中国科学技术大学

硕士学位论文

基于手机主题推荐系统的用户画像模型

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: 二〇一六年九月

University of Science and Technology of China

A dissertation for master's degree

The User Profile Based on Phone Theme Recommendation System

Author :	Lei Hu
Speciality :	Information Security
Supervisor :	Prof. Wuyang Zhou
	Dr. Sihai Zhang
Finished Time :	September 1, 2016

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

始于二十世纪九十年代的信息爆炸，使得用户越来越难有效的从茫茫多的数据中获取所需信息，因此，推荐系统凭借其精准的定位和” 千人千面” 的个性化服务受到人们的青睐和研究者的重视。本论文讨论了如何构建一个基于手机主题推荐系统的用户画像模块和用户兴趣探索模块。

传统的个性化推荐系统面临着诸多挑战，其最根本的问题是如何根据企业的商业目标和业务特点来优化推荐系统，具体到手机主题行业，推荐系统面临着包括社交化、长尾性、冷启动、动态推荐等一系列综合问题。由此，笔者提出并实现了一种适用于手机主题的用户画像模型，实践证明其能在很大程度上提升推荐系统的推荐质量。本文的主要工作和贡献有:

- 实现了推荐系统的用户画像模块。利用信息检索（Information Retrieval）技术从用户注册信息获取到用户的人口属性、职业、地理位置、性别等信息并标签化，不同标签的来源，标签的本身，以及标签与用户之间的共现关系决定着这个标签的初始权重，然后根据用户行为构建相应的 AB 测试产出标签的实际权重，权重越高则认为该标签对用户影响越大。AB 测试显示推荐系统利用用户画像标签进行推荐能显著提升诸如点击转换率等重要指标。

- 实现了推荐系统的用户兴趣探索模块。用户兴趣探索通过特征提取技术和用户满意度量化算法，定期更新用户兴趣标签和标签对应的权重。首先，利用用户兴趣特征向量和商品特征向量计算出用户-商品的相关分数。然后，利用用户行为（购买、评分、点赞、划屏频率等）量化用户满意度。一次成功的用户兴趣标签探索，首先应该有很低的相关分数和很高的满意度，其次兴趣标签应该是一个小众兴趣标签。用户兴趣探索能够实时更新用户的兴趣标签，帮助推荐系统持续满足用户的不断变化的需求。

- 利用时间因子衰减模型融合用户的长期兴趣和短期兴趣：用户画像针对的是用户的静态信息，代表了用户的长期兴趣，用户兴趣探索针对的是用户的动态信息，代表了用户的短期兴趣，衰减模型法的本质是利用自然遗忘规律拟合用户真实的兴趣衰减过程。

关键词： 推荐系统 长尾效应 动态兴趣 用户画像建模 用户兴趣探索

ABSTRACT

Information explosion in the new age let it's hard for users to get valuable infomation from the vast amounts of data, so the recommended system begin to go to the middle of the stage because it's precise forecast and Personalized service. So we here to discuss how to modeling users profile model and users interested exploration model for a android phone theme application recommended system.

There are so many weekness of the traditional recommended system, the most import one is how to sell more products, specific for android phone application, the recommended system need to solve Socializing problem, cool start problem, dynamic recommend based on timeline and so on. So the author proposed and implemented users profile model and users interested exploration model which include:

- Realized the use profile model of recommended system, we use information retrieval technology to get use basic information like occupation, location, gender from user registration information, different tag has different weight depending on the way they got, the path of they transfer and the relation between use and tags, the more weight of tag the high of credibility the tag has. AB test show that recommended system can improve click conversion rate rapidly.

- Realized the users interested exploration model of recommended system, which using feature extraction technology and user satisfaction scoring algorithm, we maintain a dynamic interesting tags vector space for all user. first, we can get user-item-scores by product users interesting vector metric and items feature metric. Then get the users satisfaction based on users history actions like buying, rateing, clicking and so on. one successful exploration means it has low user-item-relation-scores and high user satisfaction, and the tag also is minority. Experiments show that with the users interested exploration model, the recommended system has more long-tail effect.

- Sucessfully put user long term interesting and short term interesting into one model using linear decay algorithm, users profile model contains static infomation of users, users interested exploration model contains dynamic infomation of users interesting, this papar come up with the strategy to balance the static infomation and the dynamic infomation.

Keywords: recommend system, long-tail, dynamic, user profile, user interest explore

目 录

表格索引

插图索引

第一章　绪论

1.1　研究背景与意义

互联网自二十世纪九十年代从诞生、发展，到现在已经演化为人类社会的必需品。随着互联网的发展，用户的信息检索模式也发生了翻天覆地的变化，早期用户可以毫不费力的直接记住寥寥无几的网站、网址，轻松实现上网需求；随着网站呈指数的发展，网址数量大大超出人脑记录的容量，于是 Yahoo! 公司首次提出并实现了分类目录系统的概念，其本质还是通过人工将网站分门别类，因此其创新还是属于量变，没有达到质变。随着网站进入到爆发式增长，人工进行分类目录法变得越来越不现实了，于是产生 Google 为代表的搜索引擎，搜索引擎实现了互联网的质变，通过程序自动化的实现了检索网站、爬取内容、存储数据，实现了亿万级的数据积累，只有基于如此庞大的信息，才能为哪怕是最普通的用户提供及时、准确、快速信息获取服务，人类社会一定程度上填平了所谓的“信息鸿沟”。但是，后互联网时代又是个性化时代 [1–9]，需要一种系统精确刻画每个用户的兴趣爱好并能不动声色的在主页上表现出来，我们把这种提供个性化服务的系统系统统称为推荐系统。推荐系统是一种比搜索引擎更人性化、个性化的系统服务，不需要用户主动提供关键词，因此它能满足用户的更多的潜在需求，尤其当用户自己都无法精准描述自身需求的时候 [10]。第一代推荐系统以亚马逊为代表，作为一个电子商品平台，一方面有数以万计的商品需要被用户了解、熟悉和购买，另一方面有数以亿计的用户无法找到称心如意的商品。推荐系统通过构建用户和商品之间的桥梁，每年为亚马逊贡献近三十个百分点的创收！由此可见推荐系统能帮助用户快速发现有用的商品信息，具体来讲，首先推荐系统通过分析用户的历史行为每个用户进行独一无二的画像建模 [11]，目的有俩个：1，熟悉每个用户和他们的潜在需求；2，把拥有相同品味的用户归为一类群体，这样一来所有人的需求总和就可能是其中一个人的潜在需求，方便企业卖出更多的商品。随着用户终端设备的普及，出现了诸如淘宝、美团、滴滴、今日头条等互联网平台，几乎包办了人们衣食住行的方方面面，人们因为可选择性太多而出现了“选择性困难”的症状，这其实就是信息过载时代的具体表现形式。也就是说，在这个数据爆炸时代，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，都面临着日益严峻的挑战，现代人每天面临着从各种不必要的数据中找到有用的商品，其实是在浪费生命。每一个有追求、有理想的现代人，真的需要好好的设计人生，以一种精要的方式摒弃不必要之事，而这也是林语堂先生所说的生之智慧。笔者曾有过这样的一种购物体经历：笔者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。其实，作为互联网电子商家的翘楚—淘宝，一直在思考如何让自己平台下的优秀商品不埋没在大数据洪流中，因此，淘宝技术团队一直把个性

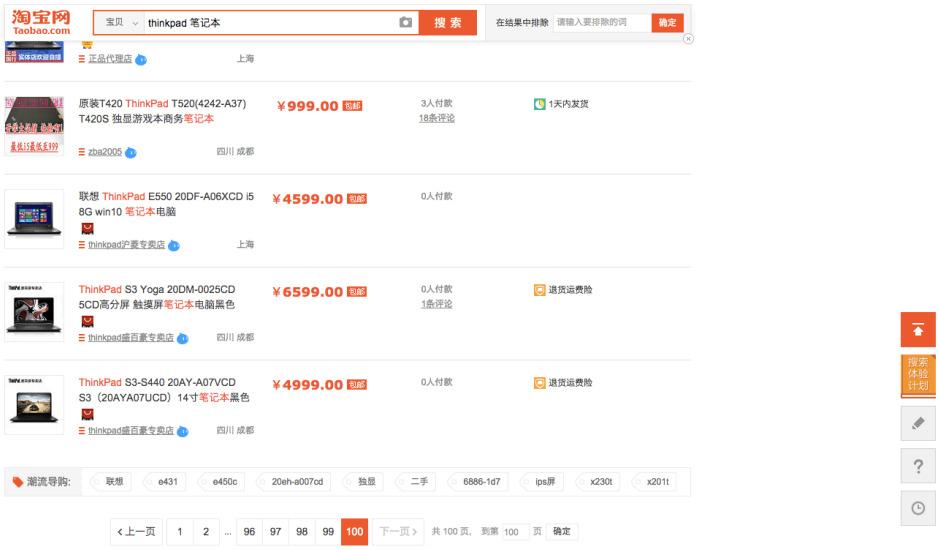

图 **1.1**　淘宝购物搜索图

化推荐系统视为解决用户-商品合理匹配的终极杀手锏。对于一个电商平台，原则上商品肯定是多多益善，但对于一个个性化推荐系统，则遵守更少，但更好的原则，推荐系统代表了一种自律的、精要的生活方式，据笔者所知，已经有很多互联网企业已经或者正在开发符合其企业文化的推荐系统，如笔者曾供职过的小米、今日头条和滴滴出行，其中小米的广告部门很早就利用推荐算法实现旗下各个业务线的智能广告投放，而今日头条利用推荐系统每日为用户定时推荐文章，滴滴出行则利用推荐系统为每个乘客和网约车做定向最优推荐。对于传统的推荐系统，首先，需要积累足够多的商品信息，因为只有尽可能的在基于所有的商品大局观上，才有可能得出比较正确的商品推荐候选集合；其次，需要尽可能积累用户的行为样本数据，因为这些样本将会是推荐统计假设检验的唯一数据标准，统计学之所以常让人意外，就是因为人们只能得到部分样本，而部分样本只是包含了事情的部分信息，于是就有扭曲事情本质的趋势，因此，精度是推荐系统最重要的指标之一，后文会详细介绍如果通过用户画像和用户兴趣 [12–16] 提升推荐系统的精度；最后就是甄别，哪些商品对哪些用户有着非同寻常的吸引力，其实对于一个用户来讲，平台上存在的绝大多数商品，包括数据、资源和他人观点，都没有什么价值，只有少数商品效果非凡，影响巨大，推荐系统的核心就是算法 [28]，通过算法甄别无意义的多数，只留下有意义的少数。总之，通过算法分析用户兴趣，分析商品特性，对用户跟所有商品的关联度打分、排序、取 topN，然后给出推荐结果。

但是，传统的推荐系统也有一些问题，典型的有数据稀疏问题、新用户问题、马太效应、实时推荐问题和用户兴趣变动问题。数据稀疏问题的本质就是商品信息数据过于膨胀，即使是骨灰级用户也没办法穷尽百分之一的商品，因此大多数的用户-商品相关值都是零，这不利于推荐系统做出正确的推荐结果；新用户问题又叫冷启动问题，指一个用户刚刚注册登录，推荐系统没有与此人相关的

信息，于是就没有方法做出推荐；马太效应是指越热门的商品越有被推荐的趋势，这种情况其实不是一件好事，因为：1，商品营收不平衡会增加平台的风险性，如果平台大多数营收的贡献来自于极少类明星商品，一旦这类商品发生问题，平台也会有问题；2，根据 2/8 原则，冷门商品虽然营收少，但它们的基数大，潜力无限；实时推荐问题是指用户从浏览到购买这段时间一般很短，而推荐系统需要打时间差，在用户点击之后、购买之前的这段时间做出推荐，但这是很难实现的，一个客观原因是对所有商品打分、排序、取 topN，最后找到用户感兴趣的商品，在工程上是需要一些时间的；用户兴趣变动问题是指用户兴趣是一个动态的过程，有可能随着季节周期性变动，有可能随着年龄发散性变化，推荐系统需要及时收集数据，保持对用户兴趣的最优拟合。

基于这些问题的存在，笔者基于推荐系统实现了用户画像模块和用户兴趣探索模块，用以解决传统推荐系统所面临的种种问题，并帮助推荐系统做出更好的推荐结果。用户画像模块其实就是回归了问题的本质：以人为本，用数据说话。通过分析、收集所有与用户有关的数据，为每个用户建立、维护一个独一无二的用户画像，用户画像的最大优点在于它能主动收集用户的基本人口数据、长期兴趣和短期兴趣，而且用户画像中的信息是动态更新的，也就是说随着时间的推移，用户的兴趣在逐渐改变，用户画像里的兴趣标签也会随之改变，最大程度上保证了用户兴趣的连续性和变化性；用户兴趣探索模块包括三个原则：1，用户和潜在感兴趣商品的关联度很低，这保证了探索的商品都是用户从前没有看见过的；2，用户满意度很高，即通过量化用户行为，包括点击次数、滑屏次数、滑屏频率、滑屏时长、点赞、分享等行为，得出用户的满意度；3，潜在商品的标签是小众的、冷门的标签，因为热门商品是没必要也不需要做探索的。

1.2　推荐系统的简介

推荐系统的研究其实是个交叉学科，因为其跟很多早期的基础领域的研究相关，比如认知科学 [17]，信息检索 [29] 和预测理论 [18]。随着数据时代的到来，研究人员开始研究如何利用用户对商品的行为数据来预测用户的兴趣，同时为用户提供推荐服务 [19]。近些年来，推荐系统越来越开始成为一个专门的研究课题，到 2005 年左右为止推荐系统的研究还是集中在基于 user、item 的协同过滤算法 [48]，在工业界目前应用最具有影响力的算法应该就是亚马逊的协同过滤算法 [20]。推荐系统推荐有俩个原则：1，给用户的商品不能与用户购买过的商品重复；2，不能与用户刚浏览过的商品太相关。推荐系统的函数形式化定义：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，如亿级别的顾客以及百万级别的商品。设函数 u() 可以计算主题 s 对用户 c 的推荐度 R，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 1.1。

$$\forall c \in C, S^* = argmax_{s \in S} u(c, s) \tag{1.1}$$

1.2.1 推荐系统的产生与发展

随着计算机存储技术以摩尔定律指数增长，信息的传播也开始爆发式的迅猛发展起来，我们这个时代的人类社会进入了一个崭新的大数据信息时代：互联网和物联网几乎无处不在，影响人类的衣食住行等方方面面，因此颠覆性的更改了人们的生活方式，在典型的互联网共享经济平台，一个用户既代表了消费者，也代表了生产者，就如笔者在滴滴出行的角色，平时上下班打车，属于运力消费者，周末开车做网约车司机，则变身为运力生产者。但是不好的一方面也开始浮现，曾几何时笔者发现自己社交账号开始多起来，数量之多以至于没办法记住每个账号的密码。这就是 Web 2.0 时代的一个副作用—让人们疲于奔命的把生活浪费在刷各种动态、信息，忙于各种无脑点赞、分享，而没有时间思考。社交化网络媒体如微信、微博的异军突起，导致互联网中的信息数据中充满了广告和噪声，而普通用户一般缺少过滤、屏蔽噪声的主观意愿和技术能力，不仅使其信息检索的时间成本巨大，也会让其在茫茫多的数据海洋里迷失自我，这就是信息过载问题的根源所在 [21, 22]。作为非常重要的技术手段，推荐系统和搜索引擎为用户解决信息过载提供了不可或缺的保障。俩者的不同之处在于：搜索引擎是分散的、被动的，用户需要先输入关键词，搜索引擎根据关键字在服务器后台进行信息检索，利用算法获得最优的匹配信息并展示给用户。但我们更多时候遇到的问题是，并不能精确描述自己的需求，而这就是推荐系统的强项，因为推荐系统是主动收集用户平日中一点一滴的数据，以至于用户不需要提供明确的需求，推荐系统只是通过分析用户的历史行为数据就可以“猜出”用户的意图，因此，如果我们把推荐系统和搜索引擎看作为两个互补的技术手段，那么效果一定很棒。

推荐系统最开始的概念，应该是在 1995 年由美国人工智能协会 [23] 上的 Robert Armstrong 教授首先提出，不仅如此，Armstrong 教授还不遗余力的推广了一个推荐系统的原型系统。受其启发，推荐系统的研究工作开始起步并发展壮大。第一个商用推荐系统应该属于 Yahoo 网站的个性化入口 MyYahoo。到了 21 新世纪，随着电子商务的风起云涌，推荐系统的研究与应用开始水涨船高，包括 eBay、taobao、Amazon、youtube[24] 等各大电子商务网站都有了自己的推荐系统，其中，Amazon 公司称其网站中百分之三十的营业额的流量入口来自于推荐系统。2006 年美国的 Netflix[25] 在网上公开了一个推荐算法竞赛，设立了丰厚的奖金，选手通过利用 Netflix 公开了的真实网站中的一部分数据，包含用户对电影的评分，利用数据挖掘算法预测哪些用户会购买哪些电影。2014 年阿里举办了阿里大数据竞赛，笔者和小伙伴有幸参加并顺利进入决赛，阿里公开了其部分用户三个月的浏览、收藏、购买商品数据，选手可以利用阿里天池计算资源做出预测，阿里大数据竞赛有效地推动了学术界和产业界对推荐算法的兴趣，很多有效的算法在此阶段被提了出来。

随着互联网企业深入人心的发展、壮大，推荐系统在电子商务中的优势地位也越来越明显。国内比较典型的电子商务平台网站有淘宝网、网易云音乐、爱奇

艺 PPS 等。在这些电子商务平台中，网站提供的商品数量不计其数，网站中的用户规模也是亿级别的。据双十一官方统计，天猫商城中的商品数量已经超过了 5000 万。试想下，在如此庞大商品数量的电商网站中，如果用户仅仅依靠搜索引擎输入关键字查询，只是过滤了百分之九十九的商品，对剩下百分之一的商品还是没辙。淘宝网在这一块做的很好，其手机 app 主页的推荐系统能够根据用户浏览行为 [26] 及时的为用户推荐商品，笔者发现淘宝的推荐结果已经成为大部分用户的主要购买入口，总的来说，目前比较成功的电子商务网站中，都在利用推荐系统这只会下金鸡蛋的母鸡，在用户购物的同时为用户推荐一些商品，从而提高商品的销售额。另一方面，随着以 IOS、Android 系统为代表的物联网引领了移动互联网的发展潮流。在用户在接入移动互联网过程中，其经纬度信息可以被非常准确地被获取，因此出现了大量的基于用户位置信息推荐系统。国外比较著名的有 Uber 和 Coupons。国内著名的有滴滴出行和美团网。美团网这种基于互联网的外卖平台，会利用位置服务为用户可推荐当前位置的餐馆、酒店、影院、旅游景点。线上交易，线下消费，之后为自己在现实世界中的体验打分，分享自己的经验与感受，形成线上下单-线下消费-线上评价的生态闭环。只是当笔者使用美团基于位置的美食服务时，同样也会遭遇信息过载问题，矛盾在于商家太多了，而笔者只能一次去一家餐厅就餐，这时美团平台的推荐系统会根据笔者的历史消费信息，加上笔者的偏好、口味、消费能力等，为笔者推荐当前位置下最可能感兴趣的餐厅。

随着网络社交的深入人心，用户不再满足于单纯的获取信息，而是与网络上的其他用户进行关注、聊天和互动。国外著名的社交网络就有 Twitter、Facebook 等，国内的社交网络有微信、微博等。在社交网站中用户不再是一个静止端点，而是与他人有错综复杂关系的社交网络。对于微信来说，最重要的资源应该就是用户之间的关联。这其中的关系可能是多层次的、多维度的、按时间序列走的，关联的因素可能是是亲人、好友、同学、同事，也可能只是网络中的萍水之交，如都是 QQ 黄金会员。因此，用户之间的关联应该有一个权重，表明了用户之间的紧密度、信任度，一个用户的好友可能是另一个好友的亲戚，因此推荐系统有助于帮助用户挖掘潜在的熟人。

同样自推荐系统诞生后，学术界对其的关注度一直不小。从 1999 年开始，在美国每年由计算机学会负责召开电子商务研讨会，会中已经发表了数以千计的推荐系统论文。在 2001 年，ACM 信息检索专业组考虑把推荐系统独立拆分，作为会议诸多独立研究主题之一。2001 年同年，在人工智能联合大会上，推荐系统也单独列为一个主题。2011 年的 KDD CUP 竞赛中，两个竞赛题目分别为音乐评分预测和识别音乐是否被用户评分 (www.kddcup2011.org)。2012 年的 KDD CUP 竞赛中，两个竞赛题目分别为腾讯微博中的好友推荐和计算广告中的点击率预测。(www.kddcup2012.org)

1.2.2　推荐系统的应用

作为 IT 数据挖掘算法工程师，笔者经常听到同事开玩笑的说：推荐系统就像万金油，抹哪哪灵。对于诸如 linkedin 的社交网络，推荐系统改变用户扩展人脉的模式和方法，而这是基于一种假设：你的朋友的朋友有可能就是你熟悉的人。对于诸如淘宝的电商平台，搜索提供的静态体验，并不足以让用户产生购买欲望，推荐系统加强了交互，包括用户和商品、用户和用户，一个人的消费可能带动一群人模仿，成就了一种极致的营销模式。对于诸如滴滴出行的网约车平台，给定某一时刻、起始经纬度、终点经纬度、车型，根据推荐算法一定会有一个最优派单，使得司机和乘客所得的好处，远远大于平台的抽成费用，形成了我们所说的三方共赢，只有“倒霉”的传统出租车利益受损的局面。总体说来，一个成功的个性化推荐系统的应用主要表现在以下几个方面：

(1) 将潜在用户转变为购买者：用户在浏览的同时并不意味着一定是要消费，也许只是看看，遇到合适就买，没有就算。个性化推荐系统的职责之一是能够洞察用户的潜意识，帮助用户找到其感兴趣的商品，从而促成购买过程。

(2) 提高平台的连带销售能力：有时候个性化推荐系统需要一点联想能力，如用户购买了手机，那么推荐手机壳就是一种明智的联想，会让用户产生“你懂我”的感觉。

(3) 提高客户对平台忠诚度：个性化推荐系统就是那个时时刻刻为用户着想的机器人，每一次推荐只是那么一点点，不是很多，但都很好，这种依赖就是俞军先生所说的体验壁垒，对于用户来讲，从滴滴出行换到 Uber，功能还是原来的功能，只是用户习惯的打车方式都变了，以至于无法接受 Uber。

1.3　用户画像的简介

用户，指企业的潜在消费者，是构成现有用户的大部分群体的统称。画像，是对一个用户的可视化、客观的描述。用户画像就是能够客观、可视化地描述潜在消费者的模型。用户画像建模的关键工作就是为用户打上合适的标签，标签通常是人为规定，且具有高度精炼的特征标识，如消费能力、偏好、年龄、性别等，将所有用户标签综合起来，抽象出本质，如忠诚度、消费度、满意度等，基本就可以勾勒出该用户的商业消费轮廓。

1.3.1　用户画像的产生背景

当互联网步入信息时代后，用户行为数据的极大丰富性给企业及消费者的消费行为带来一系列问题与变革。最大的问题在于电子商务的用户数量相比传统商务，膨胀了成百上千个数量级，单纯依靠人工方式已对其无解，2015 上半年，我国网民已达到 6.68 亿，预计年底能够顺利突破 7 亿，其中使用手机上网

人群占整体 88.9%，而手机上网存在着独特性、唯一性和私密性的特点，每个人的手机都是一套独特的生态系统。最大的变革莫过于，消费者的一切行为信息都是可数字化，随着大数据工程技术的日益精湛，带宽、计算资源、存储资源也变得极大丰富起来。这使得企业有能力把专注点回归到问题的本质，即利用信息化管理方式为每位用户建立一个档案，根据用户的生活习惯、消费行为和社会属性等信息，抽象出的一个标签化的用户模型用以精准刻画用户，基于此进而充分挖掘用户潜在的商业价值，随着用户使用时间越长，模型就越能积累多的数据，也越能精确把握用户的消费习性，反过来越能促使用户的消费行为，形成一个良性循环，至此用户画像的概念也就深入企业和用户之心。

大数据时代的用户行为数据就像是做饭的米，如果想让其变成香喷喷的白米饭，还面临很多问题：1、用户行为数据通常包含了很多的噪声，包括用户无目的地的浏览数据、用于营销目的的分享数据、用于作弊的刷单数据等等，这些数据并不能够代表用户的真实意图甚至有时代表的是相反的意愿；2、用户行为数据通常需要将其所包含的意义抽象化，才有利用价值，比如根据用户最近一个月的浏览、购买记录，通过分析、抽象、挖掘得出用户的活跃度、消费能力和忠诚度，这才能为算法所用；3、用户行为是是一个不断迭代的行为，如何均衡新旧行为数据的权重比，是一个很严肃的问题，比如一个用户上一个月购买不断，最近一个月却很少登录，那么我们应该怎么归因用户的这种行为？以及如何刻画这个时期的用户消费状态？如果用户处于将要流失的状态，又该如何做；4、不管到哪，我们总会遇到与自己志同道合的其他用户，我们其实还是比较关心这些人的选择，如果能拿来做为自己的参考也不是一件坏事，因此，如果存在一种机制，可以将有相同特征的用户抽象成一个代表，进行交叉推荐，则既方便用户消费又能促进企业营收。基于以上种种问题，我们确定选用了用户画像。

1.3.2　用户画像的应用

用户画像建模的过程，就是数据清洗、数据分析、数据挖掘，最后得出用户的抽象概念的过程，用户画像的本质就是了解企业的用户，然后完善产品运营提升用户体验，提升盈利，用户画像可以为包括推荐系统、运营推广、策略制定等提供数据支持。除此之外，用户画像可以帮助企业寻找潜在目标用户，在与用户的交互上了解其偏好，促成购买，实现精准运营和营销，用户画像改变了以往闭门造车式的商业交易模式，通过事先调研用户需求反馈，设计制造出更适合用户的产品。具体来讲，用户画像的应用包括：

- 完善及扩充用户信息：用户画像代表了用户的信息全貌，因此寻找足够多的数据是用户画像建模的前提条件。我国在各方面都是很大的长尾市场，互联网很大程度上弥补了信息的不对称，移动互联网又让把信息在精准送达到任意一个用户面前，尽管如此，根据 2/8 原则还是导致了大多数的用户和商品的数据是空缺着的。同时，在实际中用户的信息也可能提供得不

尽完整，如对于没有填写性别信息的用户，用户画像可以通过用户兴趣探索模块，生成用户数据，可见，用户画像不仅消费数据，也可以生成数据。

- 打造健康的生态圈：在掌握用户信息的基础上，电子商务平台就可以对自身的状况进行分析，从相对宏观的角度刻画用户种群的分布，从基础上把握市场的生态环境，挖掘出商品的最大价值，帮助企业提高收入。例如笔者曾经发现，通过与当前热门电影保持同步，通过适时发布引导疯传引爆点、跟进推广周边手机主题，可以很好的带动用户的消费行为，用户的消费与此同时也刺激了第三方设计师紧跟时尚潮流，尽可能第一时间发布引领流行的作品。

- 支撑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。在实际场景中，影响用户对商品的使用黏度的因素很多，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网上商店到退出使用，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，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。

- 市场安全领域的应用：有时候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。

1.4　工程背景

小米科技有限公司作为国内发展较快的互联网企业，活跃用户过亿，移动端用户比例高，有着大量的用户和丰富的用户行为，这些为推荐系统的应用和优化提供了不可或缺的条件，我们基于 MIUI 主题应用商店开发的手机主题推荐系统，作为用户和主题包之间的桥梁，体现出超强的变现能力。但现有的手机主题推荐系统也面临着一些问题。

(1) 新用户冷启动问题。当前使用的推荐算法，包括最近邻的协同过滤算法、PageRank 排序算法、关联规则挖掘是根据给定用户对某些物品的行为数据，

给每个用户推荐 Top-N 个其最喜欢的物品，当一个新用户进入一个站点时，我们对他的兴趣爱好还一无所知，这时如何做出推荐是一个很重要的问题。现有的机制是向用户推荐那写普遍反映比较好的物品，也就是说，推荐完全是基于物品的，这就会使热门的商品越来越热, 冷门的商品越来越冷，代价就是加剧了热门商品的马太效应。

(2) 数据稀疏问题，通过观察我们发现只有约 20% 的用户有过多于 5 款/日主题的浏览记录，意味着大多数用户的消费处于待挖掘状态。与此同时，我们发现只有约 20% 的主题包有过多于 10 次/日的浏览次数，意味着大多数主题包的消费处于待挖掘状态，又是一个“蛋和鸡”的问题：要形成好的推荐，首先需要有大量的用户行为支持，这样才能得到足够多的推荐数据，这里问题的关键在于推荐系统如何首先能在数据稀疏的情况下给出优质的服务，打破闭环。

(3) 不断变化的用户喜好，这个问题主要分为俩类：1、用户一直喜欢某种类型的主题包，只是长时间没有机会接触，如一位男性用户喜欢美少女主题包款式，虽然不会主动查找，但如果不经意看到一款制作精美的美女主题包，可能还是会购买，这就是用户的长期兴趣。2、用户之前喜欢某种类型的主题包，之后转为喜欢另外一类主题包，如用户刚开始喜欢清纯系，后来转为温柔系，这时如果向用户推荐温柔系主题包更有可能被其接受，这就是用户的短期兴趣。

(4) 重复推荐的问题，手机主题包属于电子虚拟商品，它的特性是第一次下载需要购买，之后下载则免费，现有的推荐系统会重复推荐用户之前购买过的主题，导致占用有限的推荐位来显示无法变现的信息，并且会给用户一种不专业、不智能的体验。

(5) 其他问题，如推荐商品长尾性 [30] 有待加强、隐性喜好 [27] 难以挖掘、偏激的用户和另类的产品、推荐系统的作弊行为、用户请求量大等。这些问题相对来讲影响范围小，本论文不做过多讨论。

我们发现，如果在底层数据仓库层和推荐系统之间加一个用户画像模块，会有效提升推荐系统的各项性能。1、对于新用户冷启动问题、数据稀疏问题，关键是收集足够多的用户基本信息，在没有或者只有少量用户行为的情况下依靠用户画像对用户推荐比较合理的主题。2、对于不断变化的用户喜好，我们通过用户画像存储用户用户长期，通过用户兴趣探索获得用户短期兴趣，并针对手机主题市场的特点，利用线性衰减算法融合用户画像和用户兴趣探索，使得推荐结果能兼顾俩者。3、对于重复推荐的问题，我们在用户画像中维护一个白名单，用来存储用户曾购买过的所有主题信息，格式为 (userId,itemId,buyTime) 这样的三元组, 避免向用户推荐已购买过的主题。除此之外，我们也通过探索用户小众

兴趣提升推荐系统的长尾发掘能力，加强了对小众主题包的推荐力度。主要思路是分析用户所有的行为数据，针对占大多数的冷门主题 (即包含小众标签的主题) 会赋予一个倾斜因子，这样会使得冷门主题更有可能被探索出来。

总之，我们采用构建用户画像的办法分析、处理、挖掘现有的用户信息，尽可能多的识别用户基础特征和兴趣偏好，达到精细化推荐的目的，包括后续定向广告投放、市场营销等功能需求也都是围绕建立更细致、准确的人群画像而展开。

1.5　推荐系统开源项目介绍

工欲善其事，必先利器，关于大数据，有很多令人兴奋的事情，但如何分析、利用好如此多的数据也带来了很多困惑。好在开源观念盛行的今天，有一些在大数据领域领先的免费开源技术可供利用。

- Redis：Redis 是一个 remote 类型的内存数据库，它不仅性能强劲，可扩展性好，而且还具有高效的复制特性，生来就是为解决实际问题而设计的数据模型。在实际工程中，笔者利用 redis 做俩件事情：1、存储计数指标，包括用户浏览、下载、购买等行为的次数，以天为单位，因为内存有限只存储最近一个月的数据；2、存储最近俩周的行为数据，按照 timelines 格式存储，天然支持时间排序。key 有用户 id，商品 id，交易 id，value 格式为商品类型: 商品价格: 折扣，score 为下单时间戳。

- Apache Hadoop：Hadoop 是由大名鼎鼎的 Apache 基金会所开发、维护的一个分布式系统，是第一款开源的用于分布存储大型数据集的开源框架，助力各企业迅速从海量数据中挖掘出“金子”。在实际工程中，每日凌晨，把当日的 MySQL DB 中的数据复制一份到 hdfs 文件，partition 为当天时间。

- Apache Hive：Hive 是著名社交网络公司 facebook 开源的在 Hadoop 上的数据仓库基础构架。Hive 的使用方式如传统的 DB，类似 SQL 查询语言。同时，hive 也帮助用户屏蔽具体的 MapReduce 开发，因此十分适合大量数据的统计分析。实际工程中，笔者利用 hive 计算超过一个月的数据计算，其小时级别的计算时间限制了其只能适用于离线计算。

- Apache Spark：Spark 是加州大学伯克利分校所开源的基于内存的通用并行计算框架，同时 spark 如 hadoop 一样容易扩展，因此适和完成数据挖掘需要大量计算迭代的任务。实际工程中，笔者利用 spark 自带的机器学习 mllib 库做推荐系统的模型训练。

- Apache Kafka：Kafka 是美国求职社交公司 linkedin 开发、开源的一种高吞吐量的分布式发布订阅消息系统，可以轻松处理现有所有大交易规模的用

户行流数据，包括今日头条、滴滴出行级别的交易量都是利用 kafka 做导流。kafka 的一个应用特点是实时性高、吞吐量巨大，任何要求实时处理的应用场景，Kafka 都是一个可行的解决方案。实际工程中，笔者利用 kafka 流作为 redis 数据的上游数据源。整体数据流结构为：app->mysql db->mysql binlog->kafka-> 清洗、去重 kafka->redis。从 app 到 redis 理论延迟为毫秒级别，其中的 kafka 是关键实现部件。

1.6　论文结构

本文的其余正文内容由以下章节组成：

- 第二章首先介绍了推荐系统基本概念，然后详细介绍了用户画像和用户兴趣探索。
- 第三章主要讨论了如何利用用户画像建模解决推荐系统的冷启动问题，从而改善推荐系统的新用户留存率。最后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　基于用户画像的推荐系统综述

2.1　引言

自从 1992 年著名的施乐公司的科学家们为了解决困扰已久的信息负载问题，第一次从概念上提出协同过滤的算法模型。1998 年，林登及其同事们成功申请了 item 协同过滤技术的专利，经过多年的工程实践，美国电商亚马逊公司的工程师们骄傲的宣称：在公司所有的销售量，推荐系统占比已经占到整个 Gross Merchandise Volume 的百分之三十以上。不久之后的美国公司 Netflix，因为其创始人与前任公司签署有若干年内不得从事同行工作的限制，于是通过举办推荐算法优化竞赛绕开限制，用以开发出更好的推荐算法。此次竞赛吸引了数以千计的团队参与角逐，期间进行了上百种的算法模型组合、优化的尝试，虽然 Netflix 公司为冠军团队支付了百万美金，但回报是 Netflix 推荐系统的快速发展以及营收的俩位数增长。其中冠军团队凭借 Sigular Value Decomposition 和 Gavin Potter 跨界引入的心理学方法进行的组合算法模型，在诸多优秀团队中脱颖而出。其中，矩阵分解的核心是将一个非常稀疏的用户评分矩阵 R 分解为两个更小的矩阵：只包含 User 特性的矩阵 P 和只包含 Item 特性的矩阵 Q，利用 P 和 Q 相乘的结果 R’ 来拟合原来的评分矩阵 R，使得矩阵 R’ 在 R 相同位置之间的损失函数值尽量的小，通过定义一个 R 和 R’ 之间的距离定义 (一般为曼哈顿距离)，如果矩阵 R’ 是正定矩阵，那么把矩阵分解转化成梯度下降求解的局部最优解，就是全局最优解。与此同时，Pandora、LinkedIn、Hulu 等网站在个性化推荐领域都展开你争我抢的竞争势头，使得推荐系统在各个细分行业、垂直领域开始全面开花，都有了不少爆发性进展。但是，对于拥有全品类的综合性购物电商、广告营销，推荐系统的进展还是缓慢，主要原因是因为不同类型的商品，消费者的心态也是不同的，例如大型家电，消费者肯定是先看了又看、选了又选，从价格、定位、功能到噪声比、性价比，大多数都会先做足了调查，才会购买；与此相反，对于日常用品消费者可能眼睛都不眨就购买了，对于这俩种极端的消费情况，推荐系统需要做出截然不同的推荐策略，具体的，单个模型在母婴品类的推荐效果还比较好，但在其他品类就可能很差，很多时候需要根据场景、推荐栏位、品类等不同，设计不同的推荐模型。同时由于用户兴趣随时间会不停的变动，需要一种机制，使得推荐系统能定期对数据进行评估、分析，要命的是对于不同类型的商品有不同的更新频率，这就对推荐系统提出了更加智能化的挑战。还有，如果定期更新模型，则可能会因为计算资源的限制导致无损害推荐的实时性，因为模型训练也要相当 cpu 计算时间，而传统的 Hadoop 的方法实在是无法进行大的更新频率，spark 框架又因为昂贵的内存限制了其计算容量，最终业务会到达一个数据量，此时的推荐效果会因为实时性问题达到第一个计算瓶颈。推荐算法包括基于人口统计学的推荐 [32]、基于商品内容的推荐 [42]、基于 user/item 的协同过

滤 [33] 的推荐等。基于内容的推荐 [34] 对物品冷启动问题免疫，但是无法解决用户冷启动问题 [35]，还有过拟合的问题：即在训练集上有比较好的表现，但在实际应用中效果往往不尽人意，推荐系统的通用性和移植性往往比较差，适合针对细分行业下的商品做推荐，一旦换了产品类型，往往需要构建新的模型。基于邻域的协同过滤算法，虽然没有领域知识要求，算法通用性好，但存在有冷启动问题、数据稀疏性问题。

由此，笔者在实际工程中，针对传统推荐算法的种种弊端，选择了用户画像。伟大的数学家、计算机学家 Knuth 先生说：如果遇到一个不好搞定的问题，那么就该添加一层中间层，用以屏蔽掉问题。实际上，用户画像作为底层数据仓库和上层推荐系统的缓冲层，起的就是这种作用。

2.2　用户画像的研究现状

2.2.1　用户画像的组成部分

基于内容和用户画像的个性化推荐，有两个实体：内容和用户。需要有一种文本机制联系这两者的东西，我们定义其为标签。内容特征文本化为标签即为内容特征化，用户兴趣文本化标签则称为用户特征化 [36–40]。因此，对于基于用户画像的推荐，主要分为以下几个关键部分：

(1) 标签库

标签是联系用户与物品、内容以及物品、内容之间的纽带，也是反应用户兴趣的重要数据源。标签库的最终用途在于对用户进行行为、属性标记。是将其他实体转换为计算机可以理解的语言关键的一步。标签库则是对标签进行聚合的系统，包括对标签的管理、更新等。在用户画像的过程中有一个很重要的概念叫做颗粒度，就是我们的用户画像应该细化到哪种程度。举一个极端的例子，如果“用户画像”最细的颗粒度应该是细到每一个用户每一具体的生活场景中，但是这基本上是一个不可能完成的任务，同时如果用户画像的颗粒度太大，对于产品设计的指导意义又相对变小了，所以把握好画像的总体丰富程度显得异常重要了。可通过调查问卷的形式来减小颗粒度。一般来说，标签是以层级的形式组织的。如体育为一级维度、篮球为二级维度、NBA 篮球为三级维度等。

(2) 内容特征化

内容特征化即给商品打标签。目前有两种方式：人工打标签和机器自动打标签。针对机器自动打标签，需要采取机器学习的相关算法来实现，即针对商品描述文本，生成一系列标签，为商品选取其中匹配度最高的几个标签。这不同于通常的分类和聚类算法 [41]。可以采取使用分词 + Word2Vec 来实现，过程：将文本语料进行分词，以空格，tab 隔开都可以，使用结巴分词。使用

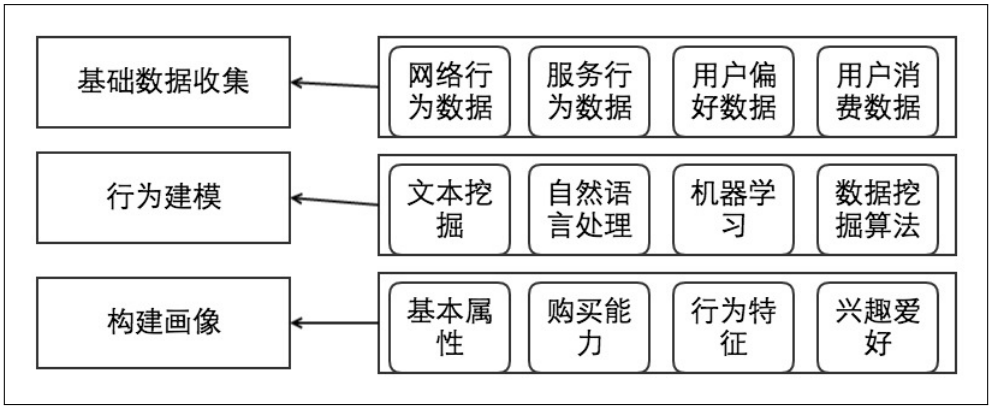

图 **2.1**　用户画像的构建周期示意图

word2vec 训练词的相似度模型。使用 tfidf 提取内容的关键词 A，B，C。对每个现存的标签，计算关键词与此标签的相似度之和。取出 TopN 相似度最高的标签即为此商品的标签。如对于《小羊肖恩》主题包，现有儿童、动漫俩个标签，描述文本有：一部史诗般的二次元欢乐片。经计算“二次元”关键字与现有标签相似度最高，则更新二次元到此商品的标签库中。

(3) 用户特征化

用户特征化即为用户打文本标签。通过用户的行为日志和一定的模型算法得到用户的每个标签的权重。用户对内容的行为：点赞、不感兴趣、点击、浏览。对用户的反馈行为如点赞赋予权值 1，默认为 0，不感兴趣为-1；对于用户的浏览行为，则可使用点击、浏览作为权值。对商品发生的行为可以认为对此商品所有标签的行为。用户的兴趣是时间衰减的，即离当前时间越远的兴趣比重越低。时间衰减函数使用 1/[log(t)+1]，t 为事件发生的时间距离当前时间的大小。要考虑到热门内容会干预用户的标签，需要对热门内容进行降权。

2.2.2　用户画像的构建周期

用户画像，即用户信息标签化，就是企业通过收集与分析消费者社会属性、生活习惯、消费行为等主要信息的数据之后，获得用户的数据标签库。构建周期如图 2.1。

(1) 数据收集

数据收集大致分为四类：1、网络行为数据包括页面浏览量、活跃人数、访问时长、浏览注册转化率、注册活跃转换率等。服务内行为数据：点击浏览路径、网页停留时长、滑屏次数、滑屏频率、滑屏时长。用户内容偏好数据：点击、浏览、收藏内容、评价、评分、评论内容、社交内容、品牌偏好等。用户交易数据（交易类服务）：购买率、折扣率、导流率、流失率等。收集到的

数据没必要是百分之百的准确，大体差不多即可。应用中，具体就是在数据清洗阶段过滤一部分不靠谱的异常值，验证、更新数据这块需要在后面的阶段中建模来再判断，比如某用户在性别一栏填的女，但其语言数据显示其为男的概率更大，根据业务再选择丢弃数据还是更新数据。

(2) 行为建模

该阶段是对收集到数据进行建模，目标是抽象出用户的文本标签，这个阶段不应该再纠结数据的正确性，而是应该注重大概率事件，通过统计学假设检验尽可能地排除用户的偶然行为。这时也要用到数据挖掘算法模型，对用户的行为进行回归预测，比如已有一个线性回归函数：y = kx + b，X 代表用户行为，y 是函数拟合的用户喜好度，y' 是用户真实偏好，我们通过不断的训练数据，利用参数 k 和参数 b 来得出最新损失函数下的值，用以精确模拟 y'。

(3) 用户画像基本成型

该阶段是行为建模的深化，需要利用用户的基本属性，如性别、地域、年龄，得出用户更高层的抽象概念：消费能力、忠诚度、活跃度、社交爱好等。因为用户画像永远也无法百分百地拟合现实中的一个人，只能做的就是不断地去减小拟合的损失函数，因此，用户画像需要根据变化的基础数据不断修正已有的更高层的抽象概念，尽可能模拟用户的变化趋势。

(4) 数据可视化

最后是数据可视化分析，这部分是最能体现推荐系统的产出，因为人类对数据不如对图画来的敏感，在此步骤中一般是针对群体做进一步的抽象，按照消费习惯、消费能力、消费偏好把用户归类为一类人，比如可以根据用户对价格的敏感度细分出高价值用户、核心用户、高忠诚用户。而决策层所做出的评估也应该是基于某一群体的潜在价值分布。典型的用户画像如图 2.2

2.2.3 用户画像的建模

用户画像的建模包括内容标签化和标签权重量化。建模过程：1、内容分析，从原先的物品描述信息中提取有用的信息用一种规范化的标签表示，有时候这种信息源自于作者提供的描述，有时候源自于用户的评价，不管如何，都需要人工审核验证正确性；2、上传、记录用户注册信息，生成用户基本信息，这些信息基本是不会变化的；上传、记录用户行为数据，这些数据是不断变化着的，通常是采用数据挖掘算法从潜在物品集合中取出若干个结果表示用户喜好的模型。例如，一个网页推荐系统，可以通过分析用户过往浏览过的文章，得出用户喜欢浏览类似于范冰冰的花边新闻，如果用户点击了所推荐的文章，则说明

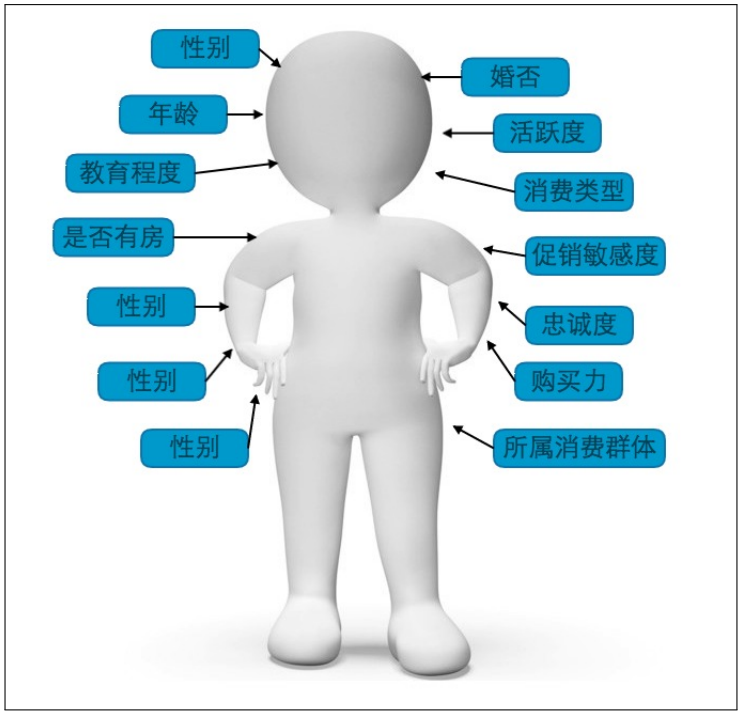

图 **2.2**　用户画像示意图

分析正确，否则需要根据反馈重新训练模型，从而实现一个反馈-推荐-反馈的闭环；3、推荐系统得出推荐集合后往往需要取 topN，因为推荐系统的本质在精不在多。通过定义一个距离算法，匹配用户标签和商品标签的相关度，相关度一般正则为 0-1 之间，结果是一个二元的离散量：<pid,score>。根据相关度将生成一个用户潜在感兴趣的物品评分列表，然后去掉用户之前看过的商品，取 topN 即可。例如在电影用户画像的建模中，首先分析用户打分比较高的电影的共同特性，包括导演、演员、风格等，这些电影的标签就会成为此用户画像的一部分，根据打分的多少，给定一个合适的权重值。用户-标签用矩阵 A 表示，电影-标签用矩阵 B 表示，A 乘 B 得出矩阵 C，C 代表了用户与电影之间的相关度，固定一个用户，对所有相关度不为零的电影做排序，取 topN 即是推荐结果。用户画像建模的根本在于用户标签的获取和权重的定量分析。

对于商品描述，也可以做进一步的处理，丰富商品的标签集合。其实和文本处理类似，笔者选择使用目前应用最广泛的方法：TF-IDF 方法。设有 N 个文本文件，关键词 k_i 在 n_i 个文件中出现，设 f_{ij} 为关键词 k_i 在文件 d_j 中出现的次数，那么 k_i 在 d_j 中的词频 TF_{ij} 定义为：$TF_{ij}=f_{ij}/max_z f_{zj}$，其中分母中的最大值是通过计算这个文本 j 中所有关键词出现的频率得出。附图给出了 3 个短文和 5 个关键词，以关键词人为例，该关键词在文本 1 中出现了 1 次，而文本 1 中出现次数最多的关键词是事，一共出现了 2 次，因此 $TF_{11}=0.5$。一个关键词经常在许多文件中出现，则该关键词能表示文件的特性的意义就会较小，试想我们考察关键词 i 出现次数的逆，也就是 $IDF_i=\log(N/n_i)$，这个想法和 Adamic-Adar 指数思路基本相似，关键词 i 在文本文件 j 中的权重于是可以表示为 $w_{ij}=TF_{ij}*IDF_i$，

而文件 j 可以用一个向量 $d_j=(w1_j, w2_j, \cdots, wk_j)$，其中 k 是整个文本库中关键词的个数。一般而言，向量应该是一个稀疏向量，即其中很多元素都为 0 。如果把用户今日点击、浏览、购买的商品抽象成一个标签向量，则可以通过用户标签向量-商品标签向量的点乘得出一个数值，从所有数值中把相似性最大的那个产品的标签更新给该用户画像，第二大相似性的产品标签权重减半更新给该用户画像，以此类推，完成用户画像的建模过程。

文本1：不做软事，不说硬话，对事不对人。
文本2：多少事，从来急；天地转，光阴迫。一万年太久，只争朝夕。
文本3：青春之所以幸福，就因为它有前途。

关键字包括人、事、硬话、一万年、朝夕、青春、幸福、前途

2.2.4 用户画像和推荐系统的评测

首先，用户画像作为一个工具，只用在运用到某一场景才有意义，并能评估出其产出，因此本节主要介绍推荐系统的评测，根据推荐系统的表现好坏才能评估出用户画像的推荐质量，本节介绍评测推荐系统常用的实验方法。

(1) 离线实验，从日志系统中直接取得用户最近单位时间的行为数据，然后将这些数据分成俩部分：训练数据和测试数据，一般来讲俩者的比例大致为：8 比 2，然后利用训练数据集迭代拟合用户的兴趣模型，在测试集上进行回归测试。过程简单、容易模式会管理，不需要人为干预，有很多的数值计算的开源软件库可以用：如 google 公司出品的 TensorFlow。能方便快捷的测试大量不同的算法。

(2) 用户调查，又叫问卷调查。离线实验得出的是客观规律下的准确率，但是客观的准确率不等于用户实际的满意度，一般来说问卷调查需要只需要在小范围之内进行，即可得出大差不差的用户满意度调查，优点是可操作性强。

(3) AB 测试，标准的 AB 测试是指通过一定的规则把类似的用户群随机分成俩组，采用旧模型的分组叫对照组，采用新模型的分组叫实验组。通过对用户展示不同的模型，得出用户的使用指标，关键是各种转化率，这样仅仅通过对比俩者的转化率即可得出各个模型的优劣。策略实验的难点在于如何找到合适的实验设计方案。通过时间交错能够在一定程度上减少由时间片带来的误差，这样就有一个难题: 如何选择合适长度的时间片。策略实验往往伴随着携带效应 (carry-over effects), 也就是上一个时间片的策略会对下一个时间片带来影响。笔者和同事们提出一个方案，当选择适当大的时间片的时候，通过 A/A 测试的数据调整 A/B 的结果，具体来说，如果 A/A 的结果是 0.4%, A/B 的结果是 1.2%. 那么我们认为 A/A 是真实的时间片之间的差异, 我们需要用 A/B - A/A 去调整时间片带来的影响，

2.3　用户画像在推荐系统的应用现状

Amazon 的仓库里堆着数百万图书，Netflix 的服务器中存储有数万部电影，淘宝平台上的小卖家总共拥有 8 亿件物品，除此之外，这三家公司都保留有数以亿计的用户行为数据。互联网电子商务开始积累了海量的用户数据，然后因为数据量过于庞大，有用信息如金矿中的金子一样很难挖掘利用，与此同时，用户发现常常需要面对过多的选择。心理学研究证实过多的选择会使人犹豫不决，导致消极等待，最终可能放弃消费的决定，这个问题严峻到可以造成肉眼可见的用户流失。近代统计学理论的发展加上最近几年的数据科学和数据挖掘工程的进步，为电子商务平台提供更有效的应对方案：推荐算法。推荐系统在帮助用户解决信息过载问题的同时，提升了企业价值。如今的企业不再局限于传统的推荐功能，通过建立完备的用户画像，推荐系统可以帮助企业更了解用户，在推广、反作弊、精细化运营等领域中发挥重要的作用。

目前使用最广的推荐系统，主要是基于内容做推荐，根据 RecSys 大会（ACM Recommender Systems）中与会者的反馈，已经有不少公司和研究者先行一步，尝试基于用户画像做推荐。利用用户的画像，结合空间、时间、天气、环境、经纬度等上下文信息，可以给用户带来不一样的感受。用户画像是一种更高级的工具，在解决把数据转化为商业价值的问题上更甚一筹，相当于从海量数据中挖出俩倍的金银。用户画像中包含着高质量多维的数据，用以记录用户长期的行为，据此还原用户真实的消费特征、教育背景、兴趣偏好。科学中国网曾在《大数据揭秘：淘宝上的假货、次品都卖给了谁?》中报道了淘宝不良商家如果利用买家信息欺骗消费者 [43]：1、分析数据看人下刀，宰用户没商量，真相就是消费者的消费记录、购买记录、客单价等都将作为参考数据被系统识别，商家会根据这些记录评估消费者能不能分辨假货，再把假货卖给对方。2、看退货率，专欺负老实人，消费者的退货率、投诉率也会被识别到系统里，这些数据帮助商家判断用户好不好惹，退货率低于百分之十的用户，会收到更多次品产品。3、看收货地址，决定给用户发什么货，一些淘宝店家还会根据用户收货地址所在城市，决定给用户发什么货。要是用户所在城市没有该品牌的专卖店，或者用户没有购买过该品牌的产品，那系统将会放心的把假货或者仿品发给用户。利用用户画像人们可以做到如此精准的销售，当然上述例子是用户画像极其错误的用法。

2.3.1　基于用户画像的推荐系统的商业应用

作为全球社交网站中的翘楚，Facebook 在很早的时候就预言到了大数据 + 推荐系统 + 用户画像的无限前景。Facebook 自己的推荐系统就是需要利用分布式计算框架快速的帮助用户找到他们可能感兴趣的人、文章、分析、用户组等。Facebook 是个伟大的公司，一直为开源软件贡献着一份力量，最近在其官网就公布了 facebook 自己的推荐系统原理、性能及使用情况 [44]。Facebook 的推荐系统

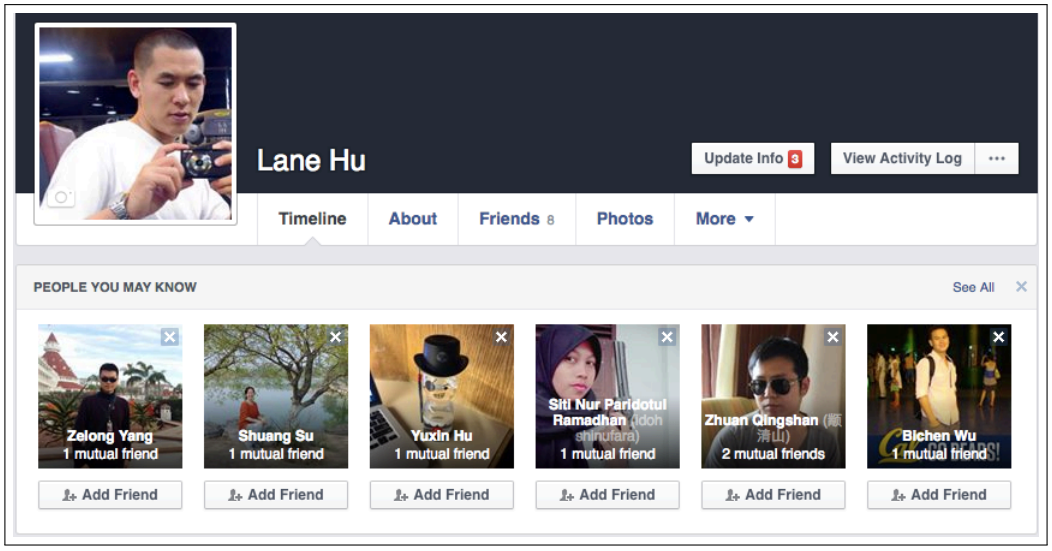

图 **2.3**　Facebook 个性化推荐用户界面

需要面对的数据量应该是所有互联网公司中的数一数二，约包含了 1000 亿级别的评分数、10 亿级别的用户数以及百万级别的虚拟商品，如何在如此庞大的数据规模下，仍然保持良好性能已经成为世界级的难题，而 facebook 解决了。即使是采用现在流行的分布式计算框架，Facebook 仍然不可能穷举每一对用户-物品的评分。团队需要寻找效率更高、耗时更少的算法来获得每个用户 topN 的推荐物品，然后再利用推荐系统计算用户对其的评分，这与我们之前解释的恰好相反。解决方案是利用 ball tree 数据结构存储商品的权重向量。all tree 可以贡献搜索过程 10-100 倍的加速率，使得推荐结果能够在合理时间内完成，典型的以空间换时间策略。最后，通过分析 Facebook 给出了一些实验的结果，表明，Facebook 的系统比传统系统要快 10 倍左右。因此可以轻松愉快的处理 1000 亿级别的评分数据。目前，该方法已经用到 Facebook 的多个 app 应用中，包括用户、用户组的推荐。进一步的，为了能够减小系统负担，Facebook 只是把稀疏度超过 100 的用户考虑为候选推荐集合。在初始迭代中，Facebook 推荐系统直接把用户历史上喜欢过的主页、群组以及不喜欢的群组都作为输入。最重要的是 Facebook 还利用 ALS 算法，从用户获得间接的反馈，这样算是完成推荐-反馈-优化-推荐的一个完美闭环。未来 Facebook 会继续优化推荐系统，持续改进部分关键模块，包括社交图、用户跳转路径、自动化参数调整以及较好的机器负载均衡策略等。Facebook 推荐主页如图 2.3。

Facebook 的用户画像进展也十分可观，几乎是与推荐系统同步发展。2011 年 12 月，Facebook 发布了里程碑式的大数据产品——Timeline，通过开发 API 接口，允许用户自行编辑个人的时间轴：在什么时间、什么地点做了什么，遇到了谁，可以说在这条时间线记录这个人的全部生活故事。Timeline 通过帮用户回忆自己的点点滴滴的同时，完成了用户数据捕获、存储，而一旦拥有了这些历史数据，Facebook 就可以做进一步的数据分析、挖掘，这时的 Facebook 就如同和你

图 **2.4**　豆瓣电台个性化推荐用户界面

从小长大的小伙伴，一个懂你的陌生人。可以说用户留下的数据越多，Facebook 就越了解这个人，投放的广告就会更加精准，最终 facebook 利用庞大的用户数据生态赚足了钱。

豆瓣网是国内互联网行业中的小清新，美誉度很高，这是一家致力于帮助用户发现美好事物为己任的公司 [45]。不用费力设置播放列表，也不用费心思考要听啥，打豆瓣电台的推荐栏目，就能获得意想不到的快乐。如初恋般的音乐体验，让用户和音乐不期而遇，豆瓣电台坚持找到符合用户口味音乐。通过高度匹配的推荐结果，豆瓣电台为音乐爱好者提供了这样一种崭新的音乐盛宴，音乐本来就是件轻松的事，豆瓣电台回归了音乐最初定位。豆瓣电台的推荐算法综合用户的各种音乐行为 [46]。在豆瓣音乐中，通过量化音乐标签，谁喜欢哪些歌手，在听哪些，想听哪些，乐评，豆列等指标，会有相关的权重算法算出一个数值，最开始的时候只是一个最简单的计算公式，在经过多次产品迭代和用户反馈后，得出更靠谱的权重值累加算。其中权重最多的应该是电台本身的红心、踩、跳过、这些显性行为数据。豆瓣电台糅合了包括数据清洗、分析、挖掘、整合、用户画像建模、编辑与运营、后台架构等等大量的因素，如此庞大的架构中，即便是推荐算法也只是现实的一部分。豆瓣电台推荐页如图 2.4。

豆瓣电台的用户画像结构大致有俩快：享受时间和购买时间，用户画像的目标用户群体是在线观众，通过用户购买时间差区别并得出分类标签。如通过分析得出此类用户大多数是周末购买音乐工作日静下心慢慢享受。用户年龄、职业和地址，这是根据用户的经纬度和注册信息来对用户进行分类的用户画像建模。用户的经纬度分为一线城市，如北京、上、二线城市，如武汉、深圳、厦门、其他三类，如合肥、呼和浩特。年龄分为小于 25 岁、26 到 35 岁、36 到 45 岁和 46 岁以上四类。据统计结果表明：按人群经纬度分布，大致与橄榄球相似，二线城市的人群占中间的大部分，其他城市人数飞速增长；按用户年龄分布，九十后用户占主体地位。同时对情侣关系的用户推荐喜欢度接近的音乐。按活跃程度分布主要分为 3 档：100 次以下；100-300 次；300 次以上。也可以同时考虑多个维度，包括活跃度、经纬度、年龄段，进行用户画像建模。

2.3.2 推荐系统的主要方法

推荐系统主要有俩种思路：评分预测和 Top-N 预测，核心的目标都是找到最适合用户的候选集合 s，从候选集合里挑选目标集合是一个非常复杂的非线性优化问题，通常采用的方案是用局部最优近似非线性最优，通过定义一个的损失函数，选取 Top-N [47]。

(1) 协同过滤的推荐

推荐系统的算法基于统计学、概率论、线性代数、微积分技术，找出用户最有可能喜欢的商品，应该是现代互联网电商的明星应用。目前用的比较广泛的推荐算法还属协同过滤推荐算法，其基本思想是根据与他兴趣相近的用户的选择，得出推荐商品候选集，取 topN 推荐给目标用户，用维度为 m×n 的矩阵表示所有用户对所有物品的兴趣值，这个值应该是根据用户历史行为数据得出，值越高表示这个用户越喜欢，利用特殊值 0 表示没有接触过。图中行向量表示某个用户对所有商品的喜爱程度，列向量表示某个商品对所有用户的吸引程度，因此单个元素 U_{ij} 表示用户 i 对物品 j 的喜欢程度。协同过滤分为两个阶段：预测阶段和推荐阶段。预测阶段是基于所有原始集商品，预测这个用户有没有可能对其感兴趣，量化为一个数值，只要值不为零即可归为候选集中；推荐是根据预测结果，先去重后去除消费过的商品，然后根据某种算法去 TopN 推荐给用户。按照用户-商品数值的得出类型，协同过滤算法分为基于内容的和基于模型俩大类 [48]。

协同过滤算法的基本思路是基于一个假设: 如果某类用户群对相同商品的打分比较类似，则表明他们的品味从某种程度上类似，由此可以推出在其它类项目的打分也应该类似才对。协同过滤推荐系统先定义好距离计算公式，然后搜索与目标用户相似的其他潜在类似用户，并根据类似用户的打分来量化潜在用户对指定商品的评分，最后选择评分最高的商品列表推荐给用户，同时可以给出令人信服的推荐理由：如某某人也购买过、评价过该商品。这种算法的优点很多：计算简单、精确度较高，能够自圆其说，因此被广泛采用。总之，基于 User 的协同过滤推荐算法的核心，就是先通过距离计算公式得出类似邻居，然后将最近邻的好评过的商品推荐给目标用户，很简单。

例如，在表 2.1所示的用户一商品评分矩阵中，行向量代表用户，列向量代表电影。表中的数值代表用户对电影的评价量化后的值。现在需要预测用户 Hanmeimei 对电影《教父》的评分 (用户 maggie 对电影《X-Files 要你相信》的评分是缺失的数据)。由表 2.1不难发现，Lane 和 Pony 对电影的评分非常接近，Lane 对《暮色 3: 月食》、《唐山大地震》、《X-Files 要你相信》的评分分别为 3、4、4，Hanmeimei 的评分分别为 3、5、4，他们之间的相似度最高，因此 Lane 是 Hanmeimei 的最接近的邻居，Lane 对《教父》的评分结果对预测值的影响占据最大比例。相比之下，用户 Jackson 和 maggie 不是 Hanmeimei

的最近邻居，因为他们对电影的评分存在很大差距，所以 Jackson 和 maggie 对《教父》的评分对预测值的影响相对小一些。

表 **2.1**　用户-物品表

	暮色 3：月食	唐山大地震	X-Files 要你相信	教父
Jackson	4	4	5	4
Marry	3	4	4	2
maggie	2	3		3
Hanmeimei	3	5	4	

尽管有这么多的优点，协同过滤算法也存在两大问题：1、数据稀疏性。一个大型的电子商务平台一般有百万级别的物品，用户可能接触到的商品占所有商品的百分之一不到，因此用户之间购买过的物品重叠性非常小，以至于没办法做推荐，一个办法是利用算法添补部分值 [49]。2、扩展性较差，因为一般来讲，电子商务平台中的商品变动很小，用户流入流出、日益增加、变动很大，基于用户的协同过滤算法需要不停的跟新迭代保证跟上用户变动的步伐。遇到这种情况，可以考虑基于商品的协同过滤算法，其基本思想类似于基于用户的协同过滤算法，只是相似性计算对象是商品，而商品一般变动很小可以忽略不计。如果我们知道物品 a 和 b 相似，而一般喜欢 a 的用户也喜欢 b，如果用户 A 喜欢 a，那么我们有很大把握得知 A 也应该喜欢 b，推荐了准没错。而物品之间的相似性比较固定，因此可以一次性计算出物品的相似度，将结果存储到 redis 中，推荐时查询 redis 即可。

2.3.3　推荐系统评测的测量指标

推荐系统存在三个参与方：用户、物品提供者和平台。好的推荐系统总体来说是一个能令三方共赢的系统。那么如何评价推荐系统功效呢？从用户角度，推荐系统必须满足用户的需求，推荐的应该是那些令用户感兴趣的、之前又没有遇到过的商品，即推荐精度。推荐系统还应该有预测用户行为的功能，通过历史展望未来，帮助用户发现那些他们原本没机会发现的小众商品，即长尾效应。最后推荐系统也应该能引导用户兴趣，推荐一些商品，虽然与用户兴趣无关，但是用户看见可能会产生兴趣的商品，即惊喜度。从平台角度，推荐系统能够让平台的营收上一个台阶。

(1) 用户满意度

用户满意度是最难量化的指标，也是最关键的指标。推荐系统的本意就是让用户满意。量化用户满意度可以采用用户问卷调查，还有一种更直接的方式，就是在推荐结果的侧栏设置俩个按钮，方便用户在线实时反馈意见，据笔者所知豆瓣的推荐物品旁边都有这类按钮，而亚马逊另辟蹊径，利用一些关键

性指标衡量用户对推荐系统的满意度，一般用点击率，用户停留时间，转化率等指标来度量。

(2) 预测准确度

如果是评分机制，则一般通过计算预测结果集合与用户实际消费集合直接的重合度，得出推荐系统的准确度。如果是 Top-N 推荐，则涉及到关键指标：召回率和准确率。准确率指在所有的推荐结果中有多少个是对的，其所占的比重，以推荐结果集合个数当除数。召回率则是指用户实际消费商品集合中，有多少物品出现在推荐结果中，已用户实际消费商品集合个数当除数。

(3) 覆盖率

就是指推荐系统有没有照顾小众商品，而不是一个劲的推荐热门商品。方法就是统计推荐结果的类型个数，比上所有商品类型个数，得到的商越大代表覆盖率越好，其他方法就涉及到信息学中的熵和基尼系数。

(4) 多样性

针对某一个用户，推荐结果中要变化多端，不能一根筋的推荐一种类型。比如电影，如果用户即格斗类的电影，同时又喜欢爱装小清新，那么推荐列表中就应该是两个类型的集合，除此之外，适当添补一些小众电影，三者比例按用户的爱好来推荐，比如用户爱格斗片多一点，文艺片也喜欢，历史片只是偶尔，那么推荐结果中最好也跟这个比例大差不差。

(5) 新颖性

如果系统推荐的物品其实是用户知晓的，那么这就是一次失败的推荐，完全失去了推荐的意义。一般来讲，用户都是期望推荐一些自己暂时还不知道的商品或者没看过没买过的商品。方法之一是过滤掉用户已经看到过、购买过、点击过的物品，除此之外，物品的平均流行度与其新颖度成反比，越冷门的物品越会给用户新颖的感觉。比如用户是周星驰的粉丝，那么推荐《临岐》就是一个很棒的选择，因为很少人知道这是周星驰出演的。

(6) 信任度

如果一个用户信任推荐系统，那么他不仅会频繁的选择查看推荐结果，还有适时的与推荐系统互动，包括反馈、评价、提建议等。如果用户信任推荐系统，从而获得更好的个性化推荐，这是一个良性循环。

(7) 实时性

有时候一个推荐系统的实时性的重要性大过天 [50]，试想一下，如果一个用户要买睡袋，但不知道哪款睡袋好，推荐系统如果这是恰当好处的推荐结果，

那么对于用户是很有意义的一件事情。反之，等用户买都买了，推荐系统在作出推荐，只会让用户难堪。

2.4 本章小结

本章简单概述了用户画像的研究现状，讨论了相关的建模过程。然后介绍了推荐系统的主要任务和问题，并从商业应用和学术研究两个角度介绍了推荐系统研究的现状，最后讨论了推荐系统的主要评测指标。

第三章　手机主题推荐系统整体设计与实现

3.1　前言

小米主题应用拥有成千上万款主题包，而一个用户整个活跃周期只能接触不到十分之一的主题，所以我们现在面临的一个问题是，如何帮助用户发现新的主题，这些主题同时满足俩个条件：1、不能和用户之前看过的、购买过的主题包重复。2、不能和用户之看买过的、购买过的主题不相关，而这也是我们开发的手机主题推荐系统所要达到的目标之一。除此之外，手机主题推荐系统要达到的目标之二是帮助第三方设计师推广其作品。手机主题应用本身既不生产主题包，也不消费主题包，存在的价值就在于为提供平台，能让用户、设计师和广告商从中受益。每个设计师都希望更多的用户体验、使用他们的主题。得益于个性化推荐系统的投入使用，我们现在可以把更多的主题包直接推送给那些潜在消费者面前。

本章节主要介绍如何介绍手机主题推荐系统的完整架构。手机主题推荐由推荐模块、用户画像模型、用户兴趣探索模块组成。推荐过程流程为：首先，推荐系统把用户画像模型中兴趣需求信息和推荐主题模型中的特征信息匹配，然后使用排序算法进行计算筛选找到用户可能感兴趣的推荐主题，最后推荐给用户。

3.2　手机主题推荐系统设计

推荐系统框架如图 3.1。最顶层显示的是推荐系统对外服务的客户端。由于不同展位的输入输出参数差异较大，因此这一层没有做过多的抽象，每个展位有自己特定的接口 Json 定义，接口层通过调用 ElasticSearch 搜索服务引擎实现秒级别的用户推荐结果列表。推荐系统利用离线方式更新 ElasticSearch 搜索服务器数据。用户画像模块除了作为推荐系统的输入数据外，也可以直接作为 Elastic-Search 的输入。用户兴趣探索模块定期扫描活跃用户和上架主题包，通过分析用户行为日志更新用户画像。从接口层接受到的每次响应请求会被记录成用户行为数据，包括请求的一些必要的上下文信息以及用户及主题包的特征信息。借助 HBase、Hive、MYSQL 等数据平台对原始日志进行处理，从而得到需要的各种数据及模型：包括用户的画像信息，用户之间的相似度，item 之间的相似度。在推荐系统的候选集生成这一块，重度使用了传统的 user based，itembased 协同过滤算法，协同过滤算法需要在用户行为较丰富的情况下才能奏效。而对于那些行为稀少的用户和新用户，需要根据平台的特点进行做好冷启动策略。对于 Spring Boot API User 输入、输出数据格式分别如Listing 3.2和Listing 3.2所示。

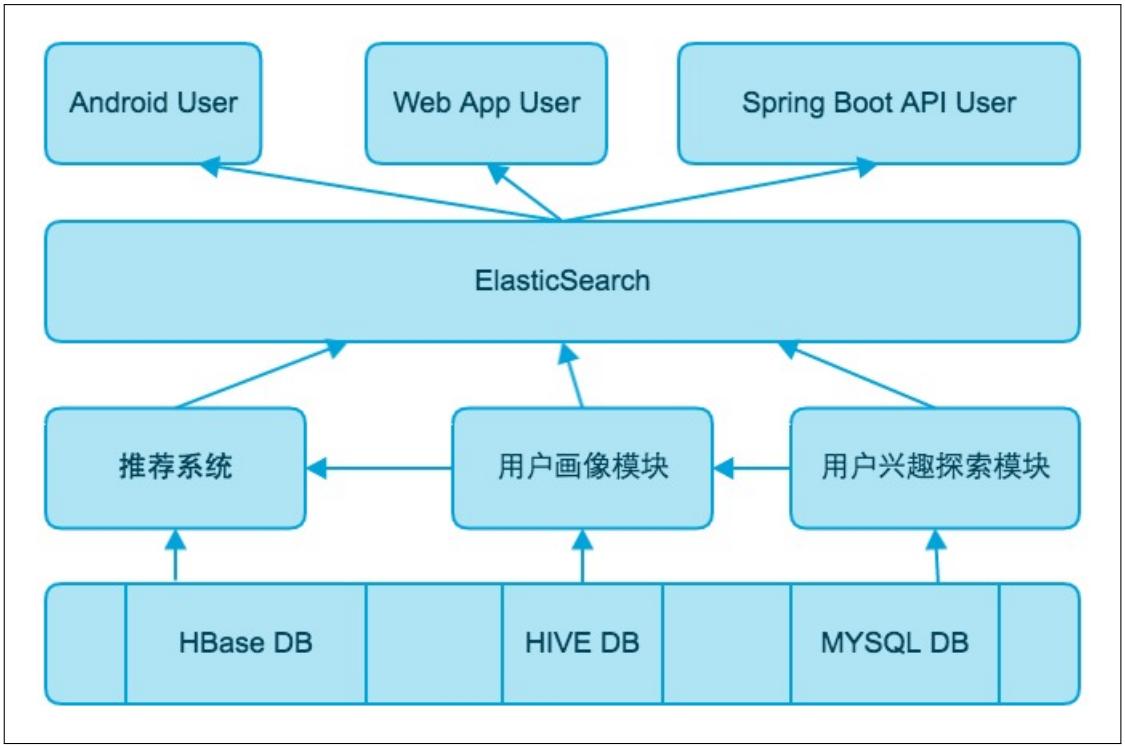

图 **3.1**　推荐系统引擎框架总览图

```
    {"user_id":"123",
      "dims":{"type":"normal", "free_or_charge":"mixed"},
      "white_list":{"id":1141},
      "max_number":"8",
      "start_date":"2016-07-01},
      "end_date":"2016-08-01}
```

```
  {"code":0,
  "message":"successfully",
  "data":{
    "total":111,
    "themes":[{
      "id":1141,
      "subscribe":1,
      "business":null,
      "keytype":"market:orderid",
      "tag":null,
      "name":"lovely baby",
      "displayName":"小可爱",
      "description":"家庭，儿童欢乐多",
      "author":"摆渡车1024",
      "sigModel":2,"type":2,"dependency":null,
      "createTime":1462447261000,
      "dims":null}
       ]}}}
```

3.2.1 数据采集和日志格式化

我们的数据采集来源包括有移动端埋点和用户请求。目前常见的前端埋点技术有三类：代码触发埋点、可视化触发埋点和延迟埋点，根据手机主题商店的业务特点和用户规模，我们选用可视化触发埋点，当用户在 UI 上点击了某个可埋点的控件时，会自动触发回掉函数调用接口发送相应事件的 log 信息，可视化触发埋点不同于代码触发埋点，其理念是把核心代码和配置、资源分开，在 APP 启动的时候通过网络更新配置和资源即可，不必每一个埋点都需要写代码，埋点产生的数据量很少，且只针对特定人群，所以用来做 AB 测试的数据源。用户请求是指用户每一次行为都会被上传、存储到服务器。获取数据后根据数据来源和存储方式，将日志格式化为：public 日志、nginx 日志、binlog 日志和 passport 日志，public 日志存放手机端用户请求 log，nginx 日志存放 Web 端用户请求，binlog 日志是将 MySQL 内容同步到 NoSQL DB 的数据，passport 日志存储用户验证信息等数据。

3.2.2 用户画像的构建

当上述日志格式化生成，通过每日定时任务扫描 passport 日志就可以获取新注册用户并为其在 ElasticSearch 创建一个 topic，同时利用移动端埋点功能获取到用户手机 IMEI 号、经纬度等基本信息，利用用户注册手机号或者邮箱账号获取用户的通信录和好友信息，借助好友信息完善此用户的用户画像，除此之外有时可以借助第三方接口获取用户的基本信息。用户画像的构建相对来说比较简单，只涉及到标签，没有产生权重。

3.2.3 商品标签的构建

小米手机主题应用商店里的主题包大多数是由第三方设计师创建、当设计师上传成品到官方产品库时会被要求填写作品标签，官方审核员也会更改、删除、添加一些标签，作品上架后用户在浏览、购买时产生的评论文本也会生成一些标签，商品标签的构建也只涉及到标签生成，没有产生权重。

3.2.4 候选集的生成

通过用户与商品的交互行为矩阵，我们最终得到了带有标签权重的候选集，具体算法是利用 Item-based Collaborative Filtering 算法生成候选集，定义 N_u 表示用户 u 之前喜欢的主题集合，则用户 u 对主题 i 的偏好度根据式 3.1可得，

$$p(u,i) = \sum_{j \in N(u)} r(u,j)s(i,j) \tag{3.1}$$

其中，$r_{u,j}$ 表示用户 u 对主题 j 的偏好度，$s_{i,j}$ 表示主题 i 和主题 j 之间的相似度。Item based Collaborative Filtering 算法定义俩个主题之间的相似度由集中在这个俩个主题的用户行为数据计算得出。N_i 为看过主题 i 的用户集合，N_j 为为看过主题 j 的用户集合，因此，主题 i 和主题 j 的相似度计算公式为式 3.2

$$s(i,j) = \frac{|N(i) \cap N(j)|}{\sqrt{|N(i) \parallel N(j)|}} \tag{3.2}$$

根据式 3.2可知，如果有很多用户同时看了主题 i 和主题 j，那么主题 i 和主题 j 之间的相似度就会很高，不幸的是，这也会导致所有热门主题与所有主题的相似度都很高，导致推荐结果包含热门主题包过多。我们的解决思路是对热门主题包降权，同时控制热门主题所占推荐列表的比例。

3.2.5　排序

排序主要是对候选集的生成的标签权重做排序，但会加入一些倾斜因子，如用户活跃度、主题包的热度、经纬度等因子，最终根据标签权重 + 倾斜因子的排序得到推荐结果。

3.3　用户画像与用户兴趣探索

众所周知用户的需求是动态变化着的，不管是随着季节周期性变动，还是随着年龄发生非逆转变化，都意味着一些标签需要删除掉，一些标签需要加进来。如图 3.1所示，用户画像的数据来源包括：原始的用户行为数据和用户兴趣探索模块，前者只是更新那些显而易见的标签，而后者负责在海量数据中挖掘出那些稍纵即逝的用户行为并准确分析用户的意图，由此可见用户兴趣探索对活跃用户效果相对较好，并且针对小众主题包进行挖掘的效果很棒，可以明显提升推荐结果的多样性。除此之外，我们利用基于时间窗口的遗忘机制解决了将新发现的用户兴趣和原有兴趣合并为用户的新兴趣的问题，时间窗口机制与自然遗忘规律相似，排前面的标签时效性最好，排后面的标签时效性差，将会被优先淘汰。通过设置时间窗口的大小、时间窗口的滑动速率，可以间接控制新、旧兴趣的比例。

3.4　用户画像与推荐系统

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，这就是我们要引进用户画像的根本原因。用户画像模块和兴趣探索模块的关系如图 3.2所示。

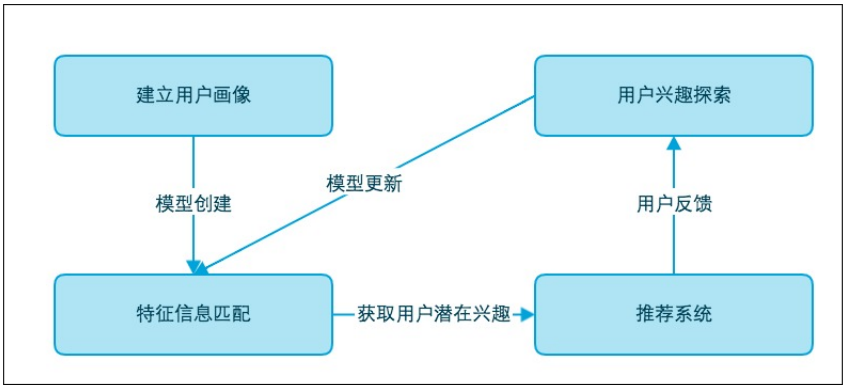

图 **3.2**　用户画像数据流图

一个好的用户画像需要有一个完整的标签体系，包括标签的数量、质量和粒度。其中标签的数量直接影响推荐系统的结果完整度，标签的质量直接影响推荐系统的精度，而标签的粒度会影响推荐系统的用户满意度。完善的标签体系更像一个金字塔，一级是最基本的概念标签，如动漫、运动等，数量被控制在几十个左右，二级标签是上层标签的扩展，如美少女动漫、搞笑动漫、篮球运动、足球运动，三级标签就是指具体化了的标签，如街头篮球、NBA 篮球、CBA 篮球等。对于活跃度高的用户标签倾向于下沉，推荐结果数据多、精度高，活跃度低的用户反之。对于一个新创建的用户画像，刚开始只包含最基本的用户人口信息，随着用户行为数据的累积会逐渐丰富起来。

3.5　本章小结

本章首先介绍了整个推荐系统结构，然后介绍了数据集的生成、格式，以及基于数据集的用户画像建模和商品标签建模，最后介绍了用户画像和用户兴趣探索，用户画像和推荐系统中的交互关系。

第四章　用户画像模块

4.1　引言

小米手机主题用户画像是根据用户社会属性、生活习惯和消费行为等信息而抽象出的一个标签化的用户模型。构建用户画像的核心工作包括：1、给用户贴标签，而标签是通过对用户信息分析而来的高度精炼的特征标识。2、对每个用户标签赋予一定权重以代表该用户对该标签的偏好度。图 4.1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。小米手机主题用户画像的标签是结构化的，最下层是用户基础信息，包括姓名、年龄、经纬度、职业等，中层是用户的兴趣标签，如正太控、动漫控、运动达人等，最上层是抽象标签，如高、中、低忠诚度用户，高、中、低价值用户等，所以对于用户画像的建模过程，就是不断的丰富标签和抽象标签的过程。具体工作包括：从各个维度上对人群细分，完善画像体系，同时针对每类细分人群，丰富群体特征。

4.2　用户画像数据类型

在个性化服务的用户画像建模中，一个完整、成熟的用户画像是包含基础静态数据类型、基础行为数据类型和高维数据类型。

4.2.1　基础静态数据类型

当一个新用户注册时会填写人口基本信息，通过 json 格式从客户端传回服务器，格式Listing 4.2.2

```
{"registerLog": {
  "userId": "001",
  "gender": "male",
  "profession": "student",
  "phone": "null",
  "borthday": "19860820",
  "isWeiboUser": "no",
  "isWeixinUser": "yes",
  "city": "北京市",
  "timestamp": "1453700393",
  "...": "..."
}}
```

有的用户会利用微信、微博提供的第三方免登陆 API，第三方数据可以用来交叉验证用户填写的基础信息数据。用户每次登陆时应用程序还会获得其手机品牌、操作系统等信息。因此，通过解析 server log 得到基础静态数据形式:

图 4.1　用户画像标签示例图

表 4.1　用户-基础静态数据矩阵表

用户 id	性别	年龄	职业	电话号码	手机运营商	是否为微博用户	...
001	女	23	学生	13948572214	移动	是	...
002	男	30	学生	15811036703	移动	是	...
...	...	...	...	...	...	...	...

4.2.2　基础行为数据类型

基础行为数据是指用户的一些行为，包括购买，试用，浏览，评价等的统计量，用户行为数据格式如Listing 4.2.2

```
{"actionLog": {
  "userId": "001"
  "actions": [{
     {"itermId": "0822"},
     {"actionType": "jumpIn"},
     {"stayTime": "32000"},
     {"clickNum": "2"},
     {"scrollNum": "5"},
     {"timestamp": "1453701393"},
     {"...": "..."}
  }]
}}
```

基础行为数据作为用户行为统计量可以反映用户的活跃度、消费能力和用户类型。基础行为数据形式如：

4.2.3　高维数据类型

高维数据即用户的抽象标签，是用户画像模型从基础静态数据和基础行为数据统计、分析、抽象出来，用来衡量用户某一方面的价值，如用户信用是指是

表 4.2　用户-基础行为数据表

用户 id	购买	试用数	浏览	未支付订单数	活跃时间段	日浏览时长	...
001	2	7	118	0	20:00-22:00	120	...
002	0	3	7	1	13:00-14:00	60	...
...	...	...	...	...	...	...	...

否有过作弊行为、退款次数过多等综合评估，用户价值是指购买次数、单笔消费额、消费频率的综合评估。高维数据可以用矩阵来表示：

表 4.3　用户-高维数据表

用户 id	信用	价值	忠诚度	活跃度	价格敏感度	奖励敏感度	...
001	高	高	高	高	低	低	...
002	中	中	高	高	高	高	...
...	...	...	...	...	...	...	...

4.3　用户画像建模

用户画像建模的过程就是原始数据进过处理、分析得到可信度高的用户标签信息的过程，对于不同类型的用户数据其建模的侧重功能点也有所区别。

4.3.1　基础静态数据建模

用户基础静态数据的特点是数量不多，但在推荐系统中所占的权重较大，因此对其可信度要求较高，在对基础静态数据建模的时候主要实现俩个功能：根据上下文信息补全为为空的标签和根据上下文信息校验已有的标签。

标签补全以用户性别标签为例，新用户注册时如未填写性别信息其值会默认设为 Null，方便用户画像建模时判断。主要思路是通过分析用户上下文信息，包括第三方登入数据、用户语音和头像获得用户真实的性别，如以上方法都未成功获取用户性别，程序会利用线性回归算法挖掘出一个最有可能的性别标签值，代码：

```
public String getUserGender(String log) {
    Gson gson = new Gson();
    UserProfile userProfile = gson.fromJson(log,
        UserProfile.class);

    if (userProfile.gender != null) {
        return userProfile.gender;
    }

    String useId = userProfile.useId;
    //通过第三方应用登陆数据得到用户信息
    UserProfile thirdPartUP =
        gson.fromJson(getThirdPartUserInfo(useId),
        UserProfile.class);
```

```
    if (thirdPartUP.gender != null) {
        return thirdPartUP.gender;
    }

    //通过分析用户语音数据得到用户信息
    UserProfile voiceUP =
        gson.fromJson(getUserVoiceUserInfo(useId),
        UserProfile.class);
    if (voiceUP.gender != null) {
        return voiceUP.gender;
    }

    //通过线性回归算法挖掘出用户信息
    UserProfile lrUP =
        gson.fromJson(getLinearRegressionUserInfo(useId),
        UserProfile.class);
    return lrUP.gender;

}
```

标签校验是指虽然相关信息已经被填写，但程序认为其值具有随意性，需要根据上下文信息加以确认并校验，标签校验由于考虑的因素较多导致计算量大，使得其应用场景较少，还是以用户性别标签为例，代码如Listing 4.3.1

```
public String getRightUserGender(String log) {
int[] count = {0, 0};
Gson gson = new Gson();
UserProfile userProfile = gson.fromJson(log,
    UserProfile.class);

if (userProfile.gender != null) {
    if (userProfile.gender.equals("male")) {
        count[0]++;
    } else {
        count[1]++;
    }
}

String useId = userProfile.useId;
UserProfile thirdPartUP =
    gson.fromJson(getThirdPartUserInfo(useId),
    UserProfile.class);
if (thirdPartUP.gender != null) {
    if (thirdPartUP.gender.equals("male")) {
        count[0]++;
    } else {
        count[1]++;
    }
}

UserProfile voiceUP =
    gson.fromJson(getUserVoiceUserInfo(useId),
    UserProfile.class);
if (voiceUP.gender != null) {
```

```
        if (voiceUP.gender.equals("male")) {
            count[0]++;
        } else {
            count[1]++;
        }
    }

    UserProfile lrUP =
        gson.fromJson(getLinearRegressionUserInfo(useId),
        UserProfile.class);
    if (lrUP.gender.equals("male")) {
        count[0]++;
    } else {
        count[1]++;
    }
    if (count[0] >= count[1]) {
        return "male";
    } else {
        return "female";
    }
}
```

4.3.2　基础行为数据建模

基础行为数据建模更新频率较快，计算量较大，因此采用离线方式利用 sql 语句从 hive 表中得出用户在一段时间区间内特定行为的统计数据。需要注意一些用户行为的延迟性，如购买行为，从下单到支付成功可能跨越若干天，因此约定订单量以支付时间为准，有时候遇到网络故障相同订单会被用户提交多次，需要利用 distinct 做去重操作。统计特定用户某段时间的订单量的 sql 语句如Listing 4.3.2

```
set hiveconf:ymdwithline=2016-04-06;
set hiveconf:userId=525108009;

select count(distinct a.order_id) score
from theme_dw.dw_v_order_base
where concat_ws('-',year,month,day) between
    date_sub('${hiveconf:ymdwithline}',5) and
    '${hiveconf:ymdwithline}'
and userId='${hiveconf:userId}'
and finish_time like '${hiveconf:ymdwithline}%'
```

4.3.3　高维数据建模

高维数据建模的数据来源包括基础静态数据、基础行为数据，数据类型包括累计量和趋势量，累计量包括用户浏览总数、用户购买总数等，趋势量是指用户最近登录时间、最近购买时间等，利用数据挖掘分类算法得出一个训练模型，需要注意的是用户行为类型、发生时间、发生位置会影响模型的权重计算，即

weight = (行为类型 + 时间上下文 + 空间上下文) × 时间衰减因子。其中，用户行为类型包括浏览、购买、搜索、评论、购买、点击赞、收藏等，我们定义购买权重计为 5，而浏览仅仅为 1。空间上下文是指用户跳转入口方式，我们定义搜索入口权重 3，排行榜入口为 2。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。时间衰减因子根据半衰期公式得出，如所示式 4.1，其中 T 取值为 1，t 为行为发生时间距离当前时间的天数。

$$score = (\frac{1}{2})^{(t/T)} \tag{4.1}$$

以用户活跃度为例，由于日活跃变动过大，月活跃过于滞后，因此按周统计，模型选择线性回归算法，模型输入为基础静态数据、基础行为数据，模型输出为一个 int 型整数，值为 [1，2，3]，分别对应不活跃、较活跃、活跃。代码，代码如Listing 4.3.3

```
public int getActivityScore(String userId) throws Exception {
    String userBaseInfo = getUserBaseInfo(userId);
    String userActionLog = getUserActionLog(userId);
    Gson gson = new Gson();
    String score = getLinearRegressionActivityScore(
      gson.fromJson(userBaseInfo, UserProfile.class),
      gson.fromJson(userActionLog, UserActions.class));
    double activityScore = Double.parseDouble(score);
    if (activityScore >= 66) {
        return 3;
    } else if (activityScore >= 33) {
        return 2;
    } else {
        return 1;
    }
}
```

4.4　实验与分析

本节的研究目标是如何利用用户画像给新注册用户做出准确的 Top-N 推荐并提升用户留存率。严谨的 AB 测试流程应该先分析 AA 测试，得出实验本身自带的误差，然后利用这个误差因子修正 AB 测试结果，最终得到统计结果，但 AA 测试会严重拖慢节奏，所以本节只重点介绍 AB 测试。实验最终用户人群为北京地区，所有从 2015 年 9 月 1 号到 2015 年 9 月 7 号这段时间注册的用户，去除用户注册信息不完整后用户数为 20 万，对照组和测试组 a 和测试组 b 人群比例为 33.3：33.3：33.4，对照组用户人群的推荐结果没有利用用户画像，测试组 a 人群的推荐结果只包含热门主题，测试组 b 人群的推荐结果利用了用户画像，对照组和测试组 b 的推荐候选集为全部主题，测试组 a 的推荐候选集为 Top 20% 热度的主题。。实验从 2015 年 9 月 8 号开始到 2015 年 10 月 8 号结束，周期为一个月。用户画像建模如Listing 4.4。

```
{
//静态数据
user_id             int   comment '用户id',
user_name           int   comment '用户name',
user_age            int   comment '用户age',
create_time         string comment '账号创建时间',
city_id             int   comment '城市id',
city_name           string comment '城市名',
phone               int    comment '手机号',
os_version          stringt comment '操作系统及版本',
phonetype_serial    string comment '手机品牌及型号',
education_level     string comment '学历',
school              string comment '学校',

//行为数据
click_num int comment '点击次数',
last_click_time int comment '最近点击时间',
buy_num int comment '购买次数',
last_buy_time int comment '最近购买时间',
try_use int comment '试用次数',
last_tryuse_time int comment '最近试用时间',
browse_num int comment '浏览次数',
last_browse_time int comment '最近浏览时间',
browse_total_time int comment '浏览总时长',
login_num int comment '登陆总次数',
login_total_time int comment '登陆总时长',
comment_num int comment '评论总次数',

//高维数据
use_time            int    comment '使用时间段',
not_use_time        int    comment '沉默天数',
friendship          list<bigint> comment '好友关系',
friend_group        list<bigint> comment '好友圈',
coupon_sensitivity_score decimal(20,4) comment
   '券敏感及阈值',
purchase_will_score decimal(20,4) comment '消费意愿',
loyal_score         decimal(20,4) comment '忠诚度',
credit_score        decimal(20,4) comment '活跃度'
}
```

4.4.1　评测指标

本节使用线上 A/B 测试方案 [51]，利用用户留存率来评测推荐系统应对冷启动问题的效果。用户留存数是指在某段时间开始使用 App 应用，经过一段单位时间后仍然继续使用该 App 应用的用户，用户留存率是指用户留存数占当时新增用户的比例，计算单位取天，用户留存率研究对象为新注册用户，反映了推

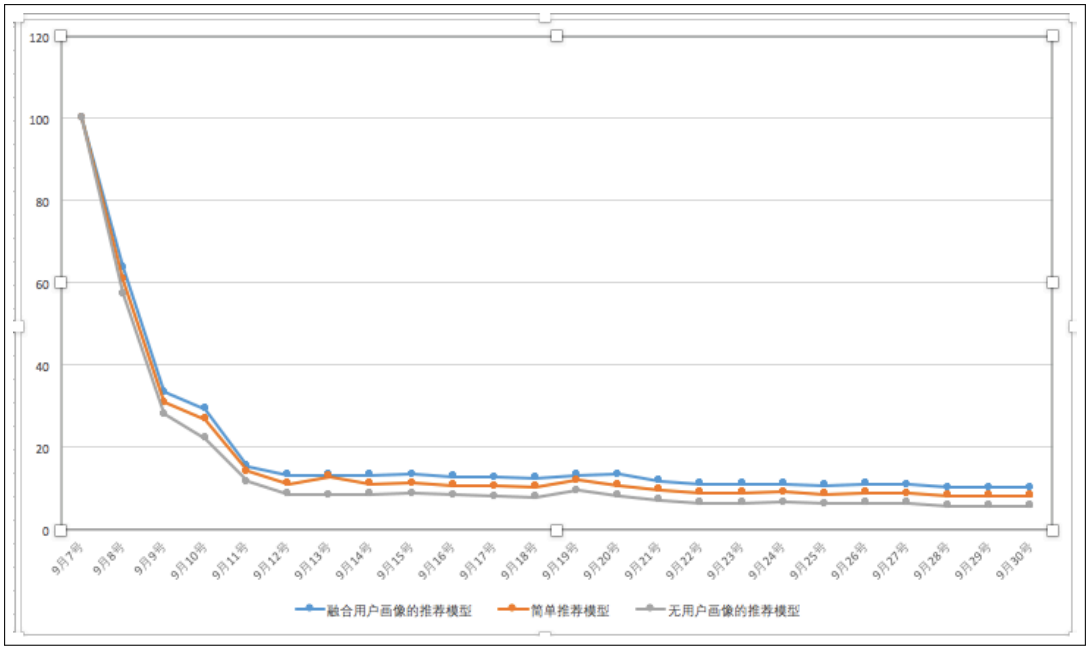

图 **4.2**　新用户留存率实验对比图

荐系统的转换能力，即由初期的不稳定的用户转化为活跃、稳定、忠诚的用户。

4.4.2　对比模型

图 4.2展示了不同模型的实验结果。推荐算法使用了开源软件 spark MLlib 的 LogisticRegressionWithLBFGS 模块，我们对比了单纯的推荐模型、推荐热门商品的简单推荐模型和融合了用户画像的推荐模型在新注册用户数据集上的用户留存率。图中，横坐标是时间变量，单位为天，纵坐标是用户留存率，每一条曲线代表了一个模型的用户留存率随时间变化的曲线。通过观察曲线可以发现用户留存率随时间流动呈指数分布，头三天就流失了约 90% 的新用户，从第四天用户留存率开始停留在一个比较稳定的阈值，为减少误差头三天的数据不记入统计数据，统计结果显示，融合了用户画像的推荐模型的留存率是 10.3%，比推荐热门商品的简单推荐模型的留存率 8.19% 高了 2.11 个百分点，相对于单纯的推荐模型的留存率 5.76% 高了 4.54 个百分点。由此可见用户画像能够很好的解决冷启动问题并得到较高的新注册用户留存率。

4.5　本章小结

用户画像对于推荐系统来讲，主要几个方面的提升：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。但是用户画像只是反映了用户长期的兴趣，所以无法动态的反映用户短期兴趣，因此我们引入了用户兴趣探索模块，将在下一章节件详细介绍。

第五章　用户兴趣探索

5.1　引言

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数据量化后得出的。但就小米主题市场而言，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的数据来源包括用户画像和商品特征表。用户画像包括用户基本信息和兴趣标签等，商品特征表包括分类、属性标签等，用户兴趣探索过程分为几个步骤：首先，利用用户历史行为 (评论，停留时长，评分，点赞，购买等) 量化用户满意度，然后利用用户兴趣特征向量与商品特征矩阵得出相关分数，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。这里涉及到的概念包括用户满意度的量化、用户和商品的关联度、商品属性标签的长尾性。

5.2　用户行为数据的存储和处理

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，每天最多也只产生上百条行为记录；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 HDFS 分布式文件集群存储数据。

HDFS 为海量的数据提供了存储，而 Hive 支撑了海量的数据统计。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。从体系架构到数据定义到数据存储再到数据处理，HDFS 分布式文件集群和 Hive 为海量用户行为的分析和用户兴趣探索提供了可能。

5.2.1　数据预处理

数据预处理是数据挖掘过程中一个重要步骤，主要工作包括字段去重、无效日志过滤、多表字段的连接等。如统计 2015 年 09 月 06 号 userId 为 001 的投诉数，数据预处理过程：

```
set hiveconf:ymdwithline=2015-09-06;
set hiveconf:metric=complaint_order_num;
set hiveconf:user_id=001;

select '${hiveconf:metric}' as metric, count(a.order_id) as
   score
from (
   //去重
   select distinct order_id
   from theme.dw_v_order_base
   //以时间范围date_sub('${hiveconf:ymdwithline}',5) and
      '${hiveconf:ymdwithline}'为条件过滤掉不符合条件的订单
   where concat_ws('-',year,month,day) between
      date_sub('${hiveconf:ymdwithline}',5) and
      '${hiveconf:ymdwithline}'
   //无效订单过滤
   and order_id!=null
   //以用户id为条件过滤掉其他订单
   and user_id=${hiveconf:user_id}
) a
inner join (
   //order_id 字段去重
   select distinct order_id
   from theme.g_comment_complaint
   //type = 3表示用户投诉
   where concat_ws('-',year,month,day) =
      '${hiveconf:ymdwithline}' and type = 3
   //多表字段的连接，如果有一个表有投诉记录，就算一次投诉。
   union
   select distinct order_id
   from theme.dwd_kefu_phone_complaint
   where concat_ws('-',year,month,day) =
      '${hiveconf:ymdwithline}'
) b
on a.order_id = b.order_id
inner join (
   select order_id
   from theme.pay_info
   where ymd = ${hiveconf:ymdwithline}
      //status=1代表当前订单状态为已支付
      and status=1
) d
on a.order_id = d.order_id
group by metric;
```

5.3　用户兴趣探索模型

用户兴趣探索主要功能模块包括：1，兴趣标签探测，在分析用户行为数据时，如果某些主题标签是这个用户画像没有的，那么这些标签会作为标签探索候选集。2，长尾标签提取，遍历标签探索候选集，如果不属于小众标签集的标签将会被过滤掉。3，用户满意度量化，根据用户所有对某一个主题的行为数据得出这个用户对这个主题的满意度。4，标签权重的更新，不管是不是一次成功的兴趣标签探索，都要对用户画像标签的权重做更新，更新算法利用了线性衰减思想。本章首先介绍一些基本概念，包括长尾标签的定义、用户满意度的量化等。然后详细介绍用户兴趣探索功能模块的实现。

5.3.1　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。包括浏览，点击，下载，试用，购买，评论。本文所指的用户行为都是指用户在某手机主题上的行为。

用户兴趣。用户兴趣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」，「NBA，0.1」，「性感，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。用户兴趣是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。

小众标签集。小众标签集是指出现频率低的主题标签的集合，代码：

```
public HashSet<String> getLongTailTags() throws Exception {
    Map<String, String> tagsCount = new TreeMap<>();

    //获取所有主题包
    Map<String, Object> allThemes = getAllThemes();
    for (Map.Entry<String, Object> theme :
        allThemes.entrySet()) {
        String themeName = theme.getKey();
        //获取当前主题的所有标签
        Object themeTags = ((Map<String, Object>)
            theme.getValue()).get("tags");
```

```
            for (String tag : (Set<String>) themeTags) {
                //出现一次，tag 对应的count加1
                tagsCount.put(tag, tagsCount.get(tag) + 1);
            }
        }

        //这里将map.entrySet()转换成list
        List<Map.Entry<String, String>> list = new
            ArrayList<Map.Entry<String, String>>(tagsCount
                .entrySet());
        //然后通过比较器来实现排序
        Collections.sort(list, new Comparator<Map.Entry<String,
            String>>() {
            //升序排序
            public int compare(Map.Entry<String, String> o1,
                Map.Entry<String, String> o2) {
                return o1.getValue().compareTo(o2.getValue());
            }
        });

        HashSet<String> out = new HashSet<>();
        //取频率最小的那80%标签作为小众标签
        double threshold = list.size() * 0.8;
        for (int i = 0; i <= threshold; i++) {
            out.add(list.get(i).getKey());
        }

        return out;
    }
```

用户满意度量化。用户满意度量化是指根据用户作用在主题上的不同行为动作及其参数值，参数值包括动作类型、次数和时长，得到一个衡量用户满意度的分数。

标签集中度 (tagFocus)。标签集中度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=allThemeNum，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，标签集中度公式如式 5.1，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，因此它应该权重较小。

$$tagFocus = \log \frac{|tagInThemeNum|}{|allThemeNum|} \tag{5.1}$$

标签热度 (tagPopular)。标签热度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。热度公式如式 5.2。

$$tagPopular = \log \frac{|peopleLikeTagNum|}{|allPeople|} \tag{5.2}$$

5.3.2　兴趣标签探测功能模块

首先候选标签是用户画像中没有的标签，如用户 001 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，那么程序可以检测汽车标签对于用户 001 是从未遇到过的标签，于是汽车标签将会是潜在的探索标签。事实上用户兴趣探索过程可以在很短的时间内完成，基于 hive + HDFS 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。标签探索算法：

```
public Set<String> tagExplore(String userId, String itemId)
    throws Exception {
    Gson gson = new Gson();
    //获取当前用户对当前主题的所有行为，只计算前一天的行为
    List<UserActions> actions =
        getActionsByUserIdAndItemId(userId, itemId);
    //获取用户详细信息
    String userInfo = getUserBaseInfo(userId);
    UserProfile userProfile = gson.fromJson(userInfo,
        UserProfile.class);
    Map<String, Double> userTags = userProfile.tags;

    Set<String> out = new HashSet<>();
    for (UserActions action : actions) {
        //获取主题详细信息
        Map<String, Object> itemBaseInfo =
            getItemBaseInfo(action.itemId);
        Set<String> tags = (Set<String>)
            itemBaseInfo.get("tags");
        for (String tag : tags) {
            if (!userTags.containsKey(tag)) {
                out.add(tag);
            }
        }
    }
    return out;
}
```

5.3.3　长尾标签抽取功能模块

长尾标签是指这个标签的集中度和热度之比大于一个阈值，且在小众标签集中。长尾标签提取算法。

```
public Set<String> getEffectTags(String userId, String
    itemId) throws Exception {
    Set<String> out = new HashSet<>();
    //获取所有长尾标签
    HashSet<String> longTailTags = getLongTailTags();
    //获取所有当前用户画像没有的标签
    Set<String> rawTags = tagExplore(userId, itemId);
    for (String tag : rawTags) {
        if (!longTailTags.contains(tag)) {
            continue;
        }

        //获取标签的集中度
        long tagFocusScore = getTagFocusScore(tag);
        //获取标签的热度
        long tagPopularScore = getTagPopularScore(tag);
        if (tagFocusScore / tagPopularScore <= threshold) {
            continue;
        } else {
            out.add(tag);
        }
    }

    return out;
}
```

5.3.4　用户满意度量化功能模块

从对用户的行为数据分析量化用户满意度，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 5.1列举的用户行为为实际使用的行为类型，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，通过数据挖掘算法过滤掉行为数据中的噪音，这样使分析更加精确。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，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，就需要进行归一化处理使得数据取值在 [0，10] 范围中，代码：

```
public Map<String, String> getUseSatisfyScore(String userId,
    String itemId) {
```

```
    //获取当前用户对当前主题包的所有行为
    List<UserActions> actions =
       getActionsByUserIdAndItemId(userId, itemId);
    double score = 0.0;
    int clickNum = 0;
    int scrollNum = 0;
    for (UserActions action : actions) {
       if (action.actionType.equals("buy") ||
          action.actionType.equals("tryUse") || action
             .actionType.equals("favor")) {
          return new HashMap<String, String>() {{
             put("score", "1");
             put("msg", "very like");
          }};
       } else if (action.actionType.equals("down")) {
          return new HashMap<String, String>() {{
             put("score", "0");
             put("msg", "not like at all");
          }};
       }

       if (action.actionType.equals("click")) {
          clickNum++;
          if (clickNum <= 5) {
             score += 0.2;
          }
       } else if (action.actionType.equals("scroll")) {
          scrollNum++;
          //滑动屏幕一次且停留时长超过3秒,说明用户对内容感兴趣
          if (scrollNum <= 5 && action.duration * 1000 > 3000)
             score += 0.5;
       } else if (action.actionType.equals("share")) {
          score += 1.5;
       } else if (action.actionType.equals("comment")) {
          score += 1.0;
       } else if (action.actionType.equals("star")) {
          //用户评分,值为1到5星
          if(action.starLevel>=4)
            score += action.starScore;
       }
    }

    //正则化
    score = (score-MIN)/(MAX-MIN)
    HashMap<String, String> ret = new HashMap<>();
    ret.put("score", String.valueOf(score));
    ret.put("msg", "user intereting in this item");
    return ret;
}
```

表 **5.1**　用户行为权重对应表

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	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

5.4　用户画像和用户兴趣探索的融合

随着时间的变化，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，通过调节时间窗口大小和更新周期，体现不同的时效性。我们可以把用户画像权重想象成一个自然冷却的过程：

- 任一时刻，用户画像中的标签都有一个当前温度，温度最高的标签权重值最高。
- 如果该用户对某主题发生了一些正向标签，如点赞，该文章包含的标签在用户画像中的温度就会上升，否则温度下降。
- 随着时间流逝，所有标签的温度都逐渐冷却，通过时间窗口向前滑动实现。

这样假设的意义在于我们可以照搬物理学的牛顿冷却定律 (Newton's Law of Cooling)，建立标签权重与时间之间的函数关系：本期分数 = 上期分数 - 冷却系数 * 间隔天数，构建一个线性衰减的过程。其中，冷却系数决定了标签融合的更新率，如果想放慢更新率，冷却系数就取一个较小的值，否则就取一个较大的值。

标签权重的线性衰减算法结合了手机主题用户长期兴趣和短期兴趣，根据时间因素权重自动进行衰减，能准确反映用户兴趣的变化趋势。该模型是指用户对兴趣标签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，该标签将被用户画像所淘汰。

```
    public void tagLinearecay(String userId) throws Exception {
      //获取当前用户当前所有有过行为的主题包
      Set<String> items = getAllItems(userId);
      //获取当前用户的画像
      UserProfile userProfile = getUserProfile(userId);
      for (String item : items) {
         //获取当前用户对当前标签的满意度值
         Map<String, String> useSatisfyScore =
            getUseSatisfyScore(userId, item);
         //threshold为逻辑回归算法训练出的阈值
         if (Double.parseDouble(useSatisfyScore.get("score")) >
            threshold) {
            //获取所有成功探索的标签
            Set<String> effectTags = getEffectTags(userId, item);
            for (String effectTag : effectTags) {
               userProfile.tags.put(effectTag, 5);
            }
         }
      }
      //得到用户行为中所有的主题标签
      Set<String> allActionTags = getAllActionTags(userId);
      for (Map.Entry<String, Double> userTag :
         userProfile.tags.entrySet()) {
         String tag = userTag.getKey();
         double score = userTag.getValue();
         if (!allActionTags.contains(tag)) {
            //将标签偏好值减少 0.5，进行衰减。
            score -=0.5;
            if(score<=0){
               //如果当前标签权重降低0以下，则移除该标签
               userProfile.tags.remove(tag);
            }else {
               userProfile.tags.put(tag, score);
            }
         }else {
            //do nothing
         }
      }
   }
```

5.5 实验与分析

5.5.1 数据集准备

实验中我们利用 2013 年 9 月到 2013 年 10 月的用户行为数据和所有关联的手机主题包。这个数据集包含了 110739 个用户在这段时间对主题包的标签行为，数据集中包含了 8936 个主题包。该数据集每行是一条记录，每条记录由四个部分组成：用户 ID，行为类型，行为属性值，主题 ID，日期，每一条记录代表了某个用户在某个时间点对某个主题包进行了某种行为。保证数据集具有一定的稠密程度，我们去除了用户行为记录少于 10 条的所有用户，最终用户集包含 10646 个用户，2033600 条用户行为记录，可见数据集的稀疏度还是在 97.86% 以上。

5.5.2 评测指标

使用线上 A/B 测试方案，利用点击购买转化率来评测推荐系统应对马太效应的效果。根据统计我们知道 20% 的热门商品在占了 80% 的曝光机会的同时却只占 50% 的销售量，这时因为虽然热门商品销量很好但其整体数量偏少，很难满足大多数消费者的需求。相反，占据 80% 的小众商品虽然曝光率低，但凭借其庞大数量和多样性，可以满足不同消费者的需求。因此如果适度对小众商品增加曝光机就会可以提升所有商品的销售量，即提升手机主题包的点击购买转换率。

5.5.3 对比模型

无兴趣探索模块的推荐模型，在实验中作为基准模型。对照模型包括融合了兴趣探索模块的推荐模型和推荐热门商品的简单推荐模型。

5.5.4 实验结果

我们对比了无兴趣探索模块的推荐模型、推荐热门商品的简单推荐模型和融合了兴趣探索模块的推荐模型在 2015 年 9 月到 2015 年 10 月的有过至少一次销售记录的商品数 itemCount。图 5.1展示了不同模型的实验结果。图中，横坐标是时间变量，单位为天，纵坐标是 itemCount，每一条曲线代表了一个模型的 itemCount 随时间变化的曲线。通过观察曲线可知，融合了兴趣探索模块的推荐模型的 itemCount 月平均数是 3136，推荐热门商品的简单推荐模型的 itemCount 月平均数是 1935，无兴趣探索模块的推荐模型的 itemCount 月平均数是 2679。实验说明融合了用户兴趣探索的推荐模型相对其他模型有更好的多样性。

我们对比了无兴趣探索模块的推荐模型、推荐热门商品的简单推荐模型和融合了兴趣探索模块的推荐模型在 2015 年 9 月到 2015 年 10 月的点击购买转化率。图 5.2展示了不同模型的实验结果。图中，横坐标是时间变量，单位为天，纵坐标是点击购买转化率，每一条曲线代表了一个模型的点击购买转化率随时间变化的曲线。实验结果显示，融合了兴趣探索模块的推荐模型相对其他模型

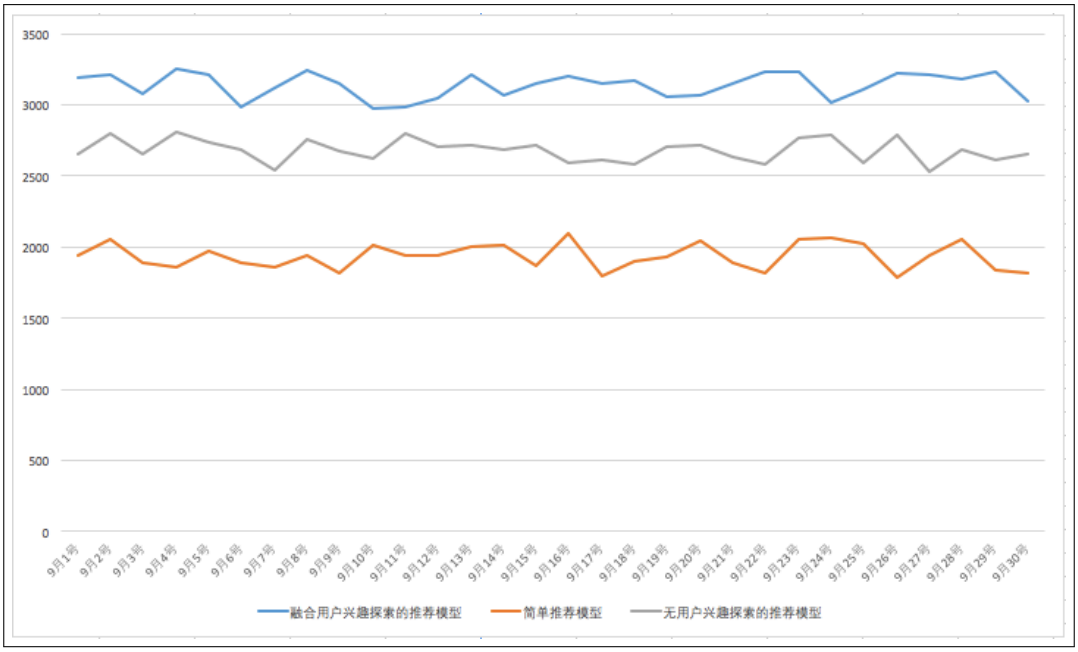

图 **5.1**　推荐多样性实验对比图

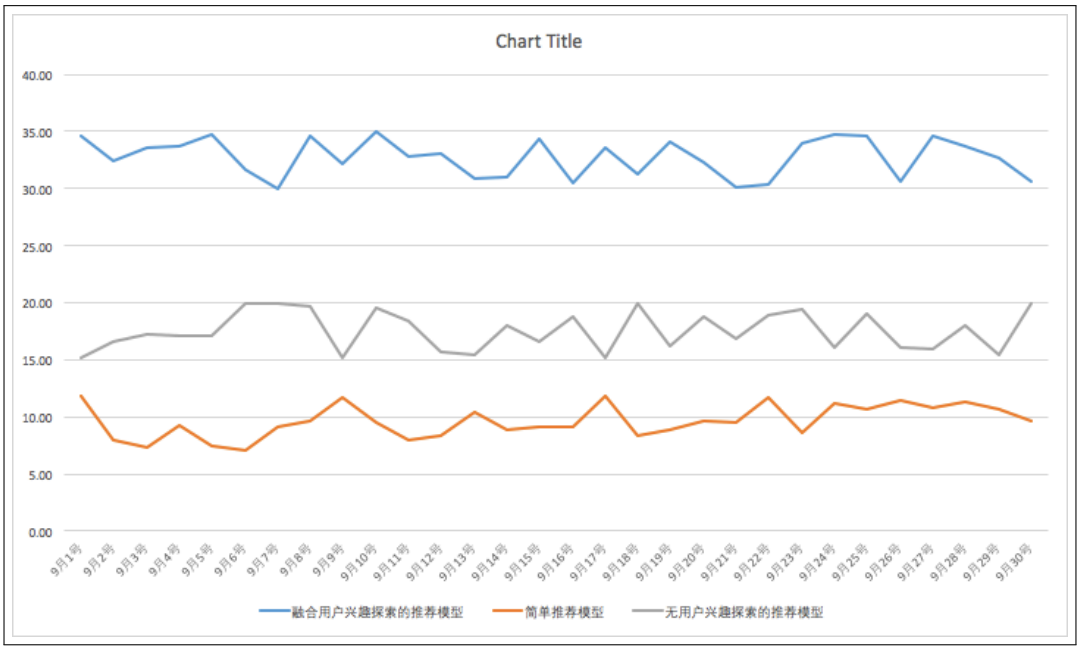

图 **5.2**　转化率实验对比图

有更高的点击购买转化率。融合了兴趣探索模块的推荐模型的平均点击购买转化率是 32.74%，比推荐热门商品的简单推荐模型的平均点击购买转化率 9.63%高了 23.11 个百分点，相对于无兴趣探索模块的推荐模型的平均点击购买转化率 17.54% 高了 15.2 个百分点。由此可见用户兴趣探索能够很好的提升点击购买转化率。

5.6 本章小结

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模块的组成内容，包括兴趣标签探测功能模块、长尾标签抽取功能模块、用户满意度量化功能模块，然后介绍了用户画像和用户兴趣探索的融合，最后给出了用户兴趣探索实验结果。

第六章　结束语

如果说过去的十年是搜索技术大行其道的十年，那么个性化推荐技术将成为未来十年中最重要的革新之一。目前几乎所有大型的电子商务系统，如 Amazon、阿里、小米、滴滴等，都不同程度地使用了各种形式的推荐系统。一个好的推荐系统需要满足的目标有：个性化推荐系统必须能够基于用户之前的口味和喜好提供相关的精确的推荐，而且这种口味和喜欢的收集必须尽量少的需要用户的劳动。推荐的结果必须能够实时计算，这样才能够在用户离开网站前之前获得推荐的内容，并且及时的对推荐结果作出反馈。实时性也是推荐系统与通常的数据挖掘技术显著不同的一个特点。一个完整的推荐系统由三部分构成：用户画像模块，用户行为挖掘模块、推荐引擎模块。用户画像模块记录了用户长期的信息，刻画用户的基础类型。用户行为挖掘模块负责记录能够体现用户喜好的行为，比如购买、下载、评分等。这部分看起来简单，其实需要非常仔细的设计。比如说购买和评分这两种行为表达潜在的喜好程度就不尽相同完善的行为记录需要能够综合多种不同的用户行为，处理不同行为的累加。推荐引擎模块的功能则实现了对用户行为记录的分析，采用不同算法建立起模型描述用户的喜好信息，通过推荐引擎模块实时的从内容集筛选出目标用户可能会感兴趣的内容推荐给用户。因此，除了推荐系统本身，为了实现推荐，还需要一个可供推荐的内容集。在经典的协同过滤算法下，内容集甚至只需要提供 ID 就足够，而对于手机主题推荐系统来说，由于需要对内容进行特征抽取和索引，我们就会需要提供更多的领域知识和标签属性。

推荐系统是一种联系用户和内容的信息服务系统，一方面它能够帮助用户发现他们潜在感兴趣的内容，另一方面它能够帮助内容供者将内容投放给对它感兴趣的用户。推荐系统的主要方法是通过分析用户的历史行为来预测他们未来的行为。因此，时间是影响用户行为的重要因素。关于推荐系统动态特性的研究相对比较少，特别是缺乏系统性的研究。对动态推荐系统的研究，无论是从促进用户兴趣模型的理论角度出发，还是从实际需求来看，都具有重要的意义，本文的研究工作正是在这一背景下展开。

6.1　研究工作总结

本文对推荐系统特别是与用户画像相关的动态推荐系统的相关工作做了总结和回顾之外，主要的工作包括以下几个方面：

- 设计了用户画像模型：按照用户属性和行为特征对全部用户进行聚类和精细化的客户群细分，将用户行为相同或相似的用户归类到一个消费群体，这样就可以将推荐平台所有的用户划分为 N 个不同组，每个组用户拥有相同或相似的行为特征，这样电商平台就可以按照不同组的用户行为对其进

行个性化智能推荐。在现有用户画像、用户属性打标签、客户和营销规则配置推送、同类型用户特性归集分库模型基础上，未来将逐步扩展机器深度学习功能，通过系统自动搜集分析前端用户实时变化数据，依据建设的机器深度学习函数模型，自动计算匹配用户需求的函数参数和对应规则，推荐系统根据计算出的规则模型，实时自动推送高度匹配的营销活动和内容信息。

- 设计了用户兴趣探索模型：模型能够实时根据用户行为变化的趋势, 实时的调整推荐结果排名, 从而不断改善用户在推荐系统中的体验。

- 利用线性衰减算法成功融合用户长期兴趣和短期兴趣：本文在研究用户画像建模和用户兴趣探索的基础上，结合电子商务参与者兴趣偏好变化频繁的特点，提出了基于线性衰减的用户兴趣融合模型。该模型采用一个 0 到 10 的数值表示用户偏好，表示用户对每个标签的喜好程度，权重值根据时间进行线性衰减，以反映用户兴趣的变化。

6.2　对未来工作的展望

本文对推荐系统的用户画像和用户兴趣探索模型进行了较深入的研究, 但是针对用户兴趣变化的推荐模型的实现还有很多工作要做。本人认为推荐系统有待解决的问题有:

- 用户行为的离线和在线计算的分配：用户行为每天产生的数据量很大，哪些行为需要在线实时计算反馈，哪些行为只需要离线计算即可，需要根据具体业务的特点和用户习惯赋予每种行为一个权重，然后根据权重排名决定计算方式。因此，用户行为的特征提取、分析将是我们将来工作的一个重要方面。

- 用户兴趣探索模型对推荐系统的影响：本文的所有工作基本集中在高推荐系统的点击购买转换率上。但点击购买转换率并不是推荐系统追求的唯一指标。比如, 预测用户可能会去看, 从而给用户推荐速度与激情, 这并不是一个好的推荐。因为速度与激情的热度很高, 因此并不需要别人给他们推荐。上面这个例子涉及到了推荐系统的长尾度, 即用户希望推荐系统能够给他们新颖的推荐结果, 而不是那些他们已经知道的物品。此外, 推荐系统还有多样性等指标。如何利用时间信息, 在不牺牲转换率的同时, 提高推荐的其他指标, 是笔者将来工作研究的一个重要方面。

- 推荐系统随时间的进化：用户的行为和兴趣是随时间变化的, 意味着推荐系统本身也是一个不断演化的系统。其各项指标, 包括长尾度, 多样性, 点击率都是随着数据的变化而演化。如何让推荐系统能够通过利用实时变化的用户反馈, 向更好的方面发展是推荐系统研究的一个重要方面。

最后, 希望本文的研究工作能够对动态推荐系统的发展作出一定的贡献, 并真诚的希望老师们出宝贵的批评意见和建议。

参考文献

[1] Gediminas Adomavicius and Alexander Tuzhilin. 1999. *User Profiling in Personalization Applications through Rule Discovery and Validation*. ACM, 377-381.

[2] lbrahim Cingil, Asuman Dogac and Ayca Azgin.2000. *A broader approach to personalization*. mmunications of the ACM, 43(8): 136-141.

[3] Joseph Kramer, Sunil Noronha and John Vergo.2000. *A user-centered design approach to personalization*. Communications of the ACM, 43(8)44-48.

[4] Bamshad Mobasher, Honghua Dai, Tao Luo, Yuqing Sun and Jiang Zhu.2000. *Integrating Web Usage and Content Mining for More Effective Personalization*. Electronic Commerce and Web Technologies, 1875: 165-176.

[5] Bamshad Mobasher, Robert Cooley and Jaideep Srivastava.2000. *Automatic personalization based on Web usage mining*. Communications of the ACM, 43(8): 142-151.

[6] P. Chen, H. Xie, S. Maslov, and S. Redner.2007. *Finding Scientific Gems with Google's PageRank Algorithm*. Journal of Informetrics, 1(1):8–15.

[7] C. Basu, H. Hirsh, and W. Cohen.1998. *Recommendation as Classification: Using Social and Content-Based Information in Recommendation*. In Proc. of the 15th National Conference on Artificial Intelligence (AAAI '98), 714–720.

[8] J. Teevan and S. T. Dumais and E. Horvitz.2005. *Personalizing Search via Automated Analysis of Interests and Activities*. . In Proc. of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2005), 449–456,

[9] S. M. McNee, I. Albert, D. Cosley, S. L. P. Gopalkrishnan, A. M.Rashid, J. S. Konstan, and J. Riedl.2002. *Predicting User Interests from Contextual Information*. In Proc. of the 2002 ACM Conference on Computer Supported Cooperative Work (CSCW '02), 116–125,

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

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

[12] Agichtein, E., Brill, E. & Dumais, S.T. (2006). *Improving web search ranking by incorporating user behavior information*. Proc. SIGIR, 19-26.

[13] Bilenko, M. et al. (2008). *Talking the talk vs. walking the walk: salience of information needs in querying vs. browsing*. Proc. ACM SIGIR, 705-706.

[14] Broder, A. (2002). *A taxonomy of Web search*. ACM SIGIR Forum, 36(2), 3-10.

[15] Broder, A. (2002). *A taxonomy of Web search*. ACM SIGIR Forum, 36(2), 3-10.

[16] Budzik, J. & Hammond, K. (1999). *Watson: anticipating and contextualizing information needs*. Proc. ASIS, 727-740.

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

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

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

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

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

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

[23] Liu, Yu; Li, Weijia; Yao, Yuan; Fang, Jing; Ma, Ruixin; Yan, Zhaofa. *An Infrastructure for Personalized Service System Based on Web2.0 and Data Mining*. International Conference on Intelligent Computing and Information Science. JAN 08-09, 2011.

[24] Shumeet Baluja, Rohan Seth, D. Sivakumar, Yushi Jing, Jay Yagnik, Shankar Kumar, Deepak Ravichandran, and Mohamed Aly.2008. *Video suggestion and discovery for youtube: taking random walks through the view graph*. In Proceeding of the 17th international conference on World Wide Web, WWW '08, pages 895–904.

[25] Robert M. Bell and Yehuda Koren. December 2007. *Lessons from the netflix prize challenge*. SIGKDD Explor. Newsl., 9:75–79.

[26] Sia K.C, Zhu S.Chi, Hino Tseng, B.L.2006. *Capturing User Interests by Both Exploitation and Exploration[C]*. Technical report, NEC Labs America.
[27] Thomas Hofmann and Jan Puzicha.1999. *Latent class models for collaborative filtering[J]*. In Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence, IJCAI'99,San Francisco, CA, USA, Morgan Kaufmann Publishers Inc, pages 688–693,
[28] Han Jiawei, Kamber, Micheline.2001. *Data mining: concepts and techniques[C]*. Morgan Kaufmann. 5.
[29] Jansen B.J and Rieh S.2010. *The Seventeen Theoretical Constructs of Information Searching and Information Retrieval[J]*. Journal of the American Society for Information Sciences and Technology. 61(8)
[30] O Celma. 2010. *Music Recommendation and Discovery in the Long Tail[C]*. Springer.
[31] Robin Burke. November 2002. *Hybrid recommender systems: Survey and experiments*. User Modeling and User-Adapted Interaction, 12:331–370.
[32] Henry Kautz, Bart Selman, and Mehul Shah.March 1997. *Referral web: combining social networks and collaborative filtering[C]*. Commun. ACM, 40:63–65.
[33] Jonathan L. Herlocker, Joseph A. Konstan, Loren G. Terveen, and John T. Riedl.January 2004. *Evaluating collaborative filtering recommender systems[C]*. ACM Trans.Inf.Syst, 22:5–53.
[34] 周涛.2011. 基于内容的推荐算法. http://blog.sciencenet.cn/blog-3075-459442.html.
[35] Andrew I.Schein, Alexandrin Popescul, Lyle H.Ungar, David M.Pennock. 2002. *Methods and Metrics for Cold-Start Recommendations[C]*. New York City, New York: ACM. 253–260.
[36] Adams, Suellen .2005. *Information Behavior and the Formation and Maintenance of Peer Cultures in Massive Multiplayer Online Roleplaying Games: a Case Study of City of Heroes*. DiGRA: Changing Views - Worlds in Play. Authors & Digital Games research Association (DiGRA).
[37] Nabeth, Thierry (26 May 2006). *Understanding the Identity Concept in the Context of Digital Social Environments*. FIDIS Deliverables. 2. FIDIS. pp. 74–91.
[38] Suler, John (2004). *The Online Disinhibition Effect*. CyberPsychology & Behavior. 7 (3): 321–326.
[39] Marcus, Bernd; Machilek, Franz; Schütz, Astrid (2006). *Personality in cyberspace: Personal web sites as media for personality expressions and impressions*. Journal of Personality and Social Psychology. 90 (6): 1014–1031.
[40] Siibak, Andra (September 2007). *Casanovas of the Virtual World. How Boys Present Themselves on Dating Websites*. Young People at the Crossroads: 5th International Conference on Youth Research. Petrozavodsk, Republic of Karelia, Russian Federation. pp. 83–91.
[41] Hartigan, J.A.Wong, M.A.Algorithm. *A k-Means Clustering Algorithm*. Journal of the Royal Statistical Society, Series C. 1979, 28 (1): 100–108.
[42] K Yoshii.2006. *Hybrid Collaborative and Content-Based Music Recommendation Using Probabilistic Model with Latent User Preferences [C]*. In: Proceedings of the International Conference on Music Information Retrieval.
[43] 山西晚报, 科学频道.2015. 大数据揭秘：淘宝上的假货、次品都卖给了谁？. http://science.china.com.cn/2015-12/01/content_8417479.htm.
[44] Maja Kabiljo, Aleksandar Ilic. June, 2015. *Recommending items to more than a billion people[DB/OL]*. https://code.facebook.com/posts/861999383875667/recommending-items-to-more-than-a-billion-people.
[45] 稳国柱. 2015. 寻路推荐豆瓣推荐系统实践之路 *[DB/OL]*. http://www.36dsj.com/archives/35273.
[46] 知乎网网友. 2011. 豆瓣 *FM* 的推荐算法是怎样的 *[DB/OL]*. https://www.zhihu.com/question/19560538.
[47] Gediminas Adomavicius, Alexander Tuzhilin. JUNE 2005. *Towards the Next Generation of Recommender Systems: A Survey of the State-of-the-Art and Possible Extensions*. IEEE transactions on knowledge and data engineering, vol. 17, no. 6.
[48] From Wikipedia, the free encyclopedia. *Collaborative filtering*. https://en.wikipedia.org/wiki/Collaborative_filtering#Types.
[49] Daniel Lemire, Anna Maclachlan. *Slope One Predictors for Online Rating-Based Collaborative Filtering*. In SIAM Data Mining (SDM'05), Newport Beach, California, April 21-23, 2005.
[50] Yehuda Koren.2009. *Collaborative filtering with temporal dynamics[J]*. In Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining, KDD'09,New York, NY, USA., pages 447–456.

[51] Kohavi, Ron, Longbotham, Roger.2015. *Online Controlled Experiments and A/B Tests[C]*. In Sammut.

致 谢

人生就是一个关于成长的漫长故事。而在中科大求学作为本人人生体验的一部分，亦是这样的一段故事。在此的俩年半，俯仰之间，科大的“问道”、“学术”于此，让我经历了这样的三段成长：学于师友，安于爱好，观于内心。

“古之学者必有师，师者，所以传道、授业、解惑也”。师友的教诲不可能一直跟着自己，可是他们治学态度却融入了我的人生观。授课的华保健老师的严谨、郭燕老师的认真、丁菁老师的直率、席菁老师的踏实都曾触动我，并给予我前进方向上的指引。

本论文内容为数据挖掘在电商行业的工程实现，因此有一段真实的、贴近数据挖掘领域的实习经历尤为重要。感谢我在苏州国云数据公司实习的 CEO 马晓东学长，让我有机会一窥大数据行业的内幕；感谢我在小米实习的导师方流博士，感谢我在滴滴出行工作的机器学习研究院李佩博士和袁森博士，让我成为大数据挖掘工程师的梦想又更近了一步；感谢我的导师周武旸教授和张四海教授，指导我完成论文。向师友和书籍学习，是从外界汲取；只有回归到自己的内心和思绪才能沉淀．在每个夜幕深沉或是晨曦初露的时刻里，感受自己情绪的流动，反思自己的取舍得失，然后才有了融于师友和书籍时的奋进。这样的三段成长，如今已是一体，不断地相互印证与反馈！

“逝者如斯夫，不舍昼夜”。成长亦复如是，不断的和昨日的自己告别。但是，一路有你，真好！相会是缘，同行是乐，共事是福！

胡磊

2017 年 2 月 21 日