中国科学技术大学

硕士学位论文

基于用户画像的手机主题推荐系统

作者姓名: 胡磊

学科专业: 信息安全专业

导师姓名: 周武旸 教授

张四海 博士

完成时间: 二〇一五年十二月

University of Science and Technology of China

A dissertation for master's degree

The Phone Theme Recommendation System Based on User Profile

Author :	Lei Hu
Speciality :	Information Security
Supervisor :	Prof. Wuyang Zhou
	Dr. Sihai Zhang
Finished Time :	December 12th, 2015

中国科学技术大学学位论文原创性声明

本人声明所呈交的学位论文，是本人在导师指导下进行研究工作所取得的成果。除已特别加以标注和致谢的地方外，论文中不包含任何他人已经发表或撰写过的研究成果。与我一同工作的同志对本研究所做的贡献均已在论文中作了明确的说明。

作者签名：_____　　签字日期：_____

中国科学技术大学学位论文授权使用声明

作为申请学位的条件之一，学位论文著作权拥有者授权中国科学技术大学拥有学位论文的部分使用权，即：学校有权按有关规定向国家有关部门或机构送交论文的复印件和电子版，允许论文被查阅和借阅，可以将学位论文编入《中国学位论文全文数据库》等有关数据库进行检索，可以采用影印、缩印或扫描等复制手段保存、汇编学位论文。本人提交的电子文档的内容和纸质论文的内容相一致。

保密的学位论文在解密后也遵守此规定。

□公开　□保密（____ 年）

作者签名：_____　　导师签名：_____

签字日期：_____　　签字日期：_____

摘 要

本文是中国科学技术大学本硕博毕业论文模板示例文件。本模板由 ywg@USTC 创建，适用于撰写学士、硕士和博士学位论文，本模板由原来的本科模板和硕博模板整合优化而来。本示例文件除了介绍本模板的基础用法外，本文还是一个简要的学位论文写作指南。

关键词： 中国科学技术大学 学位论文 LaTeX 通用模板 学士 硕士 博士

ABSTRACT

Because of the dynamic evolution and propagation of the Internet and the popularization of social networks, huge amounts of information about the users's behaviour are accessible to commercial internet companies. Most companies can be easily approaching OOS(open source software) such as hadoop to analyse and interpreting users's behaviour data-set,which just several years ago was not available or inaccessible. Also because of the evolution and popularization of the Internet and the vast amount of data stored, it has become desirable to moderate and select the content that is being displayed to the user,Mechanisms presenting goods on application try to present content that can interest the user based on his previous queries or browsing history. commercial internet companies such as e.g. Xiaomi filter the phone theme application visiting information and show only phone themes that the user potentially be interested in, trying to sell additional goods by recommender products based on user previous purchase behaviours.

Based on my working experience on Xiaomi as internship, The biggest challenge of recommender systems in commercial internet companies is: How to optimize a recommender system in accordance with the true business objective. for commercial internet company like Xiaomi the fitness recommender system is something that mix the social part, the long-tail, the cold-start and many other factors to finally aproximate what the user really wants. It's really a fascinating and complex working.

The aim of this paper is analyse the long tail feature and cold start feature of a android phone theme recommender system with the help of user profiles and user interests exploration. In order to build the user profile, Information Retrieval (IR) and Data Mining (DM) techniques such as content-based Collaborative Recommendation and text pre-processing methods have been used; In order to approaching discovery and recommendation in long tail, user interests exploration has been refered. Because of the subjective nature of such a solution, verification process such as A/B test is also introduced.

Keywords: recommender systems, long tail, cold start, user profile, user interests exploration

目 录

表格索引

插图索引

算法索引

第一章　绪论

1.1　研究背景与意义

随着物联网和用户终端设备的发展，人们逐渐从信息的匮乏时代走进了信息的过载 (Information overload) 时代。无论是作为信息消费者的普通用户，还是作为信息生产者的提供商面临着数据爆炸时代的挑战。作为用户，如何从充斥着大量噪声的大数据中找到自己感兴趣的信息是一件非常耗时费力的事情。而作为提供商，如何让自己生产的信息不埋没在大数据洪流中而受到潜在用户的充分关注，这也是其所要解决的一个课题，很多企业已经或者正在开发适合本公司的推荐系统 (Recommender System) 来解决这一矛盾。笔者曾有过这样的一种购物体验：在淘宝商城购买一台笔记本电脑，花费了一上午的时间才浏览、比较完所有的 thinkpad 品牌商家店面，如图 1.1。而近年来淘宝的交易额增长规模巨大，2005 年淘宝交易额为 80 亿，2010 年为 4000 亿，而到 2015 年淘宝双十一单日交易额就为 912 亿元，可见未来几年内笔者的这种关键字搜索 + 逐条浏览的购物方式已经不再具有可行性。推荐系统的主要任务就是联系用户和信息，一方面协助用户发现自己潜在感兴趣的信息从而提升用户的满意度，另一方面让信息针对性的展现在只对它有兴趣的用户面前从而提升商品的转化率，于是实现了消费者和生产者的双赢。

自互联网诞生以来，用户寻找信息的方法经历了几个阶段。早期的用户主要靠直接记住感兴趣网站的网址来寻找内容，直接促使 Yahoo! 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，于是产生了搜索引擎。以 Google 为代表的搜索引擎可以让用户通

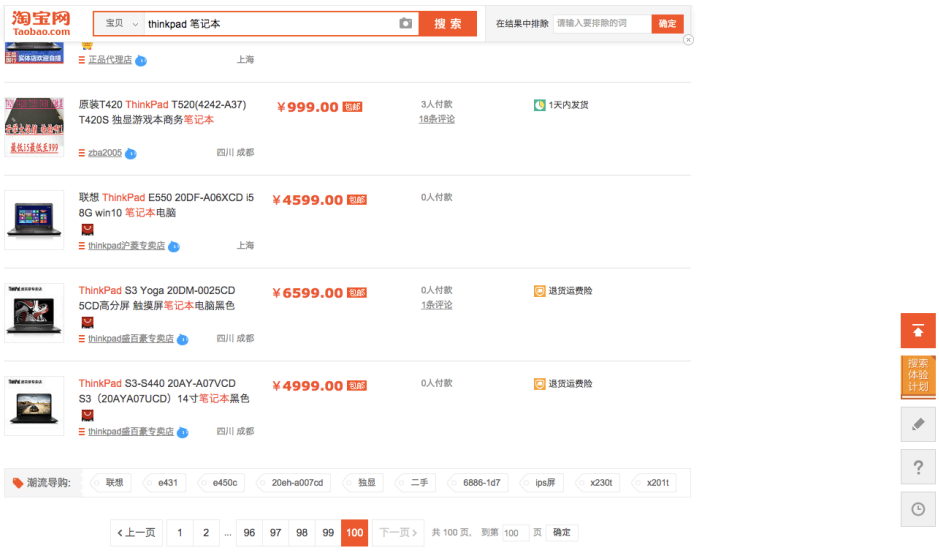

图 1.1　淘宝购物页面

过关键词找到自己需要的信息，但是，搜索引擎需要用户主动的提供显式关键词来寻找信息，因此它不能解决用户的更多的潜在需求，当用户无法精准描述自己的需求时，搜索引擎就无能为力了，于是又催生出推荐系统 [7]。以亚马逊电商官网为代表的推荐系统是一种帮助用户快速发现有用信息的工具，和搜索引擎不同的是推荐系统不需要提供明确的需求，而是通过分析用户的历史行为来给用户画像建模从而主动给用户推荐出能够满足他们兴趣和需求的信息。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。搜索引擎满足用户显式的需求，而推荐系统能够在用户没有明确目的的时候帮助他们发现潜在的需要。

现有的预测用户兴趣的方法主要有三个。内容过滤 (Content Filtering)[2] 算法认为用户会喜欢和他以前喜欢的物品在内容上相似的物品，比如用户购买了一款基于海贼王的动漫主题包，那么内容过滤算法会给他推荐另一款相似的火影忍者主题包。内容过滤算法主要利用了物品的内容数据，比如主题包的作者，标题，类型，关键词，适用人群等信息。协同过滤 (Collaborative Filtering)[12] 不依赖于用户的属性信息和物品的内容信息，而仅仅通过分析大量的用户对物品的行为数据，从社交角度找出特定的相似行为模式，据此来预测用户的兴趣并给用户做出推荐。随着以 Tencent 和 Facebook 为代表的社会网络的兴起，社会化过滤 (Social Filtering) 逐渐成为推荐领域的研究热点。社会化过滤算法 [11]，认为用户的兴趣和他的好友的兴趣会有共同点，从而可以通过分析用户好友的兴趣来预测给定用户的兴趣。除此之外，还有利用用户年龄，性别，职业，用户地理位置等属性的推荐算法。对于推荐系统来讲，每天都会有大量的新用户加入，老用户离开；大量的新内容上线，旧内容下架，与此同时用户的年龄，兴趣，社会关系，地理位置也会发生变化。因此，时间作为上下文 (Context) 信息对推荐系统来说是一种重要的要素，对用户的行为有着重要的影响。早期的推荐系统研究对时间相关的动态特性很少涉及，随着很多大时间跨度的商业网站用户行为数据的积累，以及用户画像建模的成熟，越来越多的研发人员开始聚焦、利用用户行为的动态特性。

基于用户画像的推荐系统可以更好的发掘信息的长尾效应 (The Long Tail)。长尾理论 [1] 指出，电子商务网站相对于传统零售超市的优点是电子商务网站可以给用户提供更多选择的同时，降低了商品流通成本。因为电子商务网站没有货架的成本，增加一个商品只需要在数据库中添加一行商品相关数据而已，但是随着越来越多的商品信息暴露在用户面前，必要会导致用户的不知所措。其实，大数据爆炸式的增长既是挑战也是机遇，如何充分利用大数据驱动运营，是中国互联网企业面临的一个机遇。与长尾效应相对应的是马太效应 [8]，指的是好的越好，坏的越坏，多的越多，少的越少的现象。举一个例，电子商务网站首页的商品列表常常是按照热门商品排列的，一段时间内最热门的商品一定是排在最上边的，由于热门商品列表的长度是有限的，因此那些冷门、小众的商品是不会进入列表。如果用户依赖于这个列表寻找商品，那么进入到这个列表中的商品就会

越来越热门，而进入不了这个列表中的商品就会越来越不热门。搜索引擎也有马太效应的问题 [9]，在热门搜索词排名靠前的网页会越来越热门，能够获得越来越多的外链，从而在 PageRank 算法中排名越来越高，也就更容易获得比较高的排名。推荐系统作为一种寻找信息的重要工具，也面临马太效应的挑战 [10]。但是，相比较搜索引擎来讲推荐系统是一个更具有主动性的系统，因此推荐系统能够更好的控制每个商品的展现次数，让长尾商品能够在对其感兴趣的用户面前得到充分的展示。因此，一个好的推荐系统不仅应该能够帮助用户发现有用的信息，而且需要能更好的发掘长尾效应。

1.2 研究内容

本文将集中研究推荐系统的用户画像建模和用户兴趣探索模块 [4]。利用用户画像、兴趣探索可以改善推荐系统如下几个方面的性能：

- 改善用户冷启动问题：当一个新用户注册生成时，系统一般是对其不了解的，这时可以只给其推荐热门商品，待积累足够用户行为数据时再为其做个性化推荐，但缺点是会加强马太效应。可以利用用户画像中的一些信息，如年龄，性别，职业，用户地理位置作为推荐依据，解决冷启动问题 [3]。

- 提升推荐系统的时效性：推荐系统的任何在线推荐都要保证在 120 毫米之内计算完，其中一个解决方案是利用分层思路，即在推荐系统层和服务器日志层之间增加一个用户画像层，如图 1.2所示，通过把一些高难度，复杂性计算的算法放在离线的用户画像建模，提前计算好，在需要被调用的时候，再做一些简单的在线的计算和分装，完成时效性和性能的保证。

- 提升推荐系统的安全性：随着电子商务的迅猛发展，商家会通过各种活动形式的补贴来获取用户、培养用户的消费习惯，但同时也催生一些通过刷排行榜、刷红包的用户，严重破环了市场的稳定，侵占了活动的资源。其中一个有效的解决方案就是利用用户画像沉淀方法设置促销活动门槛，即通过记录用户的注册时间、历史登陆次数、常用 IP 地址等，最大程度上隔离掉僵尸账号，保证市场的稳定发展。

- 提升推荐系统的长尾效应：如果用 2-8 法则来解释的话，就是说百分之 20 的热门商品占据了百分之 80 的关注度，百分之 80 的小众商品只占了百分之 20 的关注度，即使某用户对某个小众商品有一些高质量的用户行为数据，这些数据也会淹没在太多的热门商品行为数据中，传统的推荐系统不能准确发现这类关联关系。而用户兴趣探索作为专门聚焦于小众兴趣标签的探索，可以提升推荐系统的长尾效应。

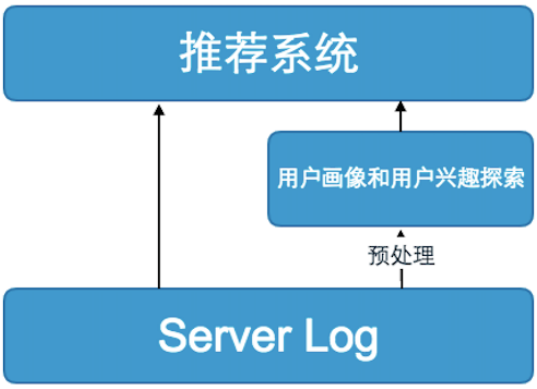

图 1.2　结合了用户画像的推荐系统结构图

1.2.1　数据集介绍

本文涉及到的数据集主要分为推荐系统的输入数据集和推荐系统的输出数据集。输入数据集由三个不同大小的数据集组成:

- 物品属性向量：用来述一个商品的性质，也经常被称为 Item Profile，包括主题名称，作者，上架时间，用户评分，编辑评星，评论，价格，以天为单位的各个时间维度 (周，月，年) 的下载量和销售量，人工 + 聚类算法打上的标签等。共包括 1.1 万个第三方开发的主题包，其中有效主题包有 9 千多个。

- 用户画像。用来述一个用户的“个性”，也就是 User Profile。包括用户注册信息：姓名，年龄，性别，职业，地理位置等，用户活跃度，用户购买力，用户兴趣探索获得的各种兴趣标签及其权重等。数据集包含了 300 万用户信息，可以用来解决推荐系统的冷启动问题。

- 用户行为数据：一般来讲，一个用户在某一时刻对某一个主题包的一次动作算做一条日志，存放在公司的 hadoop 集群上。用户行为包括显示反馈：浏览，下载，购买，评论，打分，点赞/踩，也包括隐式反馈：停留时长，划屏频率，进入方式等。这个大规模数据集包括了一个月内近 5 千万条日志，反映了用户对物品的喜好程度。

1.3　论文结构

本文的其余正文内容由以下章节组成:

- 第二章首先介绍了手机主题推荐系统基本概念和手机主题推荐系统算法模型，包括数据挖掘算法 [6] 和信息提取技术 [5] 的应用。然后根据手机主题业务特点介绍了用户兴趣动态性, 最后讨论了推荐系统的评测指标。

- 第三章主要讨论了用户画像建模，包括用户画像的数据来源，用户标签权重计算，以及用户画像建模方式。接下来从不同维度分解用户画像标签属性，最后列举了用户画像在实际生产中的应用场景，包括解决用户冷启动、用户兴趣多样性的问题，并给出了相关的实验结果及分析。
- 第四章主要讨论了如何利用用户兴趣探索跟踪用户动态并挖掘用户小众兴趣，从而提升推荐系统的长尾效应，文中给出了相关的实验结果及分析。
- 第五章主要讨论了如何设计一个实际的动态长尾推荐系统，以及动态推荐系统的各个主要模块设计和需要遵守的设计原则。
- 第六章是论文的结束语和展望，在对目前工作简要总结的基础上，提出了推荐系统下一步研究的任务和方向。

第二章　手机主题推荐系统概述

2.1　引言

当今的时代是信息过载 (Information overload) 的时代 [18]。对于一个用户来讲互联网上充斥着大量对其无用的信息，如何从这些信息里找到用户感兴趣的信息，并把这些信息推送给用户是推荐系统面临的主要问题。推荐系统通过对用户的历史行为进行挖掘，对用户画像进行建模预测用户未来的行为。

推荐系统的研究和很多早期的研究相关，比如认知科学 (cognitive science)[14]，信息检索 (information retrieval) 和预测理论 [16]。随着互联网的兴起，研究人员开始研究如何利用用户对物品行为数据来预测用户的兴趣并给用户做推荐 [17]。推荐系统开始成为一个比较独立的研究问题。到 2006 年为止推荐系统的研究主要集中在基于邻域的协同过滤算法，目前工业界应用最广泛、最知名的算法应该就是亚马逊开发并使用的协同过滤算法 [19]。

近年来很多研究人员意识到推荐的时效性和多样性对于用户的体验度非常重要，而长尾效应对提高商品销售量有非常大的帮助。创建用户兴趣画像则是其中比较有效的解决途径之一，因为度量用户对物品的喜好不仅取决于用户的喜好和物品的属性，也取决于用户所处的环境，或者称做上下文 (Context)，上下文信息有很多类型，其中时间是一种重要的上下文信息，用户在不同的时间可能喜欢不同的物品，物品在不同的时间也有不同的流行度。因此推荐系统应该是一个动态系统，随着时间的变化会给用户不同的推荐结果 [20]。

本章接下来的内容，首先论述现有手机主题推荐系统的作用和其面临的问题，接下来详细介绍推荐系统的算法模型。

2.2　手机主题推荐系统基本概念

推荐系统通过分析用户-主题交互行为数据，对用户潜在感兴趣的主题打分并按优先度推荐，这里涉及到的推荐主题只局限于那些用户还没有接触过的主题，主题推荐首页如图 2.1所示。推荐系统的具体功能包括：

- 增加主题的销售量，具体反应在各种统计数据如购买量，点击转换率等的提升。
- 提升主题销售的多样性，多样性对应于长尾效应，多样性意味着推荐系统把正确的主题推送给了正确的用户，不论这个主题是热门的还是冷门的。
- 提升用户满意度，一个好的推荐系统能更好的明白每个人的兴趣点，做到推荐结果的千人千面，其含义包括：1，不同人对应的推荐结果不同；2，同一个人不同时间的推荐结果也应该有所不同。

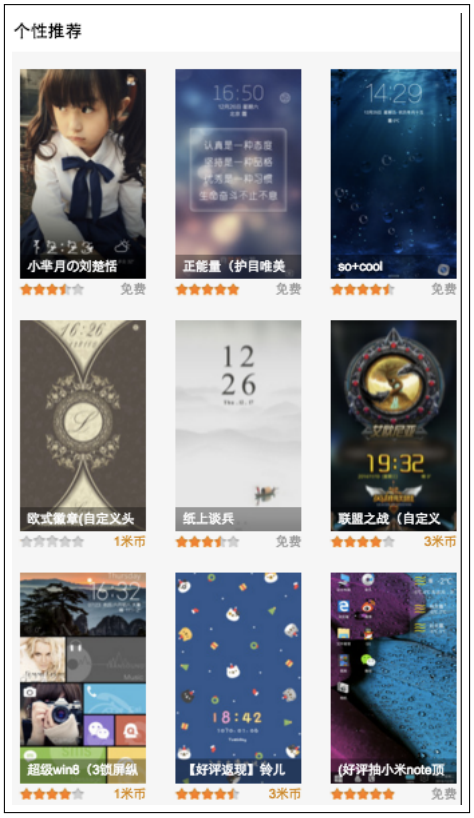

图 **2.1**　主题推荐页面

于此同时，现有的手机主题推荐系统也存在如下问题：

- 对于一个刚刚注册成功的新用户，推荐系统往往对其所知甚少，有些时候推荐系统需要在用户购买、评价信息匮乏的情况下做出推荐。
- 每一分每一秒都可能有新的用户行为数据产生，而这些用户行为数据应该能够尽快作为输入传送给推荐系统作为推荐结果的参考，实现用户兴趣的无缝体验。
- 对于一些活跃用户，在其频繁的用户行为数据集中，可能隐藏着若干条针对小众主题的高质量行为，推荐系统应该能够准确识别出这些信息并适当的向用户展示更多相似类型的小众主题。
- 针对单个用户的推荐可能需要海量的数据去计算，相反，对于大多数主题其用户行为数据十分稀疏。在如何提升算法效率的同时能有效利用计算机存储空间，是推荐系统领域一个非常热门的课题。

2.3　手机主题推荐系统算法模型

2.3.1　协同过滤算法

协同过滤的根本原理是，人们可以从和自己有相同品味、习性的人群那里获得高质量的推荐。协同过滤算法主要研究如何聚类具有相似兴趣特征的人群并基于此做出推荐，因为算法本身是基于用户社交群体，因此往往会涉及到大规模的用户行为数据的计算。协同过滤的应用领域也很广：电子商务，金融信贷，搜素引擎，互联网企业，网络社区等需要对用户提供个性化体验的服务商。因为中国现有的人口国情，协同过滤算法往往需要面对亿万级用户和海量的用户-主题交互数据。作为输入数据，一个用户是以一个 N 维度的向量来表示，N 代表所有的主题数量。向量内容可以为正也可为负，分别表示了用户喜欢、讨厌该主题的程度。对于热门主题，给其打分的用户会很多，其分数应该乘以一个因子 u 得到有效的分数，u 代表所有给其打分的用户个数的倒数，大多数用户向量是稀疏的。在协同过滤算法中关键性的一步就是要选择测量的距离，描述集合相似度算法有欧氏距离、闵可夫斯基距离、汉明距离等，这里选择余弦距离公式 (cosine similiarity)，公式描述如下，其中 $similarity_{uv}$ 代表用户 u 与 v 之间的兴趣相似度，N(u) 表示用户 u 曾经喜欢过的物品集合，N(v) 表示用户 v 曾经喜欢过的物品集合。

$$similarity_{uv} = \frac{|N(u) \cdot N(v)|}{||N(u)|| \cdot ||N(v)||} \tag{2.1}$$

然后利用相似度算法把用户分类成独立的集合，每个用户有且只属于其中的一个集合，对于每个集合，取这个集合最受欢迎的 top K 个主题，作为推荐内容推荐给集合的所有用户。大多数情况下协同过滤算法面都临着一个问题：最坏情况下需要遍历所有的用户和所有的主题，算法计算复杂度为 O(MN)，M 是用户数 N 是主题数，解决方法可以借助一种简单的降维思想加以解决：通过去掉那些非常冷门的主题对 N 做降维，通过去掉那些非常不活跃的用户对 M 做降维，计算维度下降的代价是降低了推荐系统的准确性。

协同过滤推荐包括 User-Based CF 和 Item-Based CF 俩种算法，User-Based CF 更多的是挖掘用户之间的社交属性，而 Item-Based CF 更多的是挖掘物品之间的特征属性。这里根据手机主题业务的特殊性选择用 Item-Based CF 算法，原因包括：

- 现有手机主题共有 1.1 万款，线上主题约 9 千多款，在计算主题相似度过程中只需要维护一个 1 万 *1 万的矩阵，大约只占用了 4-5GB 内存，一台服务器即可承受；相反，主题用户有 300 万，意味着有 300 万 *300 万的矩阵，计算量太大，需要用一个 spark 集群完成计算，从经济效益的角度讲不可取。

- 手机主题一般来讲周上线不足 100 多款，周更新不到 5%，且可以利用增量计算方法更新 item-item 矩阵；相反，主题用户周增加量为 10 万数量级，加上用户兴趣、社交、互动的动态变化，增量计算无能为力，导致每次更新 user-user 矩阵的时候计算量很大。
- 如果用 ItemCF，会只推荐与相似领域的主题的东西给用户，这样在有限的推荐列表中就可能包含了一定数量本领域不热门的 item，所以 ItemCF 推荐长尾的能力比较强，代价是推荐多样性不足，但是对整个系统而言，因为不同的用户的主要兴趣点不同，所以系统的 coverage 也会很大。
- ItemCF 的算法还可以为推荐结果做出理性的解释。如一个用户之前购买过魔兽世界的主题包，推荐系统会给其推荐魔兽争霸主题包并附上说明：因为用户曾经买过类似的主题包, 并且评价分数不错。

2.3.2　聚类模型算法

聚类分析（Cluster analysis）是对于统计数据分析的一门技术，和分类算法一个主要的区别就是聚类不需要人工参与打标签，基于聚类和 SlopeOne 预测的协同过滤方法，也可以在一定程度上解决传统协同过滤算法用户评分矩阵稀疏和冷启动问题，在降低用户评分矩阵稀疏性的同时提高目标用户最近邻居的查询速度。聚类是把相似的对象通过静态分类的方法分成不同的组别或者更多的子集（subset），这样让在同一个子集中的成员对象都有相似的一些属性，聚类结果不仅可以揭示数据间的内在联系与区别，还可以为进一步的数据分析与知识发现提供重要依据。在结构性聚类中关键性的一步就是要选择测量的距离。一个简单的测量就是使用曼哈顿距离，它相当于每个变量的绝对差值之和。该名字的由来起源于在纽约市区测量街道之间的距离就是由人步行的步数来确定的。聚类模块可以是对用户兴趣属性相似度做聚类，也可以对用户社交属性相似度做聚类，或者俩种兼有。

在现实社会中人们的兴趣和选择往往受到身边亲朋好友的影响。在互联网中随着诸如国内的腾讯，国外的 Twitter 等社会网络网站的兴起，如何利用用户的社会属性做推荐是近几年推荐领域比较热门的研究问题。基于社会网络的推荐算法被称为社会化推荐 (Social Recommendation)。近几年在工业界已经有了很多社会化推荐系统。最简单的社会化过滤算法是基于邻域的算法 (Neighborhood-based Method)。给定用户 u, 令 F(u) 为用户 u 的好友集合,N(u) 为用户 u 喜欢的物品集合。那么用户 u 对物品 i 的喜好程度定义为用户 u 的好友中喜欢物品 i 的好友个数，如公式 2.2。

$$P_{ui} = \sum_{v \in F(v) i \in N(u)} 1 \tag{2.2}$$

综合聚类模型利用了用户-用户的社会网络属性和用户-物品的兴趣属性，利用随机游走 (Random Walk)[23] 算法给用户做社会化推荐。Ma 在提出了一个矩

阵分解的算法来分解用户的社会网络矩阵和用户物品喜好矩阵，计算出用户的特征向量和物品的特征向量，并最终利用特征向量的点乘度量用户对物品的兴趣，综合模型相比较单独用社交属性和单独用兴趣属性的优点如下：

- 一般情况下用户物品矩阵的稀疏度比较高，因此仅仅利用用户-物品的兴趣属性来做协同过滤会有数据稀疏问题，造成推荐精度较差。但如果我们能通过用户的社交行为获得他的社会网络信息，就可以根据他朋友的历史行为来预测他的兴趣。

- 在现实社会中，人们的选择确实会受社会关系的影响，如果适当引入社会化过滤也可以增加推荐结果的用户的惊喜度和多样性。

聚类算法在许多领域受到广泛应用，包括机器学习，数据挖掘，模式识别，图像分析以及生物信息，手机主题推荐使用了一个叫 K-均值法聚类算法，K-均值算法表示以空间中 k 个点为中心进行聚类，对最靠近他们的对象归类。

input : k
output: k 个集合

```
1 while true do
2 |   选择聚类的个数 k。
3 |   任意产生 k 个聚类，然后确定聚类中心，或者直接生成 k 个中心。
4 |   对每个点确定其聚类中心点。
5 |   再计算其新的聚类中心。
6 |   如果新旧聚类中心没有变化，跳出循环。
7 end
8 return k 个中心点
```

算法 **2.1:** k means

2.3.3　SlopeOne 算法

SlopeOne 是一系列应用于协同过滤的算法的统称。由 Daniel Lemire 和 Anna Maclachlan 于 2005 年发表的论文中提出，该算法的特点就是实现简单而高效。举例，如表 2.1所示主题 2 和 1 之间的平均评分差值为 (2+(-1))/2=0.5. 因此，主题 1 的评分平均比主题 2 高 0.5。同样的，主题 3 和 1 之间的平均评分差值为 3。因此，如果我们试图根据小敏对主题 2 的评分来预测她对主题 1 的评分的时候，我们可以得到 2+0.5 = 2.5。同样，如果我们想要根据她对主题 3 的评分来预测她对主题 1 的评分的话，我们得到 5+3=8。

为减少过拟合的发生而实现基于 SlopeOne 的协同过滤算法，该方法运用更简单形式的回归表达式 (f(x)=x+b) 和单一的自由参数，而不是一个主题评分和另一个主题评分间的线性回归 (f(x)=ax+b)。该自由参数只不过就是两个主题评分间的平均差值，在某些实例当中它比线性回归的方法更有效。基于聚类算法和

表 **2.1**　SlopeOne 示例

顾客	主题 1	主题 2	主题 3
小明	5	3	2
小磊	3	4	未知
小敏	未知	2	5

SlopeOne 预测的协同过滤方法还可以很有效的解决数据稀疏性和推荐系统冷启动问题，实现步骤如下:

- 根据余弦相似度公式计算主题 i 和主题 j 的属性相似性，记为 sims(i，j)。
- 根据现有主题的既得评分，组成前述的评分矩阵，计算两个主题之间的评分相似性，记为 simr(i，j)。
- 将前两步所得结果进行线性组合，组合结果作为最终的综合相似性，记为 sim(i，j)：sim(i，j) = αsims(i，j)+(1-α)simr(i，j)。
- 对 sim(i，j) 按从大到小排序。取相似度最大的前 k_n 个主题作为邻居主题，从而得到目标主题的邻居主题集 $I = i_1，i_2，i_3 \cdots k_n$，在这个邻居主题集的基础上，对目标用户运用加权 SlopeOne 算法进行预测，将预测评分填入空缺的评分矩阵。

2.3.4　标签传播算法

Hoffman 曾提出了隐语义模型 (Latent Class Model)[21]，根据模型，我们可以认为用户并不是直接对主题产生兴趣，而是对主题所包含的几个属性特征感兴趣，而物品属于不同的类别，模型会通过用户行为学习出这些类别以及用户对类别的爱好程度，在此基础上研究人员提出矩阵分解模型，Last.fm 在线音乐平台的标签传播算法是典型的矩阵分解模型。基于内容推荐的标签传播算法通过跟踪用户所有的在线、离线、本地、远程行为，对每个用户维护一个反应其兴趣的画像，这个画像会记录用户个性化兴趣标签，让用户对自己使用的主题不断的反馈兴趣标签，然后实时的通过用户的反馈来更换用户的推荐列表，让用户越来越多的看到满意的主题包，功夫熊猫主题包的标签一共分成这么几类: 心情 (Mood)，迎合的潜在用户人群 (Audience)，用户评价 (Praise)，铃声风格 (Style)，人物态度 (Attitude)，背景画面 (Background Picture)，这些属性包括了主题方方面面的信息，可以准确描述一款主题包。推荐系统的标签一般由专职员工标注 + 用户标注 + 算法标注完成。用户标注相对于专职员工标注的特点是标签更加多样更能反映用户的长尾兴趣，但缺点是质量不高，有时候会有很多错误的标注。因此在用户标签系统中，一种方式是给用户提供一些适合于这个物品的标签供用户选择。这样做的目的有两个，首先是用户不用打字只通过点击候选标签就能

完成标注的过程，其次可以提高用户标注的质量。当用户给了物品标签后就可以将标签作为物品的关键词，利用内容过滤的算法给用户做推荐。

标签传播算法著名的应用有国外的 Pandora，Lastfm 和国内的虾米音乐。标签传播算法的运行步骤：

- 输入数据为半标签化的用户社交图，每个节点代表一个用户，上面标注了该用户接触过的所有主题包标签，每一个标签都对应着一个权重，一般来讲，一些用户行为包括购买、点赞对应的权重高一些，点击、浏览对应的权重低一些，其他行为的权重居中。
- 针对一个用户节点，把其所有的标签传播到与其相邻的其他用户节点，累加并正则化所有标签的权重。
- 多次迭代直到社交图上的标签达到收敛状态，输出用户社交图。

2.3.5　最近点击模型

点击模型（Click Model）是对用户点击行为的建模。根据用户的历史点击信息，对用户的兴趣和行为进行建模，以对用户的未来点击行为进行预测。针对用户最近 n 天的行为有 3 项特征 (Feature)，分别是点击量、停留时长、滑屏频率. 现有 2 个类别 (Category)，分别为用户是否购买。利用朴素贝叶斯分类器 (假设特征彼此独立) 计算出概率最大的那个分类，也就是求下面这个算式的最大值：

$$P(C|F1F2F3) = P(F1|C)P(F2|C)P(F3|C)P(C) \tag{2.3}$$

举例，假如已知某用户最近 3 天对某主题点击量为 6 次、停留时长为 6 秒，滑屏次数为 8 次，请问该用户是否会购买主体？这里面临的一个的问题是停留时长是连续变量，因此假设用户停留时长服从正态分布，通过训练样本预先计算出均值和方差，得到正态分布的密度函数。假设，没有购买的用户平均停留时长为 5.855、方差为 0.035，则没有购买的用户停留时长为 6 秒的概率为：

$$\frac{1}{\sqrt{2\pi\delta^2}}exp\left(\frac{-(6-5.855)^2}{2\delta^2}\right) \approx 1.5789 \tag{2.4}$$

其他特征概率计算类似。计算结果显示，用户没有购买的概率比购买的概率高出将近 10000 倍，所以判断该用户对该主题不感兴趣。实验发现，对于拥有不同购物性格的用户，最近天数 n 的取值也不一样。有些用户总是在短时间内比较了少量的商品就下单，那么他的购物性格便是冲动型，n 的取值要短些；有些用户总是在反复不停的比较少量同类商品最后才下单，那么他的购物性格便是理性型，n 的取值要长些；有些用户总是长时间大量的浏览了很多商品最后才下单，那么他的购物性格便是犹豫型，n 的取值要超长。

2.3.6　其他推荐算法和技术

用户的人口统计学特征包括了年龄，性别，工作，学历，居住地，国籍，民族，等等。这些特征对预测用户的兴趣也有很重要的作用。对于手机主题市场更为明显，不同性别的人群兴趣不同，不同年龄的人群兴趣也不同。喜欢 AK48 等美女类型的主题大多是男性，喜欢刘德华等男明星类型的主题大多是女性；喜欢动漫主题的人群大多数是 90 后，喜欢汽车主题的人群大多是 70、80 后。基于用户人口统计特征的推荐的最大好处，是可以解决注册用户的冷启动 (Cold Start) 的问题，当一个用户刚刚注册还没有任何行为的时候，我们就可以根据他注册时提供的年龄性别等人口统计特征数据来预测他的兴趣。Krulwich 在文中 [22] 研究了如何利用大量的用户人口统计特征数据和用户行为数据来构建用户的兴趣模型的方法。首先 Krulwich 根据美国用户看电视、购物的行为将用户根据他们的人口统计学特征分成 62 个先验的聚类。首先找到他的聚类然后给他推荐这个聚类里的其他用户喜欢的物品。但是基于人口统计特征的推荐的主要缺点是它的推荐粒度太粗。如果我们只有用户的年龄和性别的数据，那么对于相同年龄和性别的人的推荐结果将是完全一样的。因此没有实现彻底的个性化。另外，很多用户在提交自己的年龄性别信息时，出于对自己隐私的考虑不会提供真实的信息。

2.4　手机主题推荐系统的动态特性

现实世界的一切事物都处在变化之中。用户的兴趣、物品的属性都是在不断的变化，一个系统中每天会有大量的新用户新物品加入；时间作为一种重要的上下文信息 (Context)，不同的时间用户也会有不同的兴趣，比如用户在白天和晚上的兴趣可能不同，周末和工作日的兴趣可能不同，不同的季节用户的兴趣也会有所不同。因此，合理的利用时间信息，对推荐的精准度和用户的满意度将会有很大的提升。而传统的推荐系统在设计时并没有主动的考虑到时间因素，推荐系统的动态效应表现在：

- 用户偏好随时间变化 (User bias shifting): 用户可能在某一天只对他喜欢的物品评分，某一天可能只对他不喜欢的物品评分。因此用户某一天的平均分是随时间变化的。

- 物品偏好随时间变化 (Item bias shifting): 物品的受欢迎程度也是随时间变化的。一款主题包在刚上线的时候的因为用户关注度小平均评分会很高，随着时间的推移，越来越多的用户参与到评分中，会使其慢慢接近真实的评分。

- 用户兴趣随时间变化 (User preference shifting): 用户在不同的时候可能有不同的兴趣，比如小孩都喜欢动漫主题包，但当他长大了可能喜欢汽车主题

包。

- 季节效应: 用户行为会受季节效应的影响。主题推荐中主要的季节效应有暑期的效应，以及一些纪念日的效应 (比如国庆纪念日前后，抗日题材的主题包会受到较多的关注)。

为保持推荐系统的动态特性，工业界一般用数据追加的方式进行增量计算。推荐系统利用 hadoop 集群可以在 2 个小时内完成最近 24 小时数据的增量计算并将结果追加到现有的计算结果中，耗费的这 2 个小时可以用更少的时间进行增量计算并做数据追加。

2.5　推荐系统评测

一个好的推荐系统应该起三赢的作用：用户-找到自己感兴趣的东西；第三方设计师-增加了销量；网站-得到好的发展，提升了推荐质量。但是什么样的推荐系统才算是好的统计系统呢？这时需要有一个测评方法，常用的推荐系统测评方法见表 2.2。一个完整的推荐系统一般存在 3 个参与方：用户、物品提供者和网站平台. 好的推荐系统不仅仅能够准确预测用户的行为，而且能够扩展用户的视野，帮助用户发现那些他们可能会感兴趣，但却不那么容易发现的东西。同时，推荐系统还要能够帮助第三方设计师将那些被埋没在长尾中的好主题介绍给可能会对它们感兴趣的用户从不同角度出发。

表 **2.2**　推荐系统评测方法

方式	优点	缺点
离线实验	只需要数据集，可测试大量算法。	无法计算商业上关心的指标，如点击率、转化率等。
调查问卷	可获得感性指标，实验风险小。	成本高，参与用户少，统计意义不显著，双盲实验设计困难。
A/Btest	可获得点击率，转化率等运营指标。	准备周期长，设计复杂。

评测指标包括统计性指标：准确率 (Precision)、召回率 (Recall)、F 值 (F-Measure) 等，也包括用户感性指标：准确度、覆盖度、新颖度、惊喜度、信任度、透明度等. 一般来说评测维度分为如下几种用户维度，如果能够在推荐系统评测报告中包含不同维度下的系统评测指标，就能帮我们全面地了解推荐系统性能，找到一个看上去比较弱的算法的优势，发现一个看上去比较强的算法的缺点。

2.5.1　统计性指标

统计性指标包括准确率、召回率和 F 值，用来评价结果的质量。其中精度是检索出相关文档数与检索出的文档总数的比率，衡量的是检索系统的查准率；召回率是指检索出的相关文档数和文档库中所有的相关文档数的比率，衡量的

是检索系统的查全率。正确率、召回率和 F 值是在鱼龙混杂的环境中，选出目标的重要评价指标，其定义为：正确率 = 提取出的正确信息条数 / 提取出的信息条数，两者取值在 0 和 1 之间，数值越接近 1，正确率就越高。召回率 = 提取出的正确信息条数 / 样本中的信息条数，两者取值在 0 和 1 之间，数值越接近 1，召回率就越高。F 值综合了正确率和召回率的结果，见公式 4.1。当 F 值较高时则能说明试验方法比较有效。

$$\mathrm{F1} = \frac{2 * \mathrm{Precision} * \mathrm{Recall}}{\mathrm{Precision} + \mathrm{Recall}} \tag{2.5}$$

2.5.2　用户感性指标

推荐系统的新颖性。新颖的推荐是指给用户推荐那些他们以前没有听说过的物品。在一个网站中实现新颖性的最简单办法是，把那些用户之前在网站中对其有过行为的物品从推荐列表中过滤掉。评测新颖度的最简单方法是利用推荐结果的平均流行度，因为越不热门的物品越可能让用户觉得新颖。因此，如果推荐结果中物品的平均热门程度较低，那么推荐结果就可能有比较高的新颖性。

推荐系统的惊喜度。惊喜度（serendipity）是最近这几年推荐系统领域最热门的话题，如果推荐结果和用户的历史兴趣不相似，但却让用户觉得满意，那么就可以说推荐结果的惊喜度很高，而推荐的新颖性仅仅取决于用户是否听说过这个推荐结果。

推荐系统的信任度。信任度只能通过问卷调查的方式，询问用户是否信任推荐系统的推荐结果。提高推荐系统的信任度主要有两种方法。首先需要增加推荐系统的透明度（transparency），而增加推荐系统透明度的主要办法是提供推荐解释。只有让用户了解推荐系统的运行机制，让用户认同推荐系统的运行机制，才会提高用户对推荐系统的信任度。其次是考虑用户的社交网络信息，利用用户的好友信息给用户做推荐，并且用好友进行推荐解释。这是因为用户对他们的好友一般都比较信任，因此如果推荐的主题是好友购买过的，那么他们对推荐结果就会相对比较信任。

2.5.3　其他系统性指标

推荐系统的覆盖度。覆盖度描述了推荐系统对物品长尾的发掘能力，一般通过所有推荐物品占总物品的比例和所有物品被推荐的概率分布来计算，比例越大、概率分布越均匀则覆盖率越大。

推荐系统的多样性。多样性能显著影响用户的体验。用户的兴趣是广泛的，用户可能既喜欢看《猫和老鼠》动漫的主题包，也喜欢看成龙电影的主题包。为了满足用户广泛的兴趣，推荐列表需要能够覆盖用户不同的兴趣领域，即推荐结果需要具有多样性，多样性描述了推荐列表中物品两两之间的不相似性，因此多样性和相似性是对应的。

推荐系统的实时性。有些主题包具有很强的时效性，比如圣诞节、情人节主题包，所以需要在物品还具有时效性时就将它们推荐给用户。推荐系统的实时性包括两个方面。首先，推荐系统需要实时地更新推荐列表来满足用户新的行为变化。实时性的第二个方面是推荐系统需要能够将新加入系统的物品推荐给用户。这主要考验了推荐系统处理物品冷启动的能力。

推荐系统的健壮性，健壮性是指推荐系统对数据异常的可控性，首先给定一个数据集和一个算法，可以用这个算法给这个数据集中的用户生成推荐列表。然后用常用的攻击方法向数据集中注入噪声数据，然后利用算法在注入噪声后的数据集上再次给用户生成推荐列表。最后通过比较攻击前后推荐列表的相似度评测算法的健壮性。如果攻击后的推荐列表相对于攻击前没有发生大的变化，就说明算法比较健壮。

2.6 本章小结

本章简单概述了推荐系统的主要任务和面临着的问题。从学术研究和商业应用两个角度介绍了推荐系统常用的算法，包括协同过滤、SlopeOne 等若干种不同的推荐算法，在实际应用中，例如逻辑回归、协同过滤等算法本来被寄予厚望，却效果不佳；本来不起眼的最近点击和最近关注模型，反而能实现将近 100% 的转化提升，这说明针对自身业务场景搭建推荐系统模型很重要。然后介绍了推荐系统的动态特性和工业界常用的解决方案。最后讨论了推荐系统的主要评测指标，在后面的各章中将会利用这些评测指标对不同的改进模型进行评测。

第三章　用户画像建模

Alan Cooper（交互设计之父）最早提出了用户画像（persona）的概念:“Personas are a concrete representation of target users”。Persona 是真实用户的虚拟代表，是建立在一系列真实数据 (Marketing data，Usability data) 之上的目标用户画像。通过用户历史行为去了解用户，根据他们的目标、行为和观点的差异，将他们区分为不同的类型，然后每种类型中抽取出典型特征，赋予名字、照片、一些人口统计学要素、兴趣标签等描述，就形成了一个人物原型 (personas)，图 3.1所示为一个典型的用户画像，标签面积越大代表其权重越高。一些大公司很喜欢用 personas 做用研究，比如阿里，腾讯，微软等，刻画每个用户，是任何一家社交类型的服务都需要面对的问题，不同的公司针对各自业务会有不同的需求，构建用户画像的动机和目标也会存在一定差异。从手机主题应用商城的角度来讲，构建用户画像的目的包括:

- 完善及扩充用户信息。用户画像的首要动机就是了解用户，这样才能够提供更优质的服务。但是在实际中用户的信息提供得不尽完整，有些是因为平台的引导机制造成的，有时候又是用户不愿意或懒得提供，而且对于用户自行输入的内容又很难进行规范化此外，一些隐性或变化频繁的信息也需要通过用户的行为挖掘出来。

- 打造健康的主题设计生态圈。在掌握用户信息的基础上，平台就可以对自身的状况进行分析，从相对宏观的基础上把握主题市场的生态环境，挖掘设计作品的最大价值，帮助设计师提高收入，图 3.2。例如通过对用户信息的聚类，能够对用户进行人群的划分，掌握不同人群的活跃程度、行为及兴趣偏好，热门主题的传播方式和流行引爆点等。

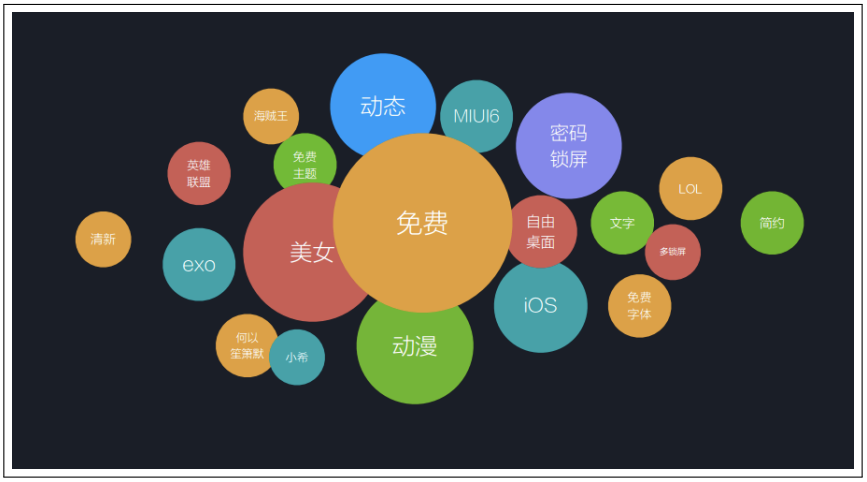

图 **3.1**　用户画像标签化

2015年Q1热销主题TOP10

序号	主题名称	销售金额（元）
1	iOS pro（好评返全款+超级自由桌面）	38万+
2	Forever love（自由桌面）	18万+
3	性感不是罪-琳	15万+
4	梵星Plus 动态星轨锁屏 密码锁屏 v5v6	14万+
5	美iOS(好评返现+强大锁屏+自由桌面）	14万+
6	I watch【至今最帅最酷的锁屏】	13万+
7	我们的爱（荧光闪耀）	9万+
8	ios8+win8(好评返现+双锁屏+自由桌面）	7万+
9	会动LOL英雄	7万+
10	【v6】喰種时代(iOS解锁+音乐界面+自由桌面)	6万+

图 **3.2**　2015 年 Q1 热销主题排行榜

- 支撑主题推荐系统的精准推荐。精准推荐的前提是对用户的清晰认知。以简单代金券发放为例，手机主题应用市场的历史数据呈现出两大类四种不同的消费习惯。代金券敏感型：发代金券才用、发代金券用的更多；代金券不敏感型：发不发都用，发代金券也不用。在推荐系统的用户画像系统中，上述四种群体会被分别冠以屌丝、普通、中产、土豪的标签。针对四类用户的运营策略也会全然不同，最直接的就是代金券的刺激频率以及刺激金额，而对“代金券”免疫的土豪群体，则更多地需要在优化服务上做文章。在实际场景中，影响用户对手机主题包的使用黏度的因素要远比代金券复杂得多，在这种情况下，利用用户画像可以对用户的“贴身跟踪”就能及时发现薄弱环节，因此从用户打开应用商店到退出使用，其间的每一步情况都被快的记录在案：哪一天退出的，哪一步退出的，退出之后“跳转”到什么软件等等。据此，用户画像也实现了用户另外一个纬度的归类，分清哪部分是忠实用户，哪部分可能是潜在的忠实用户，哪些则是已经流失的；更进一步来看流失的原因：因为代金券没有了流失？主题包质量不好流失？这些都是下一步精准推荐的依据。其实手机主题市场中的各项业务都与用户画像有着直接与间接的关系，无论是基于兴趣的推荐提升用户价值，精准的广告投放提升商业价值，还是针对特定用户群体的内容运营，用户画像都是其必不可少的基础支撑。直接地，用户画像可以用于兴趣匹配、关系匹配的推荐和投放；间接地，可以基于用户画像中相似的兴趣、关系及行为模式去推动用户兴趣和设计师的无缝对接。

- 主题市场安全领域的应用。随着手机主题市场的发展，商家会通过各种活动形式的补贴来获取用户、培养用户的消费习惯，但同时也催生一些通过

刷排行榜、刷红包的用户，这些行为距离欺诈只有一步之遥，但他们的存在严重破坏了市场的稳定，侵占了活动的资源。其中一个有效的解决方案就是利用用户画像沉淀方法设置促销活动门槛，即通过记录用户的注册时间、历史登陆次数、常用 IP 地址等，最大程度上隔离掉僵尸账号，保证市场的稳定发展。

用户画像的目的是将用户信息标签化，本文介绍针对主题应用商店本身的特点介绍用户画像的构建，该用户画像主要还是从电子商务的角度出发，完善用户信息和发掘用户兴趣，区分兴趣和购买意愿，并形式化、结构化表达出来。数据的来源也主要是主题平台本身，并没有采用更多的第三方数据。

3.1　用户画像的数据来源

手机主题用户画像的信息来源可以有如下几种方式:

- 显式用户行为。显式方法主要是通过获取用户注册信息中的有关的兴趣和偏好或允许用户自己定义和修改用户画像来实现，一般获取的是用户相对静态和稳定的属性，例如：性别、年龄区间、地域、受教育程度、学校、公司等。主题应用商店本身就有比较完整的用户注册引导、用户信息完善任务、认证用户审核等，在收集和清洗用户属性的过程中，需要注意的主要是标签的规范化以及不同来源信息的交叉验证。

- 隐式用户行为。隐式方法则是通过跟踪用户的行为和交互来评估和推测用户画像，一般获取的是用户更加动态和易变化的兴趣特征，首先，用户兴趣会受到环境、热点事件、季节等方面的影响，一旦这些因素发生变化，用户的兴趣容易产生迁移；其次，用户的行为多样且碎片化，不同行为反映出来的兴趣差异较大。

- 第三方应用数据。一些功能性应用如微信、微博提供的第三方免注册登陆 API 接口，可以直接获取第三方应用账号提供的用户基本数据。

- 自然语言处理技术。利用自然语言处理技术提取用户购买评价、评论语句中的关键词，作为用户画像标签的一部分。

在个性化服务的用户画像建模中，最常用的方式是将以上几种或多种方法结合起来，通过显式方式来获取静态用户信息如姓名、性别、职业等；通过隐式方式来获取动态用户信息如用户兴趣、爱好等；通过第三方登陆接口获取用户的分享、动态信息等；通过自然语言处理技术分析用户的当前心态、满意度、消费心情等。

3.2　标签权重计算

推荐本质上是一种个性化排序，因此在收集到一个用户可能存在的标签后，还需要给标签赋一定的权重，用来区分不同标签对于该用户的重要程度。一个标签对于特定用户的权重值可以大致表示为：标签权重 = (行为类型 + 时空上下文 + 长尾因子) × 时间衰减因子。举例，用户小磊昨天购买了一款 win8 风格的主题包，计算公式如表 3.1所示。

表 **3.1**　标签权重计算公式

标签	win8 风格，比较大众化，长尾因子记为 1
时间	昨天，衰减因子为 0.95。
行为	购买行为，记为权重 5
上下文	用户通过关键字搜索进入，最近几天有多次浏览行为，记为权重 2+2
标签权重	(5+1+4)*0.95=9.5

其中，用户行为类型一般有浏览、添加购物车、搜索、评论、购买、点击赞、收藏等，不同的行为类型具有不同的权重，如购买权重计为 5，浏览计为 1。空间上下文是指用户跳转入口方式，如通过搜索入口权重高一些，排行榜入口低一些，时间上下文是指用户之前是否接触过此类标签，接触频率等。长尾因子是指，如果标签本身是一个非常常见的词，那么它用于刻画用户的兴趣的区分性是比较差的，相反如果是一个长尾词，则区分性较强。出于这样的考虑，越是长尾词，标签的权重值会越高。标签的权重也随着时间的流逝而变化，用户的兴趣会发生转移，时间越久远，标签的权重应该相应的下降，距离当前时间越近的兴趣标签应该得到适当突出。出于这样的考虑，一般会在标签权重值上叠加一个时间衰减函数并体现不同的时效性。此外，针对用户的兴趣，还会设定一个较小的时间窗口来获取用户的短期兴趣，短期兴趣更新周期会较长期兴趣更短，兴趣更集中，但是能够比较及时地反应用户兴趣的变化。实际生产中标签权重计算需要人工参与调整，流程如图 3.3

3.3　用户画像建模方式

根据用户在建模过程中的参与程度，用户兴趣建模可以分为用户手工定制建模、示例用户建模和自动用户建模。其中自动用户建模算法如algorithm 3.1所示。

- 用户手工定制建模。指用户画像由用户自己手工输入或选择的用户建模方法，如用户手工输入感兴趣信息的关键词列表，或者是选择感兴趣的栏目等。在手机主题市场早期，用户手工定制建模是用户建模的主要方法。用户手工定制建模方法实现简单、效果也不错，但它存在以下三个方面问题:(1)

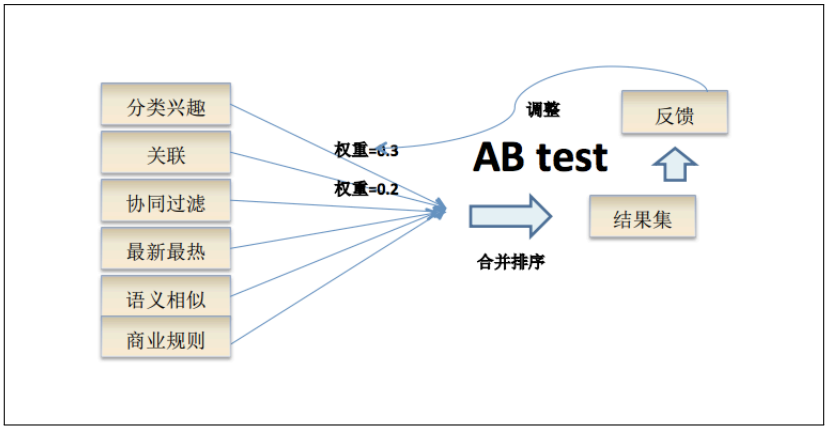

图 **3.3**　abtest 调整标签权重

input ：结构化用户注册信息和用户浏览行为
output: 初始用户兴趣模型

1 从用户购买行为和反馈表单中提取兴趣标签及其对应的权重
2 生成显式兴趣标签表，如” 动漫”：”0.8”,” 汽车”：”0.4”,” 美少女”：”0.9”...
3 根据用户浏览行为获取用户兴趣标签获得隐式权重
4 生成隐式兴趣标签表，如” 免费”：”0.8”,” 特价”：”0.4”,” 热门”：”0.9”...
5 合并显式兴趣向量和隐式兴趣向量到当前用户画像
6 如有新数据，返回第一步，否则跳出循环。

算法 **3.1:** k means

完全依赖于用户，容易降低用户使用系统的积极性。(2) 即使用户乐意手工输入用户画像，用户也难以全面，准确的罗列自己感兴趣的栏目或关键词，导致用户标签的质量有好有坏。(3) 当用户兴趣发生变化时，用户必须重新输入用户画像，这给用户带来了额外的负担。

- 示例用户建模。指由用户提供与自己兴趣相关的示例及其类别属性来建立用户画像的建模方法。由于用户对自己的兴趣和偏好等最有发言权，因而用户提供的有关自己兴趣的示例最能集中、准确地反映用户的兴趣和偏好等特点。示例一般通过要求用户在浏览过程中标注自己的兴趣兴趣度，如喜欢、赞、踩和收藏等。

- 自动用户建模. 指根据用户的浏览内容和浏览行为自动构建用户画像，自动用户建模由于无需用户主动提供信息，不会显示干扰用户，有利于提高个性化服务系统的亲和度，因此，自动建模技术当前用户画像领域热门研究方向。

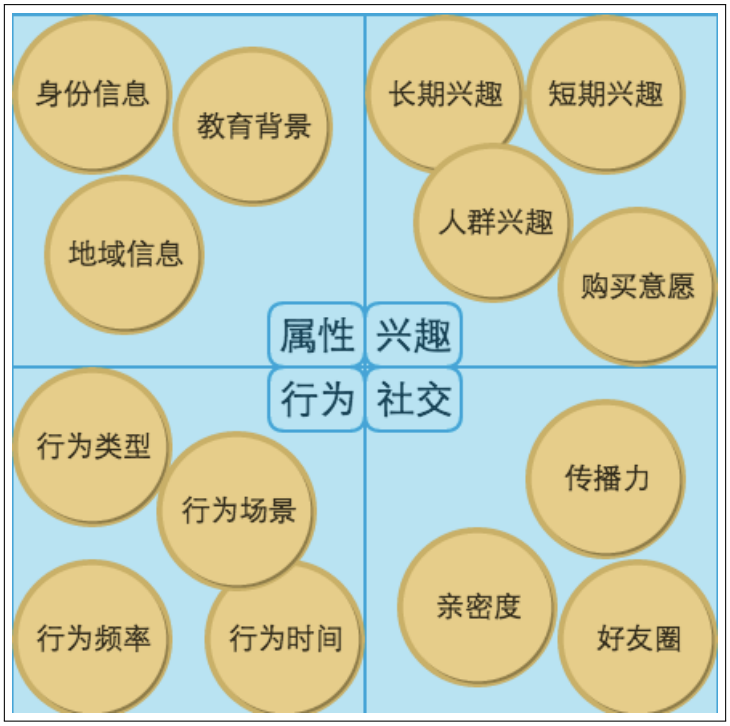

图 **3.4**　用户画像维度划分

3.4　用户画像的维度分析

一个用户可以从多个方面去刻画，也就是说用户画像可以从多个维度来考虑和构建。作为虚拟电子商品交易平台，手机主题市场的用户在平台上通过某些行为（点击、浏览、购买）生产或获取信息，也通过其它一些行为（如转发、评论、赞）将信息传播出去，信息的传播是通过用户之间的社交关系所进行的，并且在生产、消费、传播信息的过程中对信息的选择和过滤体现了用户在兴趣方面的倾向性。由此，我们可以将用户画像按照图 3.4所示的四个维度进行划分，即属性维度、兴趣维度、社交维度和行为维度。用户属性和用户兴趣是传统用户画像中包含的两个维度。前者刻画用户的静态属性特征，例如用户的身份信息（性别、年龄、受教育程度、学校等），后者则用于刻画用户在信息筛选方面的倾向（例如用户的购买能力、兴趣标签、能力标签等）。社交维度是从社交关系及信息传播的角度来刻画用户的。在社区中用户不在仅仅是一个个体，用户和用户之间的社交关系构成了一张网络，信息在这张网络中高速流动，但是这种流动并不是无差别的，信息的起始点，所经历的关键节点以及这些节点构成的关系圈都是影响信息流动的重要因素。行为维度是一个比较新的研究方向，目的是发现影响用户属性、信息变化的行为因素，分析典型用户群体的行为模式。一方面可以通过行为模式的复用来促进用户在手机主题应用平台的成长；另一方面也有利于平台认识用户，和发现新的或异常的用户行为。

3.4.1 属性维度

属性维度属于传统用户画像的范畴，即对用户的信息进行标签化。一方面，标签化是对用户信息进行结构化，方便计算机的识别和处理；另一方面，标签本身也具有准确性和非二义性，也有利于人工的整理、分析和统计。用户属性指相对静态和稳定的人口属性，例如：性别、年龄区间、地域、受教育程度、学校、公司等信息的收集和建立主要依靠产品本身的引导、调查、第三方提供等，在此基础上需要进行补充和交叉验证。

- 标签来源：不是所有的词都适合充当用户标签，这些词本身应该具有区分性和非二义性；此外，还需要考虑来源的全面性，除了用户主动提供的兴趣标签外，用户在使用的过程中的行为，构建的用户关系等也能够反应用户的兴趣，因此也要将其考虑在内。
- 权重计算：得到了用户的兴趣标签，还需要针对用户给这些标签进行权重赋值，用来区分不同标签对于该用户的重要程度。

3.4.2 兴趣维度

由于用户兴趣维度的重要性，因此有一个独立于用户画像模块的兴趣探索模块，下一章节将会详细介绍到。用户兴趣是更加动态和易变化的特征，首先兴趣受到人群、环境、热点事件、行业等方面的影响，一旦这些因素发生变化，用户的兴趣容易产生迁移；其次，用户的行为多样且碎片化，不同行为反映出来的兴趣差异较大，在用户画像建模的过程中，主要考虑如下几个方面：

- 时效性：随着时间的变化，用户的兴趣会发生转移，有些兴趣会贯穿用户使用社交媒体的全过程，而有些兴趣则是受热点时间、环境因素等的影响。
- 长尾性：对于电商领域来讲，那些冷门的用户兴趣的总和可以和那些为数不多的大众化兴趣所占的市场份额相匹配或胜出。
- 兴趣和购买意愿的区分：用户具有某方面的兴趣，只代表了他愿意接受这方面的信息，并不能代表他具有购买相关内容的意愿。例如对于一些只看不买的用户，我们认为其购买意愿很小，因此对其会尽可能多的展示免费主题。

3.4.3 社交维度

如果将主题应用平台的用户视作节点，用户之间的关系视作节点之间的边，那么这些节点和边将构成一个社交的网络拓扑结构，或称作社交图谱。消费信息就是在这个图谱上进行传播。从社交的维度建立用户画像，需要从不同的角度细致和全面地描述这个消费图谱的特征，反应影响信息传播的各层面上的因素，寻找节点之间的关联度，以及刻画图谱本身的结构特征。其中包括：

- 用户个体对消费信息传播的影响：不同用户在信息传播过程中的重要性不一样，影响大的用户对于信息的传播较影响小的用户更具有促进作用。
- 量化用户关系紧密度：存在社交关联的用户，关系越近的用户之间越容易产生相同的消费行为。
- 寻找相似的用户：消费中非对等的关系本身可以认为是一种认证，用户基于兴趣、消费态度等原因反应到线上的一种关联。那么在消费维度上的相似用户至少能反应他们在某种因素上的一致性。
- 识别关系圈：从关系图谱的本身的结构出发，从中发掘关联紧密的群体，有助于促销广告的精准投放和主题包的推广。以上关于关系建模的任务可以看作是逐步深入的，从“个体”–>“关联”–>“相似”–>“群体”的逐渐深入。

3.4.4 行为维度

分析用户的行为，建立行为模式有两个任务：针对典型个体行为进行时序分片，分析用户成长的相关因素；针对典型群体的行为进行统计，为其构建通用的用户画像。

- 典型个体的行为时序分析。所谓典型个体是指某段时间内，成长比较突出的用户。例如从一个新用户从新注册到点击过百、浏览过千需要有一个积累过程，有些用户积累较快，有些较慢，而这些积累较快的用户可以作为典型个体；或者某些用户在某一阶段消费有限，但在某时刻消费激增，无论是消费金额还是数量都变化很大，这种也可以作为典型个体。针对典型个体，需要挖掘与其用户成长相关的行为因素。基本方法是对时间进行分片，获取用户在不同时间片上的行为统计，以及在各个时间分片上的用户成长指标（点击量、购买量、点击转换比等）。在此基础上针对用户行为的统计量的变化，利用关联性分析或回归来分析用户成长与哪些因素有关。
- 典型群体行为模式分析。针对典型个体，从用户的基本信息、人口信息、兴趣维度，可以将相似的典型用户划分为同一的群体，称作典型群体，针对典型群体中的用户按照成长程度进行划分，按不同的成长阶段统计用户行为，即建立了该典型群体的行为模型。例如，对于“年龄在 20 30 岁，女性，付费用户”这样的典型群体，从日点击量、月消费额等维度将其划分到初创、成长、快速提升、成熟等阶段，针对不同成长阶段内的行为组合进行统计，结果构成该群体的行为模式。如图 3.5

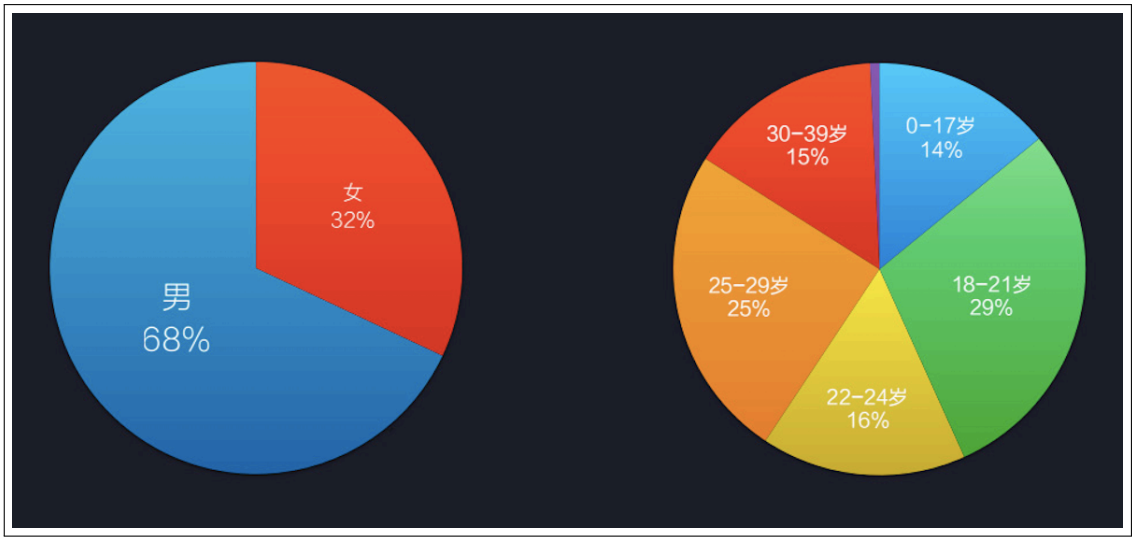

图 **3.5**　手机主题市场用户群体分布

3.5　用户画像应用场景

3.5.1　优化手机主题市场供求

改变了原有的先设计、再销售的传统模式。第三方主题设计师在设计一款新产品前，会先设定好主题类型，然后通过用户画像平台中分析该用户群体的偏好，有针对性的设计产品，从而改变原先新产品高失败率的窘境，增强销售表现。如设计一款智能手表主题，面向 28-35 岁的年轻男性，通过在平台中进行分析，发现属性 =“金属”、风格 =“硬朗”、颜色 =“黑色”/” 深灰色”、价格区间 =“中等”的偏好比重最大，那么就给新产品的设计提供了非常客观有效的决策依据。

3.5.2　提高新人留存率

工商管理有一个理论叫做，维护一个老用户的成本是获取新用户成本的五分之一甚至更低。所以如果能够把一些已经流失的用户召回来，这时候成本比拉一个新用户低得多，你做的事也会带来更大的价值。鉴于此公司启动了一个项目叫” 用户画像之拉新”，首先利用用户画像得出最近一个月没有登录过的用户数据，然后根据浏览时长分档，这是因为用户需要花自己的时间成本才能留下的最有价值的标签，之后利用用户静态标签，像姓名、职业、年龄、地域分布做进一步的细分，最后针对不同类型的用户提供不同的优惠活动。ABtest 显示，与传统一刀切的推荐相比，基于用户画像的拉新留存率提高了约 50%。

3.5.3　用户消费等级分群

大至用户终端品牌、机型、操作系统，细至屏幕分辨率、屏幕尺寸，用户画像记录了每一个用户群体的详细终端特征。哪一类人群最容易被这款应用吸引，愿意为这款应用付费？开发者经常考虑的问题可以从用户画像找到答案。每一

个用户群的价格分布、增值业务费用分布以及流量费用，包括用户详细的消费特征，比如付费频率，丰富了推荐系统的数据依据。

3.5.4 用户流失预警

一般情况用户在消费过程中会经历对如下几个期间：新鲜期，沉迷期，消退期，离开四个阶段，如何能够延长用户在应用的停留周期是需要解决的问题之一。用户画像可以辅助推荐系统进行流失用户特征分析，通过决策数算法，分析流失用户特征，建立不同原因流失的用户模型，然后通过这些特征得到当前在应用活跃用户中匹配流失概率高的用户数据。

3.5.5 反作弊

用户画像会对用户的消费能力、空闲时间、信用评级等维度进行打分；利用反作弊模型通过业务方访问收集数据，供安全部门参考。

3.6 总结

用户画像对于推荐系统来讲，主要如下几个方面的提升：提升推荐系统的精度，用户画像将用户的长期偏好融入到了推荐内容中，维护了推荐系统一致性。abtest 显示，融合了用户画像的推荐模型比单纯的推荐模型在点击转化率指标提高了约 2.8%，考虑到 300 万用户的基数，2.8% 的提升是一个很大的进步；用户画像还解决新用户的冷启动问题，对于一个新注册用户来讲，推荐系统可以利用用户画像的静态信息，然后结合商品信息进行推荐；提高推荐系统的时效性，对用户行为的离线预处理，可以节约推荐系统的大部分计算时间。但是因为用户画像存储的数据都是历史数据，所以其不能实时准确的反映用户兴趣的变化，为了解决这个问题，我们引入了用户兴趣探索模块，将在下一章节件详细介绍。

第四章　用户兴趣探索

电子商务产品的设计往往是数据驱动的，即许多产品方面的决策都是把用户行为量化后得出的。但就商品而言，那些热门主题往往只代表了用户一小部分的个性化需求，只有通过对用户行为的充分分析，才能更好的挖掘出用户的兴趣，最终提升商品的销售量。现有的推荐算法注重用户或资源间的相似性的同时却忽略了用户兴趣的动态变化，从而导致系统在时间维度上有偏离用户需求的趋势。

为了更好的探索用户兴趣，手机主题推荐系统充分利用了用户画像和商品特征表。用户画像包括基本信息和兴趣特征向量, 商品特征向量表包括分类、标签、适用人群等，给定某用户行为，用户兴趣探索过程分为如下几个步骤：首先，利用用户历史行为 (评论, 停留时长, 评分, 点赞, 购买等) 建模量化用户满意度, 然后，利用用户兴趣特征向量与商品特征向量得出相关分数，如果商品与用户的相关分数很低, 但有很高的用户满意度, 说明是一次成功的用户兴趣探索, 更新用户画像。如果是热门商品, 大量的用户都会点击, 但商品与用户不是很相关, 则认为其探索效果是有限的, 反之如果是小众商品, 考虑到长尾效应, 则可以认为其是更成功的兴趣探索。这里涉及到的关键概念包括用户满意度的量化、小众标签的定向挖掘、用户兴趣的动态化。

本章内容首先介绍海量用户行为数据的存储方式。用户行为数据拥有区别于传统数据库数据的特点有，用户行为数据量巨大，常面临 TB 甚至 PB 级的数据；含有较多的噪音；多维聚合式查询。针对这些特征，用户行为数据采用 Hbase 数据集群存储和 hadoop 集群计算。然后，介绍用户兴趣探索的算法模型；然后，介绍如何通过用户行为的分析量化用户满意度。最后，介绍小众兴趣标签的挖掘。

4.1　用户行为数据的存储

手机主题用户行为数据的特点包括：用户基数庞大。手机主题注册用户达千万级，活跃用户达百万级；用户规模增长快。月新注册用户达 10 万数量级。每个用户的行为数量较小。即使是活跃用户，每天最多也只能产生上百条行为记录，每年不超过十万条；用户行为的计算较为复杂。计算用户的两次登录间隔天数、反复购买的商品、累积在线时间，这些都是针对用户行为的计算，通常具有一定的复杂性；用户行为数据格式不规整，字段丢失率较高。根据用户行为数据的这些特点，我们采用基于 Hadop 分布式的架构。Hadop 又如下几个优点：

- 高可靠性。Hadoop 按位存储和处理数据的能力使其具有高可靠性。

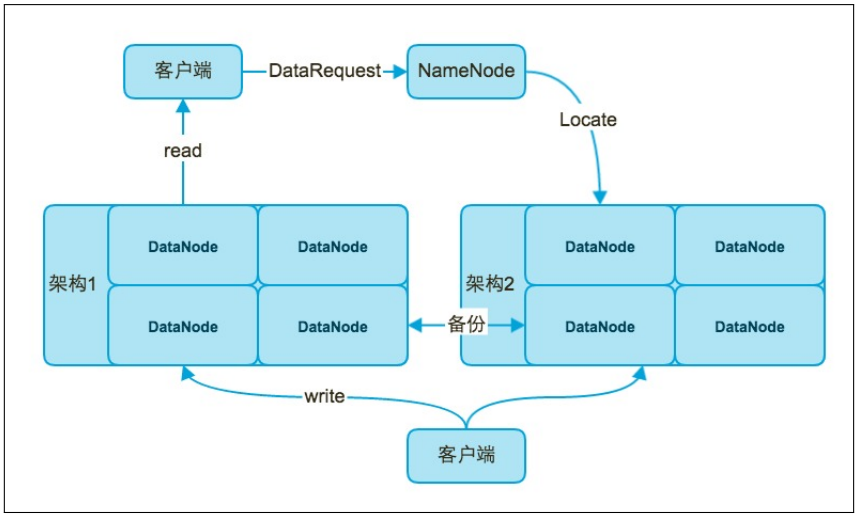

图 **4.1**　HDFS 体系结构

- 高扩展性。Hadoop 是在可用的计算机集簇间分配数据并完成计算任务的，这些集簇可以根据用户增长规模方便地扩展到数以千计的节点中。
- 高容错性。Hadoop 能够自动保存数据的多个副本，并且能够自动将失败的任务重新分配。
- 高效性。Hadoop 能够在节点之间动态地移动数据，并保证各个节点的动态平衡，因此处理速度非常快。
- 低成本。hadoop 是开源的，项目的软件成本因此会大大降低。

Hadoop 的框架最核心的设计是 HDFS 和 MapReduce。HDFS 为海量的数据提供了存储，则 MapReduce 为海量的数据提供了计算。接下来首先介绍 HDFS 的体系架构，然后介绍 MapReduce。

4.1.1　HDFS 的体系架构

HDFS 采用主从 (Master/Slave) 结构模型，一个 HDFS 集群是由一个 NameNode 和若干个 DataNode 组成的 (在最新的 Hadoop2.2 版本已经实现多个 NameNode 的配置)。NameNode 作为主服务器，管理文件系统命名空间和客户端对文件的访问操作。DataNode 管理存储的数据。HDFS 支持文件形式的数据。从内部来看，文件被分成若干个数据块，这若干个数据块存放在一组 DataNode 上。NameNode 执行文件系统的命名空间，如打开、关闭、重命名文件或目录等，也负责数据块到具体 DataNode 的映射。DataNode 负责处理文件系统客户端的文件读写，并在 NameNode 的统一调度下进行数据库的创建、删除和复制工作。NameNode 是所有 HDFS 元数据的管理者，用户数据永远不会经过 NameNode，HDFS 体系结构图如图 4.1所示。

其中，NameNode、DataNode、Client。NameNode 是管理者，DataNode 是文件存储者、Client 是需要获取分布式文件系统的应用程序。HDFS 作为分布式文件系统在数据管理方面设计了多重容错冗余：一个 Block 会有三份备份，一份在 NameNode 指定的 DateNode 上，一份放在与指定的 DataNode 不在同一台机器的 DataNode 上，一根在于指定的 DataNode 在同一 Rack 上的 DataNode 上。备份的目的是为了数据安全，采用这种方式是为了考虑到同一 Rank 失败的情况，以及不同数据拷贝带来的性能的问题。

- 文件写入：首先，Client 向 NameNode 发起文件写入的请求；然后，NameNode 根据文件大小和文件块配置情况，返回给 Client 它管理的 DataNode 的信息；最后，Client 将文件划分为多个 block，根据 DataNode 的地址，按顺序将 block 写入 DataNode 块中。

- 文件读取：首先，Client 向 NameNode 发起读取文件的请求；然后，NameNode 返回文件存储的 DataNode 信息；最后，Client 读取文件信息。

4.1.2　MapReduce 体系架构

MR 框架是由一个单独运行在主节点上的 JobTracker 和运行在每个集群从节点上的 TaskTracker 共同组成。主节点负责调度构成一个作业的所有任务，这些任务分布在不同的不同的从节点上。主节点监视它们的执行情况，并重新执行之前失败的任务。从节点仅负责由主节点指派的任务。当一个 Job 被提交时，JobTracker 接受到提交作业和配置信息之后，就会将配置信息等分发给从节点，同时调度任务并监控 TaskTracker 的执行。JobTracker 可以运行于集群中的任意一台计算机上。TaskTracker 负责执行任务，它必须运行在 DataNode 上，DataNode 既是数据存储节点，也是计算节点。JobTracker 将 map 任务和 reduce 任务分发给空闲的 TaskTracker，这些任务并行运行，并监控任务运行的情况。如果 JobTracker 出了故障，JobTracker 会把任务转交给另一个空闲的 TaskTracker 重新运行。

HDFS 和 MR 共同组成 Hadoop 分布式系统体系结构的核心。HDFS 在集群上实现了分布式文件系统，MR 在集群上实现了分布式计算和任务处理。HDFS 在 MR 任务处理过程中提供了文件操作和存储等支持，MR 在 HDFS 的基础上实现了任务的分发、跟踪、执行等工作，并收集结果，二者相互作用，完成分布式集群的主要任务。Hadoop 上的并行应用程序开发是基于 MR 编程框架。MR 编程模型原理：利用一个输入的 key-value 对集合来产生一个输出的 key-value 对集合。MR 库通过 Map 和 Reduce 两个函数来实现这个框架。用户自定义的 map 函数接受一个输入的 key-value 对，然后产生一个中间的 key-value 对的集合。MR 把所有具有相同的 key 值的 value 结合在一起，然后传递个 reduce 函数。Reduce 函数接受 key 和相关的 value 结合，reduce 函数合并这些 value 值，形成一个较

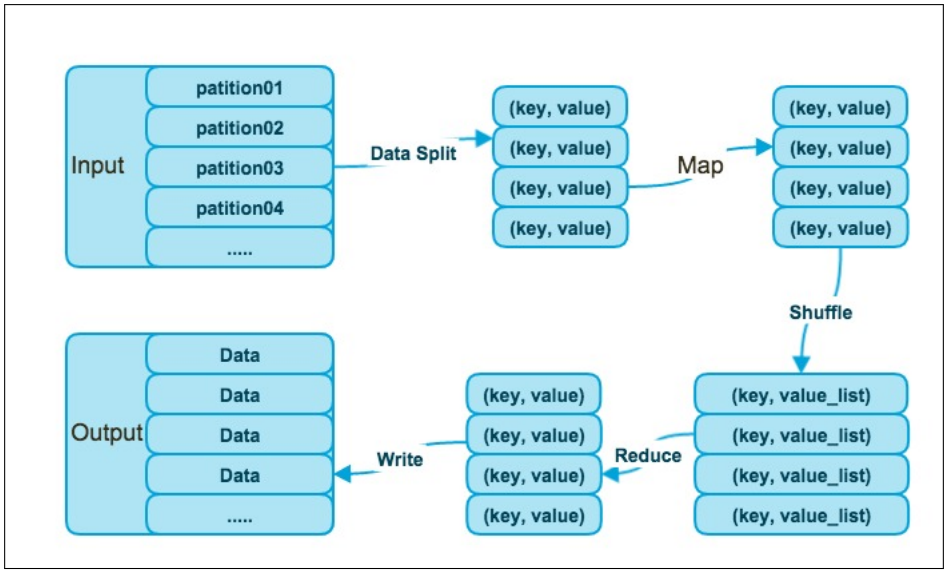

图 **4.2**　MapReduce 数据流

小的 value 集合。通常我们通过一个迭代器把中间的 value 值提供给 reduce 函数，这样就可以处理无法全部放在内存中的大量的 value 值集合了。

MapReduce 体系中数据流动过程：首先，大数据集被分成众多小的数据集块，若干个数据集被分在集群中的一个节点进行处理并产生中间结果。然后，单节点上的任务，map 函数一行行读取数据获得数据的 (k1,v1)，数据进入缓存，通过 map 函数执行 map(基于 key-value) 排序执行后输入 (k2,v2)，有时候在 map 之后 reduce 之前有一个数据合并 (Combine) 操作，即将中间有相同的 key 的对合并，Combine 能减少中间结果 key-value 对的数目，从而降低网络流量。最后，不同机器上的 (k2,v2) 通过 merge 排序的过程，reduce 合并得到，(k3,v3)，输出到 HDFS 文件中。数据流示意图如图 4.2所示。值得一提的是，Map 任务的中间结果在做完 Combine 和 Partition 后，以文件的形式存于本地磁盘上。中间结果文件的位置会通知主控 JobTracker，JobTracker 再通知 reduce 任务到哪一个 DataNode 上去取中间结果。所有的 map 任务产生的中间结果均按其 key 值按 hash 函数划分成 R 份，R 个 reduce 任务各自负责一段 key 区间。每个 reduce 需要向许多个 map 任务节点取的落在其负责的 key 区间内的中间结果，然后执行 reduce 函数，最后形成一个最终结果。有 R 个 reduce 任务，就会有 R 个最终结果，很多情况下这 R 个最终结果并不需要合并成一个最终结果，因为这 R 个最终结果可以作为另一个计算任务的输入，开始另一个并行计算任务。这就形成了多个输出数据片段 (HDFS 副本)。

4.1.3　Hbase 数据管理

Hbase 作为 Hadoop 数据仓库。与传统的 mysql、oracle 还是有很大的差别。NoSql 数据库与传统关系型数据的区别有如下几个方面：

- Hbase 适合大量插入同时又有读的情况。输入一个 Key 获取一个 value 或输入一些 key 获得一些 value。

- Hbase 的瓶颈是硬盘传输速度。Hbase 的操作包括往数据里面 insert 数据，update 的实际上也是 insert，只是插入一个新的时间戳的一行。Delete 数据也是 insert，只是 insert 一行带有 delete 标记的一行。Hbase 的所有操作都是追加插入操作。Hbase 是一种日志集数据库。存储方式像是日志文件一样，是批量大量的往硬盘中写，通常都是以文件形式的读写。所以读写速度就取决于硬盘与机器之间的传输有多快。而 Oracle 的瓶颈是硬盘寻道时间。其经常的操作时随机读写。要 update 一个数据，先要在硬盘中找到这个 block，然后将其读入内存，在内存中的缓存中修改，过段时间再回写回去。硬盘的寻道时间主要由转速来决定的，所以形成了寻道时间瓶颈。

- Hbase 中数据可以保存许多不同时间戳的版本。数据按时间排序，因此 Hbase 特别适合寻找按照时间排序寻找 Top n 的场景。找出某个人最近浏览的主题，最近购买的 N 款主题包，N 种行为等等，因此 Hbase 在互联网应用非常多。

- Hbase 的局限。只能做很简单的 Key-value 查询。它适合有高速插入，同时又有大量读的操作场景。而这种场景又很极端，并不是每一个公司都有这种需求。在一些公司，就是普通的 OLTP(联机事务处理) 随机读写。在这种情况下，Oracle 的可靠性，系统的负责程度又比 Hbase 低一些。而且 Hbase 局限还在于它只有主键索引，因此在建模的时候就遇到了问题。

- Oracle 是行式数据库，而 Hbase 是列式数据库。列式数据库的优势在于数据分析这种场景。数据分析与传统的 OLTP 的区别。数据分析，经常是以某个列作为查询条件，返回的结果也经常是某一些列，不是全部的列。在这种情况下，行式数据库反应的性能就很低效。

4.1.4　Hive 数据管理

Hive 是建立在 Hadoop 上的数据仓库基础架构。它提供了一系列的工具，用来进行数据提取、转换、加载，是一种可以存储、查询和分析存储在 Hadoop 中的大规模数据机制。可以把 Hadoop 下结构化数据文件映射为一张成 Hive 中的表，并提供类 sql 查询功能，除了不支持更新、索引和事务，sql 其它功能都支持。可以将 sql 语句转换为 MapReduce 任务进行运行，作为 sql 到 MapReduce 的映射器。提供 shell、JDBC/ODBC、Thrift 等接口。优点是成本低可以通过类 sql 语句快速实现简单的 MapReduce 统计。作为一个数据仓库，Hive 的数据管理按照使用层次可以从元数据存储、数据存储和数据交换三个方面介绍。

- 元数据存储。Hive 将元数据存储在 RDBMS 中，有三种方式可以连接到数据库：1）内嵌模式：元数据保持在内嵌数据库的 Derby，一般用于单元测试，只允许一个会话连接；2）多用户模式：在本地安装 Mysql，把元数据放到 Mysql 内；3）远程模式：元数据放置在远程的 Mysql 数据库。

- 数据存储。首先，Hive 没有专门的数据存储格式，也没有为数据建立索引，用于可以非常自由的组织 Hive 中的表，只需要在创建表的时候告诉 Hive 数据中的列分隔符和行分隔符，这就可以解析数据了。其次，Hive 中所有的数据都存储在 HDFS 中，Hive 中包含 4 中数据模型：Tabel、ExternalTable、Partition、Bucket。Table 类似与传统数据库中的 Table，每一个 Table 在 Hive 中都有一个相应的目录来存储数据。例如：一个表 zz，它在 HDFS 中的路径为：/wh/zz，其中 wh 是在 hive-site.xml 中由用户设定的数据仓库的目录，所有的 Table 数据 (不含 External Table) 都保存在这个目录中。Partition 类似于传统数据库中划分列的索引。在 Hive 中，表中的一个 Partition 对应于表下的一个目录，所有的 Partition 数据都存储在对应的目录中。例如：zz 表中包含 ds 和 city 两个 Partition，则对应于 ds=20140214，city=beijing 的 HDFS 子目录为：/wh/zz/ds=20140214/city=Beijing;Buckets 对指定列计算的 hash，根据 hash 值切分数据，目的是为了便于并行，每一个 Buckets 对应一个文件。将 user 列分数至 32 个 Bucket 上，首先对 user 列的值计算 hash，比如，对应 hash=0 的 HDFS 目录为：/wh/zz/ds=20140214/city=Beijing/part-00000; 对应 hash=20 的，目录为：/wh/zz/ds=20140214/city=Beijing/part-00020。ExternalTable 指向已存在 HDFS 中的数据，可创建 Partition。和 Table 在元数据组织结构相同，在实际存储上有较大差异。Table 创建和数据加载过程，可以用统一语句实现，实际数据被转移到数据仓库目录中，之后对数据的访问将会直接在数据仓库的目录中完成。删除表时，表中的数据和元数据都会删除。ExternalTable 只有一个过程，因为加载数据和创建表是同时完成。世界数据是存储在 Location 后面指定的 HDFS 路径中的，并不会移动到数据仓库中。

- 数据交换。用户接口包括客户端、Web 界面和数据库接口，元数据通常存储在关系数据库如 Mysql，Derby 中。

本小节主要介绍了 Hadoop 分布式计算平台最核心的分布式文件系统 HDFS、MapReduce 处理过程，以及数据仓库工具 Hive 和分布式数据库 Hbase，基本涵盖了 Hadoop 分布式平台的所有技术核心。从体系架构到数据定义到数据存储再到数据处理，Hadoop 分布式存储、计算平台为海量用户行为的分析和用户兴趣探索提供了可能。接下来的章节先介绍用户行为数据的分析，包括数据预处理和异常数据监测，然后介绍用户兴趣探索模块，包括算法模型、用户满意度量化、小众兴趣标签的挖掘。

4.2　用户行为数据的的预处理

数据预处理是数据挖掘过程中一个重要步骤，当原始数据存在不一致、重复、含噪声、维度高等问题时，更需要进行数据的预处理，以提高数据挖掘对象的质量，最终达到提高数据挖掘所获模式知识质量的目的。

4.2.1　背景

随着手机主题市场交易规模的逐步增大，积累下来的业务数据和用户行为数据越来越多，这些用户数据往往是电子商务平台最宝贵的财富。目前在手机主题推荐系统中大量地应用到了机器学习和数据挖掘技术，例如个性化推荐、搜索排序、用户画像建模等等，为企业创造了巨大的价值。本节主要介绍在用户兴趣探索实践中的数据预处理与特征挖掘方法。数据预处理主要工作是:

- 从原始数据，如文本、图像或者应用数据中清洗出特征数据和标注数据
- 对清洗出的特征和标注数据进行处理，例如样本采样，样本调权，异常点去除，特征归一化处理等过程。最终生成的数据主要是供模型直接使用。

4.2.2　特征提取

用户兴趣探索的任务包括: 探索用户的兴趣广度、兴趣深度、兴趣变动趋势。依据这些信息，推荐系统就能知道在面对某一个用户时要推荐哪几类型商品，每类商品所占的比例，未来几天推荐内容会有哪些变化。在确定了目标之后，接下来需要确定使用哪些数据来达到目标。提取哪些特征数据可能与用户是否点击购买相关，一方面可以借鉴一些业务经验，另一方面可以采用一些特征选择、特征分析等方法。从业务经验来判断，可能影响用户是否点击下单的因素有:

- 用户历史行为。对于老用户，之前可能有过点击、购买等行为。
- 用户实时兴趣。
- 用户满意度。上面的特征都是比较好衡量的，用户满意度可能是更复杂的一个特征，具体体现在用户评分、评价、购买后使用频率、时长等。
- 是否热门，商品评价人数，购买数等。

在确定好要使用哪些数据之后，还需要对使用数据的可用性进行评估，包括数据的获取难度，数据的规模，数据的准确率，数据的覆盖率等。

- 用户历史行为。只有老用户才会有行为，新用户是没有的。
- 数据获取难度。获取用户 id 不难，但是获取用户年龄和性别较困难，因为用户注册或者购买时，这些并不是必填项，即使填了也不完全准确。如果一些特征需要通过其他预测模型交叉验证的话，就存在着模型精度的问题。

- 数据覆盖率。数据覆盖率也是一个重要的考量因素，例如地理位置特征，并不是所有用户的距离我们都能获取到，PC 端的就没有地理位置，还有很多用户禁止使用它们的定位功能。

- 用户实时行为。如果用户刚打开 app，还没有任何行为，同样面临着一个冷启动的问题。

- 数据的准确率。有时候用户购买一款主题，不一定是其真心喜欢，可能是因为遇到限时半价、购买返现等活动。

4.2.3 特征获取方式

特征提取方式分为在线提取和离线提取。

离线特征获取方案。离线可以使用海量的数据，借助于分布式文件存储平台，例如 HDFS 等，使用例如 MapReduce，Spark 等处理工具来处理海量的数据等。

在线特征获取方案。在线特征比较注重获取数据的延时，由于是在线服务，需要在非常短的时间内获取到相应的数据，对查找性能要求非常高，可以将数据存储在索引、key-value 存储等，也可以使用 Kafka 等处理工具。Kafka 是一种分布式的，基于发布/订阅的消息系统。主要设计目标如下：1）以时间复杂度为 O(1) 的方式提供消息持久化能力，即使对 TB 级以上数据也能保证常数时间的访问性能；2）高吞吐率。即使在非常廉价的商用机器上也能做到单机支持每秒 100K 条消息的传输；3）同时支持离线数据处理和实时数据处理；4）分布式系统，易于向外扩展。所有的 producer、broker 和 consumer 都会有多个，均为分布式的。无需停机即可扩展机器。

4.2.4 用户行为数据预处理

根据不同业务数据的预处理方式也不同，一般来讲原始服务器日志数据脏数据的形成原因包括：缩写词不统一，数据输入错误，不同的惯用语，重复记录，丢失值，不同的计量单位，过时的编码等。相应的，数据预处理内容包括数据清理、数据集成、数据变换、数据归约、数据离散化。

1）数据清理包括格式标准化、异常数据清除、错误纠正、重复数据的清除。对于手机主题用户数据来讲，引起空缺值的原因主要是用户设备异常造成的，有些时候是因为与其他已有数据不一致而被删除或数据的改变没有进行日志记载。根据数据空缺情况的不同有不同的处理方式：

- 忽略元组。当一个记录中有多个属性值空缺、特别是关键信息丢失时，已不能反映真实情况，它的效果非常差。

- 去掉属性。缺失严重时，已无挖掘意义。

- 人工填写空缺值。但是工作量大且可行性低。
- 默认值。比如使用 unknown 或-∞。
- 使用属性的平均值填充空缺值。
- 预测最可能的值填充空缺值。使用贝叶斯公式或判定树这样的基于推断的方法。

2）数据集成就是将多个数据源中的数据整合到一个一致的存储中，需要注意以下几个情况：

- 模式集成。整合不同数据源中的元数据时的实体识别问题，比如匹配俩个表中的用户 ID，A.custId=B.customerNo。
- 检测/解决数值冲突。对现实世界中的同一实体, 来自不同数据源的属性值可能有所不同，如同表示停留时长，A 表单位是秒，B 表单位为毫秒。
- 多表之间的数据冗余。同一属性在不同的数据库中会有不同的字段名，有些时候冗余可以被相关分析检测出来，计算公式如所示，其中 $\overline{A}$ 和 $\overline{B}$ 表示为字段 A 和 B 的平均值，$\sigma_A \sigma_B$ 表示其的标准差。仔细将多个数据源中的数据集成起来, 能够减少或避免结果数据中的冗余与不一致性, 从而可以提高挖掘的速度和质量。

$$r_{A,B} = \frac{\sum(A - \overline{A})(B - \overline{B})}{(n-1)\sigma_A \sigma_B} \tag{4.1}$$

3）数据变换包括数据的平滑变换、数据聚集和数据规范化。所谓规范化是指将数据按比例缩放, 使之落入一个小的特定区间，有如下几种方式：

- 最小-最大规范化。如式 4.2所示，原始数值范围为 [min,max]，通过公式映射到新区间 [newMin,newMax]，v' 表示属性 v 的公式映射。

$$v' = \frac{v - min}{max - min}(newMax - newMin) + newMin \tag{4.2}$$

- z-score 规范化。Z-score 表示原始数据偏离均值的距离长短，而该距离度量的标准是标准方差，如果统计数据量足够多，Z-score 数据分布可以满足，68% 的数据分布在“-1”与“1”之间，95% 的数据分布在“-2”与“2”之间，99% 的数据分布在“-3”与“3 之间”。如式 4.3所示，其中 v 是原始数据，v’ 为 v 的映射，mean 是全部数据的均值，σ 为标准方差。

$$v' = \frac{v - mean}{\sigma} \tag{4.3}$$

- 小数定标规范化，小数定标规范化通过移动数据 A 的小数点位置进行规范化，小数点的移动位置依赖数据 A 的最大值。如式 4.4所示，其中 j 是使 $Max(|v'|) < 1$ 的最小整数。

$$v' = \frac{v}{10^j} \tag{4.4}$$

4）数据归约是数据字段从源数据集中得到数据集的归约表示。数据仓库中往往存有海量数据, 在其上进行复杂的数据分析与挖掘需要很长的时间，通过数据归约使得数据小得多, 且可以产生相同的分析结果，需要注意的是用于数据归约的时间不应当超过或“抵消”在归约后的数据上挖掘节省的时间。数据归约策略包括:

- 数据立方体聚集。最底层的方体对应于基本方体，基本方体对应于感兴趣的实体。在数据立方体中存在着不同级别的汇总，每次较高层次的抽象将进一步减少结果数据。数据立方体提供了对预计算的汇总数据的快速访问，所有尽可能对于汇总数据的查询使用数据立方体。

- 维归约，通过删除不相干的属性或维减少数据量，维归约又属性子集选择和启发式俩种实现方式。属性子集选择是指找出最小属性集, 使得数据类的概率分布尽可能的接近使用所有属性的原分布，同时减少出现在发现模式上的属性的数目, 使得模式更易于理解。启发式的方法有：逐步向前选择；逐步向后删除；向前选择和向后删除相结合；判定归纳树；基于统计分析的归约如主成分分析、回归分析等。

6) 数据离散化, 即连续属性的范围划分为区间，减少给定连续属性值的个数，区间的标号可以代替实际的数据值。

- 概念分层。通过使用高层的概念替代底层的属性值，如用青年、中年、老年代替年龄数据值。

- 分箱（binning）。分箱技术递归的用于结果划分, 可以产生概念分层。

- 直方图分析（histogram）。直方图分析方法递归的应用于每一部分, 可以自动产生多级概念分层。

- 聚类分析。将数据划分成簇, 每个簇形成同一个概念层上的一个节点, 每个簇可再分成多个子簇, 形成子节点。

- 基于熵的离散化。

- 自然划分。将数值区域划分为相对一致的、易于阅读的、看上去更直观或自然的区间，划分步骤：如果一个区间最高有效位上包含 3,6,7 或 9 个不同的值, 就将该区间划分为 3 个等宽子区间；如果一个区间最高有效位上包

含 2,4, 或 8 个不同的值, 就将该区间划分为 4 个等宽子区间；如果一个区间最高有效位上包含 1,5, 或 10 个不同的值, 就将该区间划分为 5 个等宽子区间；将该规则递归的应用于每个子区间, 产生给定数值属性的概念分层；对于数据集中出现的最大值和最小值的极端分布, 为了避免上述方法出现的结果扭曲, 可以在顶层分段时, 选用一个大部分的概率空间，如 5%-95%。

4.3　用户兴趣探索的算法模型

用户兴趣探索就是不断学习用户所感兴趣的内容反馈给个性化推荐模型去加强推送相关内容，本节首先介绍用户兴趣模型的基本概念，然后介绍算法模型的组成结构：用户异常兴趣探测，用户小众兴趣标签的挖掘和用户满意度量化，用户兴趣衰减算法。

4.3.1　基本概念概述

实体域。当我们想基于用户行为分析来建立用户兴趣模型时，我们必须把用户行为和兴趣主题限定在一个实体域上。个性化推荐落实在具体的推荐中都是在某个实体域的推荐。对于手机主题应用市场来说，实体域包括所有的主题，背景图片，铃声，闹铃等。

用户行为。浏览，点击，下载，试用，购买，评论等都可是用户行为。本文所指的用户行为都是指用户在某实体域上的行为。比如用户在手机铃声产生的行为。用户兴趣。用户的兴趣维度，同样是限定在某实体域的兴趣，通常以标签 + 权重的形式来表示。比如，对于手机主题，用户兴趣向量可以是「动漫，0.6」,「体育，0.1」,「情感，0.7」等分类标签。值得一提的是，用户兴趣只是从用户行为中抽象出来的兴趣维度，并无统一标准。而兴趣维度的粒度也不固定，如「体育」,「电影」等一级分类，而体育下有「篮球」,「足球」等二级分类，篮球下有「NBA」,「CBA」,「火箭队」等三级分类。我们选取什么粒度的兴趣空间取决于具体业务模型。

兴趣空间。在同一层次上兴趣维度的集合，比如手机主题中，可以用「热门」,「游戏」,「限时特价」,「科技」来构成一个程序员兴趣标签空间，也可以用「二次元」,「萝莉」,「魔幻」,「纯真」,「召唤兽」·····「法术」等构成一个动漫兴趣标签空间。

4.3.2　用户异常兴趣探测算法

统计学中的数据异常值是一个测量变量中的随机错误或偏差，信息安全学中的数据异常是指引起不正确属性值的原因包括恶意 hack 行为、数据输入等。传统的的异常兴趣检测是基于统计。首先根据现有用户画像建立数据统计模型，异常是那些模型不能完美拟合或是相对远离预测值的对象，对于常用的回归模型，如图 4.3所示。但缺点是用户兴趣概率分布模型计算比较耗费计算资源。

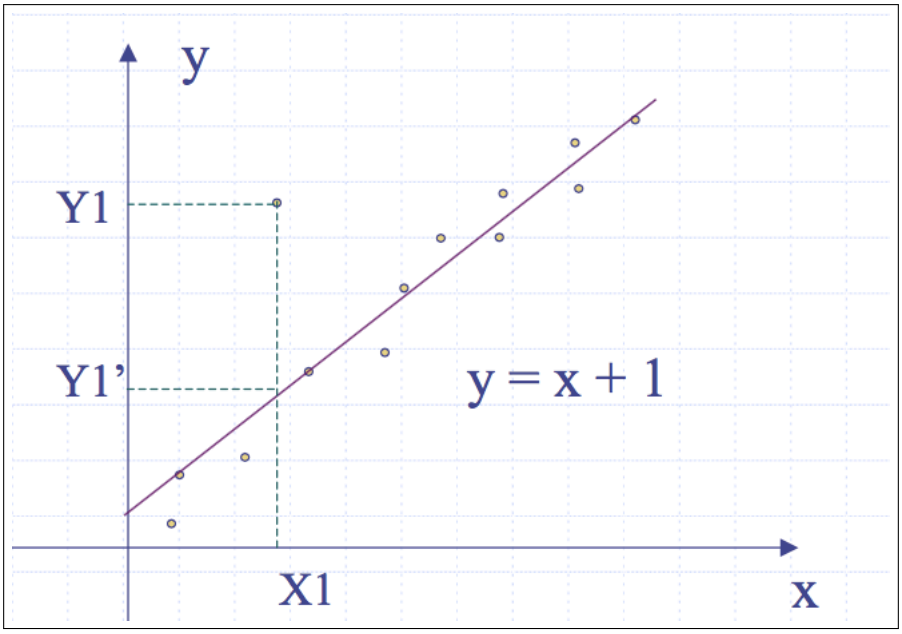

图 **4.3**　回归异常值检测

本文中涉及到的异常是指用户兴趣行为的差异化趋势，比如，用户小磊每次都会浏览动漫、美少女主题，但是有一天却购买了一款汽车手机主题，那么程序可以检测到这个异常情况，然后将汽车标签更新到用户画像中，并作为个性化推荐的依据。事实上用户兴趣迭代过程可以在很短的时间内完成，基于 hive + MapReduce 平台的时长维度为天，而基于 kafka + spark 平台可以将时长维度降到小时级别。用户异常兴趣检测算法要从用户的行为和偏好中发现新的兴趣标签，并基于此给予推荐，工作内容包括收集用户的最新的行为数据是并分析得出异常标签，算法如algorithm 4.1所示

Input: 用户画像数据 userProfile , 用户显示、隐式行为数据 logUsers
Output: 用户异常兴趣标签 newUsersTags

```
1  init newUsersTags;
2  for (user_i in logUsers) do
3      for (tag_j in user_i) do
4          if (tag_j.weight == 0) then
5              //若标签权重已经为 0, 该用户兴趣标签将被删除
6              remove tag_j;
7          end
8          else
9              //成功探测到新用户兴趣标签
10             temp.get(user_i).set(tag_j);
11         end
12     end
13 end
14 return temp;
```

算法 **4.1:** 用户异常兴趣探测

4.3.3　长尾标签抽取算法

标签集中度 (tagFocus) 是指，如果某个标签在一类主题中出现的频率高，其他主题类型很少出现，则认为此兴趣标签具有很好的类别区分能力。这是因为包含兴趣标签 t 的主题越少，也就是 n 越小，则说明标签 t 具有很好的兴趣区分，则其探索权重越大。如果某一类主题包 C 中包含兴趣标签 t 的个数为 tagInThemeNum，而其它类包含 t 的总数为 tagInOtherNum，则所有包含 t 的主题数 n=tagInThemeNum+tagInOtherNum，当 m 大的时候，n 也大，标签权重值会小，就说明该标签 t 类别区分能力不强。实际上，如果一个标签在一个类的主题中频繁出现，则说明该标签能够很好代表这类主题的特征，这样的标签应该给它们赋予较高的权重，并选来作为该类主题的特征向量以区别于其它类主题。热度 (tagPopular) 指的是某一个给定标签在用户画像中出现的频率。例如在 300 万用户总数中，十分之一的用户标签中有” 火影” 标签，那么其热度为 0.1，除此之外有些标签如” 精品”，” 气质” 等标签占了总词频的 80% 以上，而它对区分主题类型几乎没有用。我们称这种词叫“应删标签”。即应删除词的权重应该是零，也就是说在度量相关性是不应考虑它们的频率。标签集中度公式如式 4.5所示，我们很容易发现，如果一个标签只在很少的主题包中出现，我们通过它就容易锁定搜索目标，它的权重也就应该大。反之如果一个词在大量主题包中出现，我们看到它仍然不很清楚要找什么内容，因此它应该小。热度公式如式 4.6所示。长尾标签抽取算法如algorithm 4.2所示。

$$\mathrm{tagFocus} = \log \frac{|\mathrm{tagInThemeNum}|}{|\mathrm{tagInThemeNum} + \mathrm{tagInOtherNum}|} \tag{4.5}$$

$$\mathrm{tagPopular} = \log \frac{|\mathrm{peopleLikeTagNum}|}{|\mathrm{allPeople}|} \tag{4.6}$$

4.3.4　用户满意度量化算法

要从用户的行为和偏好中量化用户满意度，并基于此实现兴趣标签探索，如何收集用户的偏好行为成为用户兴趣探索效果最基础的决定因素。用户有很多方式向系统提供自己的偏好信息，而且不同的应用也可能大不相同。表 4.1列举的用户行为都是比较通用的，设计人员也可以根据实际情况添加特殊的用户行为，并用他们表示用户对物品的喜好。一般来讲我们提取的用户行为一般都多于一种，根据不同行为反映用户喜好的程度将它们进行加权，得到用户对于物品的总体喜好。显式的用户反馈比隐式的权值大，但比较稀疏，毕竟进行显示反馈的用户是少数；而隐式用户行为数据是用户在使用应用过程中产生的，它可能存在大量的噪音和用户的误操作，我们可以通过经典的数据挖掘算法过滤掉行为数据中的噪音，这样可以是我们的分析更加精确。然后是归一化操作，因为不同行为的数据取值可能相差很大，比如，用户的浏览数据必然比购买数据大的多，

```
Input: 用户画像数据 userProfile , 用户显示、隐式行为数据 logUsers
Output: 长尾兴趣标签 longTailTags
init longTailTags;
for (user_i in logUsers) do
    for (tag_j in user_i) do
        weight_ij=tag_j.tagFocus/tag_j.tagPopular;
        if (weight_ij <= threshold) then
            //若标签权重小于阈值, 该用户兴趣标签将被删除
            remove tag_j;
        end
        else
            //成功探测到新用户兴趣标签
            longTailTags.get(user_i).set(tag_j);
        end
    end
end
return longTailTags;
```

算法 **4.2:** 长尾兴趣探测

如何将各个行为的数据统一在一个相同的取值范围中，从而使得加权求和得到的总体喜好更加精确，就需要我们进行归一化处理使得数据取值在 [0,1] 范围中。算法如algorithm 4.2所示。

```
Input: 用户显示、隐式行为数据 logUsers
Output: 用户行为权重 userActionWeight
init userActionWeight;
for (user_i in logUsers) do
    for (action_j in user_i) do
        //获取用户此次行为的偏好权重并做归一化
        weight_ij=getWeigth(action_j); if
        (user_i exsit in userActionWeight) then
            //对用户行为做加权处理。
            double remaind=userActionWeight.get(user_i);
            userActionWeight.get(user_i).set(remaind+action_j * weight_j);
        end
        else
            userActionWeight.get(user_i).set(action_j * weight_j);
        end
    end
end
return userActionWeight;
```

算法 **4.3:** 用户满意度量化算法

表 **4.1**　用户行为和其权重

用户行为	类型	特征	作用	权重
评分	显式	整数量化的偏好，可能的取值是 [0, 5]	通过用户对物品的评分，可以精确的得到用户的满意度，但是噪声比较大，比如遇到好评返现活动	1
分享	显式	布尔量化的偏好，取值是 0 或 1	通过用户对物品的投票，可以精确的得到用户的喜好度，同时可以推理得到被转发人的兴趣取向（不太精确）	2
评论	显式	一段文字，需要进行文本分析，得到偏好	通过分析用户的评论，可以得到用户的情感：喜欢还是讨厌	1
赞/踩	显示	布尔量化的偏好，取值是 0 或 1	带有很强的个人喜好度	3
购买、试用	显式	布尔量化的偏好，取值是 0 或 1	用户的购买是很明确的说明这个项目它感兴趣。	3
点击流	隐式	包括滑屏频率，滑屏次数，屏停留时长，用户对物品感兴趣，需要进行分析，得到偏好	用户的点击一定程度上反映了用户的注意力，所以它也可以从一定程度上反映用户的喜好。	1
停留时长	隐式	一组时间信息，噪音大，需要进行去噪，分析，得到偏好	用户的页面停留时间一定程度上反映了用户的注意力和喜好，但噪音偏大，不好利用。比如说用户在浏览一个主题的时候，丢下手机和同学出去踢球去了，页面提留时长可能会很长	1

4.3.5　标签权重的线性衰减

实际中使用的是基于时间衰减的用户兴趣检测模型。该算法的特点是基于向量空间模型的用户画像建模，结合手机主题用户兴趣偏好变化频繁的特点，根据时间因素权重自动进行衰减，以此反映出用户兴趣的变化。该模型是指用户对资源项目的评分仅代表评价当时的兴趣度，随着时间的推移，用户对该资源项目的评分将规律性地自动衰减，当项目评分衰减到 0 时，该资源项目将被兴趣模型所淘汰。评分衰减可以按照线性规律进行，如图 4.4所示。算法描述如algorithm 4.4所示。

4.4　用户兴趣探索评估方法

评价方法分为以下俩种：线下测试和线上测试。首先介绍线下测试基本概念，然后具体介绍工业界常用的线上 A/B 测试。

```
Input: 用户画像模型中所有用户兴趣哈希表 users
Output: 更新后的所有用户兴趣哈希表 users
for (user_i in users) do
    //考虑到断电或意外关机等原因导致系统中断运行的特殊情况，算法加入了临时变量 temp
    temp.add(user_i.profile);
end
for (user_i in logUsers) do
    for (tag_j in user_i) do
        if (tag_j == 0) then
            //若标签权重已经为 0, 该用户兴趣标签将被删除
            remove tag_j;
        end
        else if (tag_j exsit in temp(user_i)) then
            //若存在新的评分值, 则更新为新标签权重
            temp.get(user_i).set(tag_j);
        end
        else
            //否则的话，将偏好值减少 0.5，进行衰减
            temp.get(user_i).set(tag_j-0.5);
        end
    end
end
return temp;
```

算法 **4.4:** 用户画像线性衰减

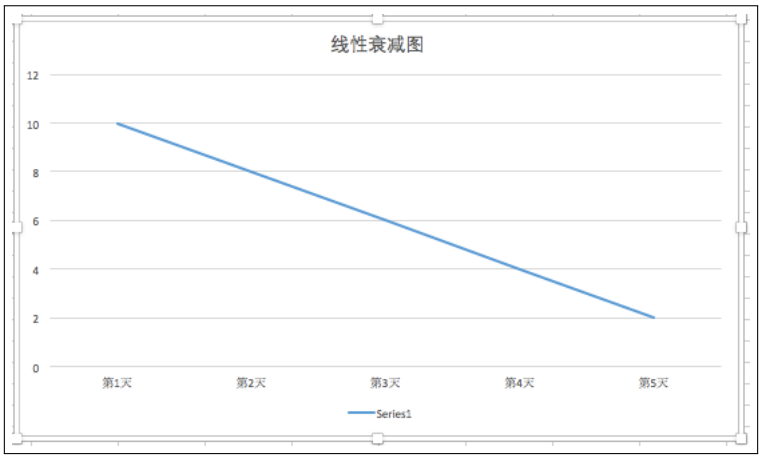

图 **4.4**　线性衰减模型

4.4.1 线下测试

笔者曾参加过 2014 年阿里举办的大数据竞赛，当时主要用线下测试评估算法模型优劣，总结出的基本准则是：不要过早优化模型调参和模型融合，这两部分应当留到中后期来做；不要一次性添加大量特征，最好一部分一部分添加，这样会对添加的特征效果有个大体的认识。线下测试具体步骤如下：

- 选定数据集选择和应用相关的数据集。数据需要是无偏的（unbiased），通过随机抽样能满足要求。将现有数据集分成训练集 (train set) 验证集 (validation set) 测试集 (test set)。其中训练集用来估计模型，验证集用来确定网络结构或者控制模型复杂程度的参数，而测试集则检验最终选择最优的模型的性能如何。一个典型的划分是训练集占总样本的 50 %，而其它各占 25 %。样本少的时候可以留少部分做测试集。然后对其余 N 个样本采用 K 折交叉验证法。就是将样本打乱，然后均匀分成 K 份，轮流选择其中 K - 1 份训练，剩余的一份做验证，计算预测误差平方和，最后把 K 次的预测误差平方和再做平均作为选择最优模型结构的依据。

- 建立算法模型.

- 准确度的评估、反馈。准确度的评估方法有 Mean Absolute Error 和 Root Mean Squared Error，对于一个元素是 user-item 对 (u，i) 的集合 T，实际评分为 r_{ui}，预测评分为 $\hat{r}_{ui}$，无论是通过 MAE 还是 RMSE 计算，最终的结果值越小证明结果越准确。但从公式可以看出，RMSE 通过平方扩大了偏离量，同样的两组结果用 RMSE 得出的差异值将比 MAE 更大。对应公式如下：

$$MAE = \frac{1}{|T|} \sum_{(ui)\in T} |\hat{r}_{ui} - r_{ui}| \tag{4.7}$$

$$RMSE = \sqrt{\frac{1}{|T|} \sum_{(ui)\in T} (\hat{r}_{ui} - r_{ui})^2} \tag{4.8}$$

4.4.2 线上 A/B 测试

在太平洋东部加拉帕戈斯（Galapagos）的一个小岛上有一种名叫达尔文雀的鸟，一部分生活在岛的西部，另一部分生活在岛的东部，由于生活环境的细微不同它们进化出了不同的喙，如图 4.5所示，这被认为是自然选择学说上的一个重要例证。同样一种鸟，究竟哪一种喙更适合生存呢？自然界给出了她的解决方案，让鸟儿自己变异（设计多个方案），然后优胜劣汰。具体到达尔文雀这个例子上，不同的环境中喙也有不同的解决方案。上面的例子包含了 A/B 测试最核心的思想：多个方案并行测试；每个方案只有一个变量（比如鸟喙）不同；以某种规则优胜劣汰。评判用户画像模型的效率高低，主要是看该模型带来的点击

图 **4.5**　达尔文雀

率、转换率等指标数据，其他统计量见表 4.2所示。理论上评测推荐系统的指标有用户满意度、预测准确度、覆盖率、多样性、新颖度、惊喜度、信任度、实时性、健壮性等。然而商业开发中，评测推荐结果只看重一个指标：点击转化率。能够提升商业价值，给业务带来更多利益的推荐系统，就是好的推荐系统。

表 **4.2**　A/B 测试主要评估指标

指标	描述
访客数	访客数就是指一天之内到底有多少不同的用户访问了你的网站。访客数要比 IP 数更能真实准确地反映用户数量。
浏览量	即 Page View，浏览量和访问次数是呼应的。用户访问网站时每打开一个页面，就记为 1 个 PV。同一个页面被访问多次，浏览量也会累积。
点击转化率	点击转化率计算公式: 点击转化率 = 成交笔数/浏览量 *100%，成交笔数影响着成交金额，所以点击转化率成为了衡量推荐系统效果的重要数据之一。
停留时长	停留时长是用户访问网站的平均停留时间，是衡量网站用户体验的一个重要指标。如果用户不喜欢主题包的内容，可能稍微看一眼就关闭页面了，那么停留时长就很短; 如果用户对页面的内容很感兴趣，停留时长就很长。
跳出率	跳出率是指访客来到网站后，只访问了一个页面就离开网站的访问次数占总访问次数的百分比，跳出率越低说明流量质量越好，用户对网站的内容越感兴趣。
其他指标	各种辅助性指标如点击量/用户，购买量/用户，下载量/用户等。

A/B 测试对用户画像建模的作用有三个：特征提取，一些标签对用户的兴趣有强相关作用，如性别标签，有些标签是弱相关作用，如用户职业标签，A/B 测试需要筛选出强相关标签，过滤掉弱相关标签；权重量化，根据 A/B 测试实验显示，发现用户画像中的最近点击标签、最近关注标签所占权重比想象中的要大；标签组合，有些标签是冗余的，只需从中选一即可。A/B 测试具体实现步骤如下：

- 方案设计。实验之前需得到一个基准版本，然后把又争议的标签按照优先度列举出来决定是否实验。真正的 A/B 测试只应一次改动一个地方，这意味着标签选择、权重量化、标签组合要分开来测试。
- 确定数据评估方案。根据实验内容不同评估它们好坏的标准也不同，如果是标签选择那么衡量的主要指标是点击量，如果是权重量化那么衡量的主要指标是点击转换率。
- 流量分配。为了试实验所得数据具备统计意义，能准确反映用户的真实行为，需要对流量设置一个下限。除此之外，为了使各个方案具有可比性，A、B 俩个方案的流量必须是相等的。
- 测试周期。根据所需测试的项目的不同测试周期也有所不同，如添加一个地理标签需要的测试周期以天为单位，如果涉及到多个标签的权重变动则需要测试周期以周为单位。
- 评估结果。适者胜出，其代表的数据作为下一轮回 A/B 测试的基准版本。
- 建立通用的数据评估题型。在经过各种类型 A/B 测试实验后，已经积累很多的评估指标，有必要把这些指标抽象出来形成一个通用的数据评估模型，减少以后实验的重复设计评估指标的时间。

4.5 总结

这一章主要研究了标签动态变化的对推荐系统的影响，实际中用户同时会受到社会因素和个人因素的影响, 但这两种因素在会产生不同强度的影响。在快速变化的系统中, 用户行为更加会受到社会因素的影响, 而在变化相对较慢的系统中, 用户行为则更加受到个人因素的影响。本章首先介绍了用户行为数据的存储方式以及基于此的用户行为数据的的预处理。然后介绍了用户兴趣探索的组成内容，包括用户异常兴趣探测、长尾标签抽取、用户满意度量化、标签权重的线性衰减。最后给出了用户兴趣探索评估方法，包括离线和在线俩种。下一章节主要介绍如何把用户画像和兴趣探索融入到推荐系统中，从而搭建出一个具有长尾性、实时性的推荐系统。

第五章　动态推荐系统设计

5.1　前言

推荐系统的形式化定义如下：设 C 是所有用户的集合，S 是所有可以推荐给用户的主题的集合。实际上，C 和 S 集合的规模通常很大，如上百万的顾客以及上亿种歌曲等。设效用函数 u（）可以计算主题 s 对用户 c 的推荐度（如提供商的可靠性（vendorreliability）和产品的可得性（productavailability）等），即 $u = C \times S \rightarrow R$，R 是一定范围内的全序的非负实数，推荐要研究的问题就是找到推荐度 R 最大的那些主题 S*，如式 5.1

$$\forall c \in C, S^* = argmax_{s \in S} u(c, s) \tag{5.1}$$

除了推荐系统自身如冷启动、数据的稀疏性等问题，还有一个关注点就是推荐系统的时间效应问题。比较常见的时间效应问题主要反映在用户兴趣的变化、物品流行度的变化以及商品的季节效应，这些问题都可以利用用户画像解决。本章节主要介绍如何搭建一个具有长尾性、实时性的动态推荐形态。动态推荐形态由 3 个重要的模块组成：用户画像、兴趣探索模块、推荐主题建模模块、推荐算法模块和评测指标模块。通用的推荐系统模型流程如所示。推荐系统把用户模型中兴趣需求信息和推荐主题模型中的特征信息匹配，同时使用相应的推荐算法进行计算筛选，找到用户可能感兴趣的推荐主题，然后推荐给用户。

用户画像模块对应着用户长期兴趣，用户兴趣探索对应着用户短期动态兴趣。短期兴趣的特点是临时、易变；长期兴趣的特点是长久、稳定；用户的短期兴趣可能会转化为长期兴趣，所以需要在推荐时综合考虑长期兴趣和短期兴趣。考虑到推荐系统的时间效应问题，将输入数据集归结为一个四元组，即用户，物品，行为，时间，通过研究用户的历史行为来预测用户将来的行为。需要解决以下俩个问题：动态评分预测、时效性的影响。首先，动态评分预测问题。数据集可以选用比较直观的显性反馈数据集，即 (用户，物品，评分，时间)，研究是这样一个问题，给定用户 u，物品 i，时间 t，预测用户 u 在时间 t 对物品 i 的评分 r。对于该类问题，与时间无关的评分预测问题算法主要有以下几种：用户兴趣的变化，如年龄增长，从儿童长成青少年壮年；生活状态的变化，由以前的小学生到大学生；社会事件的影响如俩会等。此外还有季节效应问题，一些在春季很流行的，在夏季节未必就很流行。该问题的解决有待进一步思考。对于时效性的影响，每个在线系统都是一个动态系统，但它们有不同的演化速率。比如说，新闻，手机主题更新很快，但音乐，电影的系统演化的却比较慢。

本章首先介绍用户画像和兴趣探索模块，其中兴趣探索模块需要根据业务的演化速率来调整迭代深度。然后介绍推荐主题模块，之后介绍推荐算法模块和指标体系，最后做总结。

5.2　用户画像和兴趣探索模块

目前基于用户画像的推荐，主要用在基于内容的推荐，从最近的 RecSys 大会（ACM Recommender Systems）上来看，不少公司和研究者也在尝试基于用户画像做 Context-Aware 的推荐（情境感知，又称上下文感知）。利用用户的画像，结合时间、天气等上下文信息，给用户做一些更加精准化的推荐是一个不错的方向。一个好的推荐系统要给用户提供个性化的、高效的、动态准确的推荐，那么推荐系统应能够获取反映用户多方面的、动态变化的兴趣偏好，推荐系统有必要为用户建立一个用户兴趣探索模型，该模型能获取、表示、存储和修改用户兴趣偏好，能进行推理，对用户进行分类和识别，帮助系统更好地理解用户特征和类别。推荐系统根据用户画像进行推荐，所以用户画像对推荐系统的质量有至关重要的影响。建立用户画像模型之前，需要考虑：模型的输入数据有哪些，如何获取模型的输入数据；如何考虑用户的兴趣及需求的变化；建模的对象是谁以及如何建模；模型的输出是什么。用户画像模型的输入数据主要有以下几种：

- 用户属性，分为社会属性和自然属性，包括用户最基本的如用户的姓名、年龄、职业、收入、学历等信息。用户注册时的对自然属性和社会属性进行初始建模。

- 用户手工输入的信息：是用户主动输出给系统的信息，包括用户在搜索引擎中打出的关键词，用户评论中发布的感兴趣的主题、频道。还有一类重要的信息就是用户反馈的信息，包括用户自己对推荐结果的满意程度；用户标注的浏览页面的感兴趣、不感兴趣或感兴趣的程度等。

- 用户的浏览行为和浏览内容：用户浏览的行为和内容体现了用户的兴趣和需求，它们包括浏览次数、频率、停留时间等，浏览页面时的操作（收藏、保存、复制等）、浏览时用户表情的变化等。服务器端保存的日志也能较好地记录用户的浏览行为和内容。

- 推荐对象的属性特征：不同的推荐对象，用户建模的输入数据也不同。网页等推荐对象通常考虑对象的内容和用户之间的相似性，而产品等推荐对象通常考虑用户对产品的评价。为提高推荐质量，推荐对象的相关的属性也要考虑进去，比如除网页内容以外，还要考虑网页的发布人、时间等。产品类的对象还要考虑产品的品牌、价格、出售时间等。

用户行为的权重排序。用户显式行为数据记录了用户在平台上不同的环节的各种行为，这些行为一方面用于候选集触发算法中的离线计算（主要是点击、浏览），另外一方面，这些行为代表的用户兴趣强弱不同，因此在训练重排序模型时可以针对不同的行为设定不同的权重值，以更细地刻画用户的行为强弱程度。此外，用户的购买、试用等行为还可以作为重排序模型的交叉特征，用于模

型的离线训练和在线预测。负反馈数据反映了当前的结果可能在某些方面不能满足用户的需求，因此在后续的候选集触发过程中需要考虑对特定的因素进行过滤或者降权，提高用户体验；同时在重排序的模型训练中，A/B 测试结果可以作为不可多得的负例参与模型训练。用户画像是刻画用户属性的元数据，其中有些是直接获取的基础数据，有些是经过挖掘的二次数据，这些属性一方面可以用于候选集触发过程中对标签进行加权或降权，另外一方面可以作为重排序模型中的用户维度特征。通过对数据的挖掘可以提取出一些关键词，然后使用这些关键词给主题打标签，用于主题的个性化展示。

用户行为的获取方式。模型输入数据的方式有显式获取、隐式获取和启发式获取三种方式。显式获取用户兴趣偏好的方法是简单而直接的做法，能准确地反映用户的需求，同时所得的信息比较具体、全面、客观，结果比较可靠。缺点就是数量稀少，原因用户不太愿意花时间来向商家表达自己的喜好，并且这种方法灵活性差，答案存在异质性，当用户兴趣主题改变时需要用户手动更改系统中用户兴趣。同时该方法对用户不是很人性化。解决人性化问题是推荐系统未来的一个研究方向，来研究用户能够接受的评价方式是什么，比如能够有耐心进行几次评分。利用固定负担模型来计量用户评价的负担，将人性化设计问题转化为最优化问题来研究。隐式获取法是指系统通过记录用户行为数据，通过权重排序获取用户的兴趣偏好，用户的很多动作都能暗示用户的喜好，包括查询、浏览页面和文章、标记书签、反馈信息、滑屏等。隐式的跟踪可以在建立用户画像基本数据的同时不打扰用户的正常消费活动。这种方法的缺点就是跟踪的结果未必能正确反映用户的兴趣偏好。同时系统若过度跟踪用户的历史记录，有时会引发用户隐私问题，而放弃对当前推荐系统的使用。上述获取兴趣偏好的方法有时受用户教育背景、职业和习惯等因素的限制，用户有时意识不到自己的兴趣主题，因此能为用户提供启发式信息，如领域术语抽取和相似度物品聚类，可以实现领域知识的复用，为用户间的协同提供支持，提高用户兴趣获取质量。用户的兴趣和需求会随着时间和情景发生变化，用户画像模块要考虑到用户长期兴趣偏好和短期兴趣偏好，还要考虑兴趣的变化，目前很多研究关注了用户的长期兴趣，建立了静态用户画像模型，但用户兴趣探索模型也越来越受到关注。结合长期和短期兴趣的动态建模将是未来的一个研究方向，如图 5.1所示。

用户画像更新采用了时间窗方法和遗忘机制来反映用户兴趣的变化。目前的更新机制无法及时跟踪用户兴趣的变化，just-in-time 型有更强学习效率和动态变化适应能力的建模也是未来的重要研究方向。

5.3　推荐主题模块

推荐主题分为单用户建模和群组建模，单用户建模针对个体用户进行建模，比如基于主题内容的推荐，群组建模是针对一类用户进行建模，比如基于商品的协同推荐。

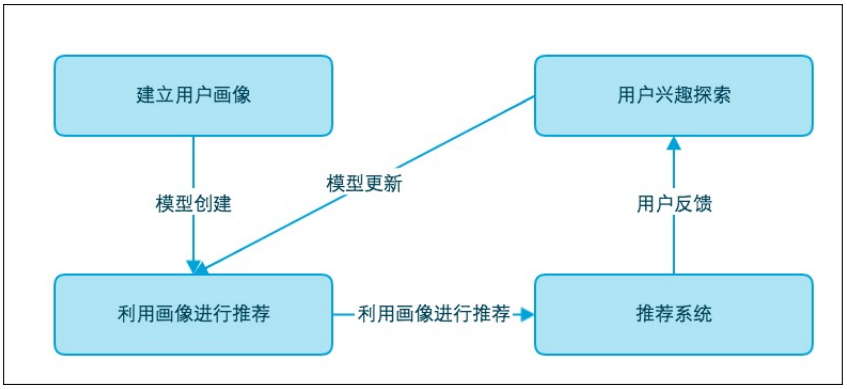

图 **5.1**　用户画像的使用

应用于不同的领域的推荐系统其推荐的主题也各不相同，如何对推荐主题进行描述对推荐系统也有很重要的影响。和用户画像一样，要对推荐主题进行描述之前要考虑：提取推荐主题的什么特征，如何提取，提取的特征用于什么目的，主题的特征描述和用户画像之间有关联。提取到的每个主题特征标签对推荐结果会有什么影响。主题的特征描述文件能否自动更新。推荐主题的描述文件中的主题特征和用户画像中的兴趣标签进行推荐计算，获得推荐主题的推荐权重，所以推荐主题的描述文件与用户画像密切相关，通常的做法是用同样的方法来表达用户的兴趣偏好和推荐主题。推荐系统推荐主题包括众多的领域，比如体育、动漫、科技、国家，还有诸如音乐、电影等多媒体资源等等。不同的主题，特征也不相同，目前并没有一个统一的标准来进行统一描述，主要有基于内容的方法和基于分类的方法两大类方法。基于内容的方法是从主题本身抽取相关信息来表示主题，使用最广泛的方法是用加权关键词矢量，该方法通过对标注主题的标签进行统计分析得出的特征向量。方法很多，比较简单的做法就是计算每个特征的熵，选取具有最大熵值的若干个特征；也可以计算每个标签的信息增量（Information gain），即计算每个特征在主题中出现前后的信息熵之差；还可以计算每个特征的互信息（mutual information），即计算每个特征和主题的相关性。在完成主题特征提取后，还需要计算每个特征的权值，权值大的对推荐结果的影响就大。基于分类的方法是把推荐主题放入不同类别中，这样可以把同类主题推荐给对该类主题感兴趣的用户了。文本分类的方法有多种，比如朴素贝叶斯（Naive-Bayes），k 最近邻方法（KNN）和支持向量机（SVM）等。主题的类型可以预先定义，也可以利用聚类算法自动产生。研究表明聚类的精度非常依赖于主题的数量，而且由自动聚类产生的类型可能对用户来说是毫无意义的，因此可以有选择的进行手工选定的类型来分类主题，在没有对应的候选类型或需要进一步划分某类型时，才使用聚类产生的类型。推荐系统推荐给用户的主题首先不能与用户购买过的主题重复，其次也不能与用户刚刚看过的主题不是太形似或者太不相关，这就是所谓的模型过拟合问题（可扩展性问题）。出现这一问题的本质上来自数据的不完备性，解决的主要的方法是引入随机性，使算法收敛到全局

最优或者逼近全局最优。针对这一问题考察了被推荐的主题的相关性和冗余性，要同时保证推荐的多样性，又不能与用户看过的主题重复或毫不相关。关于这一问题的研究是推荐系统研究的一个难点和重点。推荐系统中出现新的主题时，推荐系统尤其是协同过滤系统中，新主题出现后必须等待一段时间才会有用户浏览和评价，而在此之前推荐系统是无法对此主题进行推荐，这就是推荐系统研究的另一个难点和重点——商品冷启动问题。解决这一问题的方法就是考虑利用组合推荐方法。

5.4　推荐算法模块

推荐算法类型很多，但是各有各的局限，比较常用的有基于内容推荐，协同过滤推荐，基于关联规则推荐，基于效用推荐，基于知识推荐，组合推荐。他们的主要优缺点对比如所示。推荐算法本身是一个综合性的问题，可以简单地用最

表 **5.1**　推荐系统主要算法比较

推荐方法	优点	缺点
基于内容推荐	推荐结果直观，容易解释；不需要领域知识	稀疏问题；新用户问题；复杂属性不好处理；要有足够数据构造分类器
协同过滤推荐	新异兴趣发现、不需要领域知识；随着时间推移性能提高；推荐个性化、自动化程度高；能处理复杂的非结构化对象	稀疏问题；可扩展性问题；新用户问题；质量取决于历史数据集；系统开始时推荐质量差；
基于规则推荐	能发现新兴趣点；不要领域知识	规则抽取难、耗时；产品名同义性问题；个性化程度低；
基于效用推荐	无冷开始和稀疏问题；对用户偏好变化敏感；能考虑非产品特性	用户必须输入效用函数；推荐是静态的，灵活性差；属性重叠问题；
基于知识推荐	能把用户需求映射到产品上；能考虑非产品属性	知识难获得；推荐是静态的

基本的 Content-based，再复杂点可以 Collaborative Filtering，更深入一些诸如基于 SVD/LDA 等的降维算法和基于 SVD++ 等的评分预测算法，或者把推荐问题再转换成分类问题，或者采用以上算法前先用各种聚类算法做数据的预处理。

5.4.1　推荐算法

基于内容推荐。基于内容的推荐（Content-based Recommendation）是信息过滤技术的延续与发展，它是建立在项目的内容信息上作出推荐的，而不需要依据用户对项目的评价意见，更多地需要用机器学习的方法从关于内容的特征描述的事例中得到用户的兴趣资料。在基于内容的推荐系统中，项目或对象是通过相关的特征的属性来定义，系统基于用户评价对象的特征，学习用户的兴趣，

考察用户资料与待预测项目的相匹配程度。用户的资料模型取决于所用学习方法，常用的有决策树、神经网络和基于向量的表示方法等。基于内容的用户资料是需要有用户的历史数据，用户资料模型可能随着用户的偏好改变而发生变化。基于内容推荐方法的优点是：不需要其它用户的数据，没有冷开始问题和稀疏问题。能为具有特殊兴趣爱好的用户进行推荐。能推荐新的或不是很流行的项目，没有新项目问题。通过列出推荐项目的内容特征，可以解释为什么推荐那些项目。已有比较好的技术，如关于分类学习方面的技术已相当成熟。缺点是要求内容能容易抽取成有意义的特征，要求特征内容有良好的结构性，并且用户的口味必须能够用内容特征形式来表达，不能显式地得到其它用户的判断情况。

协同过滤推荐。协同过滤推荐（Collaborative Filtering Recommendation）技术是推荐系统中应用最早和最为成功的技术之一。它一般采用最近邻技术，利用用户的历史喜好信息计算用户之间的距离，然后利用目标用户的最近邻居用户对商品评价的加权评价值来预测目标用户对特定商品的喜好程度，系统从而根据这一喜好程度来对目标用户进行推荐。协同过滤最大优点是对推荐对象没有特殊的要求，能处理非结构化的复杂对象，如音乐、电影。协同过滤是基于这样的假设：为一用户找到他真正感兴趣的内容的好方法是首先找到与此用户有相似兴趣的其他用户，然后将他们感兴趣的内容推荐给此用户。其基本思想非常易于理解，在日常生活中，我们往往会利用好朋友的推荐来进行一些选择。协同过滤正是把这一思想运用到电子商务推荐系统中来，基于其他用户对某一内容的评价来向目标用户进行推荐。基于协同过滤的推荐系统可以说是从用户的角度来进行相应推荐的，而且是自动的，即用户获得的推荐是系统从购买模式或浏览行为等隐式获得的，不需要用户努力地找到适合自己兴趣的推荐信息，如填写一些调查表格等。和基于内容的过滤方法相比，协同过滤具有如下的优点：能够过滤难以进行机器自动内容分析的信息，如艺术品，音乐等。共享其他人的经验，避免了内容分析的不完全和不精确，并且能够基于一些复杂的，难以表述的概念（如信息质量、个人品味）进行过滤。有推荐新信息的能力。可以发现内容上完全不相似的信息，用户对推荐信息的内容事先是预料不到的。这也是协同过滤和基于内容的过滤一个较大的差别，基于内容的过滤推荐很多都是用户本来就熟悉的内容，而协同过滤可以发现用户潜在的但自己尚未发现的兴趣偏好。能够有效的使用其他相似用户的反馈信息，较少用户的反馈量，加快个性化学习的速度。虽然协同过滤作为一种典型的推荐技术有其相当的应用，但协同过滤仍有许多的问题需要解决。最典型的问题有稀疏问题（Sparsity）和可扩展问题（Scalability）。协同过滤（CF）可以看做是一个分类问题，也可以看做是矩阵分解问题。协同滤波主要是基于每个人自己的喜好都类似这一特征，它不依赖于个人的基本信息。比如刚刚那个电影评分的例子中，预测那些没有被评分的电影的分数只依赖于已经打分的那些分数，并不需要去学习那些电影的特征。

基于关联规则推荐。基于关联规则的推荐（Association Rule-based Recom-

mendation）是以关联规则为基础，把已购商品作为规则头，规则体为推荐对象。关联规则挖掘可以发现不同商品在销售过程中的相关性，在零售业中已经得到了成功的应用。管理规则就是在一个交易数据库中统计购买了商品集 X 的交易中有多大比例的交易同时购买了商品集 Y，其直观的意义就是用户在购买某些商品的时候有多大倾向去购买另外一些商品。比如购买牛奶的同时很多人会同时购买面包。算法的第一步关联规则的发现最为关键且最耗时，是算法的瓶颈，但可以离线进行。其次，商品名称的同义性问题也是关联规则的一个难点。

基于效用推荐。基于效用的推荐（Utility-based Recommendation）是建立在对用户使用项目的效用情况上计算的，其核心问题是怎么样为每一个用户去创建一个效用函数，因此，用户资料模型很大程度上是由系统所采用的效用函数决定的。基于效用推荐的好处是它能把非产品的属性，如提供商的可靠性（Vendor Reliability）和产品的可得性（Product Availability）等考虑到效用计算中。

基于知识推荐。基于知识的推荐（Knowledge-based Recommendation）在某种程度是可以看成是一种推理（Inference）技术，它不是建立在用户需要和偏好基础上推荐的。基于知识的方法因它们所用的功能知识不同而有明显区别。效用知识（Functional Knowledge）是一种关于一个项目如何满足某一特定用户的知识，因此能解释需要和推荐的关系，所以用户资料可以是任何能支持推理的知识结构，它可以是用户已经规范化的查询，也可以是一个更详细的用户需要的表示。

组合推荐。由于各种推荐方法都有优缺点，所以在实际中，组合推荐（Hybrid Recommendation）经常被采用。研究和应用最多的是内容推荐和协同过滤推荐的组合。最简单的做法就是分别用基于内容的方法和协同过滤推荐方法去产生一个推荐预测结果，然后用某方法组合其结果。尽管从理论上有很多种推荐组合方法，但在某一具体问题中并不见得都有效，组合推荐一个最重要原则就是通过组合后要能避免或弥补各自推荐技术的弱点。在组合方式上，有研究人员提出了七种组合思路：加权（Weight）：加权多种推荐技术结果。变换（Switch）：根据问题背景和实际情况或要求决定变换采用不同的推荐技术。混合（Mixed）：同时采用多种推荐技术给出多种推荐结果为用户提供参考。特征组合（Feature combination）：组合来自不同推荐数据源的特征被另一种推荐算法所采用。层叠（Cascade）：先用一种推荐技术产生一种粗糙的推荐结果，第二种推荐技术在此推荐结果的基础上进一步作出更精确的推荐。特征扩充（Feature augmentation）：一种技术产生附加的特征信息嵌入到另一种推荐技术的特征输入中。元级别（Meta-level）：用一种推荐方法产生的模型作为另一种推荐方法的输入。

5.4.2　AB 测试

产品的改变并不总是意味着进步，有时候无法评判多种设计方案中哪一种更优秀的，这时 A/B 测试就派上用场了，A/B 测试可以回答两个问题：哪个方案

好结果的可信程度 A/B 测试结果是基于用户得到的结果，用数据说话，而不是凭空想象去为用户代言，并且通过一定的数学分析给出结果的可信度。A/B 测试需要如下几个前提：多个方案并行测试；每个方案只有一个变量不同；能够以某种规则优胜劣汰其中第 2 点暗示了 A/B 测试的应用范围：A/B 测试必须是单变量，但有的时候，我们并不追求知道某个细节对方案的影响，而只想知道方案的整体效果如何，那么可以适当增加变量，当然测试方案有非常大的差异时一般不太适合做 A/B 测试，因为它们的变量太多了，变量之间会有很多的干扰，所以很难通过 A/B 测试的方法找出各个变量对结果的影响程度。在满足上述前提时，便可以做 A/B 测试了。

目标转换率变化区间估计：在做 A/B 测试的时候，抽样得到的数据并不能准确反映整体的真实水平，即样本得到的估计是有偏差的，因此需要去评估这个值可能的变化区间。例如通过区间估计得到：A 方案转换率为：6.5% ± 1.5%B 方案转换率为：7.5% ± 1.5% 方案胜出概率估计：由于最终有意义的是确立胜出的版本，然而并不是所有的实验都能做到样本足够大，区分度足够高的，因此确定版本胜出的概率，很多英文资料里面记为 Chance to beat baseline，即在给定转换率下，变体版本的实际转换率高于参展版本（默认是原始版本）的实际转换率的可能性。在实验之前需要设定一个阈值（称为置信度），某版本胜出的可能性高于这个值并且稳定时，便可以宣布该版本胜出。置信度越高，结果的可靠信越高；随着置信度的增加实验时间将会变长。

5.5　动态推荐系统底层架构

5.5.1　基于 **Spark**

基于 Spark 的方式在架构上，第一种是使用 Spark 把模型计算放在内存中，加快模型计算速度，Hadoop 中作业的中间输出结果是放到硬盘的 HDFS 中，而 Spark 是直接保存在内存中，因此 Spark 能更好地适用于数据挖掘与机器学习等需要迭代的模型计算，如表 9-2 所示。

5.5.2　基于 **Kiji** 框架

Kiji 是一个用来构建大数据应用和实时推荐系统的开源框架，本质上是对 HBase 上层的一个封装，用 Avro 来承载对象化的数据，使得用户能更容易地用 HBase 管理结构化的数据，使得用户姓名、地址等基础信息和点击、购买等动态信息都能存储到一行，在传统数据库中，往往需要建立多张表，在计算的时候要关联多张表，影响实时性。Kiji 提供了一个 KijiScoring 模块，它可以定义数据的过期策略，如综合产品点击次数和上次的点击时间，设置数据的过期策略把数据刷新到 KijiScoring 服务器中，并且根据自己定义的规则，决定是否需要重新计算得分。如用户有上千万浏览记录，一次的行为不会影响多少总体得分，不需

表 **5.2**　MR 和 spark 对比

过程	MapReduce	Spark
collect	在内存中构造了一块数据结构用于 map 输出的缓冲	没有在内存中构造一块数据结构用于 map 输出的缓冲，而是直接把输出写到磁盘文件
sort	map 输出的数据有排序	map 输出的数据没有排序
merge	对磁盘上的多个 spill 文件最后进行合并成一个输出文件	在 map 端没有 merge 过程，在输出时直接是对应一个 reduce 的数据写到一个文件中，这些文件同时存在并发写，最后不需要合并成一个
copy 框架	jetty	netty 或者直接 socket 流
对于本节点上的文件	仍然是通过网络框架拖取数据	不通过网络框架，对于在本节点上的 map 输出文件，采用本地读取的方式
copy 过来的数据存放位置	先放在内存，内存放不下时写到磁盘	一种方式全部放在内存；另一种方式先放在内存
merge sort	最后会对磁盘文件和内存中的数据进行合并排序	对于采用另一种方式时也会有合并排序的过程

要重新计算，但如果用户仅有几次浏览记录，一次的行为，可能就要重新训练模型。Kiji 也提供了一个 Kiji 模型库，使得改进的模型部署到生产环境时不用停掉应用程序，让开发者可以轻松更新其底层的模型。

5.5.3　基于 Storm

最后一种基于 Storm 的实时推荐系统。在动态推荐上，算法本身不能设计的太复杂，手机主题推荐系统的数据库是 TB 级别，实时读写大表比较耗时。可以把算法分成离线部分和实时部分，利用 Hadoop 离线任务尽量把查询数据库比较多的、可以预先计算的模型先训练好，或者把计算的中间数据先计算好，比如，线性分类器的参数、聚类算法的群集位置或者协同过滤中条目的相似性矩阵，然后把少量更新的计算留给 Storm 实时计算，一般是具体的评分阶段。用 HBase 或 HDFS 存储历史的浏览、购买行为信息，用 Hadoop 基于 User CF 的协同过滤，先把用户的相似度离线生成好，用户到商品的矩阵往往比较大，运算比较耗时，把耗时的运行先离线计算好，实时调用离线的结果进行轻量级的计算有助于提高主题推荐的实时性。协同过滤算法在 storm 上计算过程为：首先程序获取用户和主题的历史数据，得到用户到主题的偏好矩阵，利用 Jaccard 相似系数（Jaccard coefficient）、向量空间余弦相似度（Cosine similarity）、皮尔逊相关系数（Pearson correlation coefficient）等相似度计算方法，得到相邻的用户（User CF）或相似商品（Item CF）。在 User CF 中，基于用户历史偏好的相似度得到邻居用户，将邻

居用户偏好的主题推荐给该用户；在 Item CF 中，基于用户对物品的偏好向量得到相似主题，然后把这款主题推荐给喜欢相似主题的其他用户。然后通过 Kafka 或者 Redis 队列，保存前端的最新浏览等事件流，在 Storm 的 Topology 中实时读取里面的信息，同时获取缓存中用户 topN 个邻居用户，把邻居用户喜欢的商品存到缓存中，前端从缓存中取出商品，根据一定的策略，组装成推荐列表。除了相似性矩阵，其他模型大体实现也相似，比如实际的全品类电商中不同的品类和栏位，往往要求不同的推荐算法，如母婴主题，如果结合商品之间的序列模式和母婴年龄段的序列模式，效果会比较好，可以把模型通过 Hadoop 预先生成好，然后通过 Storm 实时计算来预测用户会买哪些主题。

5.6　量化评估推荐系统

推荐系统还是看目的是如何的，从用户角度讲是为了更好的理解用户，减少用户查找内容的时间和次数，从产品本身角度讲，是增加单位面积单位时间内的点击数或者说内容有效。从业务角度的衡量：衡量点击和打开率，这说明用户是否对内容感兴趣。衡量通过推荐系统替代用户主动搜索或者主动浏览的次数，可以通过横向与使用其他产品对比较，比如使用推荐系统提供内容的用户搜索次数和点击浏览目录次数明显下降。衡量推荐系统的满意度口碑，刨除因为页面位置效果等因素，衡量推荐系统一个重要的就是满意度的口碑问题，这个可以通过单个用户是否有重复使用的行为，曲线是否是一直上升的来衡量，如果一直有新用户访问，但一直没有老用户重复使用，说明用户满意度有问题。

5.7　总结

推荐系统经过了相当时间的发展，同时一些重点和难点问题得到了研究者的关注，相信是未来研究的热点问题。用户兴趣偏好获取方法和推荐对象的特征提取方法的研究目前的推荐系统中实际上较少使用了用户和推荐对象的特征，即使使用很广泛的协同推荐使用的是用户的评分。主要是用户兴趣偏好的获取方法和推荐对象特征提取方法不是很适用，需要引入更精确适用的用户和对象特征。（2）推荐系统的安全性研究进行协同推荐时需要掌握用户的兴趣偏好等用户信息，但用户担心个人数据得不到有效保护而不愿暴露个人信息，这是协同推荐长期存在的一个问题。既能得到用户信息而提高推荐系统性能，又能有效保护用户信息将是未来推荐系统的一个研究方向。同时一些不法的用户为了提高或降低某些对象的推荐概率，恶意捏造用户评分数据而达到目的，这也是推荐系统存在的一个安全问题，被称为推荐攻击 [93-96]。检测并能预防推荐攻击也将是未来一个研究方向。（3）基于复杂网络理论及图方法的推荐系统研究复杂网络理论和图方法同协同推荐存在契合点，在文献中网络视频推荐问题转化为热量散播平衡态网络上的谱图分割问题，通过设计长尾发现的推荐策略引导用户

发现潜在的感兴趣的网络视频。利用复杂网络理论和图方法进行推荐也是推荐系统研究的一个方向。（4）推荐的多维度研究目前的推荐研究都是基于用户-对象二维空间进行研究的，但是用户选择某个对象以及对对象的评分在不同的情况下会有所不同，也就是推荐使用的特征维度会有所不同，研究推荐的多维度也是未来的一个研究方向。（5）稀疏性和冷启动研究稀疏性和冷启动问题是困扰推荐系统很长时间了，包括经典协同过滤算法和新出现的基于网络结构的推荐算法都存在该问题。有很多研究者对这一问题进行研究并提出解决办法，但该问题依然存在，还需要对其进行研究。（6）推荐系统性能评价指标的研究用户对算法准确度的敏感度、算法对不同领域的普适性、广义的质量评价方法等都是未来推荐系统性能评价要进行研究的目标。

参考文献

[1] O. Celma. *Music Recommendation and Discovery in the Long Tail*. Springer. 2010.

[2] Marko Balabanovi´c and Yoav Shoham. *Fab: content-based, collaborative recommendation*. Commun. ACM, 40:66–72, March 1997.

[3] Andrew I. Schein, Alexandrin Popescul, Lyle H. Ungar, David M. Pennock. *Methods and Metrics for Cold-Start Recommendations*. New York City, New York: ACM. pp. 253–260. 2002.

[4] CTEX Sia, K.C., Zhu, S., Chi, Y., Hino, K., Tseng, B.L. *Capturing User Interests by Both Exploitation and Exploration*. Technical report, NEC Labs America. 2006.

[5] Jansen, B. J. and Rieh, S. *The Seventeen Theoretical Constructs of Information Searching and Information Retrieval*. Journal of the American Society for Information Sciences and Technology. 61(8), 2010.

[6] Han, Jiawei; Kamber, Micheline. *Data mining: concepts and techniques*. Morgan Kaufmann. p. 5. 2001.recmd-system

[7] Francesco Ricci and Lior Rokach and Bracha Shapira. *Introduction to Recommender Systems Handbook*. Springer, pp. 1-35. 2011.

[8] Robert K. Merton. *The Matthew Effect in Science*. Science, 159(3810):56– 63, January 1968.

[9] Junghoo Cho and Sourashis Roy. *Impact of search engines on page popularity*. In Proceedings of the 13th international conference on World Wide Web, WWW '04, pages 20–29, New York, NY, USA, ACM. 2004.

[10] Daniel M. Fleder and Kartik Hosanagar. *Recommender systems and their impact on sales diversity*. In Proceedings of the 8th ACM conference on Electronic commerce, EC '07, pages 192–199, New York, NY, USA, ACM. 2007.

[11] Henry Kautz, Bart Selman, and Mehul Shah. *Referral web: combining social networks and collaborative filtering*. Commun. ACM, 40:63–65, March 1997.

[12] Jonathan L. Herlocker, Joseph A. Konstan, Loren G. Terveen, and John T. Riedl. *Evaluating collaborative filtering recommender systems*. ACM Trans. Inf. Syst., 22:5–53, January 2004.

[13] Kohavi, Ron, Longbotham, Roger. *Online Controlled Experiments and A/B Tests*. In Sammut, Claude; Webb, Geoff. 2015.

[14] Elaine Rich. *Readings in intelligent user interfaces*. chapter User modeling via stereotypes, pages 329–342. 1998.

[15] Anne-F. Rutkowski and Carol S. Saunders. *Growing pains with information overload*. Computer, 43:96–95, June 2010.

[16] J. Scott Armstrong, editor. *Principles of Forecasting - A Handbook for Researchers and Practitioners*. Kluwer Academic, 2001.

[17] Henry Kautz, Bart Selman, and Mehul Shah. *Referral web: combining social networks and collaborative filtering*. Commun. ACM, 40:63–65, March 1997.

[18] Greg Linden, Brent Smith, and Jeremy York. *Amazon.com recommendation- s: Item-to-item collaborative filtering*. IEEE Internet Computing, 7:76–80, January 2003.

[19] Anne-F. Rutkowski and Carol S. Saunders. *Growing pains with information overload*. Computer, 43:96–95, June 2010.

[20] Yehuda Koren. *Collaborative filtering with temporal dynamics*. In Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining, KDD '09, pages 447–456, New York, NY, USA, 2009. ACM.

[21] Thomas Hofmann and Jan Puzicha. *Latent class models for collaborative filtering*. In Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence, IJCAI '99, pages 688–693, San Francisco, CA, USA, 1999. Morgan Kaufmann Publishers Inc.

[22] Bruce Krulwich. *Lifestyle finder: Intelligent user profiling using large-scale demographic data*. AI Magazine, 18(2):37–45, 1997.

[23] Mohsen Jamali and Martin Ester. *Trustwalker: a random walk model for combining trust-based and item-based recommendation*. In Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining, KDD '09, pages 397–406, New York, NY, USA, ACM. 2009.

致　谢

感谢原本科模板的作者 XPS、硕博模板的作者刘青松以及它们的维护者的辛勤工作！

感谢大家对本模板更新工作的支持！

本模板以及本示例文档还存在许多不足之处，欢迎大家测试并及时提供反馈。

ywg@USTC

在中国科技大学完成本科和硕博连读学业的九年里，我所从事的学习和研究工作，都是在导师以及系里其他老师和同学的指导和帮助下进行的。在完成论文之际，请容许我对他们表达诚挚的谢意。

首先感谢导师 XXX 教授和 XXX 副教授多年的指导和教诲，是他们把我带到了计算机视觉的研究领域。X 老师严谨的研究态度及忘我的工作精神，X 老师认真细致的治学态度及宽广的胸怀，都将使我受益终身。

感谢班主任 XXX 老师和 XX 老师多年的关怀。感谢 XXX、XX、XX 等老师，他们本科及研究生阶段的指导给我研究生阶段的研究工作打下了基础。

感谢 XX、XXX、XXX、XX、XXX、XXX、XXX、XX 等师兄师姐们的指点和照顾；感谢 XXX、XX、XXX 等几位同班同学，与你们的讨论使我受益良多；感谢 XXX、XX、XXX、XX、XXX 等师弟师妹，我们在 XXX 实验室共同学习共同生活，一起走过了这段愉快而难忘的岁月。

感谢科大，感谢一路走过来的兄弟姐妹们，在最宝贵年华里，是你们伴随着我的成长。

最后，感谢我家人一贯的鼓励和支持，你们是我追求学业的坚强后盾。

胡磊

2016 年 3 月 13 日