

PaperPass检测报告简明打印版

比对结果（相似度）：

总体：41 %（总体相似度是指本地库、互联网的综合比对结果）

本地库：26 %（本地库相似度是指论文与学术期刊、学位论文、会议论文数据库的比对结果）

互联网：25 %（互联网相似度是指论文与互联网资源的比对结果）

编号：VIP95AD2FAAEDBDFA2E0

标题：推荐系统

作者：李朝阳

长度：25007 字符(不计空格)

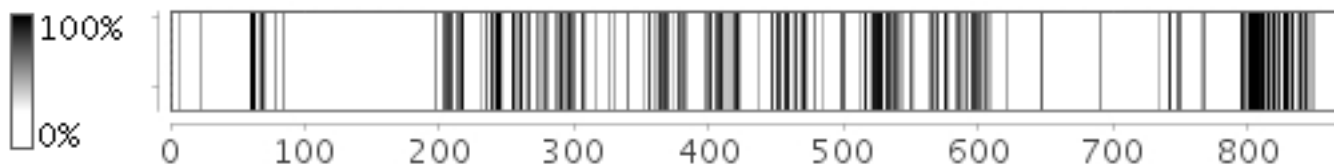
句子数：868句

时间：2015-4-13 22:34:22

比对库：学术期刊、学位论文（硕博库）、会议论文、互联网资源

查真伪：<http://www.paperpass.com/check.aspx>

句子相似度分布图：



本地库相似资源列表（学术期刊、学位论文、会议论文）：

- 相似度：8 % 篇名：《基于用户关系链的微博收听推荐系统设计与实现》
来源：学位论文 华中科技大学 2013 作者：汪晔
- 相似度：4 % 篇名：《个性化文档推荐系统的设计与实现》
来源：学位论文 华中科技大学 2013 作者：杨玲
- 相似度：3 % 篇名：《基于协同过滤模型与隐语义模型的推荐系统研究与实现》
来源：学位论文 湖南大学 2013 作者：鲁权
- 相似度：3 % 篇名：《协同过滤算法在移动电子商务推荐系统中的应用研究》
来源：学位论文 厦门大学 2013 作者：杨波
- 相似度：3 % 篇名：《结合非负矩阵分解的推荐算法及框架研究》
来源：学位论文 北京交通大学 2012 作者：周子亮
- 相似度：3 % 篇名：《企业信息安全风险分析及对策》
来源：学术期刊 《农业网络信息》 2010年7期 作者：王军英 马国青
- 相似度：2 % 篇名：《电子商务个性化推荐系统的设计与实现》
来源：学位论文 河南大学 2013 作者：李晓丽
- 相似度：2 % 篇名：《推荐系统交互性研究综述》
来源：学术期刊 《软件》 2013年8期 作者：杨淑梅

9. 相似度：2 % 篇名:《推荐技术的比较研究》
来源：学术期刊 《计算机光盘软件与应用》 2011年19期 作者: 吴振慧
10. 相似度：2 % 篇名:《基于用户标注动机与遗忘曲线的个性化推荐研究》
来源：学位论文 重庆大学 2013 作者: 陈曦
11. 相似度：2 % 篇名:《基于Linux的主机入侵检测系统设计》
来源：学位论文 河北科技大学 2009 作者: 马菊芳
12. 相似度：2 % 篇名:《奢侈品电子商务推荐的协同过滤算法研究与应用》
来源：学位论文 复旦大学 2012 作者: 李文刚
13. 相似度：2 % 篇名:《话语阴影下的个人悲剧——再评悲剧英雄威利 娄曼》
来源：学位论文 浙江大学 2010 作者: 韩会艳
14. 相似度：2 % 篇名:《中法电视访谈对比研究》
来源：学位论文 浙江大学 2010 作者: 陈锋
15. 相似度：2 % 篇名:《中欧贸易纠纷中的中国媒体话语策略研究：以文化话语学为视角》
来源：学位论文 浙江大学 2010 作者: 陈肖
16. 相似度：2 % 篇名:《论罗斯金自然观和文化观的相互融合关系》
来源：学位论文 浙江大学 2010 作者: 陈姗姗
17. 相似度：1 % 篇名:《德国诗人约阿希姆 林格尔那茨及其诗歌《库特尔 达德尔杜》,德国当前研...》
来源：学位论文 浙江大学 2010 作者: 赵建晖
18. 相似度：1 % 篇名:《中国背景下德语作为外国语课堂中语言游戏的使用--对教师和学生态度的实...》
来源：学位论文 浙江大学 2013 作者: 奥莱斯雅
19. 相似度：1 % 篇名:《A Proposed Event Schematic Approac...》
来源：学位论文 浙江大学 2012 作者: 程锦
20. 相似度：1 % 篇名:《中国的非洲政策》
来源：学位论文 浙江大学 2013 作者: 加达
21. 相似度：1 % 篇名:《改写理论视角下葛氏夫妇英译《尘埃落定》研究--兼谈对中国文学“走出去...》
来源：学位论文 浙江大学 2013 作者: 曲夏瑾
22. 相似度：1 % 篇名:《关于Leibniz流形的研究》
来源：学位论文 新疆师范大学 2007 作者: 张福娥
23. 相似度：1 % 篇名:《辛超流形和Poisson超流形上相关问题的讨论》
来源：学位论文 新疆师范大学 2007 作者: 曾辉
24. 相似度：1 % 篇名:《超图的边着色》
来源：学位论文 新疆师范大学 2008 作者: 王娜
25. 相似度：1 % 篇名:《多维度可扩展推荐引擎框架的设计与应用》
来源：学位论文 哈尔滨工业大学 2013 作者: 刘静
-

互联网相似资源列表：

1. 相似度：18 % 标题：《博士论文：动态推荐系统_文档资料库》
<http://www.03964.com/read/cc4202ad50c84a28a25e019e.html>
2. 相似度：7 % 标题：《架构设计：一种远程调用服务的设计构思 - 51CTO.COM》
<http://developer.51cto.com/art/201309/409526.htm>
3. 相似度：5 % 标题：《项亮博士论文69-第13页》
<http://3y.uu456.com/bp-7q0bbb4ssqeef8c7sfbfb3c7-13.html>

4. 相似度：2 % 标题：《非常好的协同过滤入门文章 - wentingtu - 博客园》

<http://www.cnblogs.com/wentingtu/archive/2011/12/16/2289926.html>

5. 相似度：1 % 标题：《推荐引擎材料》

<http://www.docin.com/p-584999160.html>

全文简明报告：

分类号： TP311.5

单位代码：

学 号：

硕士学位论文

中文论文题目： { 91 %：商品个性化推荐系统的设计与实现 }

英文论文题目： Product Design And implementation of personalized recommendation system

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

{ 100 %：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 :

{ 100 % : 浙江大学研究生学位论文独创性声明 }

{ 100 % : 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。 } { 99 % : 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。 } { 100 % : 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

{ 100 % : 学位论文版权使用授权书 }

{ 91 % : 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。 } { 92 % : 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。 }

{ 100 % : (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

{ 69 % : 推荐系统是解决信息过载的一种重要方式; } { 41 % : 它通过分析用户行为日志预测用户的喜好, 将来的行为以及兴趣等并为用户推荐物品. 随着 web2.0 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基于人口统计的推荐, 将而这些不能在再满足当下用户实时数据无法产生实时推荐的需求。 而且在以往的推荐系统中并没有关于整体推荐系统设计实践进行描述的文章。 因此本文基于公开用户行为数据集对推荐系统进行优化, 在原有算法的基础上添加了时间变量, 并验证算法的实效性。 本文的主要工作内容如下:

{ 69 % : 本文通过用户对物品的评分来预测用户对未评分物品进行评分。 } { 43 % : 即通过相关的方法提取用户特征, 计算用户物品的预测评分。 } 同时在考虑时间因素在预测中的影响, 判断用户的喜好程度不仅仅与物品的属性有关, { 52 % : 同时也与用户所处的上下文也有关系, 用户在不同的时间, 不同的地点喜欢的物品也是有区别的, } { 49 % : 而物品在不同的上下文中的流行度也是不同的。 }

相对于海量的物品, 用户购买的或者用户感兴趣的物品数量仅仅占总量的很少一部分。 { 47 % : 要从所有的物品中找到用户感兴趣的物品集合, 然后在对物品集合中未评分的物品进行评分。 } 本文分析了基于用户隐形数据上的Top-N推荐算法, 主要就是问了得到一个用户感兴趣的集合;

{ 43 % : 另一部分就是关于推荐系统的设计与实现, 其中包括推荐引擎, 日志系统, 展示系统。 } 推荐引擎

主要是算法的组织 and 算法的可扩展方面的设计，一个优秀的推荐系统只靠一个或者固定的几个推荐算法是不可能实现的，所以要考虑引擎的可扩展，还有就是单一业务方面的考虑；日志系统主要记录用户的行为日志（显性和隐形），还有一些用户规则如：30岁的互联网女性喜欢购买零食等；以及一些推荐结果；{59%：而展示系统主要分为两方面：} 用户行为日志的收集，通过调查问卷，物品打分以及商品浏览次数统计等方式；再有就是推荐列表的展示，远程接口提供第三方展示，插件展示等提供多种展示方式。

关键词：{49%：推荐系统，日志系统，展示系统，推荐引擎}

Abstract

Recommended system is an important way to solve the information overload. It's recommended for users log items by analyzing user behavior to predict the user's preferences, future behavior and interests, etc. With the development of web2.0, and increasing user involvement, But, the earlier recommendation system based primarily on content or recommendations based on demographics will not be able to meet the current user real-time data cannot be generated in real-time recommendation needs. In the past, and no information on the recommended system overall recommendation system design practices described in the article. Therefore, this paper based on publicly available data set of user behavior is recommended to optimize the system, including real time user dynamic recommendation.

The main contents of this paper are as follows:

Users of the goods were scored by knowing item's scores. By extracting features that users related methods to calculate user's predicted score goods. Also considering the time factor in the prediction of the determined level of user preference items is not only related to the attribute, but also with the context in which the user has a relationship, the user at different times, different places like article is also a difference, while the popularity of goods in a different context is different.

Relative to the mass of the goods, the number of items purchased by the user or the user is interested only accounts for a small part of the total. All goods in the collection to find goods of interest from the user, and then did not score for the goods in the collection of goods scored. This paper analyzes the Top-N recommendation algorithm based on user data on stealth, the main interest is to ask a set to get a user;

Relative to the mass of the goods, the number of items purchased by the user or the user is interested only accounts for a small part of the total. All goods in the collection to find goods of interest from the user, and then did not score for the goods in the collection of goods scored. This paper analyzes the Top-N recommendation algorithm based on user data on stealth, the main interest is to ask a set to get a user;

Key Words: Recommender System, Logging System, View System, Recommender Engine

目录

摘要i

Abstractii

图目录IV

表目录V

第1章 绪论6

1.1 课题背景与意义6

1.2 研究内容7

1.3 公开用户数据集合8

1.3.1 亚马逊(z.cn)数据集8

1.3.2 MovieLens电影评分数据集8

1.3.3 CiteULike论文书签数据集8

1.4 论文概要结构描述9

第2章 评分预测中用户物品特征的模型10

2.1 用户物品特征描述10

2.2 推荐系统的静态用户特征矩阵分解模型11

2.3 奇异值分解算法(SVD) 12

2.4 本章小结13

第3章 Top-N推荐用户兴趣预测的模型14

3.1 Top-N推荐简介14

3.2 问题数学定义和概括15

3.2.1 传统Top-N推荐15

3.3 基于时间的Top-N推荐15

3.3.2 隐性反馈数据的动态变化18

3.4 本章小结21

第4章 推荐系统需求分析22

4.1 推荐系统概述22

4.2 推荐引擎概述与需求分析22

4.2.1 系统目标22

4.2.2 系统功能性需求描述23

4.2.3 系统非功能性需求描述27

4.3 展示系统概述与分析27

4.3.1 展示系统需求分析27

4.4 日志系统概述与需求分析28

4.4.1 日志系统需求分析28

4.4.2 日志系统构成29

4.4.3 用户行为来源29

4.4.4 数据流程管理29

4.4.5 数据格式化输出29

4.4.6 数据持久化30

4.4.7 日志内容30

4.4.8 日志系统目标31

4.5 本章小结32

第5章 推荐系统设计与实现33

5.1 推荐引擎总体架构和J2EE技术概述33

5.1.1 架构目标33

5.1.2 整体设计框架33

5.1.3 推荐引擎功能设计34

5.1.4 用户特征模块34

5.1.5 推荐引擎结构图35

5.1.6 过滤和排名模块35

5.2 展示系统总体技术架构设计35

5.2.1 功能模块分布36

5.2.2 用户行为模块37

5.2.3 推荐结果/推荐理由展示38

5.2.4 远程接口与用户安全设计38

5.3 本章小结41

第6章 总结与展望42

6.1 推荐系统总结42

6.2 推荐系统展望42

参考文献43

图目录

图 1 1数据集所含各种类型物品数目8

图 2 1用户-物品评分矩阵12

{ 59 % : 图 3 1用户-物品关系二分图16 }

图 3 2 路径融合算法17

图 3 3 时间段图模型20

{ 66 % : 图 5 1推荐子系统架构图34 }

图 5 2推荐引擎结构图35

图 5 3视图系统的总体架构36

图 5 4展示系统功能模块36

图 5 5用户行为包设计37

表目录

{ 48 % : 表 2 1具有代表性的线性反馈数据10 }

{ 68 % : 表 4 1特征-物品相关推荐表24 }

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;

{ 77 % : 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。 } { 88 % : 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。 } { 52 % : 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。 }

{ 80 % : 搜索引擎可以通过关键词找到自己所需要的信息。 } { 72 % : 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。 } { 88 % : 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。 } { 76 % : 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。 } { 72 % : 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。 } { 76 % : 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。

{ 52 % : 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，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。 } { 68 % : 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。 } { 60 % : 但同时长尾的出现也给系统或者电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。 } { 49 % : 这种将用户和相关物品联系在一起的方式统称为个性化推荐。 }

{ 89 % : 个性化推荐系统，通过联系用户和信息，一方面帮助用户发现对自己有价值的信息，另一方面让信息能够展现在对他感兴趣的人群中， } { 100 % : 从而实现信息提供商与用户的双赢。 } 个性化推荐的实质就是解决长尾理论，而双11是国内最有名的电商促销活动。在双11大促会场上应用个性化与否，成交金额可以相差50%，这对一个对电商平台来说，绝不能忽视的增量。而个性化技术在双11中主要应用的地方有搜索个性化、推荐产品优化。在双11中如何里利用个性化推荐技术，提高消费者购物体验，增加长尾商家和商品的销售，提高平台的收益，成为一个非常关键的点。 { 48 % : 通过下面的内容，我们一起来探究这些问题。 }

研究内容

{ 51 % : 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: { 44 % : 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。

{ 48 % : 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: } { 87 % : 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, { 45 % : 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: { 78 % : 利用用户的行为数据计算用户间的相似性; } { 58 % : 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。 { 83 % : 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。 }

{ 92 % : 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。 } { 100 % : 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: { 100 % : 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。 } { 100 % : 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。 } { 100 % : 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。 } { 100 % : 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。 }

公开用户数据集

{ 83 % : 推荐系统中的用户数据集很多, 但提供了时间信息的数据集并不多。 } { 68 % : 本文主要应用了以下几个含有时间字段的数据集进行分析。 }

亚马逊 (z.cn) 数据集

从亚马逊网站上抓取的数据, 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。

{ 43 % : 图 11 数据集所含各种类型物品数目 }

MovieLens 电影评分数据集

{ 100 % : MovieLens 数据集中, 用户对自己看过的电影进行评分, 分值为 1~5。 } { 97 % : MovieLens 包括两个不同大小的库, 适用于不同规模的算法。小规模库是 943 个独立用户对 1682 部电影作的 10000 次评分的数据; } { 100 % : 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。 { 100 % : 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。 } { 100 % : 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。 } { 96 % : 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

{ 46 % : 本文除第一章之外可分为六部分 : }

{ 46 % : 第二章 描述过去的Top-N推荐, 以及改进之后的Top-N推荐 ; } { 53 % : 并分析了隐形反馈数据的动态变化。 }

{ 57 % : 第三章 通过预测算法预测给定用户对给定物品的评分 ; } { 100 % : 用目标问题所在的特定领域知识或者自动化的方法来生成、提取、删减或者组合变化得到特征。 } { 45 % : 再有就是计算用户物品评价预测值的方法介绍。 }

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

{ 57 % : 第五节 用户行为日志收集的设计与实现 }

第六章 推荐展示系统, 收集用户行为日志的接口设计以及为用户推荐展示的方式, 如网页展示, 远程调用等接口的设计与实现。

{ 70 % : 第七章 论文结束语以及展望 }

{ 54 % : 评分预测中用户物品特征的模型 }

用户物品特征描述

{ 91 % : 特征工程, 用目标问题所在的特定领域知识或者自动化的方法来生成、提取、删减或者组合变化得到特征。 } 一种可以利用领域知识生成和提取特征, 特征的提取一般通过和销售或者行业的专家进行商讨, 听取他们对特定业务的意见, 并提取相关的特性, 当然这个特性越多越好 ; { 91 % : 比如, 从购买数据里可以看出, 女性对衣服的品牌多数有固定偏好, 比如有个女性就很喜欢LOFT的衣服。 } { 100 % : 这些能对购买产生影响的因素都可以成为特征。 } { 54 % : 当然也可以通过机器学习的方式提取用户特征 ; } { 45 % : 本章主要是对简单用户商品交互特性进行建模 ; }

{ 63 % : 推荐系统的任务就是发现用户特征, 因此收集用户行为数据的系统也是推荐系统的组成部分。 } 一般通过让用户为物品评分来表达用户的兴趣, 同时用户评分行为被称为显性反馈行为, 指用户的行为明确指定倾向于指用户喜欢/不喜欢该物品。

{ 50 % : 表 2 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

{ 50 % : 门户网站对新闻的评分/评价等 }

{ 68 % : 电子商务网站对商品的评分 }

{ 45 % : 音乐网站对音乐/歌手/专辑的评分 }

{ 89 % : 显性反馈行为, 用户明确表示对物品喜好的行为。 } { 97 % : 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。 { 96 % : 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。 } { 77 % : 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)。 } { 40 % : 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。 } { 87 % : 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。 } { 97 % : 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。 }

{ 62 % : 评分预测问题的任务就是计算给用户对给定物品评分。 } { 76 % : 解决该问题主要依赖于建立用户特征模型。 } { 66 % : 传统的推荐系统在建立用户兴趣模型时一般基于静态的特征用户, 而只建立用户的静态特征模型。 }

{ 57 % : 比如, 用户近期评分行为对用户很久之前的评分行为往往具有更高的参考价值, 在对计算用户的评分的时候。 } { 45 % : 物品的流行程度也受发布时间或者更新时间的影响, 一部新发布的iPhone的之前发布的版本更加受欢迎。 } 此类的时间上下文因素在互联网中有很多, 而如何利用这些时间因素提高系统预测的精度, 设计符合用户物品特征变化的动态推荐系统, 是近年来推荐领域研究的热门问题。 { 61 % : 下面我们将就对评分预测问题建立用户物品特征模型提高评分计算的精度进行讨论。 }

本节的主要内容如下: { 85 % : 计算用户特征关系最常用的模型是基于矩阵分解的模型(也被称为Latent Factor Model)。 } { 84 % : 本章提出的相关模型也是建立在矩阵分解模型的基础之上的, 因此本节将首先介绍与时间上下文无关的矩阵分解模型。 }

{ 59 % : 推荐系统的静态用户特征矩阵分解模型 }

{ 40 % : 静态用户兴趣模型, 即与时间因素无关的用户兴趣特征模型, 矩阵分解模型也可以认为是对用户-物品二维模型的数据不全问题。 } { 46 % : 详细说明一下, 用户对物品的评分仅仅占据了用户, 物品的很小的比例。 } 无论是单个用户对物品的评价总数占物品总数的比例, 还是多个用户对一个物品的评价中多个用户与用户总数的比例; 这些都是很小的比例; { 56 % : 因此用户和物品组成的矩阵可以说是一个稀疏矩阵; } 我们现在的任务就是讲此稀疏矩阵所有坐标都给补全了。

问题数学定义

{ 49 % : 定义 $D=\{(u, i, r)\}$ 为用户 u 对物品 i 的评分的集合, 设有 N 名用户, M 件物品, 我们组成一个 $R^{N \times M}$ 的二维稀疏矩阵; } { 48 % : 在图2-1中我们看到一个简单的用户物品评分矩阵, 一行代表一个用户对所有物品的评分

，空格代表没有评分，也是需要我们计算的。} 比如，u1对i1，i4有评分，但是对于i2，i3，i5都没有评分。 { 50 %：下面我们通过奇异值分解算法来计算用户评分。 }

图 2 1用户-物品评分矩阵

奇异值分解算法 (SVD)

该算法的主要计算步骤是： 首先为每个缺失位置初始化一个初始值，比如一个用户对所有用户评分之和的平均值：

$$R(u, i) = (\sum r_{ui}) / (|N(i)|)$$

{ 55 %：其中N(i)代表用户对物品评价的个数； }

{ 49 %：或者也可以通过多个用户对一个物品的评分之和的平均值： }

$$R(u, i) = (\sum r_{ui}) / (|N(u)|)$$

{ 56 %：其中N(u)代表对物品i有评分的用户的个数。 } { 43 %：将评分不全之后的矩阵记为R' }

然后SVD将R' 分解为三个矩阵相乘：

$$R' = U \cdot T \cdot S \cdot V$$

其中U $R^{(K \times N)}$ ，V $R^{(K \times M)}$ ，S $R^{(k \times k)}$ 是三个分级或的矩阵。 S为对角矩阵，其对角线上是R' 矩阵的特征值。

然后在从R' { 55 %：中选取k个最大值组成对角阵Sk，同时也在V，U中找到k个最大值分别组成Uk，Vk。 } { 71 %：将三个新矩阵重新相乘，得： }

$$(R' = U_K \cdot S_K) \cdot V_K$$

其中R' 即为评分矩阵R的最终补全矩阵，R' (u, i)即为用户对物品的预测评分。

本章小结

本章主要是计算特定用户给明确的物品打分，通俗的说就是预测用户给某一物品的评分。 用户和物品之间没有直接的关系，但是用户有过评分记录，同时用户的好友也有评分记录，那么计算用户与物品（用户没有评分）之间的评分。

Top-N推荐用户兴趣预测的模型

Top-N推荐简介

{ 86 % : 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。 } { 72 % : 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; { 65 % : 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。 } { 91 % : 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。 { 48 % : 其中协同过滤是很多生产系统中的主要算法[,]。 }

{ 47 % : 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, 隐性反馈数据; } 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比较少; { 83 % : 而对于隐性反馈数据, 隐性反馈行为指的是那些不能明确反应用户喜好的行为。 } { 100 % : 最具代表性的隐性反馈行为就是页面浏览行为。 } { 72 % : 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。 } { 72 % : 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。 } { 75 % : 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。 } { 78 % : 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。 } { 90 % : 文献[]详细对比了隐性反馈数据和显性反馈数据的优点和缺点。 } { 100 % : 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。 } { 41 % : 本节主要描述利用隐性反馈数据来过滤和用户有关系的物品列表, 即Top-N推荐。 }

问题数学定义和概括

传统Top-N推荐

一般的, 已下的数据集 $Data = \{(user, item)\}$, 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。 { 80 % : 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件与 $user$ 最相关的物品。 }

基于时间的Top-N推荐

{ 67 % : 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。 } { 78 % : 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。 } { 78 % : 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: } { 50 % : 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。 }

{ 74 % : 推荐方法-基于图模型的协同过滤算法 }

{ 51 % : 利用二分图为用户行为建模。 } { 92 % : 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, } { 97 % : 若用户 u 对物品 i 有过行为, 就存在一条边 $e(v_u, v_i)$ E 连接用户 u 的节点 v_u 和物品 i 的节点 v_i , } 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 U_1, U_2, U_3 和 I_1, I_2, I_3, I_4 之间的关系。 { 60 % : 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 U_1 和 I_1, I_4 之间有行为; U_2 对 I_1, I_3, I_4 有行为; U_3 对 I_2, I_3 有行为。

{ 62 % : 图31用户-物品关系二分图 }

{ 47 % : 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。 { 56 % : 计算图中顶点之间距离的算法有很多 , Fouss在文[]中总结了很多 , 我们介绍一种路径融合算法[] }

路径融合算法

{ 46 % : 算法的基本思想为是 , 图上两个节点相似的判断标准是 : }

{ 70 % : 两顶点之间有许多条边相连 ; }

{ 55 % : 两顶点之间路径比较短 ; }

{ 69 % : 顶点之间路径不经过顶点出度值很大的点。 }

{ 83 % : 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。 } { 87 % : 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。 { 74 % : 图中一共有A , B , C , D四个用户和i1 , i2 , i3 , i4 4件物品。 } { 66 % : 其中用户 A喜欢物品 i1 , i3 , 用户 B喜欢物品 i1 , i3 , i4 , } { 72 % : 用户 C喜欢物品 i1 , i4 , 用户 D喜欢物品 i2 , i4。 } { 87 % : 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。 } { 76 % : 但用户A和物品i2 , i4之间的相似度却不同。 } { 90 % : 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。 } { 93 % : 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。 { 85 % : 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)。 { 83 % : 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。 } { 86 % : 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 , { 87 % : i4 , D , i2) , 也是因为B的出度值大于C的出度。 }

图 3 2 路径融合算法

{ 70 % : 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 ; } { 84 % : 第二条计算每条最短路径的权重 ; } { 91 % : 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。 }

{ 100 % : 下面将详细介绍路径融合算法。 }

{ 53 % : 首先计算两个节点之间的路径权重 , 假设P是连接V1 和 V2 之间的一条路径 , 而影响这条路径的权重的主要因素是 : } { 50 % : 两个顶点之间经过的顶点和边的数量。 } 数学定义 : $(V)^?$ $[0, 1]$ 为顶点V的权重 , $(v, v')^?$ $[0, 1]$ 为顶点v和v' 之间边的权重 ; 那么P的权重定义为 :

$$(P) = (V_n) \cdot \prod_{i=1}^{n-1} ((V_i)^? \cdot (V_i + V_{i+1})) / (|out(V_i)|) \quad (2.1)$$

其中 $(V)? [0, 1]$ 是顶点 V 的出度。从公式是得出当 n 越大，也就是边的数量越多，那么 P 的权重也就越小。 { 49 % : 同时如果 V 的出度越大，那么 P 的权重也就越小。 }

{ 43 % : 当我们可以计算一条路径的权重的时候，那么两点之间的相似度我们也是可以计算了。 } 数学定义：
{ 50 % : 计算定点 V_1 和 V_n 之间的相似度，定义 $p(V_1, V_n)$ 为 V_1 和 V_n 之间路径的集合。 } 那么 V_1 和 V_n 的相似度为：

$$d(V_1, V_n) = \frac{1}{|p(V_1, V_n)|} \sum_{P \in p(V_1, V_n)} \frac{1}{|P|} \quad (2.2)$$

{ 73 % : 在下文关于基于时间的 Top-N 推荐的问题，我们将使用图模型来将用户行为与时间信息同时建模，同时使用路径融合算法来为用户做推荐。 }

隐性反馈数据的动态变化

用户的长期兴趣和短期兴趣

{ 89 % : 用户的兴趣分为长期兴趣，短期兴趣。 } { 75 % : 比如一个武侠小说爱好者一般阅读武侠方面的文章。 } 但是，他也会关注实时的政治消息。在上述的场景中，武侠属于用户的长期兴趣，而政治新闻只是属于用户受社会热点影响而产生的兴趣，属于短期。 { 84 % : 由此可以看出，长期兴趣一般是用户自身的兴趣爱好产生的，而短期的兴趣一般是受其他方面因素影响而产生的。 } { 85 % : 一个高质量的推荐算法需要同时考虑用户的长期兴趣和短期兴趣来为用户做出推荐。 } { 72 % : 上节我们介绍了基于二分图模型的 Top-N 推荐算法，而这节将研究如何将用户的长期兴趣和短期兴趣全部加入建模到图模型中，并在此模型基础上设计适合该模型的推荐算法。 }

{ 85 % : 为了表示用户的短期兴趣，我们引入了用户的时间节点。 } { 75 % : 在前面的 Top-N 二分图模型中，用户 u 会和所有他产生兴趣的物品顶点通过带有权重的边连接。 } 但是用户不同时刻的兴趣不同。 { 71 % : 我们用 V_{ut} 节点表示用户 u 在 t 时间段的兴趣。 } { 100 % : 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。 } { 100 % : 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。 }

{ 85 % : 引入时间因素，用户物品二分图模型转化为时间段图模型 (Session based Graph Model (SGM))。 } SGM 是一个二分图，定义为 $G(U, S, I, E, w)$ 。 { 78 % : 其中 S 是用户时间节点的集合， U 是用户节点的集合， I 是物品节点的集合， $w: E \rightarrow R$ 定义了边的权重。 } { 89 % : 图4是一个 SGM 的简单例子，图中包括2个用户节点，4个 session 节点和3个物品节点。 } { 81 % : 图4表明，用户 A 对物品 i_1, i_2 有过行为，其中 A 在1时段对物品 i_1 有行为， } { 85 % : A 在2时段对物品 i_2 有行为，用户 B 对物品 i_1, i_2, i_3 有行为，其中 B 在1时段对物品 i_1 有行为， } 在2时段对物品 i_2, i_3 有行为。

{ 59 % : 在时间段图模型中，用户节点 v_u 将和用户 u 曾经发生行为的所有物品 $N(u)$ 对应的节点都有边相连，这表示了用户的长期兴趣。 } { 47 % : 而用户时间节点 V_{ut} 会和用户在 t 时刻有发生行为的所有物 $N(u, t)$ 对应的节点有边相连。 } { 73 % : 因此，如果从用户节点 v_u 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。 } { 98 % : 而如果从用户时间节点 V_{ut} 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。 } { 100 % : 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。 } { 100 % : SGM 中一共有三种不同的节点，它们的权重定义如下： }

$$(v)=\{ (1v \quad I@ \quad v \quad U@1- \quad v \quad S) \quad (2.3)$$

{ 47 % : 其中 $[0, 1]$ 控制长期兴趣和短期兴趣来影响推荐结果。 }

图 3 3 时间段图模型

基于SGM的路径融合推荐算法

{ 79 % : 本节主要讨论如何在时间段图模型上计算用户对物品的兴趣,从而设计个性化推荐算法。 } { 63 % : 在2.2.2.2节中介绍了基于用户物品二分图模型上的个性化排名推荐算法, } { 58 % : 本节将对路径融合算法稍作修改,让他们能够运行在时间段图模型上。 }

基于SGM的路径融合算法主要思想是: { 52 % : 考虑时间因素在计算用户物品的相似度的时候; } 例如下面这个简单的例子,

{ 50 % : 在图3-3中,计算用户A在2时段和物品i3的相似度。 } { 48 % : 如果考虑用户A到物品i3的长期兴趣,那么首先就要计算 $d(A, \{ i3 \})$ 的相似度,然后再计算A在2时段的相似度,最后将长期兴趣和短期兴趣相加即可得到 $s(A2, \{ i3 \})$ 的相似度。

计算过程如下:

计算A和i3之间的相似度:

$$d(A, i3) = (A, i1, B, i3) + (A, i2, B, i3) + (A, i2, B2, i3) \quad (2.4)$$

{ 42 % : 其中 (p) 见上节公式,下面计算A在时段2的时候对物品i3的相似度: }

$$d(A2, i3) = (A2, i2, B2, i3) + (A2, i2, B, i3) \quad (2.5)$$

{ 46 % : 最后计算A在2时段下对物品i3的相似度(是否感兴趣): }

$$s(A2, i3) = d(A, i3) + d(A2, i3) \quad (2.6)$$

{ 63 % : 一般来说,给定用户U和物品I,通过路径融合算法计算用户和物品之间的相似度也就是用户U对物品I的感兴趣程度,计算公式如下: }

$$s(u, i) = d(u, i) + d(u_t, i) \quad (2.7)$$

其中 $d(u, i)$ 为相似度。

本章小结

{ 72 % : 本章主要讨论在Top-N推荐中考虑用户的短期和长期兴趣,从而利用二分图模型提高推荐结果准确度。 } { 72 % : 首先开头简单描述了老的用户物品二分图模型以及基于该模型的算法。 } { 73 % : 在介绍了相关研

究的基础上，本章提出了基于SGM路径的路径融合算法，该算法通过分析节点与节点之间的路径来衡量节点之间的相似度。} {71%：然后，我们通过用户的长期和短期兴趣，并通过在二分图模型上加入一种新的节点来为这两种兴趣进行建模。} 从而提高Top-N推荐的准确度。

推荐系统需求分析

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；

{93%：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。

{88%：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。}

{92%：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。}

推荐引擎概述与需求分析

系统目标

{91%：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。

{100%：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。} {87%：对于绝大多数需求，只需要通过不同的引擎组合实现。}

{94%：可以实现推荐引擎级别的用户反馈。} {92%：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。} {94%：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，} {100%：有些用户喜欢比较新颖的推荐，有些用户喜欢专注于一个领域的推荐，有些用户喜欢多样的推荐。} {91%：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，} {100%：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

{90%：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。} {89%：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。} {75%：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。}

{92%：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。}

{67%：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。} {56%：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。} {76%：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，大多数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。} {90%：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。} {41%：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。}

{95%：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。} {46%：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。}

{54%：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。 {90%：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。}

{45%：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。 {55%：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。} {94%：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

{87%：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。

{77%：表41特征-物品相关推荐表}

Src_idReco_idItem_idWeightAdd_time

{45%：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）。 {40%：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。} {42%：比如可以根据用户的打分为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。} {64%：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，} {65%：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。}

{ 93 % : 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

{ 91 % : 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。 } { 100 % : 一般来说, 过滤模块会过滤掉以下物品。 }

{ 59 % : 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。 { 44 % : 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。 }

{ 97 % : 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。 } { 91 % : 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

{ 96 % : 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, } { 52 % : 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。 }

时间多样性

{ 44 % : 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。 { 74 % : 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。 } { 46 % : 这一点, 在本章的推荐系统设计中已经考虑到了。 } { 82 % : 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, } { 75 % : 因而推荐列表中就立即反应了用户最新行为的影响。 } { 61 % : 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。 } { 42 % : 要实现这一点, 只能通过如下方式。 }

{ 84 % : 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。 { 45 % : 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。 }

{ 96 % : 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。 }

用户反馈

{ 80 % : 排名模块最重要的部分就是用户反馈模块。 } { 100 % : 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。 }

{ 66 % : 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。 } { 68 % : 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。 } { 100 % : 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。 } { 88 % : 那么要进行点击率预测, 首先需要提取特征。 } { 61 % : 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 : }

{ 69 % : 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 ; }

{ 73 % : 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。 { 85 % : 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 ; }

{ 74 % : 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 ; }

{ 72 % : 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。 }

{ 55 % : 点击模型需要离线计算好, 在线将模型加载到内存中。 } { 44 % : 为了提高在线预测的效率, 一般只可以使用线性模型。 }

{ 87 % : 系统非功能性需求描述 }

实时性 ; { 75 % : 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。

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。

展示系统概述与分析

{ 43 % : 本系统的主要功能是推荐单子的展示, 用户行为数据的收集 ; } 推荐单子的展示不仅仅以网页的形式展现, 而且提供 web service 服务, RSS 推送等功能 ;

展示系统需求分析

用户行为收集

推荐只有拿到了用户的行为数据, 才可能进行推荐, 而用户行为从反馈的种类可分为显性和隐形, 显性数据一般能明确表达用户的兴趣, 需要持久化 ; { 100 % : 而隐形数据表达用户喜好程度较弱, 一般在内存中存储。 } 如当天在站内浏览物品的信息, 推荐结果只需要知道你浏览了什么, 然后根据浏览的物品即可做出推荐。 {

45 %：所以隐形数据在现阶段不做存储。}

推荐理由/推荐结果生成

每一种推荐都需要一个推荐的理由，也可以说没有推荐理由的推荐不是好推荐。这里暂时提供两种：1.获取推荐的具体好友信息，2.通过历史浏览记录得到的推荐结果或者理由。

远程服务

为第三方接入推荐列表提供接口服务，第三方主要是推荐结果的展示，和推荐结果反馈信息的收集。获取反馈信息分析和产出推荐规则，如IT行业的从业人员喜欢购买计算机类书。然后根据规则精确推荐结果。

安全服务

用户信息在传输过程中很容易受到攻击，为了保证用户信息的安全，在部分用户信息传输上会采用https协议进行传输；{45 %：当然在采集用户的行为数据之前，肯定是取得用户同意的。}

日志系统概述与需求分析

现阶段比较流行的网站一般都有以下特性：{44 %：访问量大，网站结构复杂，规模大等；}而且性能要求比较高，而一般用到推荐的系统往往是信息的规模达到了一个量级，仅仅几个页面不能再满足给所有用户展示所需要的信息了。{52 %：一个完善的用户收集系统，是必不可少的一部分。}它可以帮助你监控网站流量，系统是否正常，用户的行为变化等信息，我们主要是对用户行为的监控，分析，利用。当然系统级别的也是需要的，而且也有必要，保证网络的正常，系统出错提醒等。

日志系统需求分析

{57 %：日志系统的主要解决的问题是记录用户的行为日志，然后进行跟踪分析用户行为，审计用户的偏好。}在可靠的推荐系统中，用户的行为是保证成功记录下来的，不会出现遗漏的情况。{51 %：日志系统从内容方面可以分为两大类，一是业务级别的日志，也就是本系统的主要功能，主要供终端用户来分析他们业务过程，从日志数据中分析具体的业务；}{95 %：二是系统级别的日志，供开发者维护系统的稳定。}

{40 %：用户的行为日志量中不但有显性反馈数据，而且还有用户的隐形数据，比如用户点击物品的次数，}在站内停留时间等，这些数据造成日志系统的数据输出量成倍增加，从而影响整个系统的性能。如果从数据价值方面考虑的话，并不是所有的数据都是有价值的，而这些数据也增加日志信息检索的困难。

{67 %：好的日志系统的设计，必须解决以下几个方面：}

数据便于检索，可以通过SQL或者数据查询语言快速计算所需数据；

{43 %：用户行为日志数据层次清晰，组织合理。}

可控可扩展，允许用户动态修改，特别是线上生产情况下。比如新增统计某个信息提交的次数。

日志系统构成

{ 42 % : 根据具体的业务需求 , 日志系统主要分为以下几块 : } { 41 % : 用户行为来源 , 数据流程管理 , 数据格式化输出 , 数据持久化 ; }

用户行为来源

用户的行为数据可以网页中任何元素,任何内容,也可以是系统中任何页面,这里面有各种类型的用户数据,比如点击,评价,浏览次数等等,如何有效的组织这些数据,这就需要一种规则来规范数据的形式了,如我们业务仅仅需要你浏览了某一商品的次数,那么系统就收集次数。其本质就是日志系统制定规则来规范数据,过滤数据。 { 58 % : 以便系统对数据进行管理和使用。 }

数据流程管理

业务需要一件商品一天内的用户浏览量,这是我们就要控制系统在收集该类数据的时候要控制时间是24小时。数据流程管理的主要工作就是控制数据收集的时间,地点,收集的格式等等。其本质就是控制数据在系统内的执行过程。类似于Linux的任务管理。

数据格式化输出

最常见的输出方式就是从控制台输出了,无论我们学习哪种语言,第一节的内容肯定是在控制台输出 Hello World,但是生产中的输出,不仅仅是控制台输出,当然在调试的时候我们也是需要控制台输出的,但是如果我们不控制信息的级别那么 info类型的, warning类型的, error类型的都输出到控制台,对于我们 debug也是不利的。格式化的本质就是将数据按一定的方式组织在一起,以方便用户的存储,分析等。

数据持久化

{ 40 % : 用户数据根据业务的不同,对持久化的要求也是不同的,有些数据组织形似方便存储在关系型数据库中, } 而 JSON类型的数据就不那么的适合了,这时候就会选择 NOSQL类型的数据库了,比如 MongoDB; 还有就是考虑数据的大小,还有是否实时这些因素,这时就要用到内存数据库或者缓存了。

日志内容

内容分级

{ 40 % : 日志系统输出的数据不仅仅包含用户行为日志,还有系统日志,应用程序日志等, } 若想更好的管理这些日志内容,我们需要将数据进行分级,通过一个全局常量来控制,如果日志输出级别是用户行为级别,那么记录,否则放弃忽略不计。从而来控制数据的输出。也可以根据日志级别配置日志输出路径,从而将数据分流。

用户行为分类

当用户行为数据多到一定程度的时候就需要将其分类管理了,否则随着数据量的增长,查询所消耗的时间也会增加,进而影响系统的正常运行,从用户行为主动或被动角度来考虑, { 63 % : 可将数据分为显性反馈数据

，隐形反馈数据；} {51%：从对物品感兴趣和不感兴趣可将其分为正反馈，负反馈；} {43%：从数据的产生途径可分为元数据，中间值，结果这三类。}

内容格式

之前我们对内容进行了分级和分类，但是对于用户行为数据来说，粒度还是比较粗糙。我们还要根据业务需求来更加详细的划分内容，因为只有大量的用户行为数据才有统计的意义，但是大量的无用数据会造成极大的系统负担。这对于系统来说是致命的。{43%：还有就是数据的多样性，我们可以通过以下方式来解决：}

用XML格式来存储，这样可以解决数据多样性的问题，但是同时也带来了解析数据时吓人的时间复杂度。在这通过JSON来存储这些数据，不仅方便解析，而且现在很多网站在网际之间使用该格式来传递消息的。如果以上还是不能满足需求那么就根据具体的业务设计一个新的数据格式来处理这种数据，一般前两种方式就可以满足大部分需求了。

日志系统目标

易用性

{50%：易用性，是每一个软件都在追求的特性。} 易用的目的是为人提高效率。我们使用计算机的根本原因就是它能够让很多工作变得简单以及为我们节省时间，否则它就没有被我们使用的价值了。我们可以这样说，大多软件特性都是易用性的延伸，比如，国际化、扩展性以及高效性等。

性能

推荐系统如果需要实时推荐的时候，那么系统的相应时间不能超过3秒，如果纯粹从系统性能的方面来考虑，日志是不应该存在的。但是对于推荐系统那是不可能的，当打开所有任务，系统的性能将会受到极大的影响，这时候就体现出了给日志内容分级显示的尤为重要了，我们一般只开启部分任务，当然我们也可以制定一个异步任务。

国际化

如果仅仅是英文或者数字字段，不进行国际化也是可以的，但是有时候针对评论字段，那就不能不国际化，考虑国内情况；还是需要进行国际化的。{41%：当然也不需要全部的国际化，仅仅将用户行为数据国际化即可。}

本章小结

本章主要是对推荐系统整体进行需求分析，并将整体分割为三个子系统，然后对各个子系统进行详细的需求分割，划分功能模块。以下是各个子系统的描述：推荐引擎，在本节中将推荐算法划分为最小推荐模块，推荐算法有很多，同时推荐的需求也不断的变化，划分推荐引擎为最小单元有利于算法和业务的扩展。{41%：展示系统主要是划分展示的形式，以及定义各种功能接口。} 日志系统在于制定日志系统的大体规范，以保证高效获取推荐所需的各类数据。

推荐系统设计与实现

{ 55 % : 推荐引擎总体架构和J2EE技术概述 }

架构目标

本系统是为推荐系统的核心，而且兼容多种推荐算法，因此综合考虑本系统的设计目标如下：

性能要求。 要求系统能够提供良好的运行性能，能够满足大量用户的并发，并且要保证每个用户的私有信息得到保护。

系统可维护性： { 44 % : 为了适应系统的功能扩展，系统的架构必须适合扩展，符合“开-闭”原则； } 在系统的开发过程中，系统文档、指示代码、体系应遵循软件开发的规范要求，使系统具有良好的可维护性。

整体设计框架

{ 72 % : 图 5 1推荐子系统架构图 }

推荐引擎功能设计

用户特征模块

{ 43 % : 本模块主要功能就是获取用户特征和权值； } { 43 % : 通过用户的ID，或者用户的属性字段，可以是单个或者多个属性字段。 } 如性别，年龄等的组合条件。

{ 57 % : 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；

推荐引擎结构图

{ 85 % : 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。 } { 91 % : 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 5 2推荐引擎结构图

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；

{ 78 % : 展示系统总体技术架构设计 }

{ 98 % : 本系统采用目前比较流行的MVC三层架构模式，即表示层、业务逻辑层和数据库访问层。 } { 43 % : 表示层主要负责与用户的交互，即将得到的数据显示出来； } { 76 % : 业务逻辑层主要负责处理本系统的业务逻辑； } { 56 % : 数据库访问层主要负责从数据库提取所需数据。 } { 46 % : 三层结构各司其职，系统层次感分明，很好地降低了系统的耦合度。 }

下面是本系统的整体框架图：

图 5 3视图系统的总体架构

功能模块分布

图 5 4展示系统功能模块

用户行为模块

通过推荐界面提供一些按钮或者在一些特殊的图片，按钮，链接上加入监控收集用户行为数据并让用户对推荐结果进行反馈，这样才能让推荐算法不断改善用户的个性化推荐体验。 以下是用户行为接口设计：

图 5 5用户行为包设计

显性行为数据收集

程序描述：

用户购买行为收集： 用户在产生购买行为时用户的行为收集，主要包括用户购买商品的价格，数量，商品的类别，商品的种类等信息。

{ 60 % : 用户查看推荐物品行为收集： } { 52 % : 用户点击推荐栏中商品所产生的信息； }

{ 45 % : 销售排行信息产生的用户行为： } { 44 % : 热门排行，销售排行，新品推荐等用户行为收集。 }

{ 80 % : 用户对物品的评分信息收集： } { 47 % : 用户主动为物品的进行评分，评价等信息； }

隐形行为数据收集

浏览商品用户行为收集： 用户在展示页面中浏览商品，系统收集用户在浏览器中浏览商品的时间，商品信息，浏览地区等信息并存储到日志系统内；

站内搜索关键字收集： { 40 % : 用户在站内搜索框搜索站内内容，系统收集用户的内容和时间。 }

最近查看的商品的行为： { 44 %：用户在一段时间内所浏览的商品信息，也可以说浏览商品详情页的信息。

}

用户行为调查报告收集

主要提供的功能是为一些特殊物品或者用户行为信息收集提供的接口，如对MacBook的认识，价格，性能等的调查报告。

调查问卷的生成： 为特殊的需求生成调查问卷；

调查信息的收集： 收集用户调查报告的数据；

推荐结果/推荐理由展示

系统根据具体的条件如： 推荐物品的数量，推荐物品信息的选择，推荐物品更新时间间隔等因素从系统中返回推荐结果，并附加推荐解释，也就是推荐理由。

获取浏览历史推荐结果： { 42 %：根据ID（mac地址/用户ID）获取该用户关于浏览记录的推荐。 }

获取销售排行推荐结果： 根据物品的销售量将物品进行排序并输出排名最高的N件物品。

获取物品组合推荐： 物品详情页中获取与物品相关的物品，加入组合推荐栏目中。

远程接口与用户安全设计

远程接口

{ 92 %：远程接口调用是系统与系统之间的通信机制，它的另一种理解就是进程间的通信。 } { 100 %：做分布式系统的开发，远程调用技术是其核心技术。 } { 100 %：远程调用技术可以将一组计算机系统形成一个网络系统，对外提供整体服务，那么这一群的计算机系统就构成了一个更大型，性能更高的计算机系统。 } { 42 %：展现系统的远程接口设计应符合以下这些要求： }

通信技术: { 100 %：远程调用就是通过网络技术将不同系统构成一个整体，因此通信技术是其重点，通信技术我这里选择的是 Netty 技术， } { 100 %：Netty 提供异步的、事件驱动的网络应用程序框架和工具，用以快速开发高性能、高可靠性的网络服务器和客户端程序。 } { 100 %：Netty 会让我们开发通信程序变得简单，高效，其效率也是非常好的，同时它还支持多种不同的网络协议。 }

{ 100 %：序列化和反序列化技术： } { 100 %：java 的序列化技术是指将对象转换为 byte 数据，这些数据可以被还原为 java 对象， } { 100 %：这种还原的过程就是反序列化了，该机制可以自动处理不同操作系统之间的差异， } { 100 %：例如 window 下序列化的对象，可以在 linux 上进行重新构建。 } { 100 %：Java 的 JDK 里自带了一个序列化和反序列化机制，熟悉 hadoop 的人知道 hadoop 设计了一套序列化和反序列化机制， } { 100 %：为什么 hadoop 作者不选择使用 java 自带的序列化机制，这是因为 java 序列化机制非常复杂，复杂带来效率低下， } { 92 %：java 的序列化机制还有一个重要的缺点就是它序列化的数据量会非常大，因为 java 序列化的时候会附带太多该

对象的相关信息，} {100%：过大的数据量就会影响网络传输的效率，因此hadoop自己设计了一套序列化和反序列化机制，} {100%：hadoop不同节点之间的通信也是一种远程调用机制，因此我们发现好的序列化和反序列化技术对于远程调用是相当重要的。} {100%：我们公司的远程调用框架序列化技术有两种一种就是java自带的序列化和反序列化机制，} {100%：一种是hessian技术，它是一种更加高效的序列化和反序列化技术。}

压缩技术： {100%：做网络编程，最稀缺的资源就是宽带资源，如果传输数据过大，那么对数据的压缩就会显得十分重要，} {92%：这里我推荐一个压缩技术snappy，它是一种高效的压缩和解压缩包，谷歌公司内部广泛使用的一种压缩技术。}

非侵入式： {100%：这个也可以叫做松耦合，对于java的web开发，最好的解耦方式就是使用spring技术，} {100%：当我们系统里把远程调用框架引入后，配置好相关的参数，我们可以把用于远程调用的方法定义在spring的配置文件里，} {100%：那么在程序里调用的时候，利用spring直接获取这个bean，那么对于远程调用的开发就和我们在action里调用server的方法没啥区别了。}

高并发的技术： {100%：远程调用技术一定会是多线程，只有这样才能满足多个并发的处理请求，java在1.5的版本里提供了一个Executor框架，} {73%：它在线程开发里引入了任务的概念，使得多线程的程序开发会更加合理和可控，想让线程更加有效率，} {97%：池的技术也是并不可少的，apache的common-pool是一个非常好的池技术，我们可以将线程都预先创建好，} **然后放入到common-pool池里进行管理。**

负载均衡： {100%：分布式系统都离不开负载均衡，好的负载均衡可以充分利用好不同服务器的计算资源，提供系统的并发量和运算能力，} {100%：对于网站而言（我们公司现在网站服务器不是太多）少于10台服务器可以使用两种策略：} {100%：一种是简单轮询，比如有6台服务端，我们会把第一个请求给第一台服务器，第二个请求给第二台，依次类推，等6台循环完毕，又从第一台开始；} {98%：第二种是随机方式，即使用random函数，当然更多的服务器我就不知道有什么轮询机制比较好。}

用户安全

{94%：为了保障信息的机密性、完整性、可用性和可控性，必须采用相关的技术手段。} {92%：这些技术手段是信息安全体系中直观的部分，任何一方面薄弱都会产生巨大的危险。} {96%：因此，应该合理部署、互相联动，使其成为一个有机的整体。} **具体的技术介绍如下：**

加解密技术。 {100%：在传输过程或存储过程中进行信息数据的加解密，典型的加密体制可采用对称加密和非对称加密。}

VPN技术。 {90%：VPN即虚拟专用网，通过一个公用网络(通常是因特网)建立一个临时的、安全的连接，是一条穿过混乱的公用网络的安全、稳定的隧道。} {96%：通常VPN是对企业内部网的扩展，可以帮助远程用户、公司分支机构、商业伙伴及供应商同公司的内部网建立可信的安全连接，并保证数据的安全传输。}

防火墙技术。 {100%：防火墙在某种意义上可以说是一种访问控制产品。} {89%：它在内部网络与不安全的外部网络之间设置障碍，防止外界对内部资源的非法访问，以及内部对外部的不安全访问入侵检测技术。} {94%：入侵检测技术IDS是防火墙的合理补充，帮助系统防御网络攻击，扩展了系统管理员的安全管理能力，提高了信息安全基础结构的完整性。} {93%：入侵检测技术从计算机网络系统中的若干关键点收集信息，并进行分析，检查网络中是否有违反安全策略的行为和遭到袭击的迹象。}

安全审计技术。 包含日志审计和行为审计。 { 91 % : 日志审计协助管理员在受到攻击后察看网络日志 , 从而评估网络配置的合理性和安全策略的有效性 , 追溯、分析安全攻击轨迹。 } 并能为实时防御提供手段。 { 97 % : 通过对员工或用户的网络行为审计 , 可确认行为的规范性 , 确保管理的安全。 }

本章小结

{ 46 % : 本章主要内容是整个推荐系统的设计实现 , 其中包含推荐引擎 , 推荐展示系统的设计。 } 推荐引擎技术是一种新的信息传播方式 , 将推荐引擎分为两部分一部分是推荐算法模块 , 另一部分属于规则模块 , 其中有热门排行 , 还有一些规则组成 ; { 49 % : 两部分互补组成推荐的主要部分 , 再经过过滤和排名得到最终的推荐列表。 } { 42 % : 当然主要是解决推荐引擎的整体架构与技术架构实现。 } 再有就是对展示系统的详细设计 , 其中主要包含是系统的总体技术架构 , 系统功能的描述 , 分析 , 具体接口的定义等 , 其中有用户行为模块 , 推荐结果模块 , 远程接口与用户安全设计。

总结与展望

{ 40 % : 推荐系统是解决信息过载的一种解决方案 , 同时也是一种连接用户和内容的信息系统 , 一方面它帮助用户发现他们潜在的兴趣点 , } 另一方面它能够帮助信息提供者将内容投放给对相应的用户。 { 50 % : 推荐系统的常用方法是通过分析用户的历史行为来发现他们将来的行为。 } 但是在国内并没有一个能够为行业提供推荐的独立推荐公司 , 或者推荐系统 ; 百度在广告方面的推荐具有一定的优势 , 但不提供对外服务。 { 40 % : 因此本文主要是推荐系统的设计与实践 , 为提供独立的推荐系统做出一点力量。 }

推荐系统总结

{ 51 % : 本文主要的工作和贡献包括一下两方面 : }

通过 SVD 方法计算用户的与测评分 , 又使用 Top- N 推荐对隐形反馈数据计算用户-物品的权值 , 返回用户的推荐列表 , 并在其原有的算法中加入一个新的时间元素 , 形成一个合适的算法。

{ 45 % : 对推荐引擎分析 , 日志系统以及展现系统分别进行了分析和设计 ; } 从使用的相关技术到系统的技术架构 , 再到相关接口的设计 , 每一点都做了详细和深入的探索。

推荐系统展望

本文对 Top- N 做了一定的优化 , 并对整个推荐系统的实践做了相应的阐述 , 但是针对用户的组合推荐和多特征并行处理的推荐并没有相应的实践。 { 43 % : 希望在以后的实践中对推荐算法做进一步的改进 , 提高推荐算法的计算性能 , 算法推荐的精准度以及线上的相应时间。 }

参考文献