:非mbid)。ratio表示两首歌之间的时间间隔与前一首歌曲时长的比例，用以表征用户收听该首歌曲的时长比例。显然，ratio 过小表示用户刚开始收听遍跳过，ratio过大表明歌曲被完整收听而且还可能有暂停发生。

A.4.3 toptag

数据表song和artist中的字段toptag表示Lastfm网站中的用户给歌曲或曲作者所打的显著标签，数据如“tag1:count1,tag2:count2,...,tagn:countn”所示。其中，tag表示被打标签名称，count表示标签被标记次数。需要注意的是，在Lastfm中，count并非标签被应用于歌曲或曲作者的绝对次数，而是标签相对于被使用最多次的标签的相对次数。例如在描述歌曲“Collapse of History”的标签中，“industrial”被使用最多次且次数为200，而标签“Stars”被使用100次。那么，在歌曲记录对应的字段toptag中，“industrial”对应的count为100，“Stars”对应的count为50，即“industrial”:100,”starts”:50，以此类推。

A.4.4 其他字段

数据表中的其他字段都比较简单直观，这里就不再一一介绍。

A.5 应用场景

1. 使用文本分析的方法描述歌曲或者曲作者特征。
2. 分析用户所收听歌曲的序列，包括跨会话分析和会话内分析。
3. 预测用户下一首可能收听的歌曲或者曲作者。
4. 生成用户可能喜欢的播放列表。
5. 标签预测问题。
6. 其他适合的应用场景。

表 A.1: 数据集的统计信息

	Small	Whole	Session
用户数	1530	4590	1690
歌曲数	24992	62422	32218
稀疏度	99.92	99.97	99.92
最大长度	30	30	66
最短长度	10	10	20
中位长度	24	24	30

参考文献

- [1] Adomavicius, G., Tuzhilin, A., 2005. Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *Knowledge and Data Engineering, IEEE Transactions on* 17, 734–749.
- [2] Blei, D.M., Ng, A.Y., Jordan, M.I., 2003. Latent dirichlet allocation. *the Journal of machine Learning research* 3, 993–1022.
- [3] Bogdanov, D., Serrà, J., Wack, N., Herrera, P., Serra, X., 2011. Unifying low-level and high-level music similarity measures. *Multimedia, IEEE Transactions on* 13, 687–701.
- [4] Cano, P., Koppenberger, M., Wack, N., 2005. Content-based music audio recommendation, in: *Proceedings of the 13th annual ACM international conference on Multimedia, ACM*. pp. 211–212.
- [5] Chordia, P., Godfrey, M., Rae, A., 2008. Extending content-based recommendation: The case of indian classical music., in: *ISMIR, Citeseer*. pp. 571–576.
- [6] DoubanFM, . <http://douban.fm>.
- [7] Downie, J.S., 2003. Music information retrieval. *Annual review of information science and technology* 37, 295–340.
- [8] Hadoop, . <http://hadoop.apache.org/>.
- [9] Hariri, N., Mobasher, B., Burke, R., 2012. Context-aware music recommendation based on latenttopic sequential patterns, in: *Proceedings of the sixth ACM conference on Recommender systems, ACM*. pp. 131–138.
- [10] Herlocker, J.L., Konstan, J.A., Terveen, L.G., Riedl, J.T., 2004. Evaluating collaborative filtering recommender systems. *ACM Transactions on Information Systems (TOIS)* 22, 5–53.
- [11] Hoffman, M.D., Blei, D.M., Bach, F.R., 2010. Online learning for latent dirichlet allocation., in: *NIPS*, p. 5.

- [12] Hyung, Z., Lee, M., Lee, K., 2012. Music recommendation based on text mining, in: IMMM 2012, The Second International Conference on Advances in Information Mining and Management, pp. 129–134.
- [13] Kim, Y.E., Schmidt, E.M., Migneco, R., Morton, B.G., Richardson, P., Scott, J., Speck, J.A., Turnbull, D., 2010. Music emotion recognition: A state of the art review, in: Proc. ISMIR, Citeseer. pp. 255–266.
- [14] Kullback, S., Leibler, R.A., 1951. On information and sufficiency. *The Annals of Mathematical Statistics* , 79–86.
- [15] Lastfm, . <http://last.fm>.
- [16] Linden, G., Smith, B., York, J., 2003. Amazon. com recommendations: Item-to-item collaborative filtering. *Internet Computing*, IEEE 7, 76–80.
- [17] Mak, C.M., Lee, T., Senapati, S., Yeung, Y.T., Lam, W.K., 2010. Similarity measures for chinese pop music based on low-level audio signal attributes., in: ISMIR, pp. 513–518.
- [18] Makridakis, S., Hibon, M., 1997. Arma models and the box–jenkins methodology. *Journal of Forecasting* 16, 147–163.
- [19] McFee, B., Lanckriet, G.R., 2011. The natural language of playlists., in: ISMIR, pp. 537–542.
- [20] Oosterhoff, J., van Zwet, W.R., 2012. A note on contiguity and Hellinger distance. Springer.
- [21] Pandora, . <http://pandora.com>.
- [22] Park, S.E., Lee, S., Lee, S.g., 2011. Session-based collaborative filtering for predicting the next song, in: Computers, Networks, Systems and Industrial Engineering (CNSI), 2011 First ACIS/JNU International Conference on, IEEE. pp. 353–358.
- [23] Powers, D.M., 2011. Evaluation: from precision, recall and f-measure to roc, informedness, markedness & correlation. *Journal of Machine Learning Technologies* 2, 37–63.

- [24] Rentfrow, P.J., Gosling, S.D., 2003. The do re mi's of everyday life: the structure and personality correlates of music preferences. *Journal of personality and social psychology* 84, 1236.
- [25] Resnick, P., Iacovou, N., Suchak, M., Bergstrom, P., Riedl, J., 1994. Grouplens: an open architecture for collaborative filtering of netnews, in: *Proceedings of the 1994 ACM conference on Computer supported cooperative work*, ACM. pp. 175–186.
- [26] Rosen-Zvi, M., Griffiths, T., Steyvers, M., Smyth, P., 2004. The author-topic model for authors and documents, in: *Proceedings of the 20th conference on Uncertainty in artificial intelligence*, AUAI Press. pp. 487–494.
- [27] Salton, G., Wong, A., Yang, C.S., 1975. A vector space model for automatic indexing. *Communications of the ACM* 18, 613–620.
- [28] Sarwar, B., Karypis, G., Konstan, J., Riedl, J., 2001. Item-based collaborative filtering recommendation algorithms, in: *Proceedings of the 10th international conference on World Wide Web*, ACM. pp. 285–295.
- [29] Scheibehenne, B., Greifeneder, R., Todd, P.M., 2010. Can there ever be too many options? a meta-analytic review of choice overload. *Journal of Consumer Research* 37, 409–425.
- [30] Song, Y., Dixon, S., Pearce, M., 2012. A survey of music recommendation systems and future perspectives, in: *9th international symposium on computer music modelling and retrieval (CMMR 2012)*, pp. 19–22.
- [31] Storm, . <http://storm.incubator.apache.org/>.
- [32] Tang, J., Yao, L., Chen, D., 2009. Multi-topic based query-oriented summarization., in: *SDM, SIAM*. pp. 1147–1158.
- [33] Xiami, . <http://www.xiami.com>.
- [34] Yang, C.C., Chen, H., Hong, K., 2003. Visualization of large category map for internet browsing. *Decision Support Systems* 35, 89–102.

简历与科研成果

基本情况

王守涛，男，汉族，1987 年 12 月出生，安徽省阜阳市人。

教育背景

2011.9～2014.6	南京大学计算机科学与技术系	硕士
2007.9～2011.6	合肥工业大学计算机信息学院	本科

攻读硕士学位期间完成的学术成果

- [1] 吕建,徐锋,王守涛. 一种基于多维时间序列分析的个性化音乐推荐系统及其实现方法:中国. 201410077177.1[P]. 2014-03-04.

致 谢

日光荏苒，在南京大学计算机与技术系软件所攻读硕士的三年行将结束，在此对在我成长道路上谆谆教导的师长以及在我生活中给予关心和帮助的同学表达由衷的谢意。

首先，感谢我的导师徐锋教授，从起始移动互联网领域的编程实践到后来推荐系统相关知识的学习与研究，徐老师都耐心地进行指导。在我工作遇到困难的时候，徐老师总是能够给出一些非常有建设性并且有远见的建议，他总是能够从比较高的层次指出工作中存在的不足，这对于我从宏观上去把握所研究的工作内容有非常大的裨益。另外，徐老师非常健康的生活方式也值得我在以后的工作和生活中认真学习。

感谢计算机系软件所的吕建教授，陶先平教授，马晓星教授，胡昊教授，黄宇副教授，许畅副教授，曹春副教授，余萍，马俊和张建莹老师等，他们在我学习和工作中给予了许多帮助。感谢和我一起努力的同学，姚远博士，张栋栋，王瑶菁，袁明珠，周昊一，陈晓宇等，在论文和项目中的鼓励和协作。感谢徐经纬博士，朱晓瑞博士在相关工作中给出的启发性见解和指导。感谢室友王海涛，王逸恺等，让我对其他研究方向有所认知，他们科研的态度也给了我很大启发，而平时和谐愉悦的宿舍生活使得我在平时的科研和学习中充满活力，这定将成为我学生生活中最美好的记忆之一。

感谢父母一直以来对我的支持和鼓励，他们为了给我创造良好的成长环境和学习环境付出了很大的心血，他们是我坚实的后盾和持续奋斗的动力，我定不辜负他们对我的期望，在以后的工作中继续努力，成就自己的一番事业。

最后，感谢我的女朋友侯小丽的坚守和付出，无论我遇到何种挫折和坎坷，她都出现在我的身边并给予我最大的支持和鼓励，我一定会持续积极上进，为以后的更加美好的生活创造条件。