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: 二〇一六年四月

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 :	April 21th, 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

信息爆炸使得用户很难有效的从海量的数据中快速获取自己需要的信息，推荐系统凭借精准定位和” 千人千面” 的个性化服务受到互联网企业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，其中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，推荐系统需要解决社交化、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个性化推荐系统的用户画像模型，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，标签的传递路径，转发关系，标签的本身，以及标签与用户之间的共现关系决定了这个标签对应的权重，权重越高则认为该标签的可信度越高。实验显示结合了用户画像的推荐系统能显著提升推荐结果的点击转换率。

- 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，对每个用户维护一个动态变化着的兴趣标签向量空间。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。实验表明示结合了用户兴趣探索的推荐系统能显著提升推荐结果的多样性。

- 利用线性衰减算法成功融合用户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，本文提出了基于时间的线性衰减模型能有效融合用户的长、短期兴趣。

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 has improved 8% of click conversion rate.

- 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

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 [10]。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 [12] 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。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，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。而作为互联网企业，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

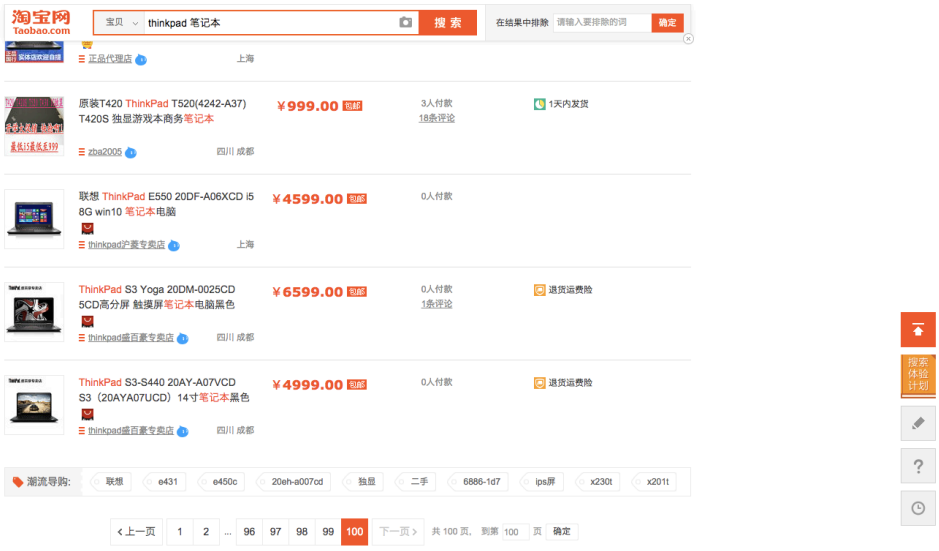

图 1.1　淘宝购物搜索图

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来解决这一矛盾。传统的推荐系统通过分析用户的数据，通过对商品打分、排序找到用户感兴趣的商品，然后推荐给他们，但是传统的推荐系统在大数据时代也面临着严峻的问题，典型的问题有数据稀疏问题、新用户问题、马太效应、实时推荐问题和用户兴趣变动问题。

基于这种现实，人们开始回归问题的本质即以人为本，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，而且此信息是动态更新的，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。因此，用户画像大大的提高了互联网企业的用户体验。用户画像的主要任务就是让推荐系统更了解用户，一方面让信息更加针对性的展现在只对它有兴趣的用户面前，提升商品的转化率，另一方面协助用户发现自己潜在感兴趣的信息，提升用户的满意度，于是实现了消费者和生产者的双赢。

1.2　推荐系统的简介

推荐系统的研究和很多早期的研究相关，比如认知科学 [13]，信息检索和预测理论 [14]。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 [15]。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 [16]。推荐系统推荐给用户的商品首先不能与用户购买过的商品重复，其次也不能与用户刚浏览过的商品太相关。推荐系统的形式化定义如：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万款手机主题。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

随着科学技术与信息传播的迅猛发展，人类社会进入了一个全新的大数据时代，互联网和物联网无处不在的影响着人类生活的方方面面，并颠覆性改变了人们的生活方式，互联网用户既代表了网络信息的消费者，也代表了网络内容的生产者。尤其是随着 Web 2.0 时代的到来，社交化网络媒体的异军突起，互联网中的信息量呈指数级增长，而由于用户的辨别能力有限，使得其在庞大且复杂的互联网信息中找寻有用信息的成本巨大，这就是所谓的信息过载问题 [17, 18]。搜

索引擎和推荐系统的出现为用户解决信息过载提供了非常重要的技术手段。搜索引擎是被动的，用户在搜索互联网中的信息时需要在搜索引擎中输入关键词，搜索引擎根据输入在系统后台进行信息匹配，将与用户查询相关的信息展示给用户。但是当用户无法精确描述自己需求时，搜索引擎就无能为力了。推荐系统是主动的，用户不需要提供明确的需求，而是通过分析用户的历史行为来对用户进行分析，从而主动给用户推荐可能满足他们兴趣和需求的信息。因此搜索引擎和推荐系统是两个互补的技术手段。

推荐系统概念是 1995 年在美国人工智能协会 [19] 上由 CMU 大学的教授 Robert Armstrong 首先提出并推出了推荐系统的原型系统——Web Watcher。随后推荐系统的研究工作开始慢慢壮大。第一个正式商用的推荐系统是 1996 年 Yahoo 网站推出的个性化入口 MyYahoo。21 新世纪推荐系统的研究与应用随着电子商务的快速发展而风起云涌，各大电子商务网站都开发、部署了推荐系统，Amazon 公司称其网站中 35% 的营业额来自于自身的推荐系统。2006 年美国的 DVD 租赁公司 Netflix[11] 在网上公开设立了一个推荐算法竞赛并公开了真实网站中的一部分数据，包含用户对电影的评分。Netflix 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。

近几年随着电子商务蓬勃发展，推荐系统在互联网中的优势地位也越来越明显。在国外比较著名的电子商务网站有 Amazon 和 eBay，其中 Amazon 平台中采用的推荐算法是非常成功的。在国内比较典型的电子商务平台网站有淘宝网、网页云音乐、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非常巨大。据不完全统计天猫商城中的商品数量已经超过了 5000 万。在商品数量如此庞大的电商网站中，如果用户仅仅根据自己的购买意图输入关键字查询只会得到很多用户很难区分的相似结果，也不便用户做出选择。因此推荐系统作为能够根据用户兴趣 [20] 为用户推荐商品的主要途径，从而为用户在购物的选择中提供建议的需求非常明显。目前比较成功的电子商务网站中，都不同程度地利用推荐系统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，随着以智能手机为代表的物联网推动了移动互联网的发展。在用户在连入移动互联网的过程中，其所处的地理位置信息可以非常准确地被获取，并由此出现了大量的基于用户位置信息的网站。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。例如，在美团网这种基于位置服务的网站中，用户可以根据自己的当前位置搜索餐馆、酒店、影院、旅游景点等信息服务。同时，可以对当前位置下的各类信息进行点评，为自己在现实世界中的体验打分，分享自己的经验与感受。当用户使用这类基于位置的网站服务时，同样会遭遇信息过载问题。推荐系统可以根据用户的位置信息为用户推荐当前位置下用户感兴趣的内容，为用户提供符合其真正需要的内容，提升用户对网站的满意度。

随着社交网络的深入人心，用户在互联网中的行为不再局限于获取信息，更

多的是与网络上的其他用户进行互动。国外著名的社交网络有 Facebook、Twitter 等，国内的社交网络有微信、米聊等。在社交网站中用户不再是单个的个体，而是与网络中的很多人具有了错综复杂的社交关系链。社交网络中最重要的资源就是用户与用户之间的这种联系。社交网络中用户间的关系是多维度的，建立社交关系的因素可能是在现实世界中是亲人、同学、同事、朋友关系，也可能只是网络中的虚拟朋友，比如都是有着共同爱好的会员成员。在社交网络中用户与用户之间的联系紧密度反映了用户之间的信任关系，用户不在是一个个体存在，其在社交网络中的行为或多或少地会受到其他用户关系的影响。因此推荐系统在这类社交网站中的研究与应用应该考虑用户社交的影响。

现如今推荐系统在很多领域得到了广泛的应用，如出租车推荐、商品推荐、美餐推荐、电影推荐和音乐推荐，几乎囊括了人类的吃住行穿四大领域，团购网站美团网早已经利用推荐系统提供面向不同业务的个性化服务：1，猜你喜欢：美团最重要的推荐产品，目标是让用户打开美团 App 的时候，可以最快找到用户想要的团购服务；2，首页频道推荐：若干频道是固定的，若干频道是根据用户的个人偏好推荐出来的；3，今日推荐个性化推送：美团的个性化推送的产品，目的是在用户打开美团 App 前，就把用户最感兴趣的服务推送给用户，促使用户点击及下单，从而提高用户的活跃度；4，品类列表的个性化排序：美团首页的那些品类频道区。

自推荐系统诞生后学术界对其关注的兴趣度也越来越大。从 1999 年开始美国计算机学会每年召开电子商务研讨会以来，发表的与推荐系统相关的论文数以千计。ACM 信息检索专业组在 2001 年开始把推荐系统作为该会议的一个独立研究主题。同年召开的人工智能联合大会也将推荐系统作为一个单独的主题。目前为止数据库、数据挖掘、人工智能、机器学习方面的重要国际会议（如 KDD、AAAI、ICML 等）都有大量与推荐系统相关的研究成果发表。同时第一个以推荐系统命名的国际会议 ACM Recommender Systems Conference 于 2007 年首次举办。在近几年的数据挖掘及知识发现国际会议举办的竞赛中，连续两年的竞赛主题都是推荐系统。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

推荐系统改变了没有活力的网站与其用户通信的方式。无需提供一种静态体验，让用户搜索并可能购买产品，推荐系统加强了交互，以提供内容更丰富的体验。推荐系统根据用户过去的购买和搜索历史，以及其他用户的行为，自主地为各个用户识别推荐内容。个性化推荐的最大的优点在于它能收集用户特征资料并根据用户特征，如兴趣偏好，为用户主动作出个性化的推荐。而且，系统

给出的推荐是可以实时更新的，即当系统中的商品库或用户特征库发生改变时，给出的推荐序列会自动改变。这就大大提高了电子商务活动的简便性和有效性，同时也提高了企业的服务水平。总体说来，一个成功的个性化推荐系统的作用主要表现在以下几个方面：

(1) 将电子商务网站的浏览者转变为购买者：电子商务系统的访问者在浏览过程中经常并没有购买欲望，个性化推荐系统能够向用户推荐他们感兴趣的商品，从而促成购买过程。

(2) 提高电子商务网站的交叉销售能力：个性化推荐系统在用户购买过程中向用户提供其他有价值的商品推荐，用户能够从系统提供的推荐列表中购买自己确实需要但在购买过程中没有想到的商品，从而有效提高电子商务系统的交叉销售。

(3) 提高客户对电子商务网站的忠诚度：与传统的商务模式相比，电子商务系统使得用户拥有越来越多的选择，用户更换商家极其方便，只需要点击一两次鼠标就可以在不同的电子商务系统之间跳转。个性化推荐系统分析用户的购买习惯，根据用户需求向用户提供有价值的商品推荐。如果推荐系统的推荐质量很高，那么用户会对该推荐系统产生依赖。因此，个性化推荐系统不仅能够为用户提供个性化的推荐服务，而且能与用户建立长期稳定的关系，从而有效保留客户，提高客户的忠诚度，防止客户流失。

1.3　用户画像的简介

用户，指企业的目标用户，或者是构成现有用户的大部分群体的统称。画像，是对一个事物的客观的、准确的、可视化的描述。用户画像就是能够客观、准确、可视化地描述目标用户的模型或方法。为用户画像的焦点工作就是为用户打标签，而一个标签通常是人为规定的高度精炼的特征标识，如年龄、性别、地域、用户偏好等，最后将用户的所有标签综合来看，基本就可以勾勒出该用户的立体画像了。

1.3.1　用户画像的产生背景

在互联网逐渐步入大数据时代后，不可避免的给企业及消费者行为带来一系列改变与重塑。其中最大的变化莫过于，消费者的一切行为在企业面前似乎都将是可视化的。随着大数据技术的深入研究与应用，企业的专注点开始回归本质，日益聚焦于怎样利用大数据来为精准刻画用户，进而深入挖掘潜在的商业价值。于是，用户画像的概念也就应运而生。大数据时代的用户画像代表了一个用户的信息全貌，为进一步精准、快速地分析用户行为习惯、消费习惯等重要信息，提供了足够的数据基础。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，抽象出一个用户的商业全貌是企业应用大数据技术的基本方式。用户画像为企业提供了足够的信息基础，能够帮助企业快速找到精准用户群体以及用户需求等更为广泛的反馈信息。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%。不同于传统 PC 上网，每个家庭共用一台设备，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。因此，将有相同特征的用户抽象成一个代表，可以极大方便开发者研究用户构成和分布，精准定义用户。中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，面临在所有互联网企业面前的问题是，如何才能将流量变现，实现产品的商业价值呢？为了充分发挥大数据的真正价值，第一步理应是整理数据。而整理数据的阶段目的是完成目标用户或者是现有用户的画像，只有得到了准确的用户画像，才能更好的达到流量变现的目的。

1.3.2　用户画像的应用

用户画像的意义在于完善产品运营提升用户体验，提升盈利，根据产品特点，找到目标用户，在用户偏好的渠道上与其交互，促成购买，实现精准运营和营销。于此同时，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。

- 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，如对于没有填写性别信息的用户，用户画像通过分析用户语音数据识别其性别，尽可能多的为推荐系统提供正确的基础特征。

- 打造健康的生态圈：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。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，热门商品的传播方式和流行引爆点等。

- 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。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。

- 市场安全领域的应用：随着电子商务的发展，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。

1.4　工程背景

小米科技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。我们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，但是代价就是加剧商品的马太效应。

(2) 数据稀疏问题，通过观察我们发现只有约 10% 的用户有过多于 10 款主题行为记录，意味着大多数主题包处于待挖掘状态，然后，这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，打破这个闭环。

(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美女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，不经济，并且会给用户一种不专业、不智能的体验。

(5) 其他问题，如推荐商品长尾性有待加强、隐性喜好难以挖掘、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，如果在数据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 (itemId,buyTime) 这样的二元组。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，针对冷门主题 (冷门主题包含的标签一般是小众标签) 的行为会赋予一个倾斜因子，这样会使得兴趣探索标签候选集中的小众标签占大多数，而如果用户对这些主题的满意度也很高，则说明这是一个成功的兴趣探索。

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，达到优化推荐系统的目的，后续需求包括广告投放、定向营销、流量变现等也都围绕建立更细致、准确的人群画像展开。

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、利用它也带来了很多困惑。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。

- Apache Hadoop：Hadoop 是一个由 Apache 基金会所开发的分布式系统基础架构，是一种用于分布式存储和处理商用硬件上大型数据集的开源框架，可让各企业迅速从海量结构化和非结构化数据中获得洞察力。Hadoop 的框

架最核心的设计就是 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。HDFS 有高容错性的特点，并且设计用来部署在低廉的硬件上；而且它提供高吞吐量来访问应用程序的数据，适合那些有着超大数据的应用程序。MapReduce 本身就是用于并行处理大数据集的软件框架，其根源是函数性编程中的 map 和 reduce 函数。它由两个可能包含有许多实例的操作组成。Map 函数接受一组数据并将其转换为一个键/值对列表，输入域中的每个元素对应一个键/值对。

- Apache Hive：Hive 是建立在 Hadoop 上的数据仓库基础构架。它提供了一系列的工具，可以用来进行数据提取转化加载，这是一种可以存储、查询和分析存储在 Hadoop 中的大规模数据的机制。Hive 定义了简单的类 SQL 查询语言，称为 HQL，它允许熟悉 SQL 的用户查询数据。同时，这个语言也允许熟悉 MapReduce 开发者的开发自定义的 mapper 和 reducer 来处理内建的 mapper 和 reducer 无法完成的复杂的分析工作，十分适合数据仓库的统计分析。

- Apache Spark：Spark 是加州大学伯克利分校所开源的类 Hadoop 的通用并行框架，Spark 拥有 Hadoop 所具有的优点；但不同于 Hadoop 的是 Job 中间输出结果可以保存在内存中，从而不再需要读写 HDFS，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 MapReduce 的算法。

- Apache Kafka：Kafka 是一种高吞吐量的分布式发布订阅消息系统，它可以处理消费者规模的网站中的所有用户行为流数据。这种用户行为流数据是在现代网络上的许多社会功能的一个关键因素。这些数据通常是由于吞吐量的要求而通过处理日志和日志聚合来解决。对于像 Hadoop 的一样的日志数据和离线分析系统，但又要求实时处理的限制，Kafka 一个可行的解决方案。其目的是通过 Hadoop 的并行加载机制来统一线上和离线的消息处理，也是为了通过集群机来提供实时的消费。

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和排序模型，包括数据挖掘算法 [27] 和信息提取技术 [28] 的应用，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 [29]，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　推荐系统和用户画像综述

2.1　引言

自从 1992 年施乐的科学家为了解决信息负载的问题，第一次提出协同过滤算法，个性化推荐已经经过了二十几年的发展。1998 年，林登和他的同事申请了 item-to-item 协同过滤技术的专利，经过多年的实践，亚马逊宣称销售的推荐占比可以占到整个销售 Gross Merchandise Volume（年度成交总额）的 30% 以上。随后 Netflix 举办的推荐算法优化竞赛，吸引了数万个团队参与角逐，期间有上百种的算法进行融合尝试，加快了推荐系统的发展，其中 Sigular Value Decomposition（奇异值分解）和 Gavin Potter 跨界的引入心理学的方法进行建模，在诸多算法中脱颖而出。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 分解为两个矩阵：User 特性的矩阵 P 和 Item 特性的矩阵 Q，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 的非零元素那些位置上的值尽量接近 R 中的元素，通过定义 R 和 R’之间的距离，把矩阵分解转化成梯度下降等求解的局部最优解问题。与此同时，Pandora、LinkedIn、Hulu、Last.fm 等一些网站在个性化推荐领域都展开了不同程度的尝试，使得推荐系统在垂直领域有了不少突破性进展，但是在全品类的电商、综合的广告营销上，进展还是缓慢，仍然有很多的工作需要探索。特别是在全品类的电商中，单个模型在母婴品类的效果还比较好，但在其他品类就可能很差，很多时候需要根据品类、推荐栏位、场景等不同，设计不同的模型。同时由于用户、SKU 不停地增加，需要定期对数据进行重新分析，对模型进行更新，但是定期对模型进行更新，无法保证推荐的实时性，一段时间后，由于模型训练也要相当时间，可利用传统的批处理的 Hadoop 的方法是无法再缩短更新频率，最终推荐效果会因为实时性问题达到一个瓶颈。推荐算法主要有基于人口统计学的推荐、基于内容的推荐、基于协同过滤的推荐等，而协同过滤算法又有基于邻域的方法、隐语义模型、基于图的随机游走算法等。基于内容的推荐解决了物品的冷启动问题，但是解决不了用户的冷启动问题，并且存在过拟合问题，即在训练集上有比较好的表现，但在实际预测中效果大打折扣，对领域知识要求也比较高，通用性和移植性比较差，换一个产品形态，往往需要重新构建一套，对于多媒体文件信息特征提取难度又比较大，往往只能通过人工标准信息。基于邻域的协同过滤算法，虽然也有冷启动问题和数据稀疏性等问题，但是没有领域知识要求，算法通用性好，增加推荐的新颖性，并且对行为丰富的物品，推荐准确度较高。基于模型的协同过滤算法在一定程度上解决了基于邻域的推荐算法面临的一些问题，在 Root Mean Squared Error（均方根误差）等推荐评价指标上更优，但是通常算法复杂，计算开销大，所以目前基于邻域的协同过滤算法仍然是最为流行的推荐算法。

2.2　推荐系统的研究现状

Amazon 的数百万图书，Netflix 的 10 万部电影，淘宝的 8 亿件在线物品，以及数以亿万计用户的资料和行为记录。互联网公司最近十年的迅猛发展伴随着海量数据的积累。然而，在线用户常常面对过多的选择而显得无所适从。心理学研究证实这类情境下的用户有时做出放弃交易的决定，从而造成大量潜在的用户流失。统计技术的发展能够为在线服务商提供更有效的推荐算法，在帮助用户走出信息过载困境、改善用户体验的同时，还能够挖掘物品长尾、提升企业价值。在今天，用户不再局限于通过搜索引擎来寻找感兴趣的信息，推荐系统无所不在地为我们发现自己的潜在需求。

2.2.1　推荐系统的商业应用

作为全球排名第一的社交网站，Facebook 就需要利用分布式推荐系统来帮助用户找到他们可能感兴趣的页面、组、事件或者游戏等。之前 Facebook 在其官网公布了其推荐系统的原理、性能及使用情况 [6]。目前，Facebook 中推荐系统所要面对的数据集包含了约 1000 亿个评分、超过 10 亿的用户以及数百万的物品，如何在在大数据规模情况下仍然保持良好性能已经成为世界级的难题。即使是采用了分布式计算方法，Facebook 仍然不可能检查每一个用户/物品对的评分。团队需要寻找更快的方法来获得每个用户排名前 K 的推荐物品，然后再利用推荐系统计算用户对其的评分，解决方案是采用 ball tree 数据结构来存储物品向量。all tree 结构可以实现搜索过程 10-100 倍的加速，使得物品推荐工作能够在合理时间内完成。最后，Facebook 给出了一些实验的结果。在 2014 年 7 月，Databricks 公布了在 Spark 上实现 ALS 的性能结果。Facebook 针对 Amazon 的数据集，基于 Spark MLlib 进行标准实验，与自己的旋转混合式方法的结果进行了比较。实验结果表明，Facebook 的系统比标准系统要快 10 倍左右。而且，前者可以轻松处理超过 1000 亿个评分。

目前，该方法已经用了 Facebook 的多个应用中，包括页面或者组的推荐等。为了能够减小系统负担，Facebook 只是把度超过 100 的页面和组考虑为候选对象。而且，在初始迭代中，Facebook 推荐系统把用户喜欢的页面/加入的组以及用户不喜欢或者拒绝加入的组都作为输入。此外，Facebook 还利用基于 ALS 的算法，从用户获得间接的反馈。未来，Facebook 会继续对推荐系统进行改进，包括利用社交图和用户连接改善推荐集合、自动化参数调整以及尝试比较好的划分机器等。Facebook 推荐主页如图 2.1。

豆瓣网在国内互联网行业美誉度很高，这是一家以帮助用户发现未知事物为己任的公司 [7]。不用设置播放列表，也不用费心想听什么，打豆瓣电台就能收听。全新的音乐体验，让用户和喜欢的音乐不期而遇，找到符合用户口味音乐，豆瓣电台坚持这样的理念制作着全新的网络电台。通过优质的推荐系统为用

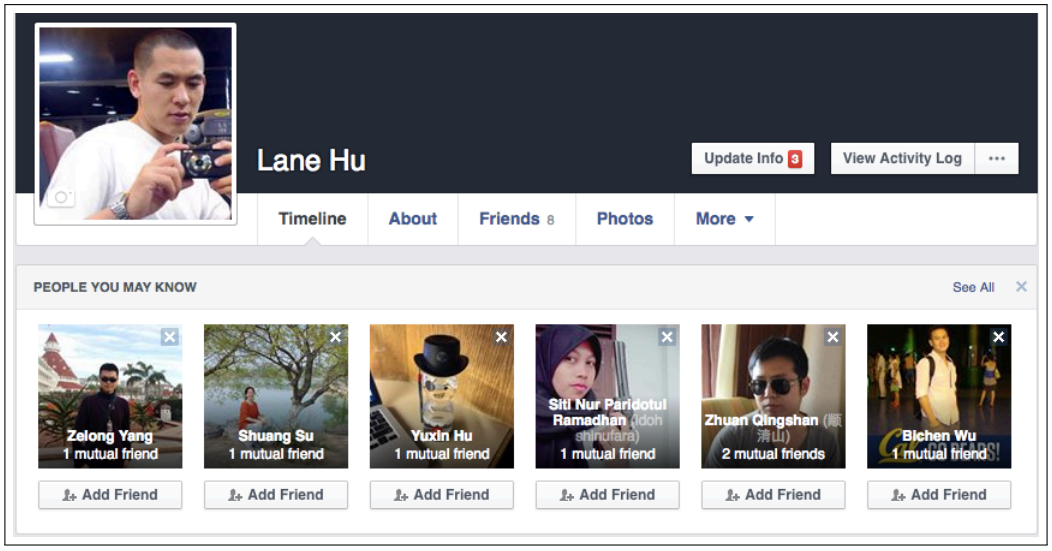

图 **2.1**　Facebook 个性化推荐用户界面

图 **2.2**　豆瓣电台个性化推荐用户界面

户提供喜欢的音乐，在优质的电台频道下听喜欢的歌曲，豆瓣电台为喜爱音乐的人提供了这样一种全新音乐体验，音乐，本是件轻松的事，豆瓣电台做的很好。豆瓣电台的私人电台会综合用户在豆瓣上的各种音乐行为做算法推荐 [8]。在豆瓣音乐中，有哪些音乐标签，喜欢哪些歌手，在听哪些，想听哪些，乐评，豆列等等，即豆瓣音乐中提供的社会化网络行为，会有相关的权重算出一个公式，最开始的时候只是一个最简单的算法，在进行一段时间数据根据和用户反馈后，有了更多的权重值累加。考虑最多的是电台本身的红心、垃圾、跳过、这些用户行为数据。豆瓣电台糅合了包括算法、数据清洗与整合、音频分析技术、用户行为分析、编辑与运营、后台架构等等大量的因素，即便是推荐算法也只是算法技术中的一部分。单论推荐算法，就最简单的算法，也会极大地受到其它因素的影响，比如单曲推荐功能、新版的上线，对于算法的学习与积累都会起到极大的正面作用。简洁、小清新的界面风格，简单又易用的操作，豆瓣电台延续了豆瓣一贯的品质，豆瓣电台推荐页如图 2.2。

2.2.2 推荐系统的主要方法

推荐系统主要是评分预测和 Top-N 预测，不论是哪一种推荐方式，其核心的目标是找到最适合用户 c 的项集合 s，从集合里挑选集合是一个非常复杂的问题优化方案，通常采用的方案是用局部最优的方式，而我们只需要定义一个的效用函数，选取 Top-N [5]。

(1) 协同过滤的推荐

推荐系统应用数据分析技术，找出用户最可能喜欢的东西推荐给用户，现在很多电子商务网站都有这个应用。目前用的比较多、比较成熟的推荐算法是协同过滤（Collaborative Filtering）推荐算法，协同过滤的基本思想是根据用户之前的喜好以及其他兴趣相近的用户的选择来给用户推荐物品。在协同过滤中，用 m×n 的矩阵表示用户对物品的喜好情况，一般用打分表示用户对物品的喜好程度，分数越高表示越喜欢这个物品，0 表示没有买过该物品。图中行表示一个用户，列表示一个物品，U_{ij} 表示用户 i 对物品 j 的打分情况。协同过滤分为两个过程，一个为预测过程，另一个为推荐过程。预测过程是预测用户对没有购买过的物品的可能打分值，推荐是根据预测阶段的结果推荐用户最可能喜欢的一个或 Top-N 个物品。协同过滤算法分为两大类，一类为基于内容的（Memory-based），一类为基于模型的（Model-based），User-based 和 Item-based 算法均属于 Memory-based 类型 [4]，这里主要介绍 User-based 的协同过滤。

有效用函数定义为用户之间的喜好上的相似性，因为基于用户的协同过滤算法是根据邻居用户的偏好信息产生对目标用户的推荐。它基于这样一个假设: 如果一些用户对某一类项目的打分比较接近，则他们对其它类项目的打分也比较接近。协同过滤推荐系统采用统计计算方式搜索目标用户的相似用户，并根据相似用户对项目的打分来预测目标用户对指定项目的评分，最后选择相似度较高的前若干个相似用户的评分作为推荐结果，并反馈给用户。这种算法不仅计算简单且精确度较高，被现有的协同过滤推荐系统广泛采用。User-based 协同过滤推荐算法的核心就是通过相似性度量方法计算出最近邻居集合，并将最近邻的评分结果作为推荐预测结果返回给用户。

例如，在表 2.1所示的用户—项目评分矩阵中，行代表用户，列代表项目 (电影)，表中的数值代表用户对某个项目的评价值。现在需要预测用户 Tom 对电影《枪王之王》的评分 (用户 Lucy 对电影《阿凡达》的评分是缺失的数据)。由表 2.1不难发现，Mary 和 Pete 对电影的评分非常接近，Mary 对《暮色 3: 月食》、《唐山大地震》、《阿凡达》的评分分别为 3、4、4，Tom 的评分分别为 3、5、4，他们之间的相似度最高，因此 Mary 是 Tom 的最接近的邻居，Mary 对《枪王之王》的评分结果对预测值的影响占据最大比例。相比之下，用户

John 和 Lucy 不是 Tom 的最近邻居，因为他们对电影的评分存在很大差距，所以 JohLn 和 Lucy 对《枪王之王》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	阿凡达	枪王之王
John	4	4	5	4
Marry	3	4	4	2
Lucy	2	3		3
Tom	3	5	4	

User-based 算法存在两个重大问题：1、数据稀疏性。一个大型的电子商务推荐系统一般有非常多的物品，用户可能买的其中不到 1% 的物品，不同用户之间买的物品重叠性较低，导致算法无法找到一个用户的邻居，即偏好相似的用户。2、算法扩展性。最近邻居算法的计算量随着用户和物品数量的增加而增加，不适合数据量大的情况使用。Iterm-based 的基本思想是预先根据所有用户的历史偏好数据计算物品之间的相似性，然后把与用户喜欢的物品相类似的物品推荐给用户。还是以之前的例子为例，可以知道物品 a 和 c 非常相似，因为喜欢 a 的用户同时也喜欢 c，而用户 A 喜欢 a，所以把 c 推荐给用户 A。因为物品直接的相似性相对比较固定，所以可以预先在线下计算好不同物品之间的相似度，把结果存在表中，当推荐时进行查表，计算用户可能的打分值，可以同时解决上面两个问题。

(2) 基于内容的推荐

有效用函数定义为用户和物品的内容上的相似性，基于内容推荐的推荐过程：1、内容分析，从原先的物品信息（例如文档、网页、新闻、产品描述）中提取有用的信息用一种适当的方式表示。2、文件学习，作用是收集、泛化代表用户偏好的数据，生成用户概要信息。通常是采用机器学习方法从用户之前喜欢和不喜欢的物品信息中推出一个表示用户喜好的模型。例如，一个基于网页的推荐系统的属性学习器能够实现一个相关反馈的方法，将表示正面和负面例子的向量与表示用户概要信息的原型向量混合在一起。训练样例是那些附有用户正面和负面反馈信息的网页。3、过滤，通过学习用户概要信息，匹配用户概要信息和物品信息，推荐相关的物品，结果是一个二元的连续型的相关判断。后者将生成一个用户可能感兴趣的潜在物品评分列表。该匹配是计算原型向量和物品向量的余弦相似度。

最初的基于内容的推荐是协同过滤技术的延续与发展 [1]，它不需要依据用户对项目的评价意见，而是依据用户已经选择的产品内容信息计算用户之间的相似性，进而进行相应的推荐。随着机器学习等技术的完善，当前的基于内容的推荐系统可以分别对用户和产品建立配置文件，通过分析已经购买或

浏览过的物品内容，建立或更新用户的配置文件。系统可以比较用户与产品配置文件之间的相似度，并直接向用户推荐与其配置文件最相似的产品——这种直接比较用户和产品相似性并进行推荐的方法，就无法纳入协同过滤的框架了。例如，在电影推荐中，基于内容的系统首先分析用户已经看过的打分比较高的电影的共性（演员、导演、语言、风格等），再推荐与这些用户感兴趣的电影内容相似度高的其它电影。基于内容的推荐算法的根本在于内容的获取和定量分析，因为在文本信息获取与过滤方面的研究较为成熟，现有很多基于内容的推荐系统都是通过分析产品的文本信息进行推荐。现在已经有一些技术可以从图片、音乐、视频中自动抽取内容信息，但是抽取后的内容多以文本、关键词（标签）、特征向量等方式表达。对这些信息的进一步处理方法，其实和文本处理是类似的。当然，文本处理发展到今天，方法已经是玲珑满目，有一部分进展我们会在下面一节中提到。本节我们介绍最为经典，也是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文本 j 中所有关键词出现的最大频率得到。附图给出了 3 个文本文件和 5 个关键词，以第一个关键词百分点为例，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流量，一共出现了 2 次，因此 $TF_{11}=0.5$。一个关键词如果在许多文件中同时出现，则该关键词对于表示文件的特性贡献较小，因此要考察一个关键词 i 出现次数的逆，也就是 $IDF_i=\log(N/n_i)$。这个想法和我们在介绍关联规则时提到的 Adamic-Adar 指数思路相似。按照这个定义，对于第一个关键词百分点，$IDF_1=\log(3/2)$。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 是整个系统中关键词的个数。一般而言，该向量中很多元素都为 0 。如果把用户购买或者浏览过的产品信息抽象成一个配置文件，也用这样的向量表示出来，则可以通过直接计算没有购买过的产品相应的文件的向量和用户的配置文件的向量的相似性，把相似性最大的产品推荐给该用户。在个性化技术研究历史中非常有名的 Fab 系统，就是使用内容推荐的典型例子。

文本1：不做软事，不说硬话。
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。

关键字包括软事、硬话、一万年、朝夕、青春、幸福、前途

总结起来，基于内容推荐的优点有：1、可以处理新用户和新产品问题。由于新用户没有选择信息，新产品没有被选信息，因此协同过滤推荐系统无法处理这类问题。但是基于内容的推荐系统可以根据用户和产品的配置文件进行

相应的推荐。2、实际系统中用户对产品的打分信息非常少，协同过滤系统由于打分稀疏性的问题，受到很大的限制。基于内容的推荐系统可以不受打分稀疏性问题的约束。3、通过列出推荐项目的内容特征，可以解释为什么推荐这些产品，使用户在使用系统的时候具有很好的用户体验。与此同时，我们也注意到，基于内容的推荐系统不可避免地受到信息获取技术的约束，例如自动提取多媒体数据（图形，视频流，声音流等）的内容特征具有技术上的困难，这方面的相关应用受到了很大限制。另外，关键词的设计往往需要领域专家的参与，否则通过自动算法获得的关键词很可能没有办法表现产品特征，反而引入过度噪音。

2.2.3 推荐系统评测的实验方法

任何推荐算法都要通过评测，才能评估它的推荐质量，本节介绍评测推荐系统常用的实验方法。

(1) 离线实验，是从日志系统中取得用户的行为数据，然后将数据集分成训练数据和测试数据，比如 80% 的训练数据和 20% 的测试数据，然后在训练数据集上训练用户的兴趣模型，在测试集上进行测试。优点只需要一个数据集即可，不需要实际的推荐系统，离线计算，不需要人为干预，能方便快捷的测试大量不同的算法。缺点是无法获得很多实际推荐系统的指标，比如点击率、转化率等。

(2) 用户调查，离线实验往往测的最多的就是准确率，但是准确率不等于满意度，所以在算法上线之前，需要用户调查一下，测试一下用户满意度。

(3) AB 测试，通过一定的规则把用户随机分成几组，并对不同组的用户采用不同的推荐算法，这样的话能够比较公平的获得不同算法在实际在线时的一些性能指标。但是缺点是周期比较长，需要长期的实验才能得到可靠的结果。

2.2.4 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，给用户推荐那些令他们感兴趣的图书。推荐系统还应该能够做到准确预测用户的行为，帮助用户发现那些他们可能感兴趣但不易本发现的物品（挖掘物品的长尾）。最后推荐系统也应该能够挖掘用户潜在的兴趣，将那些与用户兴趣无关但是用户看见之后可能会感兴趣的物品推荐给用户（后文将要说明的惊喜度）。从物品提供商角度，推荐系统要让提供商的物品都能够被推荐给对其感兴趣的用户。从平台角度，推荐系统能够让本身收集到高质量的用户反馈，不断完善推荐质量，增加用户和网站的交互（用户活跃度和粘稠度?）。

(1) 用户满意度

用户满意度是最最关键的指标，推荐系统推荐物品干嘛，就是希望推荐出来的物品能让用户满意。可以有两种方法，一是用户问卷调查，二是在线评测满意度，比如豆瓣的推荐物品旁边都有满意和不满意的按钮，亚马逊这种可以计算推荐的物品有没有被用户购买等等，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，则一般计算均方根误差和平均绝对误差。如果是 Top-N 推荐的话，则主要计算召回率和准确率。准确率就是指我推荐的 n 个物品中有多少个是对的，其所占的比重。召回率则是指正确结果中有多少比率的物品出现在了推荐结果中。两者的不同就是前者已推荐结果个数当除数，后者已正确结果个数当除数。

(3) 覆盖率

就是指推荐出来的结果能不能很好的覆盖所有的物品，是不是所有的物品都有被推荐的机会。最简单的方法就是计算所有被推荐的物品占物品总数的比重，当然这个比较粗糙，更精确一点的可以信息熵和基尼系数来度量。

(4) 多样性

推荐结果中要体现多样性，比如我看电影，我既喜欢看格斗类的电影，同时又喜欢爱装文艺，那么给我的推荐列表中就应该这两个类型的电影都有，而且得根据我爱好比例来推荐，比如我平时 80% 是看格斗类的，20% 是看文艺类的，那么推荐结果中最好也是这个比例。可以根据物品间的相似度来计算，一个推荐列表中如果所有物品间的相似度都比较高，那么往往说明都是同一类物品，缺乏多样性。

(5) 新颖性

不能说系统推荐的物品其实我都知道，那这样推荐系统就完全失去了存在的意义，一般都希望推荐一些用户不知道的物品或者没看过没买过的物品。方法一是取出已经看到过买过的物品，但这还不够，一般会计算推荐物品的平均流行度，因为通常越不热门的物品越会让用户觉得新颖。比如我爱周星驰，那么推荐《临岐》就很有新颖性，因为大家都不知道这是周星驰出演的。

(6) 惊喜度

这个和新颖度还是有区别的，惊喜度是讲我直觉想不出来为什么会给我推荐这物品，比如电影，但是我看了之后觉得很符合我的胃口，这就是惊喜度。像上面一个例子，只要我知道是周星驰演的，那可能就没什么惊喜度，因为

我知道是因为演员才给我推荐的这部电影。注：新颖性和惊喜度暂时没有什么可以度量的标准。

(7) 信任度

如果用户信任推荐系统，那么往往会增加与推荐系统的互动，从而获得更好的个性化推荐。增加信任的方法往往是提供推荐解释，即为什么推荐这个物品，做到有理有据。也可以通过类似 facebook 间的好友关系来增加信任度，一般相比于陌生人的推荐，总会选择好友给的推荐。

(8) 实时性

新闻等一些物品具有很强的实时性，一般得在具有有效性的时候进行推荐，必须考虑推荐系统处理物品冷启动的能力。

2.3 用户画像的研究现状

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。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。用户画像就是在解决把数据转化为商业价值的问题，就是从海量数据中来挖金炼银。这些高质量多维数据记录着用户长期大量的网络行为，用户画像据此来还原用户的属性特征、社会背景、兴趣喜好，甚至还能揭示内心需求、性格特点、社交人群等潜在属性。了解了用户各种消费行为和需求，精准刻画人群特征，并针对特定业务场景进行用户特征不同维度的聚合，就可以把原本冷冰冰的数据复原成栩栩如生的用户形象，从而指导和驱动业务场景和运营，发现和把握蕴藏在细分海量用户中的巨大商机。

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 [3]：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 10% 的用户，会收到更多垃圾产品。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。这里说明了利用用户画像可以做到精准销售，当然了，这是用户画像极其错误的用法。

2.3.1 用户画像的商业应用

作为中国一家大型全品类综合电商，京东利用用户画像应用服务支持公司集团全业务需求 [2]，其下游面向不同类型不同需求的人群，他们需求各不相同，从技术方案到使用方法也千差万别，因此有必要采取体系化多层次服务平台进行支持。对于公司内部，针对研发、采销、市场、客服、物流等各体系不同需求分别采取统一数据仓库、数据接口服务、产品化平台多种服务方式提供支持，针对各业务线需求场景不同，人员经验也不尽相同，用户画像的平台化给内部使用人员打造切合自身业务场景和使用经验的操作：对经验丰富的使用者提供更深入、综合参考并可自主订制或二次开发；给经验较浅的用户提供数据之外还培养其分析意识；对小白用户则可建立数据化分析运营的意识与习惯；对外部用户的支持力度也在逐步放开、加大，比如 POP 商家，可以满足商家针对自身店铺的个性化订制需求，并结合各种营销方式提供一站式服务解决方案。在京东用户行为日志中，每天记录着数以亿计的用户来访及海量行为。我们通过对用户行为数据进行分析和挖掘，发掘用户的偏好，逐步勾勒出用户的画像。用户画像通常通过业务经验和建立模型相结合的方法来实现，但有主次之分，有些画像更偏重于业务经验的判断，有些画像更偏重于建立模型。业务经验结合大数据分析为主勾画的人群，此类画像由于跟业务紧密相关，更多的是通过业务人员提供的经验来描述用户偏好。举个例子，比如：根据业务人员的经验，基于客户对金额、利润、信用等方面的贡献，建立多层综合指标体系，从而对用户的价值进行分级，生成用户价值的画像。一方面我们的产品经理可以根据用户价值的不同采取针对性的营销策略，另一方面通过分析我们的不同价值等级用户的占比，从而思考如何将低价值的用户发展成高价值的用户。

再比如，通过用户在下单前的浏览情况，业务人员可以区分用户的购物性格。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；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；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。对于不同购物性格的用户，我们可以推荐不同类型的商品，针对冲动型用户，我们直接推荐给他/她最畅销的同类商品，而理性型用户我们推荐给他/她口碑最好的商品。并且针对每一个用户，我们根据其购物性格定制了个性化的营销手段。

以建立模型为主勾画的人群，我们不能认为买过母婴类用品的用户家里就一定有小孩，因为这次购买很有可能是替别人代买或者送礼物。所以我们要判断这个用户所购买的母婴类商品是否是给自己买。根据用户下单前浏览情况、收货地址、对商品的评价等多种信息建立模型，最终判断出用户家庭是否有小孩。再根据购买的商品标签，比如奶粉的段数，童书适应年龄段等信息，建立孩子成长模型，在孩子所处不同的阶段进行精准营销。

京东拥有最全的品类，各品类间用户转化成为我们业务的一个重点。挖掘一

个品类的潜在用户，首先要找出此品类已有的用户，然后通过这些用户的行为、偏好、画像等信息对用户细分，挖掘其独有的特征，最后通过这些特征建立模型定位出该品类的潜在用户。

这一阶段主要是为了验证我们为用户描绘的画像是否准确。比如一个用户的画像是：性别男、年龄在 36 岁 45 岁之间、家里有小孩、未婚、有车一族、购买等级高。我们可以很快发现家里有小孩且未婚这一矛盾的结果。首先，我们可以判断对这个用户的画像肯定有问题的。接下来我们看这个用户的画像，似乎只有未婚这一条与其他画像格格不入。通过模型之间的验证，我们发现一些错误案例并分析原因，进而改进我们的模型。

2014 年的 618 前夕京东范产品的数据接口服务，将用户画像模型充分应用到产品当中，根据族群的差异化特征，帮助业务部门找到营销机会、运营方向，全面提高产品的核心影响力，增强产品用户体验。应用模型包括：年龄、性格、购物偏好、购买力等用户特征，诠释勾勒出用户在京东上的体貌特征，赋予一定的潮流“范儿”的概念，贴近用户。

京东数聚汇也是用户画像的一个典型应用，通过深度分析年度网购用户的行为，挖掘网络购物趣味数据，结合用户画像，从用户的购物行为入手，结合年度流行热点，分析不同地域网购人群的购物习惯和喜好，为网民展现一场京东大数据的饕餮盛宴，同时给商家和消费者提供了经营和购物参考。

2.3.2　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。需要有一个联系这两者的东西，即为标签。内容转换为标签即为内容特征化，用户则称为用户特征化。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。可以有一级维度、二级维度等。标签的来源主要有：已有内容的标签。网络抓取流行标签。对运营的内容进行关键词提取，对于内容的关键词提取，使用结巴分词和 TFIDF 即可。

(2) 内容特征化

内容特征化即给内容打标签。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，即针对一系列给定的标签，给内容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。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 word2vec 训练词的相似度模型。使用 tfidf 提取内容的关键词 A，B，C。遍历每一个标签，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内容的标签。

(3) 用户特征化

用户特征化即为用户打标签。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，不感兴趣赋予-1；对于用户的浏览行为，则可使用点击/浏览作为权值。对内容发生的行为可以认为对此内容所带的标签的行为。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。使用 click/pv 作为用户浏览行为权值即可达到此目的。此外，还需要考虑噪声的干扰，如标题党等。

(4) 隐语义推荐

有了内容特征和用户特征，可以使用隐语义模型进行推荐。这里可以使用其简化形式，以达到实时计算的目的。用户对于某一个内容的兴趣度 (可以认为是 CTR)：

$$r_{uc} = q_c * \sum_{i=1}^{n} m_{ci} * n_{ui} \tag{2.1}$$

其中 i=1…N 是内容 c 具有的标签，m_{ci} 指的内容 c 和标签 i 的关联度 (默认为是 1)，n_{ui} 指的是用户 u 的标签 i 的权重值，当用户不具有此标签时 n_{ui}=0，q 指的是内容 c 的质量，可以使用点击率 (click/page view) 表示。

2.3.3 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，完美地抽象出一个用户的商业全貌作是企业应用大数据技术的基本方式。构建周期如图 2.3。

(1) 数据收集

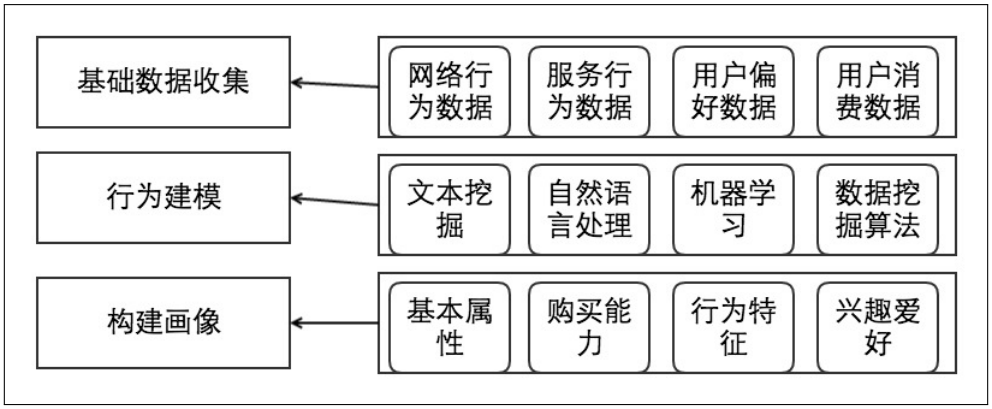

图 **2.3**　用户画像的构建周期示意图

数据收集大致分为网络行为数据、服务内行为数据、用户内容偏好数据、用户交易数据这四类。网络行为数据：活跃人数、页面浏览量、访问时长、激活率、外部触点、社交数据等。服务内行为数据：浏览路径、页面停留时间、访问深度、唯一页面浏览次数等。用户内容便好数据：浏览 / 收藏内容、评论内容、互动内容、生活形态偏好、品牌偏好等。用户交易数据（交易类服务）：贡献率、客单价、连带率、回头率、流失率等。收集到的数据不会是 100% 准确的，都具有不确定性，这就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男，但通过其行为偏好可判断其性别为女的概率更大。

(2) 行为建模

该阶段是对上阶段收集到数据的处理，进行行为建模，以抽象出用户的标签，这个阶段注重的应是大概率事件，通过数学算法模型尽可能地排除用户的偶然行为。这时也要用到机器学习，对用户的行为、偏好进行猜测，好比一个 y = kx + b 的算法，X 代表已知信息，Y 是用户偏好，通过不断的精确 k 和 b 来精确 Y。在这个阶段，需要用到很多模型来给用户贴标签。

(3) 用户画像基本成型

该阶段可以说是二阶段的一个深入，要把用户的基本属性（年龄、性别、地域）、购买能力、行为特征、兴趣爱好、心理特征、社交网络大致地标签化。因为用户画像永远也无法 100％地描述一个人，只能做到不断地去逼近一个人，因此，用户画像既应根据变化的基础数据不断修正，又要根据已知数据来抽象出新的标签使用户画像越来越立体。

(4) 数据可视化

最后是数据可视化分析，这是把用户画像真正利用起来的一步，在此步骤中一般是针对群体的分析，比如可以根据用户价值来细分出核心用户、评估某一群体的潜在价值空间，以作出针对性的运营。典型的用户画像如图 2.4

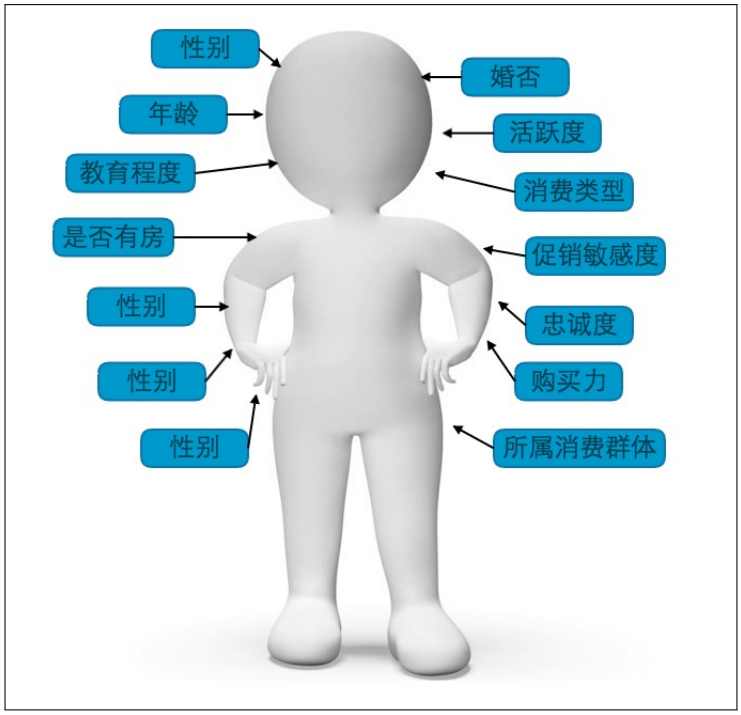

图 **2.4**　用户画像示意图

2.4　本章小结

本章简单概述了推荐系统的主要任务和问题。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，并讨论了推荐系统的主要评测指标。然后介绍了用户画像的研究现状，讨论了相关的建模过程。

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，我们的存在价值就是提供一个平台，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
她们的主题，尤其是对于刚出道的设计师。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的服务接口。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形式。接口层会调用 abtest 配置模块，对接入的流量按照 uuid、城市等维度进行分流量的配置。Abtest 配置模块之下，是推荐候选集的生成，排序和业务处理模块。候选集生成和排序模块，除了针对不同展位有不同逻辑以外，对同一展位的不同策略也有不同的逻辑。abtest 模块在配置流量策略的时候，可以根据需要单独配置候选集策略和排序策略。从接口层接受到的每次响应请求会打印一些必要的日志，记录这次请求的一些必要的上下文信息以及用户及 item 相关的特征信息，以便生成用户行为数据。这些日志通过 flume 传输到 HDFS 上面。借助 Hadoop、Hive、Spark 等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，item based 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，需要根据平台的特点进行做好冷启动策略。这里面需要注意的是，推荐系统引入了时间衰减的因子，从而使新的行为起的作用大于老的行为，从结果来看确实对于效果会有提升。

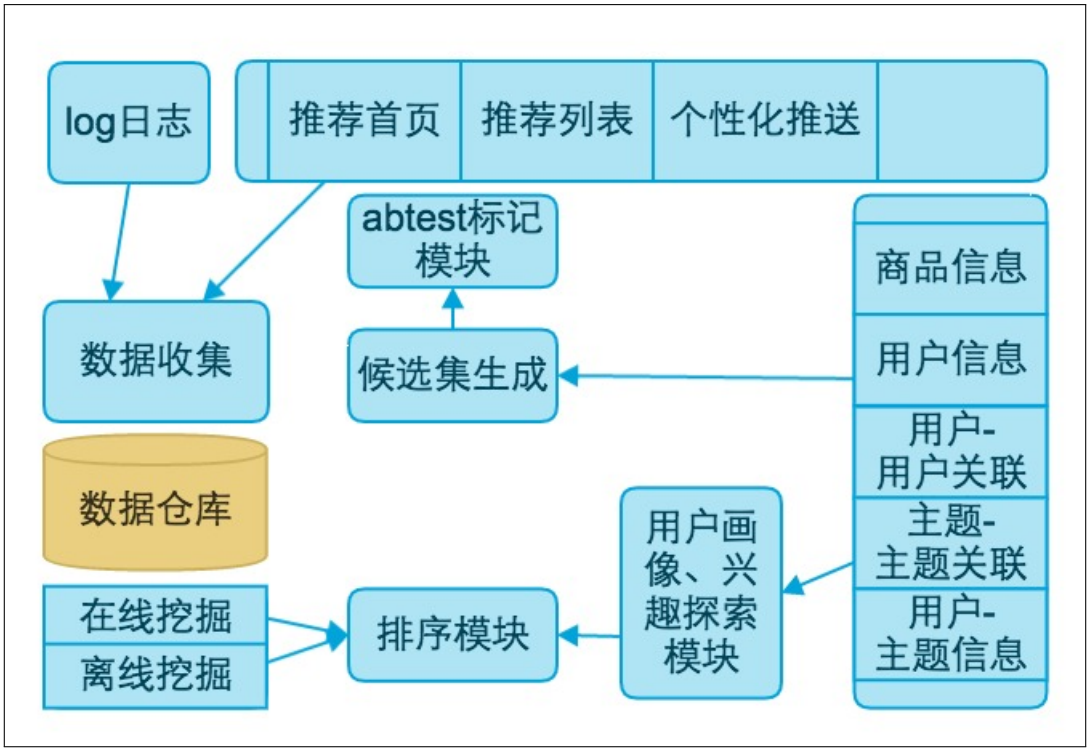

图 **3.1**　推荐系统引擎框架总览图

3.2.1　数据集

我们的数据来源有俩部分。1、主动推送，推送有两个特点，一个是异步，可以在用户没有使用 APP 的时候，将消息推送给他，所以可以作为用户召回的一种手段；另一个是快且实时，因此它也是提高用户活跃度的一种方式。主动推送能够确保优秀的新主题包的时候非常及时的到达用户。2、被动响应，就是用户打开应用，跳转到主题推荐页面，这时才会给用户做推荐。这俩种数据来源都包含着用户行为数据，而用户行为数据是我们的手机主题推荐系统主要驱动数据。用户行为数据包括俩类：隐式反馈数据和显示反馈数据，隐式数据包括用户点击、浏览、搜索关键字等，显示数据主要是用户点赞和评星，显示数据在推荐算法中的权重占比很大，因为如果一个用户给一个主题评星为 5 星，我们就知道用户确确实实是喜欢这个主题，但是如果用户仅仅是浏览了一款主题，我们是没有办法知道用户真实的想法。通过我们的统计发现，显示数据大约占了 1.2% 的比重，隐式数据占了 98.8% 的比重，所以我们的推荐系统设计、实现主要是基于隐式用户数据的。

3.2.2　候选集的生成

通过用户历史行为数据生成推荐列表，我们把相似的主题包放在候选集中。我们主要利用 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。通过 A/B 测试我们得知，根据用户最近行为作出的推荐比根据用户之前行为作出的推荐，点击转换率比例为 1.8:1，因此如果用户最近行为和之前行为有冲突，推荐系统应该倾向于根据用户最近行为作出推荐，而不是反过来。

3.2.3　排序

排序之前先过滤掉用户之前接触过的主题包，对于剩余的主题，我们利用生成候选集时得到的用户-主题相关度对主题排序，作为最终推荐结果的重要依据，需要注意的是确保每种类型的主题都要或多或少的推荐一俩款，即保持推荐多样性。最后，推荐结果会对用户给出推荐理由，如一个用户之前买过一款《似水流年》的主题包，我们给他推荐了一款《青春不老我们不散》的主题，给出的解释是：因为您之前购买过《似水流年》

3.3　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。

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。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问题有：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

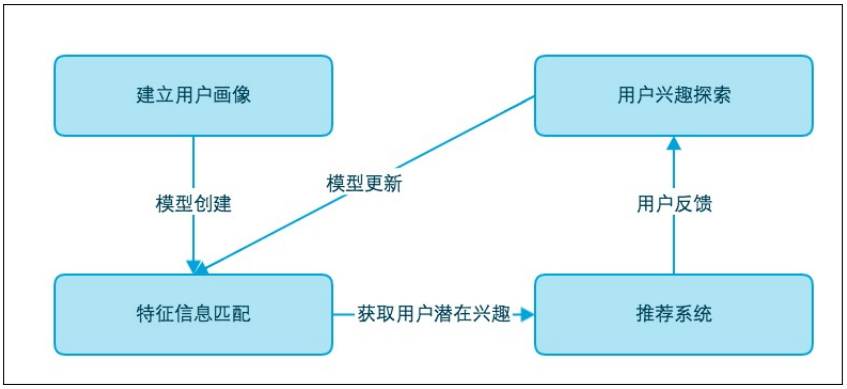

图 **3.2**　用户画像架构示意图

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构成包括：

- 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记录了用户的浏览行为和内容。

手机主题推荐系统对每个新注册用户会生成一个用户画像，刚开始只是包含最基本的用户人口信息，在维护过程中会逐渐增加用户行为和行为偏好，然后利用离线训练模型生成用户喜好的主题，详细内容将在第四章、第五章展开。

3.4　量化评估推荐系统

3.5　本章小结

我们的手机主题推荐系统还遇到一种情况是，有时候推荐时，排序较高的主题不一定是用户需要的，而排序较低的主题有可能是用户期望看到的。例如，一个用户喜欢读弗洛伊德的《梦的解析》，那么他有可能会喜欢《异梦空间》这款主题，但是这款主题很冷门，推荐系统没办法挖掘出其价值。再比如，像《似水流年》这样常月霸占免费排行榜第一、第二位置的热门主题包，我们实际上是

不需要对其作任何推广、推荐的。除此之外，我们也希望手机推荐系统能推陈出新，而不是一成不变。为了解决这些问题，本人基于手机主题推荐系统开发了用户画像功能模块和用户兴趣功能模块。

第四章　用户画像模块

4.1　引言

简而言之，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。

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。从手机主题应用的业务特点来讲，构建用户画像的目的包括：

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、成熟的用户画像应该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，格式如

```
{"registerLog": {
  "userId": "001",
  "gender": "male",
  "profession": "student",
  "phone": "null",
```

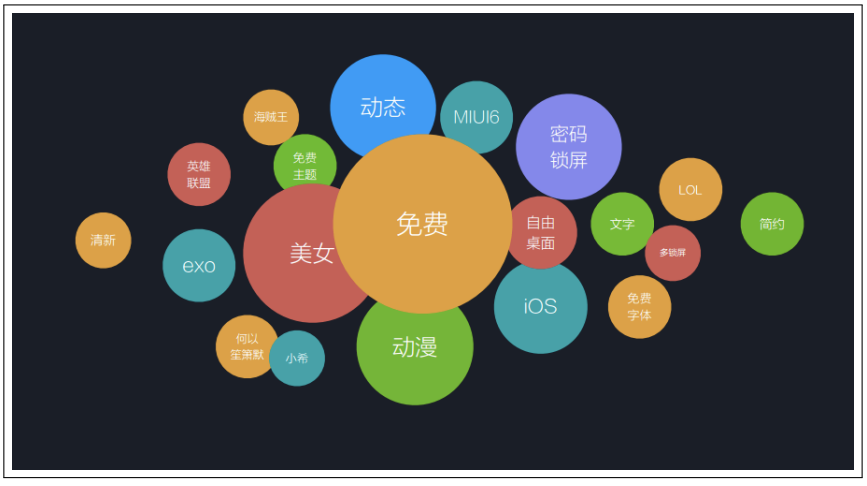

图 4.1　用户画像标签示例图

```
        "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:

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

```
      {"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:

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	...
...	...	...	...	...	...	...	...

4.2.3　高维数据类型

高维数据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：

```
    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

基础行为数据建模跟新频率较快，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：

```
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：

```
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。

4.4.1　数据集准备

手机主题应用月新注册用户超过 20 万个用户，大部分用户的第一个月的行为记录少于 10 个，我们从 2015 年 9 月 1 号到 2015 年 9 月 7 号这段时间，筛选出所有注册信息相对完整的用户数据作为实验数据集，create table 格式：

```
1   {
2   //静态数据
3   user_id             int   comment '用户id',
4   user_name           int   comment '用户name',
5   user_age            int   comment '用户age',
6   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'
    }
```

4.4.2　评测指标

本节使用线上 A/B 测试方案 [33]，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，这里的单位时间取天，实验时间区间为 2015 年 9 月 7 号到 2015 年 9 月 30 号。用户留存率研究对象为新注册用户，反映了推荐系统的转换能力，即由初期的不稳定的用户转化为活跃、稳定、忠诚的用户。

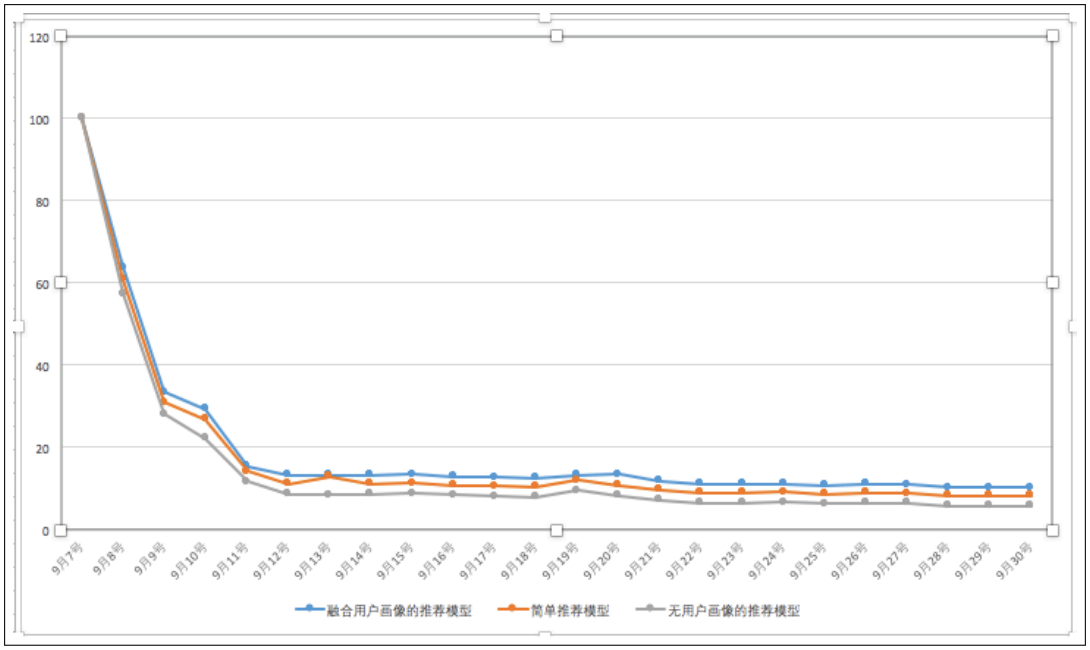

图 **4.2**　新用户留存率实验对比图

4.4.3　对比模型

基准模型为融合了用户画像的推荐模型，对照模型为单纯的推荐模型和推荐热门商品的简单推荐模型。每个推荐模型分流 10% 的用户流量，推荐算法使用了开源软件 spark MLlib 的 LogisticRegressionWithLBFGS 模块，前俩个模型的推荐候选集为全部主题，简单推荐模型的推荐候选集为 Top 20% 热度的主题。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 2015 年 9 月新注册用户数据集上的用户留存率。图 4.2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，实验结果显示，融合了用户画像的推荐模型相对其他模型有更高的留存率。截止到 2015 年 9 月 30 号，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，则 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，这个时间衰减函数被设计成、的形式，通过定义衰减幅度和周期，调节衰减的程度，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。

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 2003 年 9 月到 2003 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

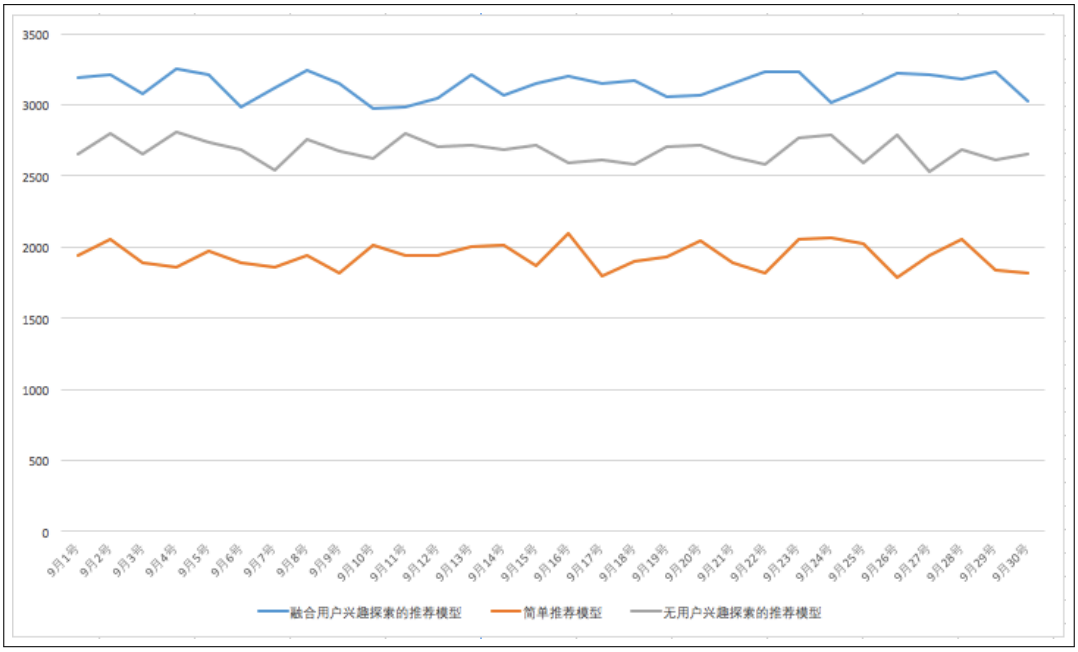

图 **5.1**　推荐多样性实验对比图

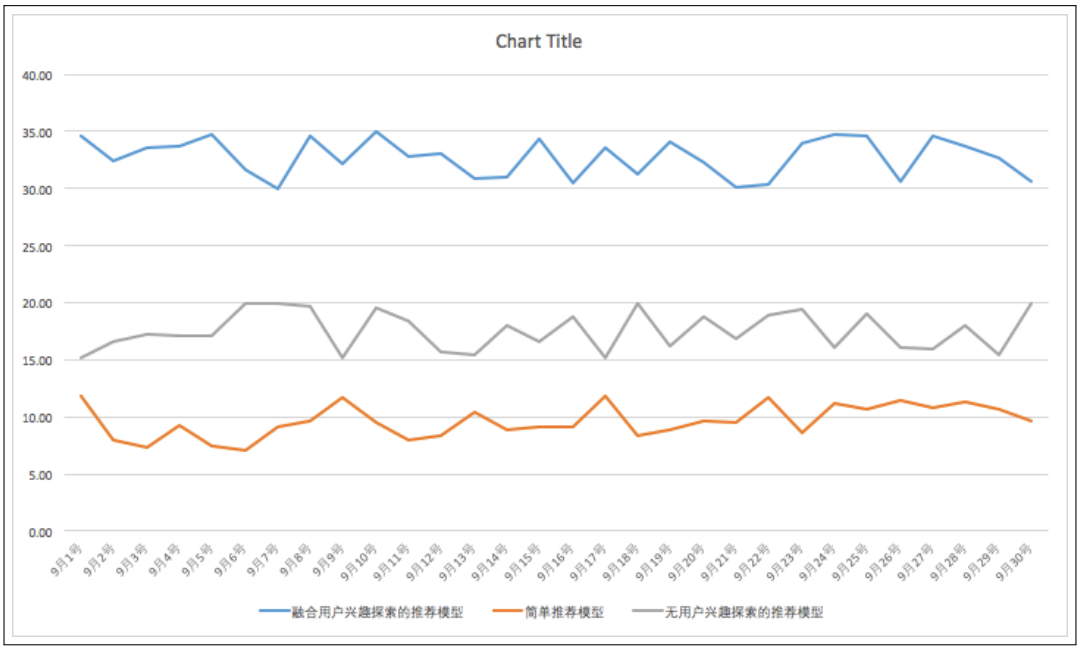

图 **5.2**　转化率实验对比图

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] 周涛.2011. 基于内容的推荐算法. http://blog.sciencenet.cn/blog-3075-459442.html.

[2] 36 大数据.2014. 大数据在京东的典型应用：京东用户画像技术曝光. http://www.36dsj.com/archives/16090.

[3] 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.

[4] From Wikipedia, the free encyclopedia. *Collaborative filtering*. https://en.wikipedia.org/wiki/Collaborative_filtering#Types.

[5] 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.

[6] 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.

[7] 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.

[8] 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.

[9] 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.

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

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

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

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

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

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

[16] 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.

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

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

[19] 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.

[20] 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.

[21] 项亮. 2012. 推荐系统实践. 图灵原创, 人民邮电出版社, 36:5–21.

[22] 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.

[23] 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.

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

[25] 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.

[26] 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,
[27] Han Jiawei, Kamber, Micheline.2001. *Data mining: concepts and techniques[C]*. Morgan Kaufmann. 5.
[28] 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)
[29] O Celma. 2010. *Music Recommendation and Discovery in the Long Tail[C]*. Springer.
[30] 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.
[31] 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.
[32] 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.
[33] Kohavi, Ron, Longbotham, Roger.2015. *Online Controlled Experiments and A/B Tests[C]*. In Sammut.
[34] Robin Burke. November 2002. *Hybrid recommender systems: Survey and experiments*. User Modeling and User-Adapted Interaction, 12:331–370.
[35] Gediminas Adomavicius and Alexander Tuzhilin. 1999. *User Profiling in Personalization Applications through Rule Discovery and Validation*. ACM, 377-381.
[36] lbrahim Cingil, Asuman Dogac and Ayca Azgin.2000. *A broader approach to personalization*. mmunications of the ACM, 43(8): 136-141.
[37] Joseph Kramer, Sunil Noronha and John Vergo.2000. *A user-centered design approach to personalization*. Communications of the ACM, 43(8)44-48.
[38] 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.
[39] Bamshad Mobasher, Robert Cooley and Jaideep Srivastava.2000. *Automatic personalization based on Web usage mining*. Communications of the ACM, 43(8): 142-151.
[40] 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.
[41] 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.
[42] 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,
[43] 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,

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，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

2016 年 4 月 23 日