分类号： TP311.5 单位代码：

学 号：



硕士学位论文



**中文论文题目 ：团购商品个性化推荐系统的设计与实现**

**英文论文题目：****Group-buying** [**Design**](http://cn.bing.com/dict/clientsearch?mkt=zh-CN&setLang=zh&form=BDVEHC&ClientVer=BDDTV3.5.0.4311&q=%E5%9B%A2%E8%B4%AD%E5%95%86%E5%93%81%E4%B8%AA%E6%80%A7%E5%8C%96%E6%8E%A8%E8%8D%90%E7%B3%BB%E7%BB%9F%E7%9A%84%E8%AE%BE%E8%AE%A1%E4%B8%8E%E5%AE%9E%E7%8E%B0)**[And](http://cn.bing.com/dict/clientsearch?mkt=zh-CN&setLang=zh&form=BDVEHC&ClientVer=BDDTV3.5.0.4311&q=%E5%9B%A2%E8%B4%AD%E5%95%86%E5%93%81%E4%B8%AA%E6%80%A7%E5%8C%96%E6%8E%A8%E8%8D%90%E7%B3%BB%E7%BB%9F%E7%9A%84%E8%AE%BE%E8%AE%A1%E4%B8%8E%E5%AE%9E%E7%8E%B0" \t "_blank)** **[implementation](http://cn.bing.com/dict/clientsearch?mkt=zh-CN&setLang=zh&form=BDVEHC&ClientVer=BDDTV3.5.0.4311&q=%E5%9B%A2%E8%B4%AD%E5%95%86%E5%93%81%E4%B8%AA%E6%80%A7%E5%8C%96%E6%8E%A8%E8%8D%90%E7%B3%BB%E7%BB%9F%E7%9A%84%E8%AE%BE%E8%AE%A1%E4%B8%8E%E5%AE%9E%E7%8E%B0" \t "_blank) of** **[personalized](http://cn.bing.com/dict/clientsearch?mkt=zh-CN&setLang=zh&form=BDVEHC&ClientVer=BDDTV3.5.0.4311&q=%E5%9B%A2%E8%B4%AD%E5%95%86%E5%93%81%E4%B8%AA%E6%80%A7%E5%8C%96%E6%8E%A8%E8%8D%90%E7%B3%BB%E7%BB%9F%E7%9A%84%E8%AE%BE%E8%AE%A1%E4%B8%8E%E5%AE%9E%E7%8E%B0" \t "_blank)** **[recommendation](http://cn.bing.com/dict/clientsearch?mkt=zh-CN&setLang=zh&form=BDVEHC&ClientVer=BDDTV3.5.0.4311&q=%E5%9B%A2%E8%B4%AD%E5%95%86%E5%93%81%E4%B8%AA%E6%80%A7%E5%8C%96%E6%8E%A8%E8%8D%90%E7%B3%BB%E7%BB%9F%E7%9A%84%E8%AE%BE%E8%AE%A1%E4%B8%8E%E5%AE%9E%E7%8E%B0" \t "_blank)** **[system](http://cn.bing.com/dict/clientsearch?mkt=zh-CN&setLang=zh&form=BDVEHC&ClientVer=BDDTV3.5.0.4311&q=%E5%9B%A2%E8%B4%AD%E5%95%86%E5%93%81%E4%B8%AA%E6%80%A7%E5%8C%96%E6%8E%A8%E8%8D%90%E7%B3%BB%E7%BB%9F%E7%9A%84%E8%AE%BE%E8%AE%A1%E4%B8%8E%E5%AE%9E%E7%8E%B0" \t "_blank)**

申请人姓名： 李朝阳

指导教师： 李际军

合作导师： 朱小军

专业学位类别： 工程硕士

专业学位领域： 软件工程

所在学院： 软件学院

**论文提交日期 年 月 日**

**学位论文书脊示例**

|  |
| --- |
| **题**  **目**  **作**  **者**  **姓**  **名**  **浙**  **江**  **大**  **学** |

3cm左右

3cm左右

**中文论文题目** （小二号仿宋体加黑）



**论文作者签名:**

**指导教师签名:**

论文评阅人1： (姓名\职称\单位,下同)

评阅人2： (隐名评阅学位论文省略)

评阅人3：

评阅人4：

评阅人5：

答辩委员会主席： (姓名\职称\单位)

委员1：

委员2：

委员3：

委员4：

委员5：

答辩日期：

**英文论文题目**（16pt Time New Roman，Bold）



**Author’s signature:**

**Supervisor’s signature:**

Thesis reviewer 1： (姓名\职称\单位,下同)

Thesis reviewer 2： (隐名评阅学位论文省略)

Thesis reviewer 3：

Thesis reviewer 4：

Thesis reviewer 5：

Chair： (姓名\职称\单位)

(Committee of oral defence)

Committeeman 1：

Committeeman 2：

Committeeman 3：

Committeeman 4：

Committeeman 5：

Date of oral defence：

浙江大学研究生学位论文独创性声明

本人声明所呈交的学位论文是本人在导师指导下进行的研究工作及取得的研究成果。除了文中特别加以标注和致谢的地方外，论文中不包含其他人已经发表或撰写过的研究成果，也不包含为获得 **浙江大学** 或其他教育机构的学位或证书而使用过的材料。与我一同工作的同志对本研究所做的任何贡献均已在论文中作了明确的说明并表示谢意。

学位论文作者签名： 签字日期： 年 月 日

学位论文版权使用授权书

本学位论文作者完全了解 **浙江大学** 有权保留并向国家有关部门或机构送交本论文的复印件和磁盘，允许论文被查阅和借阅。本人授权 **浙江大学** 可以将学位论文的全部或部分内容编入有关数据库进行检索和传播，可以采用影印、缩印或扫描等复制手段保存、汇编学位论文。

（保密的学位论文在解密后适用本授权书）

学位论文作者签名： 导师签名：

签字日期： 年 月 日 签字日期： 年 月 日

摘要

推荐系统是解决信息过载的一种重要方式; 它通过分析用户行为日志预测用户的喜好,行为,兴趣等并为用户做推荐.随着web 2.0 的不断发展, 用户的参与度越来越高, 而早期推荐系统主要基于内容或者静态用户如基于人口统计的推荐, 而用户实时数据无法产生实时推荐.或者根据用户UGC的上下时间因素进行推荐, 在这方面有许多关键点需要解决.

本文基于公开用户行为数据集对推荐系统进行研究, 包括..; 本文的主要工作内容如下:

1. 推荐系统现状, 个性化服务的需求

**关键词：**　推荐系统，协同过滤, 实时响应, 日志系统, 展示系统, 推荐引擎

Abstract

**Key Words：**,

目录

[摘要 i](#_Toc415728240)

[Abstract ii](#_Toc415728241)

[图目录 III](#_Toc415728242)

[第1章 绪论 4](#_Toc415728243)

[1.1 课题背景与意义 4](#_Toc415728244)

[1.2 研究内容 4](#_Toc415728245)

[1.3 公开用户数据集 5](#_Toc415728246)

[1.3.1 亚马孙（z.cn）数据集 5](#_Toc415728247)

[1.3.2 MovieLens电影评分数据集 6](#_Toc415728248)

[1.3.3 CiteULike论文书签数据集 6](#_Toc415728249)

[1.4 论文概要结构描述 6](#_Toc415728250)

[第2章 Top-N推荐用户兴趣预测的模型 7](#_Toc415728251)

[2.1 简介 7](#_Toc415728252)

[2.1.1 7](#_Toc415728253)

[2.2 问题数学定义和概括 8](#_Toc415728254)

[2.2.1 传统Top-N推荐 8](#_Toc415728255)

[2.2.2 基于时间的Top-N推荐 8](#_Toc415728256)

[2.2.3 隐性反馈数据的动态变化 10](#_Toc415728257)

[2.3 本章小结 12](#_Toc415728258)

[2.3.1 Hello 13](#_Toc415728259)

[第3章 评分预测中用户兴趣预测的模型 14](#_Toc415728260)

[3.1 第一节 14](#_Toc415728261)

[3.1.1 14](#_Toc415728262)

[3.2 本章小结 14](#_Toc415728263)

[3.2.1 14](#_Toc415728264)

[第4章 推荐引擎设计与实现 15](#_Toc415728265)

[4.1 第一节 15](#_Toc415728266)

[4.1.1 15](#_Toc415728267)

[4.2 本章小结 15](#_Toc415728268)

[4.2.1 15](#_Toc415728269)

[第5章 用户行为日志系统设计与实现 16](#_Toc415728270)

[5.1 第一节 16](#_Toc415728271)

[5.1.1 16](#_Toc415728272)

[5.2 本章小结 16](#_Toc415728273)

[5.2.1 16](#_Toc415728274)

[第6章 展示系统设计与实现 17](#_Toc415728275)

[6.1 第一节 17](#_Toc415728276)

[6.1.1 17](#_Toc415728277)

[6.2 本章小结 17](#_Toc415728278)

[6.2.1 17](#_Toc415728279)

[第7章 18](#_Toc415728280)

[7.1 第一节 18](#_Toc415728281)

[7.1.1 18](#_Toc415728282)

[7.2 本章小结 18](#_Toc415728283)

[7.2.1 18](#_Toc415728284)

[第8章 19](#_Toc415728285)

[8.1 第一节 19](#_Toc415728286)

[8.1.1 19](#_Toc415728287)

[8.2 本章小结 19](#_Toc415728288)

[8.2.1 19](#_Toc415728289)

[参考文献 20](#_Toc415728290)

图目录

[图 1 数据集所含各种类型物品数目 6](#_Toc415730679)

[图 2 用户-物品关系二分图 9](#_Toc415730680)

[图 3 路径融合算法 10](#_Toc415730681)

[图 4 时间段图模型 12](#_Toc415730682)

[图 3.1 流程图 15](#_Toc415730683)

[表 2.1简单的多描述分配表 2](#_Toc164669160)

# 绪论

## 课题背景与意义

随着web2.0 的快速发展, 无论是用户产生的数据, 还是物品的数量已经从匮乏到了过载的阶段, 那么如何从海量的物品中找到真正用户需要的物品;

以用户的视角来看，自从有了互联网以来，随着网络资源的逐渐增多，用户寻找信息的方式经历了几个重要阶段。早期用户主要靠直接记住感兴趣网站的网址来寻找内容，后来雅虎公司提出了分类目录系统，将网站分门别类，方便用户查询。但随着信息越来越多，分类目录也只能记录少量的网站，而且维护分类信息的费用也随着信息数量的增长也越来越高, 于是产生了搜索引擎。

搜索引擎可以通过关键词找到自己所需要的信息。但是搜索引擎需要用户提供关键字来寻找信息，因此它不能满足用户的其他需求，比如当用户无法准确描述自己需求的关键字的时候，搜索引擎就无法返回用户所需的内容了。和搜索引擎一样，推荐系统也是一种为用户快速查找有用信息的工具。和搜索引擎不同的是，推荐系统不需要提供明确的关键字，而是通过分析用户行为数据来给用户的兴趣建模从而主动推荐出能够满足用户需求的信息。因此从某种意义上说，推荐系统和搜索引擎是两个互补的工具。搜索引擎满足了用户能表达自己的意图时主动查找的需求，而推荐系统是在用户没有目的的时候帮助用户发现让他们感兴趣的内容。这样满足了在互联网有相关信息的基础上, 用户所需信息能够推送给相关用户。

以信息的视角来看，推荐系统可以更好的利用信息的长尾(The Long Tail)长尾理论[[1]](#footnote-1)，电子商务网站相对于传统超市的优点是它可以给用户提供更多的选择。因为电子商务网站货架的成本极为低廉，上架商品只需要在系统中添加一条记录。但同时长尾的出现也给系统或者点子商务带来了新的挑战，即如何将大量的商品推荐给潜在消费用户和用户实时行为系统如何实时为用户推荐。

## 研究内容

推荐系统的主要目的是，将与用户兴趣相关的信息推送给用户；这就涉及两部分：1）如何找到与用户实时行为相关的物品； 2）如何展现相关推荐并与用户交互产生实时数据；本位主要探究以上两个方面。

协同过滤算法是第一代推荐系统也是应用最为成熟的推荐系统，目前有两类主要的协同过滤算法： 基于用户的协同过滤算法和基于项目的协同过滤算法。基于用户的系统过滤算法就好比， 用户A有一批志同道合的朋友， 他们对物品I都有兴趣并做了评分， 而推荐系统就根据A的朋友的评分进行计算判断是否为A推荐物品I； 其核心思想是：利用用户的行为数据计算用户间的相似性；得到与用户相似性较高的邻居-用户好友群， 然后利用这些用户对其物品的评价来预测目标用户的喜好程度， 然后做出推荐； 基于项目的协同过滤算法是： 用户A对物品集合有一个相似的评价，当需要对某物品I进行预测评估的时候，可根据对相似物品的集合的评价来对物品I进行推荐。这种做法的优点在于对推荐对象没有特殊的要求，能处理音乐电影等难以进行文本结构化表示的对象。

基于内容的推荐是在推荐引擎应用最为广泛的推荐机制，它的核心思想是根据推荐物品或内容的元数据，发现物品或者内容的相关性，然后基于用户以往的喜好记录，推荐给用户相似的物品。这种基于内容的推荐机制的好处在于它能很好的建模用户的口味，能提供更加精确的推荐。但它也存在以下几个问题：1. 需要对物品进行分析和建模，推荐的质量依赖于对物品模型的完整和全面程度。在现在的应用中我们可以观察到关键词和标签（Tag）被认为是描述物品元数据的一种简单有效的方法。2. 物品相似度的分析仅仅依赖于物品本身的特征，这里没有考虑人对物品的态度。3. 因为需要基于用户以往的喜好历史做出推荐，所以对于新用户有“冷启动”的问题。

本文相关的推荐算法

## 公开用户数据集

### 亚马孙（z.cn）数据集

从亚马孙网站上抓取的数据， 其中包括548552种不同的产品（书籍，音乐CD，DVD和VHS录像带）的信息。每条记录包括题目， 销售排行， 相似产品列表， 产品的分类以及产品的评价。



图 1 数据集所含各种类型物品数目

### MovieLens电影评分数据集

MovieLens数据集中，用户对自己看过的电影进行评分，分值为1～5。MovieLens包括两个不同大小的库，适用于不同规模的算法．小规模的库是943个独立用户对1682部电影作的10000次评分的数据；大规模的库是6040个独立用户对3900部电影作的大约100万次评分。

### CiteULike论文书签数据集

CiteULike4是一个著名的论文书签网站。它让研究人员把自己喜欢的论文信息提交到它的网站上，同时让用户对这些论文打上标签。CiteULike公布了一个数据集，每一行是一条记录，由一个四元组构成，包括用户，论文，时间和书签。这个数据集不包含评分数据，但包含了时间信息，因此可以作为一个含有时间信息的用户隐形反馈数据集使用。

## 论文概要结构描述

本文除第一章之外可分为六部分：

* 第二章 Top-N推荐
* 第三章 基于用户邻域模型与矩阵分解
* 第四章 主要是推荐系统中推荐引擎具体实现的架构设计， 类图描述等
* 第五站 用户行为日志收集的设计与实现
* 第六章 推荐单子的展示， 与用户日志的收集
* 第七章 论文结束语以及展望

# Top-N推荐用户兴趣预测的模型

## 简介

推荐系统的主要任务是为每个用户提供一个含有N件物品的推荐列表。因此，实际生产中的推荐系统在预测用户行为时通常有两步。第一步是得到一个和用户相关的物品列表，在真正的生产中系统中包含很多件物品， 但是真正和用户相关的物品很少，我们仅仅只要基于与用户相关的物品即可；第二步是在得到用户将会评分的列表之后，预测用户会给该物品多少评分。第一个问题也被称为Top-N推荐问题，该问题是本节研究的主要问题。关于Top-N推荐的研究有很多。其中协同过滤是很多生产系统中的主要算法[[[2]](#endnote-1), [[3]](#endnote-2)]。

用户的行为数据根据是否为用户主动返回数据可将数据分为显性反馈数据， 隐形反馈数据；显性反馈数据是用户主动返回的数据， 比如， 对物品的喜欢/不喜欢，评分， 评价等， 但是返回的数量想回来说比较少； 而对于隐形反馈数据， 隐性反馈行为指的是那些不能明确反应用户喜好的行为。最具代表性的隐性反馈行为就是页面浏览行为。凡是网站大豆有隐性反馈数据(Implicit Feedback)，这些数据存储在日志系统内。比如，在视频网站中，用户可以通过打分表示对一个视频的喜好。但是，大部分用户看完视频后并不会对视频进行评价，相反用户看了什么视频，每个视频看了多长时间，这样的数据是视频网站能够收集的主要数据，而且数据量巨大。但用户观看一个视频，并不能体现出用户对这个视频的兴趣意图，比如用户看完可能觉得不错，或者觉得不好看。文献[[[4]](#endnote-3)]详细对比了隐性反馈数据和显性反馈数据的优点和缺点。隐性反馈数据虽然不如显性反馈明确，但其中也蕴含着用户的兴趣信息，如果能够做出适当的挖掘，可以很好的来预测用户的兴趣和行为。本节主要描述利用隐形反馈数据来过滤和用户有关系的物品列表， 即Top-N推荐。

### 

每个表都应有表说明[2]，表说明包含表序与表题，居中排印在表的上方；表序与表题之间空一字距；独表表示也同样要求；表与表说明不能破页。[[5]](#footnote-2)

表中不设“备注”栏，需要说明的事项可排印在表下方，表内用星号“\*”或圈码“①、②”标注在相应内容的右上角。表中的参数应表明量和单位的符号，如表中所有参数的单位相同，可标注在表的右上方或表说明之后；各栏单位不同则应标注在各栏表头内。表中不能用“同上”、“同左”一类词代替具体数字；无某项目则空白；未发现用“...”；结果为零用“0”；同一栏的数字必须按位次上下对齐。

表随正文，先见文字后见表；需要转页的表，应在续表的右上角或左上角注明“续表×”，并应重复排印表头。

图表与上下文之间各空一行。

于是我们得到表2.1中的两个描述[3]。

表2.1 简单的多描述分配表

|  |  |  |
| --- | --- | --- |
| 描述1 | ζR | (1−ζ)R |
| 描述2 | ζR（重复） | (1−ζ)R |

## 问题数学定义和概括

### 传统Top-N推荐

一般的， 如下的数据集Data = {( user, item)}，其中数据集中的每个二元组( user, item)代表用户 user对物品 item产生过行为。而Top-N推荐的任务，是通过学习数据集Data，建立用户兴趣的模型，用来给每个user推荐 N个与user最相关的物品。

可扩展内容

### 基于时间的Top-N推荐

用户的兴趣是随着时间变化的，并且用户的历史行为也随之增加。因此，推荐结果也需要随之实时更新他们兴趣的变化，来适应用户的新行为，来产生满足用户最新兴趣的推荐内容。因此，考虑了时间因素后，本节主要解决随着时间变化的Top-N推荐：在一个给定的时间段内， 根据用户行为日志行为数据为用户提供满足用户兴趣的N件物品。

#### 推荐方法-基于图模型的协同过滤算法

利用二分图为用户行为建模。用户物品二分图定义为 G(U, I, E, w)，其中I是物品节点集合，U是用户节点集合，若用户 u对物品i 有过行为，就存在一条边e(vu, vi) ∈ E连接用户 u的节点vu和物品i的节点vi，而行为的强度定义为边的权重w(e)。如图2所示， 图中定义了U1, U2, U3 和 I1, I2, I3, I4之间的关系。 从图的左半部分我们可以看到用户和物品之间的关系； 比如 U1 和 I1, I4之间有行为； U2对 I1, I3, I4有行为；U3对I2, I3有行为。

.图 2 用户-物品关系二分图

Top-N推荐的主要任务就是找到和用户有行为的物品集；反映到二分关系模型上就转化为了用户节点到物品节点是否存在路径， 有几条路径， 路径的度为多少； 比如，在图2中用户U1到I2,之间并没有直接关系，但是有 U1 → I4→U3→ I2这条路径到达I2， 也就是说U1和I2,之间有一条长短为3的关系。计算图中顶点之间距离的算法有很多，Fouss在文[[[6]](#endnote-4)]中总结了很多， 我们介绍一种路径融合算法[[[7]](#endnote-5)]

#### 路径融合算法

算法的基本思想为是，图上两个节点相似的判断标准是：

1. 两顶点之间有许多条边相连；
2. 两顶点之间路径比较短；
3. 顶点之间路径不经过出度很大的点。

根据以上三个标准，图中顶点的相似度关键取决于连接顶点的路径。算法在计算顶点的相似度的时候，首先找到顶点间的所有路径，并计算每条路径的权重，并利用全部路径权重的和来衡量顶点之间的相似度。

以图3为例。图中一共有A,B,C,D四个用户和i1,i2,i3,i4四个物品。其中用户A 喜欢物品i1,i3，用户B喜欢物品i1,i3,i4，用户C喜欢物品i1,i4，用户D喜欢物品i2,i4。如果要给用户A 做推荐，可以推荐物品i2和i4，因为这两个物品用户A还没有过行为。但用户A和物品i2,i4之间的相似度却不同。根据上面的三条准则，用户A和物品i4在图上的相似度更高。如图所示，用户A和物品i4之间的最短路径长度是3，并且长度为3的最短路径有2条。而用户A和物品i2之间的最短路径是5。因此根据第二条准则， A对i4的喜欢程度很可能大于A对i2的喜欢程度。此外， A,i4之间长度为3的路径有两条(A,i1,B,i4)和(A,i1,C,i4)。那么根据上面第三条， (A,i1,C,i4)是一条比(A,i1,B,i4)更重要的路径，因为顶点C的出度2，小于顶点B的出度。同理， A,i2之间长度为5的两条路径(A,i1,B,i4,D,i2)的重要程度也小于(A,i1,C,i4,D,i2)，也是因为B的出度值大于C的出度。



图 3 路径融合算法

通过上面简单的例子可知， 路径融合算法在计算用户对物品的喜好程度时通过下面三个步骤. 第一条找出用户顶点和物品顶点之间的最短路径；第二条计算每条最短路径的权重；第三条将所有最短路径的权重线性叠加作为最终用户对物品喜好程度的度量。

详细的路径融合算法

### 隐性反馈数据的动态变化

#### 用户的长期兴趣和短期兴趣(fix)

用户的兴趣可以分为长期兴趣和短期兴趣。比如一个历史爱好者一般阅读历史方面的文章。但是当08年奥运会开始的时候，他也会关注奥运方面的体育新闻。在这个例子中，历史属于这个用户的长期兴趣，而奥运新闻只是该用户在奥运期间受社会热点影响而产生的短期兴趣。由上面的例子可以看出，长期兴趣一般是由于用户自身的兴趣爱好产生的，而短期的兴趣一般是受社会热点的影响而产生的。一个好的推荐算法需要通过同时考虑用户的长期兴趣和短期兴趣来给用户做出推荐。上一节介绍了基于图模型的Top-N推荐算法，这一节将研究如何将用户的长期兴趣和短期兴趣同时建模到图模型中，并在模型的基础上设计推荐算法。

为了表达用户的短期兴趣，我们引入了用户时间节点。在前面的二分图模型中，用户 u会和所有他感兴趣的物品节点通过带有权重的边相连。但是用户不同时刻的兴趣不同。为了表示用户 u在 t时间段的兴趣，我们引入了 vut节点。这个节点将会和用户 u 在 t时段感兴趣的所有物品节点通过边在图上相连。这里每个时间段的时间长度将根据数据集的不同来选择不同的长度。

引入用户时间节点，用户物品二分图模型转变为时间段图模型 Session based Graph Model (SGM)。 SGM是一个二分图，它被定义为 G( U, S, I, E, w)。其中U是用户节点集合， S是用户时间节点集合， I是物品节点集合， w : E →R 定义了边的权重。图4是一个SGM的简单例子，图中包含了2个用户节点，4个session节点以及4个物品节点。图中表明，用户A对物品i1,i2有过行为，其中A在1时段对物品i1有行为， A在2时段对物品i2有行为，用户B对物品i1,i2,c有行为，其中B在1时段对物品i1有行为，在2时段对物品i2,i3有行为。

在SGM中，用户节点vu将和用户 u曾经发生行为的所有物品(这些物品的集合用N(u)表示)对应的节点都有边相连，这表示了用户的长期兴趣。而用户时间节点vut会和用户在t时刻有发生行为的所有物品(这些物品的集合被记为N(u, t))对应的节点有边相连。因此，如果从用户节点vu出发，会经过N(u)中的物品节点而最终到达用户 u没有过行为，但和 N( u)中物品相似的物品对应的节点。而如果从用户时间节点 vut出发，最终则会到达和 N( u, t)中物品相似的物品对应的节点。因此，用户节点和用户时间节点分别代表了用户的长期兴趣和短期兴趣，通过给这两种节点上赋予不同的权重，就可以控制用户的长期兴趣和短期兴趣对最终推荐结果的影响。SGM中一共有三种不同的节点，它们的权重定义如下：

其中β ∈ [0, 1]控制了长期兴趣和短期兴趣对推荐结果的影响。



图 4 时间段图模型

#### 基于SGM的推荐算法(fix)

本节主要讨论如何在时间段图模型SGM上度量用户对物品的兴趣并设计个性化推荐算法。 2.2.2.1节中介绍了用户物品二分图模型上的个性化排名算法，本节将对那些算法进行改进，让他们能够运行在SGM模型之上。

路径融合算法

## 本章小结

本章主要讨论如何利用图模型在Top-N推荐中同时考虑用户的短期兴趣和长期兴趣，从而提高推荐结果的准确度。首先本章讨论了传统的用户物品二分图模型以及基于该模型的推荐算法。在介绍相关研究的基础上，本章提出了基于路径的路径融合算法，该算法通过分析图上节点之间的路径来度量节点之间的相似度。然后，我们发现用户的兴趣分为长期兴趣和短期兴趣，并通过在用户物品二分图模型上加入一种新的节点来对这两种兴趣进行建模。两个不同数据集上的实验表明，考虑用户的长期兴趣和短期兴趣能够提高Top-N推荐的准确度，同时路径融合算法相对与其他算法也具有更高的推荐准确度。

# 评分预测中用户兴趣预测的模型(fix)

## 简介

推荐系统的主要目的是从用户的行为中挖掘出用户的兴趣，因此收集用户行为的系统是推荐系统的重要组成部分。让用户给物品评分是很多推荐系统收集用户兴趣的重要手段。用户评分行为也被称为显性反馈行为，因为评分结果明确的表示了用户对物品的兴趣。

表格 1具有代表性的线性反馈数据

|  |  |
| --- | --- |
| 网站 | 显性反馈 |
|  |  |
| 视频网站 | 对视频的评分 |
| 门户网站 | 对新闻的评分/评价等 |
| 电子商务网站 | 对商品的评分 |
| 音乐网站 | 对音乐/歌手/专辑的评分 |

来源: 虚构数据，仅作举例之用

显性反馈行为包括用户明确表示对物品喜好的行为。很多网站都使用了5分的评分系统来让用户直接表达对物品的喜好，但也有些网站使用简单的“喜欢”或者“不喜欢”按钮收集用户的兴趣。这些不同的显性反馈方式各有利弊。 YouTube最早是用5分评分系统收集显性反馈的，但后来他们的研究人员统计了不同评分的评分数，结果发现，用户最常用的评分是5分，其次是1分，其他的分数很少有用户打。因此，后来YouTube就把评分系统改成了两档评分系统（喜欢/不喜欢）。当然，我们举这个例子并不是试图说明一种评分系统比另一种好，而是要说明不同的网站需要根据自己的特点设计评分系统，而不是一味照搬其他网站的设计。YouTube的用户主要将精力放在看视频上，因此他们只有在特别不满或者特别满意时才会评分，因此二级评分系统就足够了。但如果是评论网站，用户主要将精力放在评论上，这时多级评分系统就是必要的。

评分预测问题的主要任务是预测一个给定用户对一个给定物品的评分。解决该问题主要依赖于建立用户的兴趣模型。传统的推荐系统在建立用户兴趣模型时往往忽略了用户兴趣的变化，而只建立用户的静态兴趣模型。比如，静态用户兴趣模型在计算一个用户对一个物品的评分时，往往只参考该用户对其他物品的评分，而忽视用户对其他物品评分的时间。但用户评分行为发生的时间信息对准确预测用户未来的行为有很重要的作用。

比如，用户最近的评分行为相对于用户很久之前的评分行为往往对预测用户未来的评分具有更高的参考价值。物品的受欢迎程度也受时间的影响，一部电影刚刚发布时受用户的欢迎程度和发布很久之后受用户的欢迎程度是不同的。此外，季节效应对评分预测问题也有很重要的作用，用户在不同的季节会有不同的兴趣，用户在平时和周末会有不同的兴趣，用户在不同的节日会有不同的兴趣。诸如此类的时间因素在各种类型的网站中都有很多，而如何利用这些时间信息提高推荐系统预测的精度，设计符合用户兴趣变化的动态推荐系统，是近几年推荐领域研究的热门问题。下面几节，我们将就如何针对评分预测问题建立用户兴趣的动态模型以提高评分预测的精度进行讨论。

本节的主要内容如下：

* 评分预测问题中最常用的模型是基于矩阵分解的模型(也被称为LatentFactor Model)。本章提出的动态模型也是建立在矩阵分解模型的基础之上的，因此本章将首先介绍时间无关的矩阵分解模型，以及模型优化的算法。
* 本节提出了3种主要的时间效应，包括用户偏好的变化效应，物品流行度的时间效应以及用户兴趣变化的效应，并利用矩阵分解模型将这三种时间效应建模到矩阵分解模型中。本章通过在Netflix用户电影评分数据集上的实验对比动态矩阵分解模型和时间无关模型的预测精度，证明了使用时间信息可以大大地提高评分预测的精度。

### 

对外统一的出错处理函数

根据错误的代号确定错误种类

根据错误的文件名以及行号确定

错误位置

打印错误信息

清空连接器前申请资源，退出连接器程序

图 3.1 流程图

## 问题描述

用户显性反馈数据集或用户评分数据集D = {( u, i, t, r)} 是四元组的集合。其中每一个四元组( u, i, t, r) ∈ D 代表用户 u在 t时刻给物品 i赋予评分 r。本文只考虑用户对物品只评过一次分的情况，即用户不会对一个物品反复评分。给定一个用户 u和物品 i，令 tui为用户 u给物品 i评分的时间，令 ruit为用户 u在 t时刻给物品 i评的分数。不同的网站有不同的评分体系，本章主要研究五分的评分体系，即 rui ∈ {1 , 2 , 3 , 4 , 5}。评分预测问题需要我们通过对用户历史评分数据集D进行学习，得到用户兴趣模型，对给定的用户 v ∈ U，物品 j ∈ I，预测用户 v在 t时刻对物品 j的评分 r v j t。对于任何一个预测算法，我们令

rˆv j t为算法对用户 v在 t时刻给物品 j的评分的预测值。

为了能离线的评测不同预测算法的效果，我们将数据集划分成训练集DA和测试集Dt，在训练集上训练动态用户兴趣预测模型，并在测试集上测试预测精度。本章使用均方误差(RMSE)作为评测指标，因此评分预测问题就转化为如何利用训练数据设计预测器，以最小化如下的测试集的预测误差：

为了在测试集上最小化上面的RMSE，可以通过在测试集上最小化RMSE来实现。因此，评分预测问题就转化为如果在训练集上设计预测器以最小化训练集上的预测误差。

## 推荐系统的矩阵分解模型

## 本章小结

评分预测问题是推荐系统的一个重要问题，它的主要任务是预测给定用户对给定物品的评分。用户行为和兴趣是随时间变化的，因此准确的评分预测算法需要考虑用户兴趣的动态模型。本章采用了基于矩阵分解的模型来融合用户行为的时间信息(TRSVD)。我们考虑了四种主要的时间效应：社会兴趣的变化（全局时间效应），用户偏好的变化（用户相关的时间效应），物品偏好的变化（物品相关的时间效应）以及用户兴趣的变化（用户物品同时相关的时间效应）。

# 推荐引擎设计与实现

## 推荐系统概述

推荐系统具体的可以分为三部分， 分别是推荐引擎（推荐子系统）， 日志系统， 展示系统； 系统通过展示系统收集用户的行为日志， 将用户行为日志做进一步的处理，并持久化到日志系统中； 而后由推荐引擎从日志系统中抓取数据，计算用户兴趣， 并为用户输出包含N件物品的推荐列表； 再由展示系统展示推荐列表；这是整个推荐系统的大体架构。

本节的主要部分在于：推荐引擎的架构设计与实现；

* 该部分负责从数据库或者缓存中拿到用户行为数据，通过分析不同行为，生成当前用户的特征向量。不过如果是使用非行为特征，就不需要使用行为提取和分析模块了。该模块的输出是用户特征向量。
* 该部分负责将用户的特征向量通过特征-物品相关矩阵转化为初始推荐物品列表。
* 该部分负责对初始的推荐列表进行过滤、排名等处理，从而生成最终的推荐结果。

## 系统需求分析和技术架构

### 系统目标

推荐系统需要由多个推荐引擎组成，每个推荐引擎负责一类特征和一种任务，而推荐系统的任务只是将推荐引擎的结果按照一定权重或者优先级合并、排序然后返回结果；这样做还有两个好处。

1. 可以方便地增加/删除引擎，控制不同引擎对推荐结果的影响。对于绝大多数需求，只需要通过不同的引擎组合实现。
2. 可以实现推荐引擎级别的用户反馈。每一个推荐引擎其实代表了一种推荐策略，而不同的用户可能喜欢不同的推荐策略。有些用户可能喜欢利用他的年龄性别作出的推荐，有些用户可能比较喜欢看到新加入的和他兴趣相关的视频，有些用户喜欢比较新颖的推荐，有些用户喜欢专注于一个邻域的推荐，有些用户喜欢多样的推荐。我们可以将每一种策略都设计成一个推荐引擎，然后通过分析用户对推荐结果的反馈了解用户比较喜欢哪些引擎推荐出来的结果，从而对不同的用户给出不同的引擎组合权重。

### 系统功能性需求描述

#### 生成用户特征向量

一般来说，用户的特征包括两种，一种是用户的注册信息中可以提取出来的，主要包括用户的人口统计学特征。对于使用这种特征的推荐引擎，如果内存够，可以将存储这些特征的信息直接缓存在内存中，在推荐时直接拿到用户的特征数据并生成特征向量。除了这种特征，另一种特征主要是从用户的行为中计算出来的。

一个特征向量由特征以及特征的权重组成，在利用用户行为计算特征向量时需要考虑以下因素。

* **用户行为的种类** 在一个网站中，用户可以对物品产生很多不同种类的行为。用户可以浏览物品、单击物品的链接、收藏物品、给物品打分、购买物品、评论物品、给物品打上不同的标签、和好友分享物品、搜索不同的关键词等。这些行为都会对物品特征的权重产生影响，但不同行为的影响不同，大多时候很难确定什么行为更加重要，一般的标准就是用户付出代价越大的行为权重越高。比如，购买物品需要用户掏钱，所以用户一定会三思而后行，因此购买行为最为重要。相反，浏览物品的网页代价很小，所以这种行为对反映用户的真实兴趣的影响很小。
* **用户行为产生的时间** 一般来说，用户近期的行为比较重要，而用户很久之前的行为相对比较次要。因此，如果用户最近购买过某一个物品，那么这个物品对应的特征将会具有比较高的权重。
* **用户行为的次数** 有时用户对一个物品会产生很多次行为。比如用户会听一首歌很多次，看一部电视剧的很多集等。因此用户对同一个物品的同一种行为发生的次数也反映了用户对物品的兴趣，行为次数多的物品对应的特征权重越高。
* **物品的热门程度** 如果用户对一个很热门的物品产生了行为，往往不能代表用户的个性，因为用户可能是在跟风，可能对该物品并没有太大兴趣，特别是在用户对一个热门物品产生了偶尔几次不重要的行为（比如浏览行为）时，就更说明用户对这个物品可能没有什么兴趣，可能只是因为这个物品的链接到处都是，很容易点到而已。反之，如果用户对一个不热门的物品产生了行为，就说明了用户的个性需求。因此，推荐引擎在生成用户特征时会加重不热门物品对应的特征的权重。

#### 特征-物品相关推荐

在得到用户的特征向量后，我们可以根据离线的相关表得到初始的物品推荐列表。离线相关表可以存储在MySQL中。

表格 2特征-物品相关推荐表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Src\_id | Reco\_id | Item\_id | Weight | Add\_time |
| 特征ID | 算法ID | 物品ID | 权重 | 时间戳 |

在线使用的特征物品相关表一般都不止一张。以论文之间的相关表为例，计算论文之间的相关性既可以使用第3章提出的协同过滤算法（即如果两篇论文的读者重合度很大说明两部电视剧相似），也可以通过内容计算（比如有相同的作者、关键词、相似的标题等）。即使是协同过滤，也可以根据不同的用户行为数据得到不同的相关表。比如可以根据用户的打分行为计算论文之间的相关性，也可以根据用户的浏览行为计算论文之间的相关性。总之，对于一个推荐引擎可以在配置文件中配置很多相关表以及它们的权重，而在线服务在启动时会将这些相关表按照配置的权重相加，然后将最终的相关表保存在内存中，而在给用户进行推荐时，用的已经是加权后的相关表了。

特征—物品相关推荐模块除了给用户返回物品推荐列表，还需要给推荐列表中的每个推荐结果产生一个解释列表，表明这个物品是因为哪些特征推荐出来的。

#### 过滤模块

在得到初步的推荐列表后，还不能把这个列表展现给用户，首先需要按照产品需求对结果进行过滤，过滤掉那些不符合要求的物品。一般来说，过滤模块会过滤掉以下物品。

* **用户已经产生过行为的物品** 因为推荐系统的目的是帮助用户发现物品，因此没必要给用户推荐他已经知道的物品，这样可以保证推荐结果的新颖性。
* **候选物品以外的物品** 候选物品集合一般有两个来源，一个是产品需求。比如在首页可能要求将新加入的物品推荐给用户，因此需要在过滤模块中过滤掉不满足这一条件的物品。另一个来源是用户自己的选择，比如用户选择了某一个价格区间，只希望看到这个价格区间内的物品，那么过滤模块需要过滤掉不满足用户需求的物品。
* **某些质量很差的物品** 为了提高用户的体验，推荐系统需要给用户推荐质量好的物品，那么对于一些绝大多数用户评论都很差的物品，推荐系统需要过滤掉。这种过滤一般以用户的历史评分为依据，比如过滤掉平均分在2分以下的物品。

#### 排名模块

经过过滤后的推荐结果直接展示给用户一般也没有问题，但如果对它们进行一些排名，则可以更好地提升用户满意度，一般排名模块需要包括很多不同的子模块，下面将对不同的模块分别加以介绍。

1. 时间多样性

时间多样性主要是为了保证用户不要每天来推荐系统都看到同样的推荐结果。在第5章已经提到，提高推荐系统的时间多样性要从两个地方着手。首先要保证推荐系统的实时性，在用户有新行为时实时调整推荐结果以满足用户最近的需求。这一点，在本章的推荐系统设计中已经考虑到了。如果用户有实时行为发生，那么行为提取和分析模块就能实时拿到行为数据并转化为新的特征，然后经过特征-物品相关模块转换成和新特征最相关的物品，因而推荐列表中就立即反应了用户最新行为的影响。提高推荐结果多样性的第二个方面是要在用户没有新的行为时，也要保证推荐结果每天都有变化。要实现这一点，只能通过如下方式。

* 记录用户每次登陆推荐系统看到的推荐结果。
* 将这些结果发回日志系统。这种数据不需要实时存储，只要能保证小于一天的延时就足够了。
* 在用户登录时拿到用户昨天及之前看过的推荐结果列表，从当前推荐结果中将用户已经看到的推荐结果降权。

1. 用户反馈

排名模块最重要的部分就是用户反馈模块。用户反馈模块主要通过分析用户之前和推荐结果的交互日志，预测用户会对什么样的推荐结果比较感兴趣。

如果推荐系统的目标是提高用户对推荐结果的点击率，那么可以利用点击模型（ click model）预测用户是否会点击推荐结果。点击模型在很多领域得到了广泛应用，比如搜索结果的点击预测、搜索广告的点击预测、上下文广告的点击预测。点击预测的主要问题是预测用户看到某个推荐结果时是否会点击。那么要进行点击率预测，首先需要提取特征。在推荐系统的点击率预测中可以用如下特征预测用户 u会不会点击物品i：

* 用户 u相关的特征，比如年龄、性别、活跃程度、之前有没有点击行为；
* 物品i相关的特征，比如流行度，平均分，内容属性；
* 物品i在推荐列表中的位置。用户的点击和用户界面的设计有很高的相关性，因此物品i在推荐列表中的位置对预测用户是否点击很重要；
* 用户之前是否点击过和推荐物品i具有同样推荐解释的其他推荐结果；
* 用户之前是否点击过和推荐物品i来自同样推荐引擎的其他推荐结果。

点击模型需要离线计算好，在线将模型加载到内存中。为了提高在线预测的效率，一般只可以使用线性模型。

### 系统非功能性需求描述

1. 实时性； 前面说用户的兴趣可分为长期兴趣， 短期兴趣； 特别是短期兴趣是不断发生变化， 若系统不能做到实时性那么系统将会出现推荐结果不可靠， 或者并不是用户感兴趣的。
2. 准确性； 这个主要依赖于推荐算法的选择和对用户具体场景的判断。所以推荐排名的策略就应该更加智能。

### 系统总体架构和J2EE技术概述

数学公式一般另行起排，居中书写，并用阿拉伯数字分章编号。若数学公式前有文字（如"解"、"假定"等），文字空两格写，数学公式仍居中写。数学公式序号按章编排，序号加圆括号，右顶格排。如第1章第1个数学公式序号为"公式（1.1）"。文中引用数学公式时，一般用"见公式（1.1）"或"公式（1.1）"。

#### 架构目标

#### 总体设计框架

## 系统功能设计

### 用户特征模块

#### 得到人口统计学中的行为特征

根据用户的ID从用户注册信息表中抓取用户的特征，和用户特征的权值， 这个权值可通过枚举类进行配置。

#### 计算用户行为特征

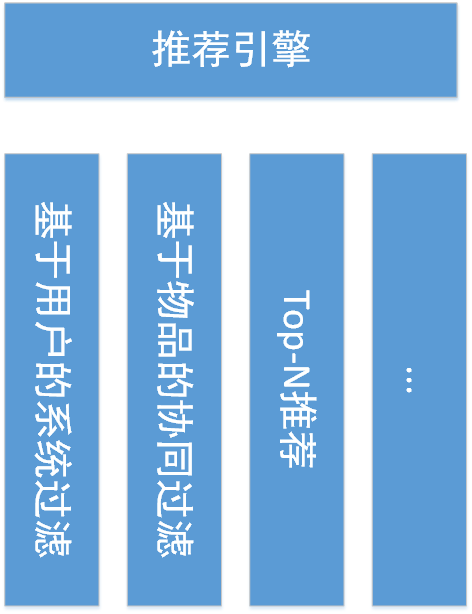
根据用户行为的种类， 产生时间（可选）， 行为的次数（可选）， 热门程度（可选）作为传入参数考虑， 返回用户的行为记录， 然后计算用户的行为特征；

## 用户引擎模块

### 推荐引擎结构图

如果要在一个系统中把上面提到的各种特征和任务都统筹考虑，那么系统将会非常复杂，而且很难通过配置文件方便地配置不同特征和任务的权重。因此，推荐系统需要由多个推荐引擎组成，每个推荐引擎负责一类特征和一种任务，而推荐系统的任务只是将推荐引擎的结果按照一定权重或者优先级合并、排序然后返回结果。

图 6推荐引擎结构图



### 推荐算法类图

## 过滤和排名模块

根据商品的打分情况， 还有用户是否购买过此商品来过滤最初的推荐列表， 然后对列表进行排名；

## 系统非功能性设计

## 本章小结

# 用户行为日志系统设计与实现

## 第一节

### 

## 本章小结

### 

# 展示系统设计与实现

## 第一节

### 

## 本章小结

### 

# 

## 第一节

### 

## 本章小结

### 

# 

## 第一节

### 

## 本章小结

### 

参考文献

1. Wiki： http://zh.wikipedia.org/wiki/%E9%95%BF%E5%B0%BE [↑](#footnote-ref-1)
2. []James Davidson, Benjamin Liebald, Junning Liu, Palash Nandy, Taylor VanVleet, Ullas Gargi, Sujoy Gupta, Yu He, Mike Lambert, Blake Livingston, and Dasarathi Sampath. The youtube video recommendation system. In Proceedings of the fourth ACM conference on Recommender systems, RecSys’10, pages 293–296, New York, NY, USA, 2010. ACM [↑](#endnote-ref-1)
3. []Greg Linden, Brent Smith, and Jeremy York. Amazon.com recommendations: Item-to-item collaborative filtering. IEEE Internet Computing, 7:76–80, January 2003. [↑](#endnote-ref-2)
4. Greg Linden, Brent Smith, and Jeremy York. Amazon.com recommendations: Item-to-item collaborative filtering. IEEE Internet Computing, 7:76–80, January 2003. [↑](#endnote-ref-3)
5. 数据来源××××× [↑](#footnote-ref-2)
6. Francois Fouss, Alain Pirotte, Jean-Michel Renders, and Marco Saerens.Random-walk computation of similarities between nodes of a graph with application to collaborative recommendation. IEEE Trans. on Knowl. and Data Eng., 19:355–369, March 2007. [↑](#endnote-ref-4)
7. Liang Xiang, Quan Yuan, Shiwan Zhao, Li Chen, Xiatian Zhang, Qing Yang, and Jimeng Sun. Temporal recommendation on graphs via long- and shortterm preference fusion. In Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining, KDD ’10, pages 723–732, New York, NY, USA, 2010. ACM. [↑](#endnote-ref-5)