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 [2]。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 [4] 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。

而近年来淘宝的交易额增长规模巨大，2005 年淘宝交易额为 80 亿，2010 年

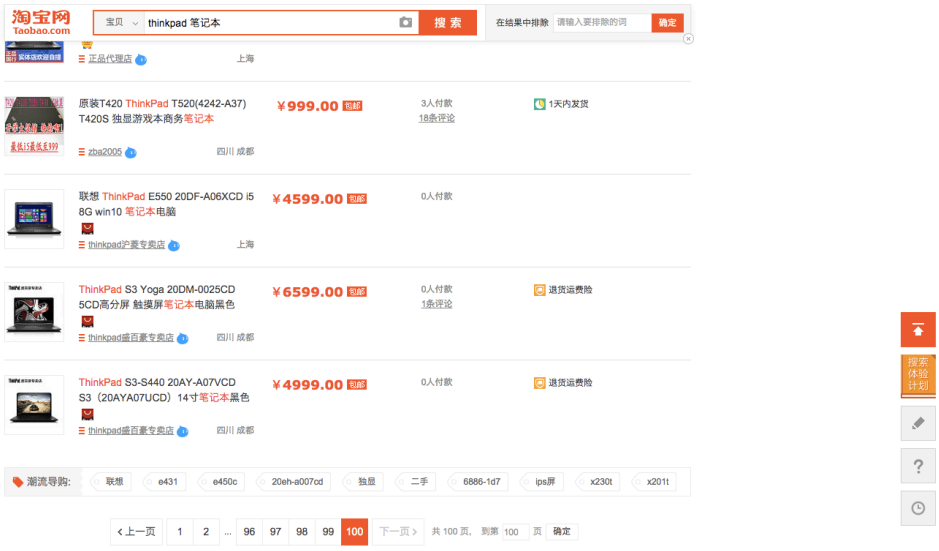

图 1.1　淘宝购物搜索图

为 4000 亿，而到 2015 年淘宝双十一单日交易额就为 912 亿元，可见未来几年内笔者的这种关键字搜索 + 逐条浏览的购物方式已经不再具有可行性。而作为提供商，如何让自己生产的信息不埋没在大数据洪流中而受到潜在用户的充分关注，这也是其所要解决的一个课题，很多企业已经或者正在开发适合本公司的推荐系统 (Recommender System) 来解决这一矛盾。

推荐系统广泛应用于电子商务领域，通过分析用户的数据，帮助用户找到喜欢和感兴趣的商品，然后推荐给他们。推荐系统的最大优点在于它能收集用户的兴趣信息并根据用户的不同偏好，主动的为用户做出个性化推荐，而且此推荐信息是动态更新的，也就是说随着时间的推移，用户的兴趣在逐渐改变，推荐系统的推荐结果也会随之改变。因此，推荐系统大大的提高了网站的用户体验，方便了用户对资源信息的查询。推荐系统的主要任务就是联系用户和信息，一方面协助用户发现自己潜在感兴趣的信息从而提升用户的满意度，另一方面让信息针对性的展现在只对它有兴趣的用户面前从而提升商品的转化率，于是实现了消费者和生产者的双赢。

1.1.1　推荐系统的定义

推荐系统的研究和很多早期的研究相关，比如认知科学 (cognitive science)[5]，信息检索 (information retrieval) 和预测理论 [6]。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 [7]。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 [8]。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。

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

1.1.2　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 [9, 10]。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 [11] 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[3] 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。

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 GMV（Gross Merchandise Volume，即年度成交总额）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 SVD（Sigular Value Decomposition，即奇异值分解，一种正交矩阵分解法）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（又称基于记忆的方法）、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商品的冷启动问题，但是解决不了用户的冷启动问题，并且存在过拟合问题（往往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，并且对行为丰富的商品，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 RMSE（Root Mean Squared Error，即均方根误差）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。

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)

1.1.3　推荐系统的作用

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。

1.1.4 推荐系统与电子商务

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 [12] 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。

1.2　大数据时代下的推荐系统

虽然推荐系统已经被成功运用在很多大型系统、网站，但是在当前大数据的时代下，推荐系统的面临的场景越来越复杂，推荐系统不仅需要解决传统的数据稀疏、冷启动和动态兴趣问题，还面临由大数据引发的更多、更复杂的实际问题，例如数以亿计的用户数目和海量用户同时访问推荐系统所造成的性能压力，使传统的基于单节点架构的推荐系统不再适用。同时 Web 服务器处理系统请求在大数据集下变得越来越多，Web 服务器响应速度缓慢制约了当前推荐系统为大数据集提供推荐。基于实时模式的推荐在大数据集下也面临着严峻考验，用户难以忍受超过秒级的推荐结果返回时间。传统推荐系统的单一数据库存储技术在大数据集下变得不再适用，急需一种对外提供统一接口、对内采用多种混合模式存储的存储架构来满足大数据集下各种数据文件的存储。并且传统推荐系统在推荐算法上采取的是单机节点的计算方式也不能满足海量用户行为数据的计算需求。大数据本身具有的复杂性、不确定性也给推荐系统带来诸多新的挑战，传统推荐系统的时间效率、空间效率和推荐准确度都遇到严重的瓶颈。

1.2.1　推荐系统的关键技术

分布式文件系统。传统的推荐系统技术主要处理小文件存储和少量数据计算，大多是面向服务器的架构，中心服务器需要收集用户的浏览记录、购买记录、

评分记录等大量的交互信息来为单个用户定制个性化推荐。当数据规模过大，数据无法全部载入服务器内存时，就算采用外存置换算法和多线程技术，依然会出现 I/O 上的性能瓶颈，致使任务执行效率过低，产生推荐结果的时间过长。对于面向海量用户和海量数据的推荐系统，基于集中式的中心服务器的推荐系统在时间和空间复杂性上无法满足大数据背景下推荐系统快速变化的需求。大数据推荐系统采用基于集群技术的分布式文件系统管理数据。建立一种高并发、可扩展、能处理海量数据的大数据推荐系统架构是非常关键的，它能为大数据集的处理提供强有力的支持。Hadoop 的分布式文件系统架构是其中的典型。与传统的文件系统不同，数据文件并非存储在本地单一节点上，而是通过网络存储在多台节点上。并且文件的位置索引管理一般都由一台或几台中心节点负责。客户端从集群中读写数据时，首先通过中心节点获取文件的位置，然后与集群中的节点通信，客户端通过网络从节点读取数据到本地或把数据从本地写入节点。在这个过程中由 HDFS 来管理数据冗余存储、大文件的切分、中间网络通信、数据出错恢复等，客户端根据 HDFS 提供的接口进行调用即可，非常方便。

分布式计算框架。集群上实现分布式计算的框架很多，Spark 作为推荐算法并行化的依托平台，既是一种分布式的计算框架，也是一种新型的分布式计算编程模型，是一种常见的开源计算框架。其基于内存的 MapReduce 算法的核心思想是分而治之，把对大规模数据集的操作，分发给一个主节点管理下的各个分节点共同完成，然后通过整合各个节点的中间结果，得到最终结果。计算框架负责处理并行编程中分布式存储、工作调度、负载均衡、容错均衡、容错处理以及网络通信等复杂问题，把处理过程高度抽象为两个函数：map 和 reduce。map 负责把任务分解成多个任务，reduce 负责把分解后多任务处理的结果汇总起来。

推荐算法并行化。大型企业所需的推荐算法要处理的数据量非常庞大，从 TB 级别到 PB 级甚至更高，腾讯 Peacock 主题模型分析系统需要进行高达十亿文档、百万词汇、百万主题的主题模型训练，仅一个百万词汇乘以百万主题的矩阵，其数据存储量已达 3TB。面对如此庞大的数据，若采用传统串行推荐算法，时间开销太大。当数据量较小时，时间复杂度高的串行算法能有效运作，但数据量极速增加后，这些串行推荐算法的计算性能过低，无法应用于实际的推荐系统中。因此，面向大数据集的推荐系统从设计上就应考虑到算法的分布式并行化技术，使得推荐算法能够在海量的、分布式、异构数据环境下得以高效实现。

1.2.2　推荐系统算法简介

现有的推荐算法类型很多，但是各有各的局限，因此推荐系统经常采用组合推荐算法，即融合了协同过滤推荐、聚类算法和其他算法的组合推荐算法。

(1) 协同过滤算法。

利用用户的历史喜好信息计算用户之间的距离，然后利用目标用户的最近邻居用户对评价的加权评价值来预测目标用户对特定手机主题的喜好程度，系

统从而根据这一喜好程度来对目标用户进行推荐。协同过滤是基于这样的假设：为一用户找到他真正感兴趣的内容的好方法是首先找到与此用户有相似兴趣的其他用户，然后将他们感兴趣的内容推荐给此用户。协同过滤正是把这一思想运用到手机推荐系统中来，基于其他用户对某一类手机主题的评价来向目标用户进行推荐。基于协同过滤的推荐系统可以说是从用户的角度来进行相应推荐的，而且是自动的，即用户获得的推荐是系统从购买模式或浏览行为等隐式获得的，不需要用户努力地找到适合自己兴趣的推荐信息，如填写一些调查表格等。

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

$$\mathrm{similarity}_{uv} = \frac{|N(u) \cdot N(v)|}{||N(u)|| \cdot ||N(v)||} \tag{1.2}$$

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

(2) 聚类算法

聚类分析是对于统计数据分析的一门技术，和分类算法一个主要的区别就是聚类不需要人工参与打标签，基于聚类的协同过滤方法，也可以在一定程度上解决传统协同过滤算法用户评分矩阵稀疏和冷启动问题，在降低用户评分

矩阵稀疏性的同时提高目标用户最近邻居的查询速度。聚类是把相似的对象通过静态分类的方法分成不同的组别或者更多的子集，这样让在同一个子集中的成员对象都有相似的一些属性，聚类结果不仅可以揭示数据间的内在联系与区别，还可以为进一步的数据分析与知识发现提供重要依据。在结构性聚类中关键性的一步就是要选择测量的距离。一个简单的测量就是使用曼哈顿距离，它相当于每个变量的绝对差值之和。该名字的由来起源于在纽约市区测量街道之间的距离就是由人步行的步数来确定的。聚类模块可以是对用户兴趣属性相似度做聚类，也可以对用户社交属性相似度做聚类，或者俩种兼有。

在现实社会中人们的兴趣和选择往往受到身边亲朋好友的影响。在互联网中随着诸如国内的腾讯，国外的 Twitter 等社会网络网站的兴起，如何利用用户的社会属性做推荐是近几年推荐领域比较热门的研究问题。基于社会网络的推荐算法被称为社会化推荐。近几年在工业界已经有了很多社会化推荐系统。最简单的社会化过滤算法是基于邻域的算法。给定用户 u, 令 F(u) 为用户 u 的好友集合,N(u) 为用户 u 喜欢的物品集合。那么用户 u 对物品 i 的喜好程度定义为用户 u 的好友中喜欢物品 i 的好友个数，如公式 1.3。

$$P_{vi} = \sum_{v \in F(v) i \in N(u)} 1 \tag{1.3}$$

聚类算法在许多领域受到广泛应用，包括机器学习，数据挖掘，模式识别，图像分析以及生物信息，最常用的 k-means 算法 [23] 表示以空间中 k 个点为中心进行聚类，对最靠近他们的对象归类。

(3) 基于内容的推荐算法。

基于内容的推荐是信息过滤技术的延续与发展，它是建立在对手机主题的标签信息上作出推荐的，而不需要依据用户对手机主题的评价意见，需要用机器学习的方法从关于内容的特征描述的事例中得到用户的兴趣资料。手机主题是通过相关的特征的属性来定义，系统基于用户评价对象的特征，学习用户的兴趣，考察用户资料与待预测手机主题的相匹配程度。用户的资料模型取决于所用学习方法，采用了综合决策树、神经网络和基于向量的组合方法。基于内容的用户资料是需要有用户的历史数据，用户资料模型可能随着用户的偏好改变而发生变化。基于内容推荐方法的优点是：不需要其它用户的数据，没有冷开始问题和稀疏问题。能为具有特殊兴趣爱好的用户进行推荐。能推荐新的或不是很流行的手机主题，没有产品问题。通过列出推荐手机主题的内容特征，可以解释为什么推荐那些手机主题。

本节利用 spark mllib 中 ALS 算法解释基于内容的推荐。首先，给出一个 (用户，主题，评分) 三元组的数据集,ALS 会建立一个 user∗product 的 m∗n 的矩

阵，其中，m 为用户的数量，n 为商品的数量。这个矩阵的每一行代表一个用户 $(u_1,u_2,\cdots,u_9)$、每一列代表一个产品 $(v_1,v_2,\cdots,v_9)$。用户的打分在 0 到 10 之间。但是在这个数据集中，并不是每个用户都对每个产品进行过评分，所以这个矩阵往往是稀疏的，所以需要做预处理将其填满，然后开始训练：假设 m∗n 的评分矩阵 R，可以被近似分解成 $U*V^T$，U 为 m∗d 的用户特征向量矩阵，V 为 n∗d 的产品特征向量矩阵，d 为用户和商品的特征值的数量。

$$\begin{Bmatrix} & u_1 & u_2 \\ p_1 & 8 & 7 \\ p_2 & 44 & 39 \end{Bmatrix} = \begin{Bmatrix} & f_1 & f_2 \\ p_1 & 0 & 1 \\ p_2 & 2 & 3 \end{Bmatrix} * \begin{Bmatrix} & u_1 & u_2 \\ p_1 & 10 & 9 \\ p_2 & 8 & 7 \end{Bmatrix}$$

对于电影类型的手机主题，可以从 d 个角度进行评价，如主角，铃声，背景，特效 4 个角度来评价，那么 d 就等于 4。矩阵 V 由 n 个 product∗d 个特征值组成。对于矩阵 U，假设对于任意的用户 A，该用户对一款手机主题的综合评分和主题的特征值存在一定的线性关系，综合评分 =(a1*d1+a2*d2+a3*d3+a4*d4)，其中 a_i 为用户 A 的特征值，d_i 为之前所说的主题的特征值。ALS 算法认为 m*n 的评分矩阵 R 可以被近似分解成 $U*V^T$，得到目标函数：

$$L(U,V)=\sum_{i,j}(R_{ij}-U_i^TV_j)^2 \tag{1.4}$$

其中 a 表示评分数据集中用户 i 对产品 j 的真实评分，另外一部分表示用户 i 的特征向量和产品 j 的特征向量，加上正则化参数 $\lambda(\|U_i\|^2+\|V_j\|^2)$ 以防止过度拟合，固定 V 对 U 求导得到公式：

$$U_t=R_tV_{ut}(V_{ut}^TV_{ut}+\lambda n_{ut}I)^{-1}, i\in[1,m] \tag{1.5}$$

其中 R_t 表示用户 i 评过的手机主题的评分向量，V_{ut} 表示用户 i 评过的手机主题的特征向量组成的特征矩阵。n_{ut} 表示用户 i 评过的手机主题数量。同理，固定 U，可以得到求解 V_j 的公式：

$$V_j=R_j^TU_{mj}(U_{mj}^TU_{mj}+\lambda n_{mj}I)^{-1} \tag{1.6}$$

R_j 表示评过手机主题 j 的用户向量，U_{mj} 表示评过手机主题 j 的用户特征向量组成的矩阵，m_{mj} 表示评过电影 j 的用户的数量。

首先用一个小于 1 的随机数初始化 V，根据式 1.5求 U，此时就可以得到初始的 UV 矩阵了，根据计算得到的 U 和式 1.6重新计算并覆盖 V，反复进行以上两步的计算，直到目标函数和小于一个预设的值，或者迭代次数满足要求则停止。

(4) 组合推荐。

由于各种推荐方法都有优缺点，手机主题推荐采用了组合推荐方式。研究和应用最多的是基于内容的推荐和协同过滤推荐的组合。最简单的做法就是分别用基于内容的方法和协同过滤推荐方法去产生一个推荐预测结果，然后用某方法组合其结果。组合推荐一个最重要原则就是通过组合后要能避免或弥补各自推荐技术的弱点。在组合方式上使用了几种组合思路：加权（Weight）：加权多种推荐技术结果。变换（Switch）：根据问题背景和实际情况或要求决定变换采用不同的推荐技术。混合（Mixed）：同时采用多种推荐技术给出多种推荐结果为用户提供参考。特征组合（Feature combination）：组合来自不同推荐数据源的特征被另一种推荐算法所采用。层叠（Cascade）：先用一种推荐技术产生一种粗糙的推荐结果，第二种推荐技术在此推荐结果的基础上进一步作出更精确的推荐。特征扩充（Feature augmentation）：一种技术产生附加的特征信息嵌入到另一种推荐技术的特征输入中。

1.2.3 推荐系统面临的问题

(1) 特征提取问题。

推荐系统的推荐对象种类丰富，例如新闻、博客等文本类对象，视频、图片、音乐等多媒体对象以及可以用文本描述的一些实体对象等。如何对这些推荐对象进行特征提取一直是学术界和工业界的热门研究课题。对于文本类对象，可以借助信息检索领域已经成熟的文本特征提取技术来提取特征。对于多媒体对象, 由于需要结合多媒体内容分析领域的相关技术来提取特征, 而多媒体内容分析技术目前在学术界和工业界还有待完善, 因此多媒体对象的特征提取是推荐系统目前面临的一大难题。此外推荐对象特征的区分度对推荐系统的性能有非常重要的影响。目前还缺乏特别有效的提高特征区分度的方法。

(2) 数据稀疏问题。

现有的大多数推荐算法都是基于用户—物品协同过滤矩阵数据, 数据的稀疏性问题主要是指用户—物品评分矩阵的稀疏性，即用户与物品的交互行为太少。一个大型网站可能拥有上亿数量级的用户和物品，用户评分数据总量在面对增长更快的“用户—物品评价矩阵”时, 仍然表现出稀疏性, 推荐系统研究中的经典数据集 MovieLens 的稀疏度仅 4.5%，Netflix 百万大赛中提供的音乐数据集的稀疏度是 1.2%。这些都是已经处理过的数据集，实际上真实数据集的稀疏度都远远低于 1%。例如，Bibsonomy 的稀疏度是 0.35%，Delicious 的稀疏度是 0.046%，淘宝网数据的稀疏度甚至仅在 0.01% 左右。根据经验，数据集中用户行为数据越多，推荐算法的精准度越高，性能也越好。若数据

集非常稀疏，只包含极少量的用户行为数据，推荐算法的准确度会大打折扣，极容易导致推荐算法的过拟合，影响算法的性能。

(3) 冷启动问题。

冷启动问题是推荐系统所面临的最大问题之一。冷启动问题总的来说可以分为 3 类：系统冷启动问题、新用户问题和新物品问题。系统冷启动问题指的是由于数据过于稀疏，“用户—物品评分矩阵”的密度太低，导致推荐系统得到的推荐结果准确性极低。新物品问题是由于新的物品缺少用户对该物品的评分，这类物品很难通过推荐系统被推荐给用户，用户难以对这些物品评分，从而形成恶性循环，导致一些新物品始终无法有效推荐。新物品问题对不同的推荐系统影响程度不同：对于用户可以通过多种方式查找物品的网站，新物品问题并没有太大影响，如电影推荐系统等，因为用户可以有多种途径找到电影观看并评分；而对于一些推荐是主要获取物品途径的网站，新物品问题会对推荐系统造成严重影响。通常解决这个问题的途径是激励或者雇佣少量用户对每一个新物品进行评分。新用户问题是目前对现实推荐系统挑战最大的冷启动问题：当一个新的用户使用推荐系统时，他没有对任何项目进行评分，因此系统无法对其进行个性化推荐；即使当新用户开始对少量项目进行评分时，由于评分太少，系统依然无法给出精确的推荐，这甚至会导致用户因为推荐体验不佳而停止使用推荐系统。当前解决新用户问题主要是通过结合基于内容和基于用户特征的方法，掌握用户的统计特征和兴趣特征，在用户只有少量评分甚至没有评分时做出比较准确的推荐。

(4) 马太效应。

马太效应（Mattnew Effect）是指强者愈强、弱者愈弱的现象，在互联网中引申为热门的产品受到更多的关注，冷门内容则愈发的会被遗忘的现象。很不幸的是推荐系统的出现加剧了互联网商品的马太效应，因为很多商品只有很少的评分，因此很难在推荐系统中应用，导致推荐结果大部分为热门商品。与马太效应相对于的是长尾理论，由美国人克里斯·安德森提出。长尾理论认为，由于成本和效率的因素，当商品储存流通展示的场地和渠道足够宽广，商品生产成本急剧下降以至于个人都可以进行生产，并且商品的销售成本急剧降低时，几乎任何以前看似需求极低的产品，只要有卖，都会有人买。这些需求和销量不高的产品所占据的共同市场份额，可以和主流产品的市场份额相比，甚至更大。

1.2.4 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。

1.2.5　推荐系统的应用案例

近几年随着社会化网络的发展，推荐系统在工业界广泛应用并且取得了显著进步。比较著名的推荐系统应用有：淘宝网的电子商务推荐系统、Youtube 的视频推荐系统 [1]、网易云音乐推荐系统以及 Facebook 好友推荐系统。个性化推荐系统具有良好的发展和应用前景。目前，几乎所有的大型电子商务系统，如 Amazon、eBay 等，都不同程度的使用了各种形式的推荐系统。各种提供个性化服务的 Web 站点也需要推荐系统的大力支持。在日趋激烈的竞争环境下，个性

化推荐系统能有效的保留客户，提高电子商务系统的服务能力。成功的推荐系统会带来巨大的效益。我们每天使用的许多网站中都可找到推荐系统。

作为全球排名第一的社交网站 (https://code.facebook.com/)，Facebook 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，代表了国外推荐系统的最高发展水平。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。Facebook 设计了一个全新的推荐系统。Facebook 团队之前已经在使用一个分布式迭代和图像处理平台——Apache Giraph。因其能够很好的支持大规模数据，Giraph 就成为了 Facebook 推荐系统的基础平台。在工作原理方面，Facebook 推荐系统采用的是流行的协同过滤技术。CF 技术的基本思路就是根据相同人群所关注事物的评分来预测某个人对该事物的评分或喜爱程度。从数学角度而言，该问题就是根据用户-物品的评分矩阵中已知的值来预测未知的值。其求解过程通常采用矩阵分解方法。MF 方法把用户评分矩阵表达为用户矩阵和物品的乘积，用这些矩阵相乘的结果 R’来拟合原来的评分矩阵 R，使得二者尽量接近。如果把 R 和 R’之间的距离作为优化目标, 那么矩阵分解就变成了求最小值问题。对大规模数据而言，求解过程将会十分耗时。为了降低时间和空间复杂度，一些从随机特征向量开始的迭代式算法被提出。这些迭代式算法渐渐收敛，可以在合理的时间内找到一个最优解。随机梯度下降算法就是其中之一，其已经成功的用于多个问题的求解。SGD 基本思路是以随机方式遍历训练集中的数据，并给出每个已知评分的预测评分值。用户和物品特征向量的调整就沿着评分误差越来越小的方向迭代进行，直到误差到达设计要求。因此，SGD 方法可以不需要遍历所有的样本即可完成特征向量的求解。交替最小二乘法是另外一个迭代算法。其基本思路为交替固定用户特征向量和物品特征向量的值，不断的寻找局部最优解直到满足求解条件。

为了利用上述算法解决 Facebook 推荐系统的问题，原本 Giraph 中的标准方法就需要进行改变。之前，Giraph 的标准方法是把用户和物品都当作为图中的顶点、已知的评分当作边。那么，SGD 或 ALS 的迭代过程就是遍历图中所有的边，发送用户和物品的特征向量并进行局部更新。该方法存在若干重大问题。首先，迭代过程会带来巨大的网络通信负载。由于迭代过程需要遍历所有的边，一次迭代所发送的数据量就为边与特征向量个数的乘积。假设评分数为 1000 亿、特征向量为 100 对，每次迭代的通信数据量就为 80TB。其次，物品流行程度的不同会导致图中节点度的分布不均匀。该问题可能会导致内存不够或者引起处理瓶颈。假设一个物品有 1000 亿个评分、特征向量同样为 100 对，该物品对应的一个点在一次迭代中就需要接收 80GB 的数据。最后，Giraph 中并没有完全按照公式中的要求实现 SGD 算法。真正实现中，每个点都是利用迭代开始时实际收到的特征向量进行工作，而并非全局最新的特征向量。因此 Giraph 中最大的问题就在于每次迭代中都需要把更新信息发送到每一个顶点。为了解决这个问题，

Facebook 发明了一种利用 work-to-work 信息传递的高效、便捷方法。该方法把原有的图划分为了由若干 work 构成的一个圆。每个 worker 都包含了一个物品集合和若干用户。在每一步，相邻的 worker 沿顺时针方法把包含物品更新的信息发送到下游的 worker。这样，每一步都只处理了各个 worker 内部的评分，而经过与 worker 个数相同的步骤后，所有的评分也全部都被处理。该方法实现了通信量与评分数无关，可以明显减少图中数据的通信量。而且，标准方法中节点度分布不均匀的问题也因为物品不再用顶点来表示而不复存在。为了进一步提高算法性能，Facebook 把 SGD 和 ALS 两个算法进行了揉合，提出了旋转混合式求解方法。

接下来，Facebook 在运行实际的 A/B 测试之间对推荐系统的性能进行了测量。首先，通过输入一直的训练集，推荐系统对算法的参数进行微调来提高预测精度。然后，系统针对测试集给出评分并与已知的结果进行比较。Facebook 团队从物品平均评分、前 1/10/100 物品的评分精度、所有测试物品的平均精度等来评估推荐系统。此外，均方根误差（Root Mean Squared Error, RMSE）也被用来记录单个误差所带来的影响。

此外，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。其中一种可能的解决方案是采用 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。另外一个能够近似解决问题的方法是根据物品特征向量对物品进行分类。这样，寻找推荐评分就划分为寻找最推荐的物品群和在物品群中再提取评分最高的物品两个过程。该方法在一定程度上会降低推荐系统的可信度，却能够加速计算过程。

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。

1.3　研究内容与研究方法

推荐系统问题之一是冷启动问题，冷启动问题有三种：用户冷启动、物品冷启动、系统冷启动，本文主要研究用户冷启动问题。经典的算法诸如最近邻的协同过滤算法、PageRank 排序算法、关联规则挖掘等算法是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，这种思路对于新注册用户来讲效果不好，因为没有用户行为数据可供分析。解决这个问题的关键是对用户画像建模，实验发现融合用户画像的热门商品推荐是解决冷启动问题的最佳方式。

推荐系统问题之二是马太效应，即热门商品越来越热，冷门商品越来越冷，在互联网指数级的爆发下信息量极大富余，这更加推动了马太效应的快速形成以及规模的无限扩大，现有的大多数推荐算法更是极大地加速马太效应的形成速度以及规模。我们提出利用用户兴趣探索解决商品的马太效应，提升推荐系统对物品长尾的发掘能力，主要思路是分析用户所有的行为数据，针对冷门商品 (冷门商品包含的标签一般是小众标签) 的行为会赋予一个倾斜因子，这样会导致兴趣探索标签候选集中的小众标签占大多数，而如果用户对其的满意度也很高，则说明这是一个成功的兴趣探索。这里涉及到的概念包括小众标签的定义和用户满意度的量化，将会在用户兴趣探索章节详细介绍。

推荐系统问题之三是用户兴趣的动态变化问题，即时效性问题。笔者一直关心的一个问题就是不同系统的用户行为究竟有什么区别，并如何根据这些区别来选择合适的时间参数来预测用户的行为。如 nytimes 的时效性很短，大部分新闻都是在第一天被很多人关注，而后面就没有人关注了，所以即使很热门的新闻，其生命周期比不热门的新闻长不了太久。其次是 blogspot，然后是 youtube，最后是 Wikipedia，根据用户兴趣时效性可得排序：NYTimes > BlogSpot > Youtube > Wikipedia。其中 Wikipedia 的斜率很接近最大理论斜率 (0.5)，这说明 Wikipedia 的热门的东西完全是因为生命周期长所以才热门，而不是因为在某天特别的火过。因此，正确把握用户兴趣变动的时效性对推荐结果影响很大，本文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，其中用户画像代表了用户长期兴趣，用户兴趣探索代表了用户短期兴趣。

1.4　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 [19] 和信息提取技术 [20] 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 [21]，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　手机主题推荐系统分析

2.1　引言

推荐系统的主要任务是给每个用户提供一个候选推荐列表，过程分为两步：1、预测用户会对哪些物品评分，2、预测用户会给该物品什么评分。本章首先介绍推荐系统的目标，然后详细介绍手机主题推荐系统的排序模块，排序模块的本质就是预测用户会对哪些物品做出多少评分。

在推荐过程中会遇到诸如冷启动问题、数据稀疏性问题，引入了用户画像就为解决这些问题：推荐系统首先利用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，然后使用相应的推荐算法进行计算筛选，找到用户可能感兴趣的推荐主题，最后推荐给用户。

除此之外还有一个关注点就是推荐系统的长尾性和时效性，长尾效应对提高商品销售量有非常大的帮助，而推荐的时效性对于用户的体验度也很重要，比较常见的时间效应问题主要反映在用户兴趣的变化、物品流行度的变化以及手机主题的季节效应。用户兴趣探索则是其中比较有效的解决途径之一，因为度量用户对物品的喜好不仅取决于用户的喜好和物品的属性，也取决于用户所处的环境，或者称做上下文。用户在不同的时间可能喜欢不同的物品，物品在不同的时间也有不同的流行度。因此推荐系统应该是一个动态系统，随着时间的变化会给用户不同的推荐结果 [22]。

2.2　手机主题推荐系统引擎模块

2.2.1　推荐系统的目标

手机主题推荐系统的目标，首先是要帮助用户快速找到所需。推荐系统作为手机主题应用平台的重要组成部分，其目标就是为用户快速找到“高品质，低价格”的商品，衡量目标是否实现就看用户看了推荐结果以后的下单转化效果。另外，推荐系统希望用户对其的认知是“无所不有”的大平台，所以也希望推荐出来的结果包含多个类型的结果，即推荐结果有多样性。目前，推荐系统的目标还主要集中在下单转化效果，随着下单率效果的大幅度提高会逐渐把重心转到多样性。

2.2.2　推荐系统框架总览

推荐系统框架总览如图 2.1。

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

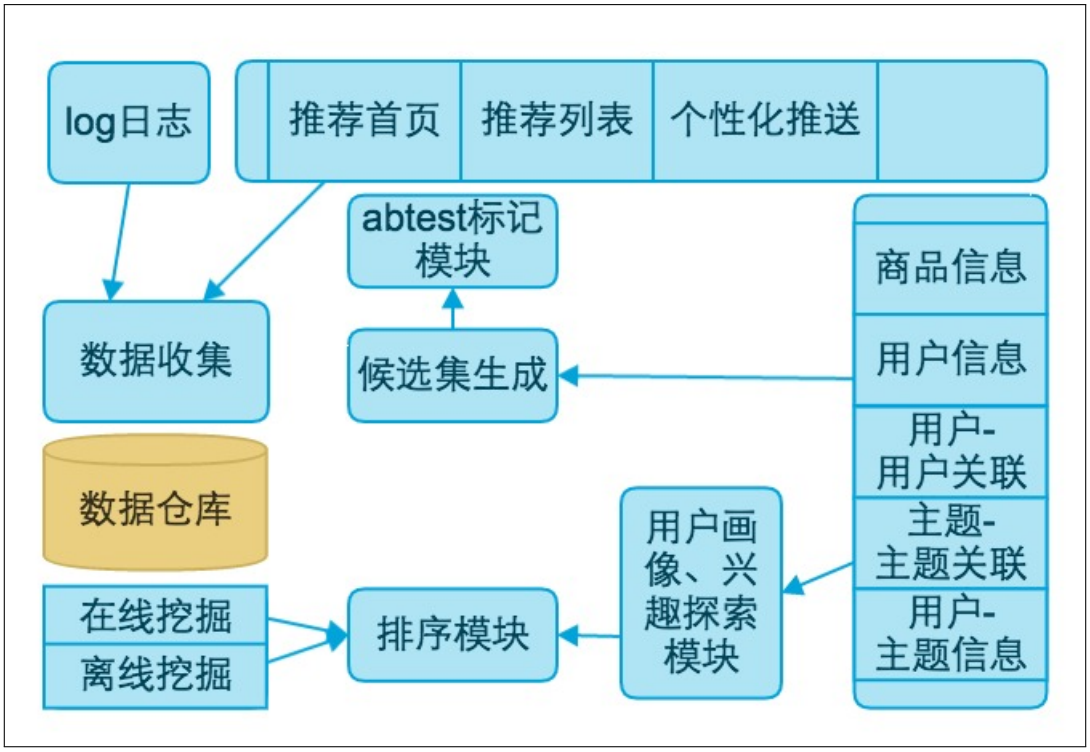

图 **2.1**　推荐系统引擎框架总览图

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。

2.2.3　排序模块

对于推荐系统的效果提高，排序比候选集的贡献要大很多。排序方面所做的主要工作：

(1) 模型及建模

目前的推荐系统的排序模型主要是 Additive Groves 模型。AG 模型是一种决

策树类型的模型，属于非线性模型。这种非线性模型的特点，是一定程度上能够自动进行特征组合的工作，不需要人工进行大量这类工作。建模方法和传统的 ctr 预估建模方法一样，是 point wise 的模型。每一个 item 对一个用户的每次展示可以作为一个样本，这个 item 是否被点击或者是否被下单作为标记。推荐系统会为这些样本抽取一些 item 特征，用户特征，上下文特征，item 与用户的交叉特征。

(2) 样本采样及 label 处理

由于推荐系统的最终目标是提高 item 的下单转化效果，所以需要重点采用用户下单行为作为标记。但是如果只用下单行为，又会导致数据较为稀疏，有很大比例的用户很长时间内是没有下单行为的。所以我们还需要使用点击行为作为标记。而对点击行为和下单行为对于训练目标的价值是不一样的，对它们需要做不同的处理。推荐系统尝试了 2 种方式，在参数取得比较合适的情况下，二者的结果效果都很好。一种方式是提高下单样本的采样比例。一种方式是提高标记值。比如下单行为的标记值为 30，点击行为的标记值为 1。

(3) 去除 position bias

item 在展示列表中的位置，对 item 的点击概率和下单概率是有非常大影响的，排名越靠前的 item，越容易被点击和下单，这就是 position bias 的含义。在抽取特征和训练模型的时候，就需要很好去除这种 position bias。推荐系统在两个地方做这种处理：一个是在计算 item 的历史平均点击率 ctr(Click-Through-Rate) 和历史平均下单率 cvr(Click Value Rate) 的时候，首先要计算出每个位置的 $\mathrm{ctr_p}$ 和率 $\mathrm{cvr_p}$，然后在计算 item 的每次点击和下单的时候，都根据这个 item 被展示的位置，计算为 $\mathrm{ctr_0/ctr_p}$ 及 $\mathrm{cvr_0/ctr_p}$；一个是在产生训练样本的时候，把展示位置作为特征放在样本里面，并且在使用模型的时候，把展示位置特征统一置为 0。

(4) 特征工程

特征工程是排序模型的最重要工作，排序带来的效果提升，大部分是由特征工程带来的。特征提取就是不断地去接触和理解业务数据，试图从中挖掘出和用户转化相关的特征。使用的主要特征包括：上下文特征：如时间，地理位置，天气，温度等。item 特征：如团购服务的价格，销量，用户评分。用户特征：用户的属性特征，如年龄，性别，婚育状态，品类偏好，价格偏好等。

2.3　用户画像模块

2.3.1　用户画像介绍

用户画像模型，用户兴趣探索模块，推荐系统模块之间的关系如图 2.2。

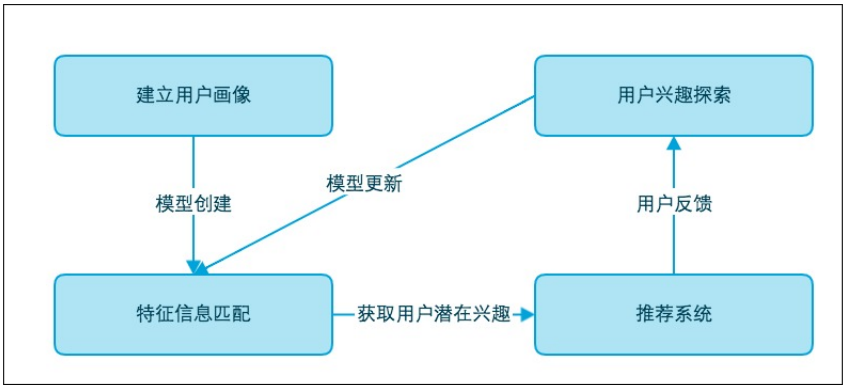

图 **2.2**　手机主题推荐系统功能模块图

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。一个好的推荐系统要给用户提供个性化的、高效的、动态准确的推荐，那么推荐系统应能够获取反映用户多方面的、动态变化的兴趣偏好，推荐系统有必要为用户建立一个用户兴趣探索模型，该模型能获取、表示、存储和修改用户兴趣偏好，能进行推理，对用户进行分类和识别，帮助系统更好地理解用户特征和类别。推荐系统根据用户画像进行推荐，所以用户画像对推荐系统的质量有至关重要的影响。建立用户画像模型之前需要考虑问题有：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。

2.3.2　用户画像数据来源

电子商务用户画像的信息来源可以有如几种方式：

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

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

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

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

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

2.3.3 用户画像构建

一个标签通常是人为规定的高度精炼的特征标识，如年龄段标签：25 到 35 岁，地域标签：北京。标签有两个重要特征：语义化和短文本，人能很方便地理解每个标签含义。这也使得用户画像模型具备实际意义。能够较好的满足业务需求。如，判断用户偏好。同时，每个标签通常只表示一种含义，标签本身无需再做过多文本分析等预处理工作，这为利用机器提取标准化信息提供了便利。人制定标签规则，并能够通过标签快速读出其中的信息，机器方便做标签提取、聚合分析。所以，用户画像和用户标签为我们展示了一种朴素、简洁的描述用户信息的方法。构建用户画像是为了还原用户信息，因此数据来源于所有与用户相关的数据。对于与用户相关数据的分类，一般采用一种封闭性的分类思想。如，世界上分为两种人，一种是懂计算机的人，一种是不懂计算机的人；客户分三类，高价值客户，中价值客户，低价值客户；产品生命周期分为，投入期、成长期、成熟期、衰退期，所有的子分类将构成了类目空间的全部集合。这样的分类方式，有助于后续不断枚举并迭代补充遗漏的信息维度。不必担心架构上对每一层分

类没有考虑完整，造成维度遗漏留下扩展性隐患。另外，不同的分类方式根据应用场景，业务需求的不同，也许各有道理，按需划分即可。

用户数据类型一般划分为静态信息数据、动态信息数据两大类。静态信息数据是指用户相对稳定的信息，如图所示，主要包括人口属性、商业属性等方面数据。这类信息，自成标签，如果企业有真实信息则无需过多建模预测，更多的是数据清洗工作，因此这方面信息的数据建模不是本篇文章重点。动态信息数据是指用户不断变化的行为信息，广义上讲，一个用户打开手机应用软件，点击了一个链接，购买了一个杯子等都属于用户行为。当行为集中到互联网，乃至电商，用户行为就会聚焦很多。用户行为可以被看作用户动态信息的唯一数据来源。用户画像的目标是通过分析用户行为，最终为每个用户打上标签，以及该标签的权重。其中标签表征了用户对该内容有兴趣、偏好、需求等等。权重表征了用户的兴趣、偏好指数，也可能表征用户的需求度，可以简单的理解为可信度，概率。

下面内容将详细介绍如何根据用户行为，构建模型产出标签、权重。一个事件模型包括：时间、地点、人物三个要素。每一次用户行为本质上是一次随机事件，可以详细描述为：什么用户，在什么时间，什么地点，做了什么事。

- 什么时间：时间包括两个重要信息，时间戳 + 时间长度。时间戳，为了标识用户行为的时间点，通常采用精度到秒的时间戳即可。浏览器时间精度，准确度最多也只能到毫秒。时间长度为了标识用户在某一页面的停留时间。

- 什么地点：用户的接触点。对于每个用户接触点。潜在包含了两层信息：网址和内容。网址定位了一个互联网页面地址，或者某个产品的特定页面。可以是 PC 上某电商网站的页面 url，也可以是手机上的微博，微信等应用某个功能页面，某款产品应用的特定画面。内容可以是单品的相关信息：类别、品牌、描述、属性、网站信息等等。其中网址决定了权重；内容决定了标签。

- 什么事：用户行为类型，对于电商有几种典型行为：浏览、添加购物车、搜索、评论、购买、点击赞、收藏等等。不同的行为类型对于接触点的内容产生的标签信息，具有不同的权重。

综合上述分析，用户画像的数据模型，可以概括为下面的公式：用户标识 + 时间 + 行为类型 + 接触点，某用户因为在什么时间、地点、做了什么事。用户标签的权重还可能随时间的增加而衰减，因此定义时间为衰减因子 r，行为类型、网址决定了权重，内容决定了标签，进一步转换为公式，标签权重 = 衰减因子 × 行为权重 × 网址子权重。

2.3.4　用户画像标签维度

一个用户可以从多个方面去刻画，也就是说用户画像可以从多个维度来考虑和构建。作为虚拟电子商品交易平台，电子商务市场的用户在平台上通过某些

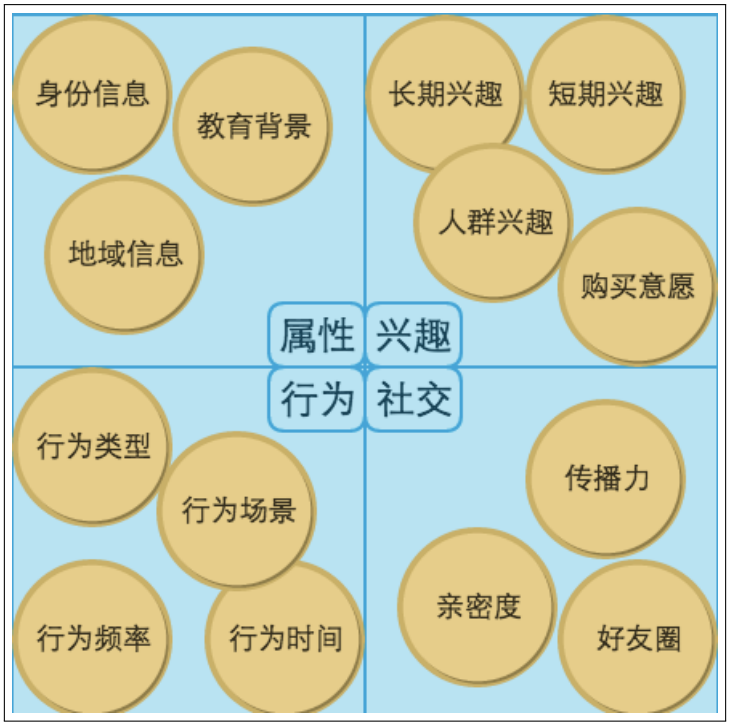

图 **2.3**　用户画像维度图

行为（点击、浏览、购买）生产或获取信息，也通过其它一些行为（如转发、评论、赞）将信息传播出去，信息的传播是通过用户之间的社交关系所进行的，并且在生产、消费、传播信息的过程中对信息的选择和过滤体现了用户在兴趣方面的倾向性。由此，我们可以将用户画像按照图 2.3所示的四个维度进行划分，即属性维度、兴趣维度、社交维度和行为维度。用户属性和用户兴趣是传统用户画像中包含的两个维度。前者刻画用户的静态属性特征，例如用户的身份信息（性别、年龄、受教育程度、学校等），后者则用于刻画用户在信息筛选方面的倾向（例如用户的购买能力、兴趣标签、能力标签等）。社交维度是从社交关系及信息传播的角度来刻画用户的。在社区中用户不在仅仅是一个个体，用户和用户之间的社交关系构成了一张网络，信息在这张网络中高速流动，但是这种流动并不是无差别的，信息的起始点，所经历的关键节点以及这些节点构成的关系圈都是影响信息流动的重要因素。行为维度是一个比较新的研究方向，目的是发现影响用户属性、信息变化的行为因素，分析典型用户群体的行为模式。一方面可以通过行为模式的复用来促进用户在电子商务应用平台的成长；另一方面也有利于平台认识用户，和发现新的或异常的用户行为。属性维度：属性维度属于传统用户画像的范畴，即对用户的信息进行标签化。一方面，标签化是对用户信息进行结构化，方便计算机的识别和处理；另一方面，标签本身也具有准确性和非二义性，也有利于人工的整理、分析和统计。用户属性指相对静态和稳定的人口属性，例如：性别、年龄区间、地域、受教育程度、学校、公司等信息的收集和建立主要依靠产品本身的引导、调查、第三方提供等，在此基础上需要进行补充和

交叉验证。

- 标签来源：不是所有的词都适合充当用户标签，这些词本身应该具有区分性和非二义性；此外，还需要考虑来源的全面性，除了用户主动提供的兴趣标签外，用户在使用的过程中的行为，构建的用户关系等也能够反应用户的兴趣，因此也要将其考虑在内。

- 权重计算：得到了用户的兴趣标签，还需要针对用户给这些标签进行权重赋值，用来区分不同标签对于该用户的重要程度。

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

- 时效性：随着时间的变化，用户的兴趣会发生转移，有些兴趣会贯穿用户使用社交媒体的全过程，而有些兴趣则是受热点时间、环境因素等的影响。

- 长尾性：对于电商领域来讲，那些冷门的用户兴趣的总和可以和那些为数不多的大众化兴趣所占的市场份额相匹配或胜出。

- 兴趣和购买意愿的区分：用户具有某方面的兴趣，只代表了他愿意接受这方面的信息，并不能代表他具有购买相关内容的意愿。例如对于一些只看不买的用户，我们认为其购买意愿很小，因此对其会尽可能多的展示免费主题。

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

- 用户个体对消费信息传播的影响：不同用户在信息传播过程中的重要性不一样，影响大的用户对于信息的传播较影响小的用户更具有促进作用。

- 量化用户关系紧密度：存在社交关联的用户，关系越近的用户之间越容易产生相同的消费行为。

- 寻找相似的用户：消费中非对等的关系本身可以认为是一种认证，用户基于兴趣、消费态度等原因反应到线上的一种关联。那么在消费维度上的相似用户至少能反应他们在某种因素上的一致性。

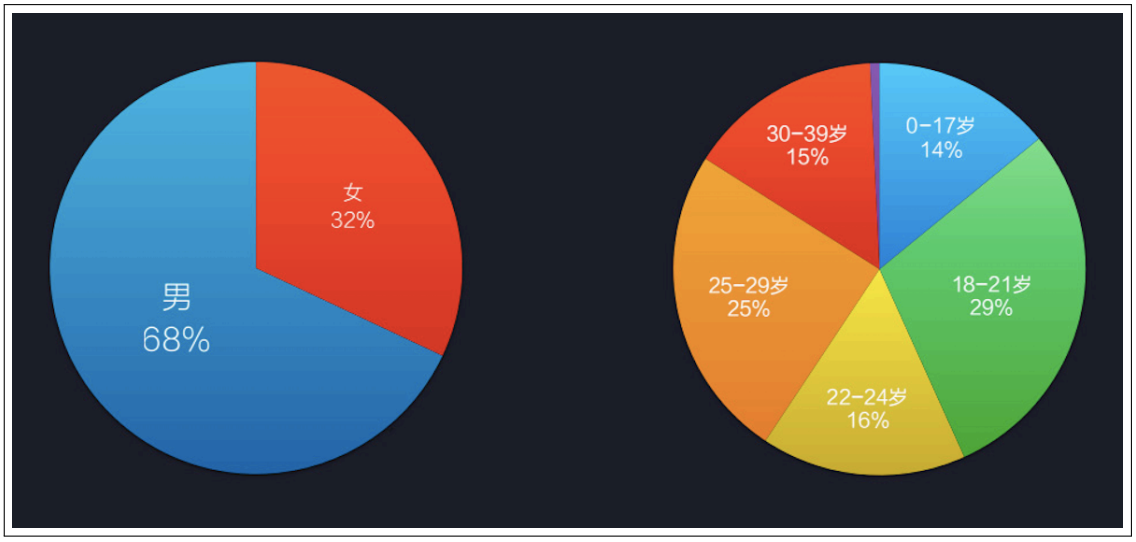

图 **2.4**　电子商务用户分布图

- 识别关系圈：从关系图谱的本身的结构出发，从中发掘关联紧密的群体，有助于促销广告的精准投放和主题包的推广。以上关于关系建模的任务可以看作是逐步深入的，从“个体”–>“关联”–>“相似”–>“群体”的逐渐深入。

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

- 典型个体的行为时序分析。所谓典型个体是指某段时间内，成长比较突出的用户。例如从一个新用户从新注册到点击过百、浏览过千需要有一个积累过程，有些用户积累较快，有些较慢，而这些积累较快的用户可以作为典型个体；或者某些用户在某一阶段消费有限，但在某时刻消费激增，无论是消费金额还是数量都变化很大，这种也可以作为典型个体。针对典型个体，需要挖掘与其用户成长相关的行为因素。基本方法是对时间进行分片，获取用户在不同时间片上的行为统计，以及在各个时间分片上的用户成长指标（点击量、购买量、点击转换比等）。在此基础上针对用户行为的统计量的变化，利用关联性分析或回归来分析用户成长与哪些因素有关。

- 典型群体行为模式分析。针对典型个体，从用户的基本信息、人口信息、兴趣维度，可以将相似的典型用户划分为同一的群体，称作典型群体，针对典型群体中的用户按照成长程度进行划分，按不同的成长阶段统计用户行为，即建立了该典型群体的行为模型。例如，对于“年龄在 20 30 岁，女性，付费用户”这样的典型群体，从日点击量、月消费额等维度将其划分到初创、成长、快速提升、成熟等阶段，针对不同成长阶段内的行为组合进行统计，结果构成该群体的行为模式。如图 2.4。

2.3.5 用户画像应用场景

(1) 优化电子商务市场供求。

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

(2) 提高新人留存率。

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

(3) 用户消费等级分群。

大至用户终端品牌、机型、操作系统，细至屏幕分辨率、屏幕尺寸，用户画像记录了每一个用户群体的详细终端特征。哪一类人群最容易被这款应用吸引，愿意为这款应用付费？开发者经常考虑的问题可以从用户画像找到答案。每一个用户群的价格分布、增值业务费用分布以及流量费用，包括用户详细的消费特征，比如付费频率，丰富了推荐系统的数据依据。

(4) 用户流失预警。

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

(5) 反作弊。

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

2.4　用户兴趣探索模块

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

- 用户偏好随时间变化 (User bias shifting)：用户可能在某一天只对他喜欢的物品评分，某一天可能只对他不喜欢的物品评分。因此用户某一天的平均分是随时间变化的。
- 物品偏好随时间变化 (Item bias shifting)：物品的受欢迎程度也是随时间变化的。一款主题包在刚上线的时候的因为用户关注度小平均评分会很高，随着时间的推移，越来越多的用户参与到评分中，会使其慢慢接近真实的评分。
- 用户兴趣随时间变化 (User preference shifting)：用户在不同的时候可能有不同的兴趣，比如小孩都喜欢动漫主题包，但当他长大了可能喜欢汽车主题包。
- 季节效应：用户行为会受季节效应的影响。主题推荐中主要的季节效应有暑期的效应，以及一些纪念日的效应 (比如国庆纪念日前后，抗日题材的主题包会受到较多的关注)。

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

2.4.1　用户行为数据存储

电子商务用户行为数据的特点包括：用户基数庞大。以电子商务网站淘宝网为例，注册用户往往以千万计，活跃用户达百万计；用户规模增长快。每个用户的行为数量较小。即使是活跃用户，每天最多也只能产生上百条行为记录，每年不超过十万条；用户行为的计算较为复杂。计算用户的两次登录间隔天数、反复购买的商品、累积在线时间，这些都是针对用户行为的计算，通常具有一定的复杂性；用户行为数据格式不规整，字段丢失率较高。根据用户行为数据的这些特点采用基于 Hadop 分布式的架构。

2.4.2 用户行为处理

对于用户的一些人口属性信息采用了显式方式直接获取，对于用户一些明显的兴趣偏好采用了隐式获取，对于用户潜在的兴趣偏好则通过关联技术启发式获取。显式获取用户兴趣偏好的方法是简单而直接的做法，能准确地反映用户的需求，同时所得的信息比较具体、全面、客观，结果比较可靠。缺点就是数量稀少，原因用户不太愿意花时间来向商家表达自己的喜好，并且这种方法灵活性差，答案存在异质性，当用户兴趣改变时需要用户手动更改系统中用户兴趣。同时该方法对用户不是很人性化。隐式获取法是指系统通过记录用户行为数据，通过权重排序获取用户的兴趣偏好，用户的很多动作都能暗示用户的喜好，包括查询、浏览页面和文章、标记书签、反馈信息、滑屏等。隐式的跟踪可以在建立用户画像基本数据的同时不打扰用户的正常消费活动。这种方法的缺点就是跟踪的结果未必能正确反映用户的兴趣偏好。上述获取兴趣偏好的方法有时受用户教育背景、职业和习惯等因素的限制，用户有时意识不到自己的兴趣主题，因此能为用户提供启发式信息，如领域术语抽取和相似度物品聚类，可以实现领域知识的复用，为用户间的协同提供支持，提高用户兴趣获取质量。

随着电子商务市场交易规模的逐步增大，积累下来的业务数据和用户行为数据越来越多，这些用户数据往往是电子商务平台最宝贵的财富。目前在电子商务推荐系统中大量地应用到了机器学习和数据挖掘技术，例如个性化推荐、搜索排序、用户画像建模等等，为企业创造了巨大的价值。数据预处理主要工作是:

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

根据不同业务数据的预处理方式也不同，一般来讲原始服务器日志数据脏数据的形成原因包括: 缩写词不统一，数据输入错误，不同的惯用语，重复记录，丢失值，不同的计量单位，过时的编码等。相应的，数据预处理内容包括数据清理、数据集成、数据变换、数据归约、数据离散化。数据清理包括格式标准化、异常数据清除、错误纠正、重复数据的清除。对于电子商务用户数据来讲，引起空缺值的原因主要是用户设备异常造成的，有些时候是因为与其他已有数据不一致而被删除或数据的改变没有进行日志记载。根据数据空缺情况的不同有不同的处理方式:

- 忽略元组。当一个记录中有多个属性值空缺、特别是关键信息丢失时，已不能反映真实情况，它的效果非常差。
- 去掉属性。缺失严重时，已无挖掘意义。
- 人工填写空缺值。但是工作量大且可行性低。

- 默认值。比如使用 unknown 或-∞。
- 使用属性的平均值填充空缺值。
- 预测最可能的值填充空缺值。使用贝叶斯公式或判定树这样的基于推断的方法。

2.4.3　用户行为权重排序

用户显式行为数据记录了用户在平台上不同的环节的各种行为，这些行为一方面用于候选集触发算法中的离线计算（主要是点击、浏览），另外一方面，这些行为代表的用户兴趣强弱不同，因此在训练重排序模型时针对不同的行为设定了不同的权重值，以更细地刻画用户的行为强弱程度。此外，用户的购买、试用等行为还作为重排序模型的交叉验证特征值，用于模型的离线训练和在线预测。负反馈数据反映了当前的结果可能在某些方面不能满足用户的需求，因此在后续的候选集触发过程中需要考虑对特定的因素进行过滤或者降权，提高用户体验；同时在重排序的模型训练中，A/B 测试结果作为负例参与模型训练。用户画像是刻画用户属性的元数据，其中有些是直接获取的基础数据，有些是经过挖掘的二次数据，这些属性一方面可以用于候选集触发过程中对标签进行加权或降权，另外一方面可以作为重排序模型中的用户维度特征。通过对数据的挖掘可以提取出一些关键词，然后使用这些关键词给主题打标签，用于主题的个性化展示。

2.4.4　用户行为建模

当我们想基于用户行为分析来建立用户兴趣模型时，我们必须把用户行为和兴趣主题限定在一个实体域上。个性化推荐落实在具体的推荐中都是在某个实体域的推荐。对于手机主题应用市场来说，实体域包括所有的主题，背景图片，铃声，闹铃等。用户行为。浏览，点击，下载，试用，购买，评论等都可是用户行为。本文所指的用户行为都是指用户在某实体域上的行为。比如用户在手机铃声产生的行为。用户兴趣。用户的兴趣维度，同样是限定在某实体域的兴趣，通常以标签 + 权重的形式来表示。比如，对于手机主题，用户兴趣向量可以是「动漫，0.6」,「体育，0.1」,「情感，0.7」等分类标签。值得一提的是，用户兴趣只是从用户行为中抽象出来的兴趣维度，并无统一标准。而兴趣维度的粒度也不固定，如「体育」,「电影」等一级分类，而体育下有「篮球」,「足球」等二级分类，篮球下有「NBA」,「CBA」,「火箭队」等三级分类。我们选取什么粒度的兴趣空间取决于具体业务模型。

实际应用中，在社交网络用户的行为一般是主动进行的，例如，自行定义或选择标签，浏览页面，使用站内产品或第三方 APP，发表博文或对其他博文内容的点赞或收藏，关注其他用户并将其关注的对象划分到自行设置的各用户组内

等。而上述这些社交网络用户的行为能够在一定程度上反映出用户的兴趣。因此，社交网络中，可以根据用户的这些网络行为来进行用户的兴趣挖掘。该阶段是用户行为数据进行建模，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。基于用户标签的兴趣挖掘方法。具体地，可以根据标签的具体内容，将标签归类到相应的兴趣类别后，再根据用户的自定义标签及其所属的兴趣类别，分析出用户的兴趣。

2.5　本章小结

用户画像模块对应着用户长期兴趣，用户兴趣探索对应着用户短期动态兴趣。短期兴趣的特点是临时、易变；长期兴趣的特点是长久、稳定；用户的短期兴趣可能会转化为长期兴趣，所以需要在推荐时综合考虑长期兴趣和短期兴趣。考虑到推荐系统的时间效应问题，将输入数据集归结为一个四元组，即用户，物品，行为，时间，数据集可以选用比较直观的显性反馈数据集，给定用户 u，物品 i，时间 t，预测用户 u 在时间 t 对物品 i 的评分 r。对于该类问题的评分预测问题主要有：用户兴趣的变化，如年龄增长，从儿童长成青少年壮年；生活状态的变化，由以前的小学生到大学生；社会事件的影响如俩会等。此外还有季节效应问题，一些在春季很流行的，在夏季节未必就很流行。对于时效性的影响，每个推荐系统都有不同的演化速率，相对来说新闻更新很快，但音乐、电影的跟新却比较慢。

第三章　用户画像模块

3.1　引言

Alan Cooper 最早提出了用户画像的概念：Personas are a concrete representation of target users。Persona 是真实用户的虚拟代表，是建立在一系列真实数据之上的目标用户画像。通过用户历史行为去了解用户，根据他们的目标、行为和观点的差异，将他们区分为不同的类型，然后每种类型中抽取出典型特征，赋予名字、照片、一些人口统计学要素、兴趣标签等描述，就形成了一个人物原型，图 3.1所示为一个典型的用户画像，标签面积越大代表其权重越高。

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：

- 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。

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

- 支撑主题推荐系统的精准推荐：精准推荐的前提是对用户的清晰认知。以简单代金券发放为例，手机主题应用市场的历史数据呈现出两大类四种不

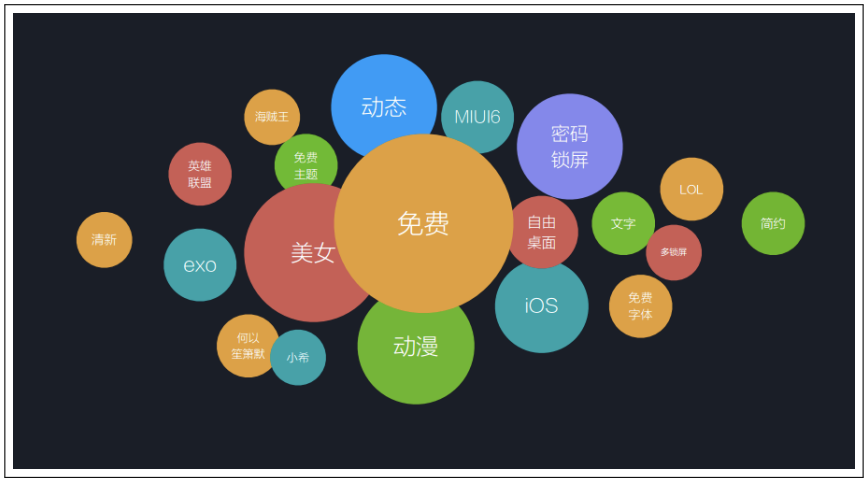

图 **3.1**　用户画像标签示例图

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

- 主题市场安全领域的应用：随着手机主题市场的发展，商家会通过各种活动形式的补贴来获取用户、培养用户的消费习惯，但同时也催生一些通过刷排行榜、刷红包的用户，这些行为距离欺诈只有一步之遥，但他们的存在严重破坏了市场的稳定，侵占了活动的资源。其中一个有效的解决方案就是利用用户画像沉淀方法设置促销活动门槛，即通过记录用户的注册时间、历史登陆次数、常用 IP 地址等，最大程度上隔离掉僵尸账号，保证市场的稳定发展。

3.2　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。

3.2.1　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",
```

```
        "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 3.1　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	...
...	...	...	...	...	...	...	...

3.2.2　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 3.2　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	...
...	...	...	...	...	...	...	...

3.2.3　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 3.3　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	...
...	...	...	...	...	...	...	...

3.3　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。

3.3.1　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";
        }
    }
```

3.3.2　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}%'
```

3.3.3　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 3.1，其中 T 取值为 1，t 为行为发生时间距离当前时间的天数。

$$score = (\frac{1}{2})^{(t/T)} \tag{3.1}$$

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;
   }
}
```

3.4　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。

3.4.1　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'
    }
```

3.4.2　评测指标

本节使用线上 A/B 测试方案 [25]，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。

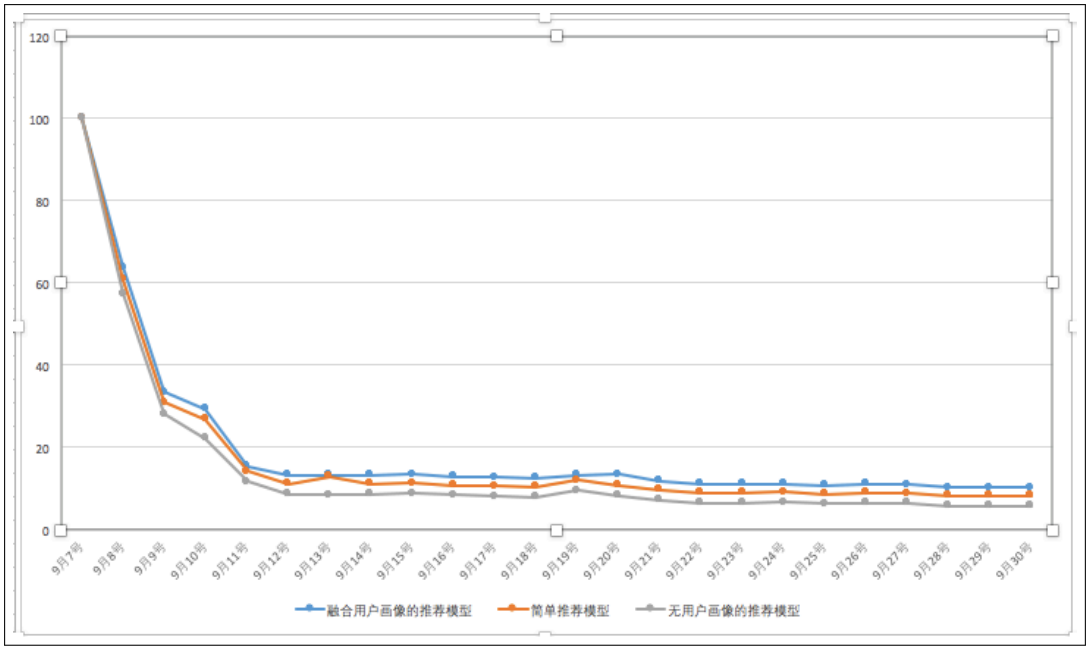

图 **3.2**　新用户留存率实验对比图

3.4.3　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 top20% 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 3.2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。

3.5　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。

第四章　用户兴趣探索

4.1　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。

4.2　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。

4.2.1　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;
```

4.3　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。

4.3.1　基本概念概述

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

用户行为。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 4.1，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。

$$tagFocus = \log \frac{|tagInThemeNum|}{|allThemeNum|} \tag{4.1}$$

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 4.2。

$$tagPopular = \log \frac{|peopleLikeTagNum|}{|allPeople|} \tag{4.2}$$

4.3.2　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;
}
```

4.3.3　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;
}
```

4.3.4　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 4.1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 **4.1**　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

4.4　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
         }
      }
   }
```

4.5　实验与分析

4.5.1　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。

4.5.2　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。

4.5.3　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。

4.5.4　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 4.1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 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

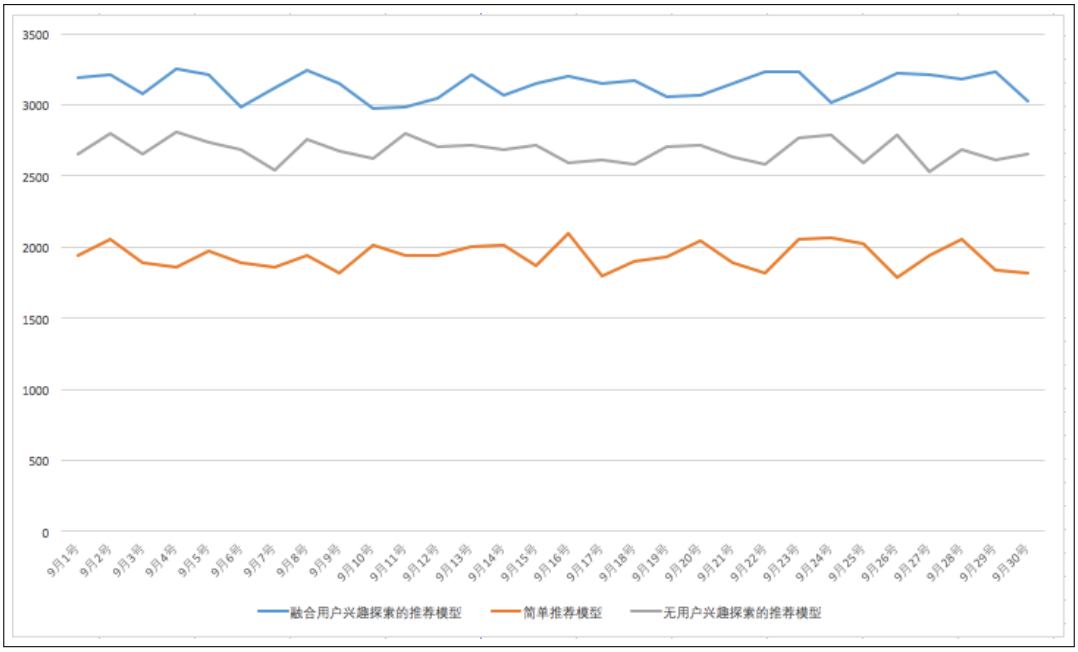

图 **4.1**　推荐多样性实验对比图

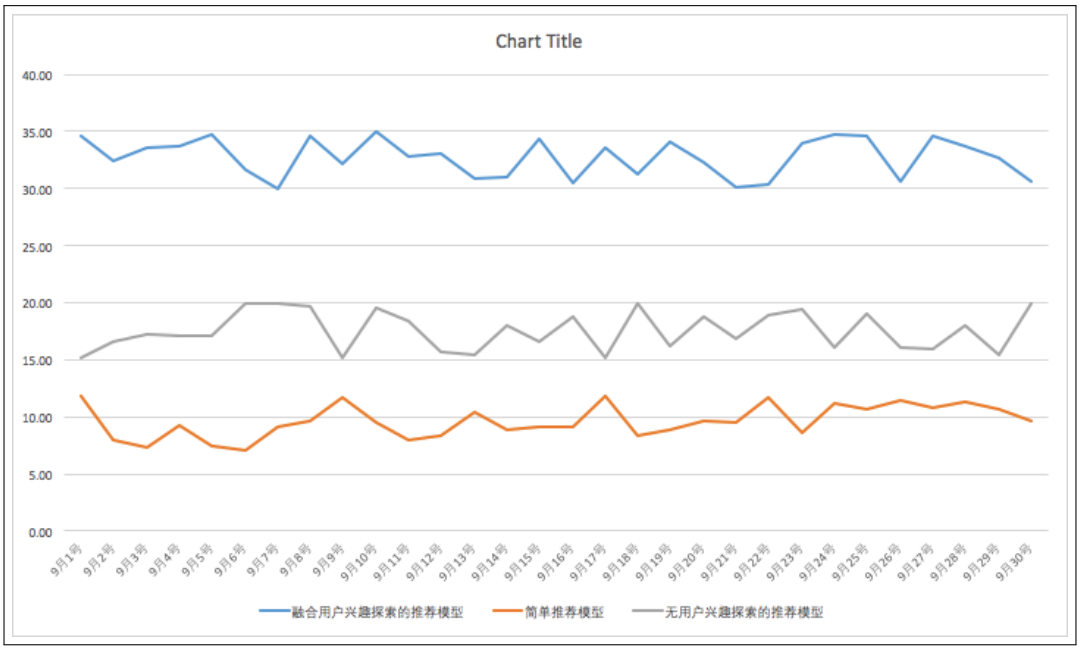

图 **4.2**　转化率实验对比图

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。

4.6 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。

第五章　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。

5.1　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。

5.2　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] Shumeet Baluja, Rohan Seth, D. Sivakumar, Yushi Jing, Jay Yagnik, Shankar Kumar, Deepak Ravichandran, and Mohamed Aly.2008. *Video sug- gestion 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.

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

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

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

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

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

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

[8] 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.

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

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

[11] 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.

[12] 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.

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

[14] 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.

[15] 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.

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

[17] 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.

[18] 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,

[19] Han Jiawei, Kamber, Micheline.2001. *Data mining: concepts and techniques[C]*. Morgan Kaufmann. 5.

[20] 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)

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

[22] 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.

[23] 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.

[24] 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.

[25] Kohavi, Ron, Longbotham, Roger.2015. *Online Controlled Experiments and A/B Tests[C]*. In Sammut.
[26] Robin Burke. November 2002. *Hybrid recommender systems: Survey and experiments*. User Modeling and User-Adapted Interaction, 12:331–370.
[27] Gediminas Adomavicius and Alexander Tuzhilin. 1999. *User Profiling in Personalization Applications through Rule Discovery and Validation*. ACM, 377-381.
[28] lbrahim Cingil, Asuman Dogac and Ayca Azgin.2000. *A broader approach to personalization*. mmunications of the ACM, 43(8): 136-141.
[29] Joseph Kramer, Sunil Noronha and John Vergo.2000. *A user-centered design approach to personalization*. Communications of the ACM, 43(8)44-48.
[30] 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.
[31] Bamshad Mobasher, Robert Cooley and Jaideep Srivastava.2000. *Automatic personalization based on Web usage mining*. Communications of the ACM, 43(8): 142-151.
[32] 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.
[33] 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.
[34] 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,
[35] 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 21 日