**密级： 保密期限：**



**硕士学位论文**



**题目： 面向用户体验的智能应用**

**使用模式分析与优化研究**

**学 号： 2013140249**

**姓 名： 尹彦龙**

**专 业： 电子与通信工程**

**导 师： 张琳**

**学 院： 信息与通信工程**

**2015年 12月 01日**

独创性（或创新性）声明

本人声明所呈交的论文是本人在导师指导下进行的研究工作及取得的研究成果。尽我所知，除了文中特别加以标注和致谢中所罗列的内容以外，论文中不包含其他人已经发表或撰写过的研究成果，也不包含为获得北京邮电大学或其他教育机构的学位或证书而使用过的材料。与我一同工作的同志对本研究所做的任何贡献均已在论文中作了明确的说明并表示了谢意。

申请学位论文与资料若有不实之处，本人承担一切相关责任。

本人签名： 日期：

关于论文使用授权的说明

学位论文作者完全了解北京邮电大学有关保留和使用学位论文的规定，即：研究生在校攻读学位期间论文工作的知识产权单位属北京邮电大学。学校有权保留并向国家有关部门或机构送交论文的复印件和磁盘，允许学位论文被查阅和借阅；学校可以公布学位论文的全部或部分内容，可以允许采用影印、缩印或其它复制手段保存、汇编学位论文。（保密的学位论文在解密后遵守此规定）

保密论文注释：本学位论文属于保密在 年解密后适用本授权书。

非保密论文注释：本学位论文不属于保密范围，适用本授权书。

本人签名： 日期：

导师签名： 日期：

面向用户体验的智能应用使用模式分析与优化研究

摘 要

当今时代，移动互联网正在快速的发展，与此同时，大数据时代也悄然而至,“移动互联网”和“大数据”变成当今当前互联网领域内最火的话题。而在这之中，和用户关系最直接的移动互联网便是移动app，而面对海量的app，如何选择合适的app，对普通用户来说是一个头疼的问题。

在这种情况下，如何在大量数据里帮助用户选择合适的app，以提升用户的使用体验并且能为用户节约使用成本变得重要起来。基于这个情况，本文将从流量使用方面，结合用户的使用偏好，为用户推荐符合自身使用习惯并且减少流量使用的app。本文将从以下两个方面展开：首先，数据分析平台的搭建，这是文章建立的基础。本文用到的数据分析平台是基于ambari的Hadoop分析平台。基于搭建的这个平台，对目标数据集进行相关的分类、处理，进而对数据做一些相关的分析。

其次，建立app推荐模型，本文提出的推荐模型分为两个方面：用户相关的使用偏好和app相关的画像。只有分析出来用户的使用偏好，才能正确的为不同的用户有针对性的推荐。在分析了用户的使用偏好之后，为用户推荐什么样的app是接下来分析的基础。根据本文的出发点，app的分析主要从流量消耗方面来考虑，其次还要关注app的流行度，根据这两个点，综合考虑，然后得出推荐的列表。除此之外，不论是用户还是app，都需要考虑时间段的问题，也就是说用户偏好使用某一类app的时间段和app被使用最频繁的时间段。

根据建立好的推荐模型，分析出用户的使用偏好，为用户推荐相似的但流量更少，流行度更高的app，改善用户的使用体验。

关键词Hadoop 用户体验 使用偏好 推荐 app

THE RESEARCH OF THE USAGE PATTERNS

AND OPTIMIZATION OF THE SMARTPHONE

APPLICATION FOR USER EXPERIENCE

ABSTRACT

Follow the development of mobile Internet, big data time arrives in quick succession. ‘Mobile Internet’ and ‘Big Data’ have become the hottest topics in the internet area. In those topics, mobile application has the most direct relationship with users. But most of the users find it difficult choosing appropriate applications to use.

Under this circumstances, it becomes more and more important that helping people find the fittest application and enhancing their use experience in massive data. Based on those information, this pager will start with the Traffic Usage, in combination with user’s use pattern and then recommend the applications which will consume less traffic and are more popular. The main work of this paper consists of two parts: first, we build the Hadoop platform through Ambari. The platform is the base of this paper. With the platform, we could do some classification and then process and analyze the dataset. Second, we build the recommend model, which consists of two parts either, user use pattern and app pattern. For one thing, in order to recommend applications for users, we analyze the user’s hobby in using applications and their use experience. For another, we study the applications to record how they been used and group them into different type. Finally under the condition that we satisfy user’s hobby and use experience, we would recommend the user more suitable applications which may consume less traffic and are more popular. One more thing, we have to study the using period of either the user or the app.

According to the recommend model, we study users’ use pattern and recommend them with the most appropriate application to enhance their use experience.

KEY WORDS Hadoop user experience use pattern recommend application

目 录

[第一章 绪论 1](#_Toc436410216)

[1.1 研究背景 1](#_Toc436410217)

[1.2 国内外研究现状 3](#_Toc436410218)

[1.3 本文主要工作 5](#_Toc436410219)

[1.4 文章的结构 6](#_Toc436410220)

[第二章 数据分析平台说明 8](#_Toc436410221)

[2.1 Hadoop集群管理工具-Ambari介绍 8](#_Toc436410222)

[2.2 Hadoop 简介 11](#_Toc436410223)

[2.3 HDFS的体系结构 12](#_Toc436410224)

[2.4 MapReduce的体系结构 15](#_Toc436410225)

[2.5 本章小结 17](#_Toc436410226)

[第三章 基本数据分析 18](#_Toc436410227)

[3.1 本文使用的数据集说明 18](#_Toc436410228)

[3.1.1 数据来源 18](#_Toc436410229)

[3.1.2 数据格式 20](#_Toc436410230)

[3.2 相关的数据分析 23](#_Toc436410231)

[3.2.1 数据总体情况分析 24](#_Toc436410232)

[3.2.2 用户方面的情况分析 28](#_Toc436410233)

[3.2.3 APP方面的情况分析 30](#_Toc436410234)

[3.3 本章小结 33](#_Toc436410235)

[第四章 推荐模型的建立 34](#_Toc436410236)

[4.1 用户使用偏好模型的建立 35](#_Toc436410237)

[4.1.1 使用时间的分析 36](#_Toc436410238)

[4.1.2 使用频度的分析 40](#_Toc436410239)

[4.1.3 用户使用时间段的分析 43](#_Toc436410240)

[4.2 APP画像的分析 44](#_Toc436410241)

[4.2.1 APP流量分析 45](#_Toc436410242)

[4.2.2 APP流行度分析 49](#_Toc436410243)

[4.2.3 APP推荐排名的建立 51](#_Toc436410244)

[4.2.4 APP活跃时间段的分析 53](#_Toc436410245)

[4.3 本章小结 53](#_Toc436410246)

[第五章 推荐结果分析 55](#_Toc436410247)

[5.1 视频类软件推荐结果分析 55](#_Toc436410248)

[5.2 下载类软件推荐结果分析 56](#_Toc436410249)

[5.3 音乐类软件推荐结果分析 57](#_Toc436410250)

[5.4 浏览器类软件推荐结果分析 59](#_Toc436410251)

[5.5 本章小结 60](#_Toc436410252)

[第六章 总结及展望 61](#_Toc436410253)

[6.1 总结 61](#_Toc436410254)

[6.2 展望 61](#_Toc436410255)

[参考文献 63](#_Toc436410256)

[致谢 66](#_Toc436410257)

[攻读硕士期间发表的学位论文 68](#_Toc436410258)

# 第一章 绪论

## 1.1 研究背景

随着安卓,ios,Windows mobile等智能手机操作系统的崛起，移动互联网时代进入了快速成长期，2015年度的移动互联网的发展大会在北京召开，会上发布了由人民网研究院主编的《中国移动互联网发展报告（2015）》蓝皮书。报告中指出，截至到2015年3月，通过移动设备例如手机，平板电脑等，进行上网的用户总数达8.99亿户，这表明大约有73.8%手机用户都会用手机上网；与此同时，全国的网民用户中有八成多是通过手机上网，这直接的决定了手机这个第一大上网终端地位。我国的人民在积极的拥护着移动互联网的发展。其实，鉴于移动设备的便携性，通过移动设备进行信息交换以及通过移动设备获取服务已经变成了主流的交互方式。移动互联网能够发展如此迅速，依靠的是各种各样的第三方应用的发展，在这种情况下，大量的应用程序进入移动互联网用户的视线。以苹果的 App Store 和 Google 的 Google Play 为例，Play Store的Android应用总量达到143万款，而App Store的iOS应用总量为121万款，并且在2015年6月份之前的12个月里，Google Play应用下载量已经达到惊的500亿次，App Store下载量也有将近250亿次。另外一个不可忽视的方面，APP应用的同质化，也就是指，同样功能的app会有几十甚至上百款，这种现象越来越严重，这对人们的生活造成了极大的不便。当今生活的节奏已经变得越来越快，app的更新换代也变得越来越快，人们想要在同质化严重的的app大海里寻找一款适合自己使用的手机APP已经变得十分困难，还有一个普遍的现象是，新发布的APP应用肯定会不断地替代旧的APP应用，而用户肯定会不断尝试新的app，进而不断的放弃旧的app，这也变成了一个不争的事实。很多人不禁会问：在这么多应用市场的这么多应用软件中，该如何挑选符合自己使用习惯并且消耗较少的流量的应用软件呢?当用户登上各种各样的应用商店时，成千上万的各种APP应用让用户看的眼花缭乱，更让用户选择的头疼，很多用户会选择盲目的下载。该如何选择正确，符合自己的使用习惯并且能花费更少的流量，确实也是一个难题。一般来讲，现在的情况下用户在选择app的时候可能会查看app相关的介绍，app的截图，以及应用商店提供的app的评论等等这些信息。App的介绍可以帮助用户大体了解app的功能。用户的评论则可以帮助用户判断APP是否符和自己的一些要求。UI界面则影响着APP应用的第一形象。开发商可以看出APP应用的整体质量，但是这个过程都是在用户亲自去查看的情况下进行的，也就是说这还得需要用户去判断。并且以视频类应用来讲，如果用户想查找一款能够替代当前使用的视频类应用，并且在移动网络下能够花费较少的流量，通过刚才的方法怕是做不到的。除此之外，在这种全新的市场氛围下,运营商也应该顺应移动互联网发展，根据自己原有的优势和机遇，做出相应的调整。在机遇与挑战的面前,运营商必须第一时间了解潜在市场的用户需求,就能适应市场,为用户提供更完善更个性化的服务,就能在竞争中占有一席之地。如何获取用户的需求信息,最根本的途径是针对用户的上网数据进行分析,即实现用户行为分析。在移动互联网潜在客户类型识别过程中,移动互联网用户行为分析可以有效、快捷地获取用户潜在需求,能够持续不断地吸引客户。对于成熟稳定的客户群体可以通过用户行为分析不断增强和改善用户服务感受,提升用户体验。

移动互联网迅速发展的今天必然也带动了大数据时代的发展。前段时间，中国信息通信研究院发布了《2015年中国大数据发展调查报告》，该报告指出在2015年，中国的大数据市场规模将会达到115.9亿元。该报告还显示，在大数据应用的相关部署方面，受访的企业当中有超过44%的企业并没有部署大数据开发平台，也没有推出相关的大数据应用。另外，还有24%的企业已经开始着手部署大数据开发平台，但还并没有开发大数据相关的应用。只有大约32%的受访企业已经实现了部分的大数据应用的开发工作。除此之外，该报告还显示，互联网行业、电信行业以及金融行业是部署大数据平台并且开发大数据应用最为集中的几个行业，其他一些传统的行业开发部署大数据相关应用的脚步进展的比较缓慢，有超过80%的批发零售业的相关企业并没有部署开发大数据应用的计划。但比较好的情况是，该项调查报告表明，95%的企业的大数据应用都处于一个增长态势。在这种形式下，随着一些企业的大数据应用带来显著的应用效果，势必会刺激其他企业的大数据相关的开发部署工作，预计未来的2016至2018年期间中国的大数据市场会始终保持高速增长的势头。以上所讲纷纷表明了一个事实，大数据的发展是一种趋势，代表着未来。人们开始讨论如何处理海量的数据，如何在海量数据里找出有价值的信息，就像刚才提到的，在大量的app使用记录里如何提取出对用户有用的信息，如何分析出用户使用app的相关行为特征，传统的数据处理方法可能无法满足我们的要求，这里就需要我们借助大数据处理的工具-Hadoop，来完成数据分析的要求。Hadoop是一个开发和运行处理大规模数据的软件平台,是Apache的一个用java语言实现开源软件框架，是利用大量计算机组成的廉价的集群，然后对大量计算机中存储的海量数据进行分布式计算。Hadoop框架的核心设计就是HDFS和MapReduce. HDFS提供了海量数据的存储方式, 而MapReduce则提供了对这些分布式存储的数据的分布式计算能力.除此之外，hadoop还具有其他的很多优势，在这里就不在赘述。

在当今时代海量的app存在的情况下，利用Hadoop大数据分析平台，通过使用Hadoop对数据进行分析处理，分析用户的app使用偏好，并根据用户的使用偏好为用户进行应用推荐，使用户能够得到更合适的应用以及更好的用户体验的同时，能够得到更少的流量消耗，这也迎合了未来的发展趋势。

## 1.2 国内外研究现状

在这个时代，推荐是永恒的话题，和搜索引擎不同，个性化推荐必须依赖用户的行为数据，只有基于用户的信息，才能进行相关推荐。推荐系统的相关应用充斥着当今互联网的各类网站中，而这些个性化推荐系统在这些网站中的主要作用都是通过处理，并且分析大量用户的相关行为日志，然后给不同用户提供针对各个用户的不同的页面展来示，从而来提高网站的点击率，增加网站的访问量，提高网站的知名度。

Google的Google Play应用商店，也会在一定程度上对用户进行应用推荐。具体方案包括：第一、根据你的好友下载并且对软件评价的情况，为你推荐类似的应用，这里的好友是你的Google社交软件Google+里的好友，他们来自世界各地，有着不同的生活习惯，甚至和我们的生活习惯千差万别，所以这种推荐并没有太大的意义；第二、应用商店需要你对你下载的应用程序进行评分，你进行评分之后，系统会为你推荐类似功能的软件，这主要是根据个人喜好来推荐，并没有考虑到软件对流量的消耗以及用户对流量的承受能力。在我国，虽然这些年流量正在变得越来越便宜，但是应用程序对流量的消耗也变得越来越多，所以移动流量问题始终是困扰移动互联网用户的一大问题。国内的应用市场，就360手机助手来说，推荐策略包括几个方面：第一，根据你下载的应用程序，告诉你下载该应用的其他用户还下载了什么，这是根据相似下载来推荐相似应用的一种策略；还有推荐的方式是下载排行等等。360手机助手也没有考虑的软件的流量问题，当然他们可能也没有相关的信息。

在以往的个性化推荐系统中，基本的流程都是，先定期的对数据进行分析处理，然后根据处理的结果对模型进行定向的更新，进而利用更行后的新的模型进行最终的个性化推荐。由于是对模型更行是定期进行的，所以推荐模型无法保持推荐结果的实时性，导致最终推荐的结果可能不是很精准。

就拿网易云音乐的用户举个例子，假如一个用户走在大街上，或者在商场里闲逛，突然无意中听到了一种以前为曾听过的曲风，并且觉得这种曲风特别好听，非常适合自己的口味，就连续听了好几首这种曲风的歌曲。如果平台的推荐系统做不到实时性的推荐的话，那么推荐系统给用户推荐的歌曲肯定是用户以前喜欢听的歌曲，而无法立即给用户推荐他刚刚听到的那类歌曲曲风的新的歌曲，这样一来推荐就失去了意义。

2014年，祝恒书在文献[7]中介绍了基于安全性和隐私方面的应用推荐，主要从app需要的权限问题入手，分析app需要的权限，例如，位置权限、读取通讯录权限、使用摄像头权限等等，从使用这些权限可能会带来的安全问题，隐私问题来讨论，并且结合这些app的特征以及流行度，为用户推荐不同安全级别的应用软件，最终为用户的隐私和个数据安全性提供了保障。2013年潘宇斌在文献[6]中主要是将社交网络和个性化推荐结合在了一起，提出了一套基于社交网络的应用推荐系统，通过分析朋友圈、社交圈子中具有相似特征，相同喜好的用户，从而为他们找到可能感兴趣的应用。

用户行为分析与研究也是本文的一个重要的出发点，只有确切的了解了用户的相关行为，才能进而基于用户的行为进行相关的处理。2014年，余泓在文献[1中]主要对移动互联网的相关服务质量以及用户的一些行为进行了分析，具体采用的方法是通过移动终端进行数据采集，然后再通过Hadoop大数据平台进行数据的存储并进一步分析，并设计相关的参数指标来评价网络质量和用户的行为。而2012年，杨艳在文献[4]提出的用户行为分析是面向下一代网络的。通过分析用户在网络中的行为，进而分析用户行为对网络性能、以及相关的阻塞率的影响。文献[6]提出了一套个性化推荐的移动应用管理系统，并采用Hadoop平台作为后台的系统支持，最后的推荐工作是基于相关的推荐算法，但是文章并没有对用户行为进行过多的分析。

2014年，Tekin, C.和Zhang, S.和van der Schaar在文献[22]中介绍了分布式系统在推荐系统中的应用，主要说明了利用分布式文件系统的特性来提升推荐的精确性。2012年，Verbert, K.和Manouselis在文献[24]中提出了一套基于情景感知的推荐系统，该推荐系统主要以情景感知为根本，进而为用户推荐。这些文献并没有移动互联网方面的移动app的相关背景，也没有提到流量方面的推荐。

本文基于hadoop平台的分布式存储功能，以及高效的mapreduce分布式数据处理能力，对海量数据进行分析处理，本文使用hadoop数据分析平台对软件的流量监控信息进行分析，对着重对用户的使用模式进行讨论，并对用户行为喜好软件建造模型，量化喜好和流量，根据软件的一些特征参数匹配出最大化满足用户喜好模型以及最小化流量使用的应用程序。

## 1.3 本文主要工作

当今时代非常热门的一个课题是移动互联网时代的海量数据，以及海量数据的后期处理工作，其实大数据技术包括一系列的流程，数据采集、数据清洗、数据存储、数据处理。本课题的研究正是基于大数据技术而来。本文从移动互联网海量用户信息和app信息出发，利用Hadoop大数据分析平台，采集、存储并且处理数据。提出了一套基于用户使用应用程序的行为偏好，为用户推荐使用流量最少，并且符合用户使用偏好的的应用程序的一套理论。设计的内容包括大数据技术中的数据采集、少量的数据清洗、数据存储以及最终的数据处理过程。其中数据集的采集部分的工作已经完成，数据在不断增加中。本文的工作主要基于Hadoop平台，从数据的备份到数据的导入，再到数据的处理，这一切都是通过Hadoop的相关功能来实现的。为了简化Hadoop平台的搭建，本文采用ambari集成管理系统，搭建了Hadoop大数据分析生态系统，包含了一系列的组件，以供在后面的开发中使用。

本文的主要内容包括用户的行为分析以及app的状态分析。为了给用户推荐合适的应用，需要先分析用户使用app的偏好，只有了解了用户喜欢使用什么样的app，才能针对用户的使用习惯为用户进行推荐。分析用户的使用偏好主要从两方面进行，一是用户在某一类app的使用时间，时间最能反映一个用户对某个应用的使用习惯，其次是用户对某一类app的使用频度，所谓频度就是表明用户使用这一类软件是否频繁。也能在一定程度上表征用户的使用习惯。将使用时间和使用频度结合就能够确立用户的app使用习惯。在了解了用户的使用偏好之后，还需要对app进行分析，本文的研究方向用户行为分析和优化，而优化的方面确立为使用流量的优化，所以分析app的相关特征的时候。主要从app的流量消耗方面进行。为保证推荐的可靠性和高质量，除了流量消耗，本文还加入了app流行度的信息。从app的流量消耗和app的流行度两个方面入手，综合考虑两方面的因素，根据投资组合理论确立最终的推荐列表。

根据最终确立的推荐列表，然后对应分析每个用户的使用偏好，就可以为用户进行相关的应用推荐。

## 1.4 文章的结构

本文从数据采集，到数据备份都做了相关的介绍，有对数据进行了基本的处理，说明本文研究的必要性，本文的结构安排如下：

第一章：绪论，主要介绍了本文的研究背景以及国内外的发展现状，然后说明了本文的主要研究工作，最后介绍了文章的工作安排。

第二章：着重介绍了Hadoop大数据分析平台，包括本文数据处理所用到到的分布式文件系统hdfs，用来存储本文所用到的数据。其次介绍了mapreduce分布式处理框架以及其运算机制。最后介绍了hive和搭建本平台用的ambari工具。

第三章：着重介绍了对数据集的基本处理工作，包括数据格式的说明。并从总体情况，用户方面和app方面三个点对数据展开了分析。从数据层面分析了本文研究的可行性和必要性。

第四章：着重介绍了推荐模型的建立。包括用户使用偏好模型的建立和app画像的建立。前者又包括使用时间和频率的分析，后者包括app流量和流行度的分析。最后基于app的流量和流行度计算出app统计排名。

第五章：推荐结果的比较。根据推荐模型对用户进行推荐，并对推荐结果进行分析比较。

第六章：总结及论文展望。主要对本文进行了总结并展望未来。

# 第二章 数据分析平台说明

## 2.1 Hadoop集群管理工具-Ambari介绍

Apache Ambari现在是一个Apache的顶级项目，早在2011年8月，Hortonworks引进Ambari作为Apache Incubator项目，制定了Hadoop集群极致简单管理的愿景。这个相关的开发社区正在不断地成长，从开始的一个小开发团队，现在成为了Hortonworks贡献者。另外一个方面，使用Ambari管理大数据集群的用户群也一直在稳步增长，许多公司机构都依靠Ambari来搭建并且管理Hadoop集群。主要的原因是Ambari提供了一个图形化的管理界面，极大的简化了集群的搭建工作，并且最大程度上通过web界面帮助用户管理集群系统。

Apache Ambari是一种基于Web的工具。 Apache Ambari 支持HDFS、MapReduce、Hive、Pig、Hbase、Zookeeper、Sqoop和Hcatalog等的集中管理。也是5个顶级Hadoop管理工具之一。

Ambari主要取得了以下成绩：

* 通过一步一步的安装向导简化了集群供应。
* 预先配置好关键的运维指标（metrics），可以直接查看Hadoop Core（HDFS和MapReduce）及相关项目（如Hbase、Hive和Hcatalog）是否健康。
* 支持作业与任务执行的可视化与分析，能够更好地查看依赖和性能。
* 用户界面非常直观，用户可以轻松有效地查看信息并控制集群。

Ambari使用Ganglia收集度量指标，用Nagios支持系统报警，当需要引起管理员的关注时（比如，节点停机或磁盘剩余空间不足等问题），系统将向其发送邮件。

此外，Ambari能够安装安全的（基于Kerberos）Hadoop集群，以此实现了对Hadoop 安全的支持，提供了基于角色的用户认证、授权和审计功能，并为用户管理集成了LDAP和Active Directory。本文处理数据用到的的Hadoop集群是通过Ambari来进行搭建和管理的。搭建好的Ambari集群功能图如图 2 - 1所示，管理界面包含和很多集群方面的信息。

图 2 - 1 Ambari集群功能图



我们的集群只使用了3台机器，其中一台是是主节点，另外两台是从节点，如图 2 - 2所示。

三台机器环境如表 2 - 1所示：

表 2 - 1 集群机器配置图

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 机器名称 | 操作系统 | 内存大小 | 处理器 | 硬盘大小 | 位数 |
| Master | Centos 6.5 | 8GB | 4核志强 | 500GB | 64位 |
| Slave2 | Centos 6.5 | 8GB | 4核志强 | 500GB | 64位 |
| Slave4 | Centos 6.5 | 16GB | 8核志强 | 1TB | 64位 |

从图 2 - 1中可以看出，本集群的名字叫做AAA，已经运行的服务包括HDFS，MapReduce，Hive，Zookeeper。其中HDFS服务包括3个node，3个node都处于启动状态，并且HDFS磁盘使用率已经达到了86.7%。其次，MapReduce也有三个节点，且都处于运行状态，进行该截图时，集群并没有进行mapreduce任务，所有界面显示0 jobs running，0 jobs waiting。Hive也处于启动状态，hive启动后相应启动了一个MySQL server进行相关数据的存储。除了介绍过的这三个主要的组件外。本Ambari系统，还启动了一个zookeeper服务。和Hadoop一样，ZooKeeper是一个分布式的系统，其主要功能是对Hadoop的其他组件的功能和服务进行协调，它能保证服务器上的各种服务能够定时的获得各种更新信息，从而最大程度上保证信息的实时性，并且最重要的一点是不管主机连接那个server，zookeeper都会保证主机得到的是同一个视图，这就是zookeeper的最终一直性特性。Zookeeper的目的就在于此。本文并没有使用到zookeeper，故在此不做多余说明。点击上图中的hosts，可以查看集群的主机信息，点开后界面如图 2 - 2所示：

图 2 - 2 集群节点信息图



从这个截图可以看出，集群共有三台主机，主机名分别为master，slave2，slave4。各主机IP地址位于同一个网段内。本集群中slave4是主节点，包含所有的集群主要信息，可以看出集群的节点是3个。

其次，点击jobs，可以查看运行过的任务信息，比如任务的名字（通常在MapReduce程序中设置），任务的类型，本集群中类型一般包括MapReduce和Hive两种（其他的集群可能还会包括pig，streaming等等），其次是本次任务的调用者（一般是HDFS），最后比较重要的信息是本次任务的输入文件大小和输出结果的大小。如表表 2 - 2所示：

表 2 - 2 集群任务历史

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Job\_id | Name | Type | Input | Output |
| mr\_201508140850\_0622 | Demo25 | Mapreduce | 23.5GB | 13.7KB |
| mr\_201508140850\_0621 | Demo24 | Mapreduce | 23.5GB | 16.9KB |
| mr\_201508140850\_0620 | Demo23 | Mapreduce | 23.5GB | 25.3KB |
| hive\_hdfs\_2015082921 | Select max(id) from db | Hive | 33.1MB | <1KB |

从Ambari的集群管理界面可以获取到大量的集群的相关信息，例如哪些节点的datanode没有启动，哪些节点的tasktracker没有启动，以及每个节点的空间使用情况，当运行一个MapReduce任务的时候，能够查看任务运行的详细信息。从图 2 - 1可以看出集群中的3个节点的hdfs服务和MapReduce服务都运行正常。一旦集群出现异常信息，便可在此界面查看相应的异常信息。其次job运行的详细过程也可以在这个管理界面中看到。这不仅极大的方便了集群的管理，还为我们查看任务运行的状态提供了方便的接口。

## 2.2 Hadoop 简介

当今时代，想要大幅度的提高单台计算机的速度已经不太可能了，因为CPU的速度由于技术上的瓶颈，已经不可能再大幅度提升。人们一直希望通过增加计算机的数量提升运算和数据处理速度，例如希望同时在300台计算机上处理数据，让处理这批数据的速度变成0.1小时。Hadoop正是为此诞生的，Hadoop是一个分布式的计算框架，它的设计目的是在大量廉价的硬件设备组成的集群上运行应用程序。Hadoop计算框架的最终目的是构建一个能够进行分布式数据存储以及分布式计算的系统，并且这个系统需要具有很高的可靠性和良好拓展性。当今社会，云计算正在变得越来越流行，基于这一情景，越来越多的个人和企业正在了解并使用这一系统。

Hadoop平台主要用来处理海量数据，它实现了MapReduce一样的编程模式和框架，能在由大量计算机组成的集群中在各台计算机上运行海量数据并进行分布式运算。它处理的海量数据能够达到PB级别，并可以让应用程序在上千个节点中进行分布式处理，处理的方式是可靠的、高效的、可伸缩的。Hadoop是数据处理过程是可靠的，如果在计算过程中，有数据丢失或者节点宕机的情况出现，集群机会启动或维护多个需要处理的数据副本，并让其他的节点继续处理未处理的数据，这将确保失败的节点处理的数据还会得到正确的处理。Hadoop数据处理能力是可伸缩的，他能够处理GB，TB，PB甚至ZB级别的数据。Hadoop工作还是高效的，由于他的工作方式是并行进行的，也就是说，集群里的每一台机器都在进行着数据处理，采用这种处理方式可以明显的加快处理数据的速度。除此之外，Hadoop依赖于社区服务器，所以他的成本很低。

Hadoop自带Java语言编写的框架，在Linux平台上运行是非常理想的。Hadoop平台上的应用程序也可以用其他语言编写，如C++，Python，Ruby等等。

Hadoop族群包括很多项目，如表 2 - 3所示：

表 2 - 3 Hadoop集群组件

|  |  |
| --- | --- |
| 项目名称 | 项目说明 |
| HDFS | 分布式文件系统，是GFS的开源实现 |
| MapReduce | 分布式并行编程模型和程序执行框架，Google公司MapReduce的开源实现 |
| Hive | 一个数据仓库，将hdfs中的数据已关系型数据库的方式进行管理，支持sql操作，方面开发人员通过sql调用mapreduce程序 |
| Hbase | 一个基于分布式的按列存储的菲关系型数据库。利用HDFS集群上的数据作为数据支持，进行数据查询的时候可以调用mapreduce进行数据处理 |
| Mahout | 一个在Hadoop上运行的机器学习类库 |
| Zookeeper | 一个分布式、可用性高的协调服务。 |

本文中用到的Hadoop组件主要是Hadoop的核心组件HDFS和MapReduce，以及Hive，其他的组件我们暂且不讨论。Hadoop最根本的优势在于分布式计算。将海量的数据分布的存储在多台计算机上，并且计算这些数据的时候由这些分布式的计算机上单独的处理每一部分数据。最终将每一步分处理的结果汇总起来，这样就得到了最终的结果，这一切都由master来操作，而master本身并不作为数据节点和计算节点。Hadoop从根本上解决了单台计算机的性能瓶颈问题。

## 2.3 HDFS的体系结构

Hadoop分布式文件系统(HDFS)的设计目的就是让其运行在多台普通机器上的分布式文件系统。本集群系统上hdfs总体上由master进行调度，而具体的数据存储则由slave2和slave4来进行。

HDFS有着高容错性的特点，并且设计用来部署在低廉的硬件上，这从本论文所采用的集群中的机器配置就可以看出来。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集的应用程序。

1. HDFS有以下几个主要特点：

处理超大文件：这也是hdfs设计的初衷，本身就是为了存储海量数据而存在的，而hdfs存储的一个超大文件可以达到很大规模，甚至能达到PB级。

集群规模动态扩展：这是hdfs灵活性的体现，集群的机器可以动态的添加，而不影响集群的数据存储情况。

流式数据读写：HDFS的设计思想就是“一次写入，多次读取”，一个hdfs中的数据块，一旦被处理会被分发到各个节点上等待被处理。

运行于廉价的商用机器集群上：这是hdfs最出名的一点，HDFS设计时充分考虑了安全性和可靠性，所以hadoop对硬件的要求比较低，可以运行在多台配置比较低的机器上。

2.HDFS的局限性：

不适合低延迟数据访问： hdfs是为了存储大数据而存在，但是hdfs不适合处理低延时的流数据，对于时间要求很高的数据处理任务来说，还是不要使用hdfs进行数据存储了。

不适合存储大量的小文件： HDFS 文件系统的文件的数量会受到限制，这导致系统不能存储大量的小文件数据，如果存储了大量的小文件，在进行后期数据处理的时候，处理过程的也会很慢。

3. HDFS结构模型如图 2 - 3所示：

图 2 - 3 HDFS结构



一个HDFS集群是由一个NameNode和若干个DataNode组成的。本集群中namenode主节点有master承担，所以master是主服务器，管理着文件系统的命名空间以及客户端对文件的访问操作记录等；DataNode就是集群中的一般节点了，负责节点数据的存储，本集群中的datanode包括slave2和slave4两个节点。而用户处理数据的时候，会通过NameNode和DataNode各个节点交互访问文件系统，首先联系NameNode获得要处理的文件的元数。而主要的文件的I/O操作则是直接和数据节点DataNode进行交互。HDFS系统里的的文件通常是按照64MB被切分成不同的数据块（Block）的，每个数据块会被尽可能的分散存储在不同的DataNode中，若干个数据块存放在一组DataNode上。

4．HDFS的数据复制与存放

Hdfs在多台机器上进行分布式的存储数据，并且数据是以数据块的形式存在，每个数据块都会存储在不同的节点上，而每个数据块的存放是有一定的规则的。

首先，HDFS数据的复制。为了防止某一数据节点的宕机，HDFS会让用户指定每个文件的副本数目，这一数据默认是3个。其中数据的复制由NameNode来管理，每个数据块的副本会存储在其他的数据节点上。

## 2.4 MapReduce的体系结构

MapReduce，简单的理解就是一种编程模式，这种模式采用的是分布式的计算方法。MapReduce致力于解决大规模数据集的问题。它可以用在非常广泛的应用程序中，既然要解决大规模数据集的问题，就要考虑从一部分数据开始，利用局部分析的方法，将大规模数据集的问题分解成小部分数据的问题。也就是我们熟悉的分而治之的数据处理方法。就像上一结HDFS里我们讲到的一样，在用户进行数据处理之前，数据集已经分布在各个节点上了。我们在处理数据时，每个节点会优先处理存储在你本地的数据来进行map处理，map处理过后，再讲数据进行合并（combine），同时进行一定的排序（sort），最后分发到reduce节点上。

MapReduce包括两个核心的操作，map和reduce。简单来说，map就是一个映射的过程，从一组数据到另外一组数据。这个过程通过map函数来实现。简单举个例子进行乘以2的映射就变成了。Reduce过程则进行对映射后的数据进行规约处理，这一过程同样通过reduce函数来实现。如果对映后的数据进行求和计算则最终的运算结果就是20。概括来讲，map的任务就是将大任务分解成多个小任务，而reduce的工作就是将每个小任务处理的结果汇总起来。当然这个过程还有很多复杂的问题，如分布式存储、工作调度、负载均衡、容错处理和相关的网络通信等。这些事情都是MapReduce框架来解决的。不需要我们用户关心这些问题。MapReduce的处理流程图如图 2 - 4所示：

图 2 - 4 MapReduce的处理流程图



从图中可以看出，数据输入之后，被分割成不同的区块，然后通过map任务，分配给不同的节点来同时处理数据的不同部分，这就达到了分布式的效果。map处理完成之后，通过reduce将结果汇总。

Mapreduce框架由jobtracker和tasktracker组成，在本集群中jobtracker由master担任，slave2和slave4是两台tasktracker，所以这样来讲，slave2和slave4既是存储数据的数据节点，又是处理数据的处理节点，由于本文的集群规模比较小，无法完美的体现分布式计算的优势。在执行mapreduce程序的时候，slave2和slave4会优先处理存储在本机器上的数据，这部分的说明了mapreduce的优势，这种处理方式很大程度上解决了大量数据的IO操作，而IO操作恰巧是mapreduce的性能瓶颈。

## 2.5 本章小结

本章主要介绍了Hadoop基本的信息，详细说明了Hadoop核心组件HDFS的基本数据存储情况和一些特点。另外也介绍了另一核心组件MapReduce的基本框架和运行原理，还结合HDFS做了数据相关的分析。并且，本章进一步介绍了本文数据分析平台的基本情况，这也是本文数据处理的基础。以便在后来使用的过程中更深的了解。有了这些技术，本文的数据分析才能够得以进行。

# 第三章 基本数据分析

## 3.1 本文使用的数据集说明

### 3.1.1 数据来源

本文使用的数据集来源于一款测速软件产生的用户相关的信息。软件的测试分为两种，一种是用户端主动发起的，需要消耗流量的主动测试，另一种是不需要消耗额外的流量，只对程序消耗的流量进行监控，可以分别对用户消耗流量的APP进行监控，可以详细记录用户详细的流量使用行为，和使用APP的具体网速。被动流量监控主要针对的是APP使用网络流量进行的监控。当网络发生变化时，这部分数据进行全部更新；当小区发生切换时被动流量数据库全部更新；当网络和小区都没有发生变化时，只针对流量变化的APP数据库进行更新。如图 3 - 1所示

图 3 - 1 数据收集流量图



程序启动后，监控程序就已经在后台运行了，这一过程每天都会产生大量的数据。当产生了相关的数据或者数据更新后，保存到手机端的数据库之后，程序会定时每隔20s触发一次上传，上传到公网服务器的数据库中，上传的数据库主要有几个表，首先是被动监控各个app后台使用流量的数据表，我们称之为app\_traffic\_db，这也是本文数据分析的主要数据来源，还有一个数据表，是用户主动测试当前网速时，向数据库上传的数据表，我们称之为testinfo\_db，接下来也会使用一点。还有其他一些软件会收集但是本文不予分析的表。当数据传到公网服务器上的数据库之后，由于公网数据库的容量有限，只有1G的空间，上传的数据很容易就能把数据库填满，导致后来的数据会上传失败。其次，考虑到数据的安全性等相关问题。我们将实验室备份到了本地的数据库里。这一过程采用增量备份的方式。也就是，每天都会获取新增的数据，将新增的数据备份的本地数据库，同时将已经成功备份到本地数据库的数据，在公网服务器的数据库里将其删除。这样公网数据库也不会出现数据长满的情况。备份到本地数据库之后，由于数据量较大，MySQL已经处理不了我们的一些需求。进一步我们将本地数据库里的数据导入了Hadoop里的HDFS里进行数据存储。在此基础上，对数据进行分析。整个数据存储的过程，如图 3 - 2所示：

图 3 - 2 系统数据存储过程



### 3.1.2 数据格式

上一节我们讨论过，本文主要分析用到的数据表是app\_traffic\_db和testinfo\_db，其中被动数据监控表app\_traffic\_db是对手机上安装的其他软件的几个监控信息，具体内容如下表 3 - 1所示：

表 3 - 1 被动数据表app\_traffic\_db字段说明

|  |  |  |
| --- | --- | --- |
| 表项 | 数据类型 | 解释 |
| id | int(10) | 自增长的关键字 |
| package\_name | text | app包名（例如：com.tencent.qq） |
| app\_name | text | app的名称（例如：QQ） |
| uid | text | app的标识id（每个app唯一） |
| network\_type | text | Wi-Fi/mobile |
| mobile\_type | text | 具体移动网络类型（  network\_type为Wi-Fi的情况下，本列值为Wi-Fi，network\_type为mobile的情况下，本列可能的值为，GPRS，edge，hspa，lte等等） |
| cell\_id | text | 移动蜂窝网小区识别码 |
| wifi\_bssid | text | 一种特殊的Ad-hoc LAN的应用的id |
| start\_time | text | 本条记录程序监控起始时间 |
| end\_time | text | 本条记录程序监控结束时间 |
| upload\_traffic | text | 当次监控软件上传流量 |
| download\_traffic | text | 当次监控软件下载流量 |
| date | text | 监控日期 |
| time\_index | text | 详细监控日期时间 |
| imei | varchar(60) | 手机imei（也是用户的唯一标识） |
| sdk\_version | varchar(20) | 本测速软件的SDK版本 |
| gps\_lat | varchar(20) | 定位纬度 |
| gps\_lon | varchar(20) | 定位经度 |
| location\_type | varchar(60) | 定位类型 |

这个数据表，相对来说数据量比较大，用户只要安装了这款软件，软件就会在后台时刻进行着监控其他软件的相关信息，并且定期上传监控的信息。本文数据处理用到比较多的数据项是app\_name（处理app相关信息的根本参考）、network\_type（根据网络类型，来区分WiFi或者移动网络，进而分析WiFi情况下和移动网络情况下用户使用情况的变化）、start\_time和end\_time（这两项是时间处理的根本依据）、upload\_traffic和download\_traffic（这是本文中的处理的部分，上行流量和下行流量）、imei（区分用户的唯一标示）。和这个数据表不同的是，另外一张数据表主动测试数据表testinfo\_db，数据量就比较小，因为只有当用户安装了这款软件并且使用软件进行了一次测速之后，软件才会上传本次测速相关的一些信息，比如测得的网速，网络类型等等，具体上传信息参见表 3 - 2：

表 3 - 2 主动数据表testinfo\_db字段说明

|  |  |  |
| --- | --- | --- |
| 表项 | 数据类型 | 解释 |
| id | int(10) | 自增长的关键字 |
| ping | varchar(10) | 时延(单位：ms) |
| ave\_downloadSpeed | varchar(10) | 平均下载速度(单位：KB/s) |
| max\_downloadSpeed | varchar(10) | 最大下载速度(单位：KB/s) |
| ave\_uploadSpeed | varchar(10) | 平均上传速度(单位：KB/s) |
| max\_uploadSpeed | varchar(10) | 最大上传速度(单位：KB/s) |
| rssi | varchar(10) | 信号强度（单位：DBM） |
| gps\_lat | varchar(20) | 测试点的经度 |
| gps\_lon | varchar(20) | 测试点的纬度 |
| location\_type | varchar(60) | 定位类型 |
| imei | varchar(60) | 手机的imei |
| server\_url | text | 测试服务器地址 |
| ant\_version | varchar(60) | app版本号 |
| detail | text | 详情 |
| time\_client\_test | varchar(20) | 客户端测试时间 |
| time\_server\_insert | varchar(20) | 服务器端插入时间 |
| networkType | varchar(10) | Wi-Fi/4G/3G/2G |
| operator\_name | varchar(20) | CMCC/CUCC/CTCC |
| wifi\_bss\_id | varchar(20) | Wi-Fi bssid |
| cell\_id | varchar(60) | plmn+lac+cid/plmn+sid+bid+nid |
| province | varchar(20) | 测试点的省份 |
| city | varchar(20) | 测试点的城市 |
| street | text | 街道地址 |
| location\_detail | text | 定位详细信息 |
| upload\_traffic | varchar(20) | 上行测试流量 |
| download\_traffic | Varchar(20) | 下行测试流量 |

## 3.2 相关的数据分析

相关的数据分析旨在更加了解数据集的情况，并且从数据集中，对用户的分析可以得到用户在流量使用上的一些信息，而对数据集中app的分析可以清楚的了解到app的相关信息。对这两方面分析的结果有助于更加深刻的认识本文的价值。前面已经讨论过相关的数据信息和数据的相关流程信息，在前面讨论的这些内容的基础上，我们开始对数据进行初步的分析。这些分析的主要步骤是，编写MapReduce程序或者直接使用Hive对数据进行处理，然后，将处理之后的结果从HDFS上面拷贝到本地服务器的本地磁盘上。为了对处理结果进行展示，能够让其他人也看到我们的处理结果，本文处理的数据结果又被上传到了公网服务器上（此服务器和之前用户上报监控数据的服务器是同一个服务器），在服务器上通过一些程序将数据展现出来，展示方式包括柱状图、饼图、折线图等等。展示地址为，<http://buptant.cn/autochart/>。这一流程如图 3 - 3所示

图 3 - 3 数据处理展示流程图

 这一部分的数据分析，从数据的总体情况，数据集中的用户情况和数据集中的app情况进行了全方位的分析，有利于更好的了解数据集的情况，对本文后面的分析有很大的帮助。

### 3.2.1 数据总体情况分析

截止到本文完成之前，数据集中统计的总的WiFi流量已经达到15TB的级别，移动网络流量也达到了将近2TB的量。数据集中涉及的用户遍布全国各地，并且用户数已经达到了好几万人，这些用户的网络包揽了中国移动、中国联通、中国电信三个运营商的各种网络以及其对应的各种WiFi网络。另外，数据集中的app数目已经达到了数十万的级别，这些app包揽了视频、音乐、浏览器、下载以及其他好几种类别。除此之外，被动监控数据表app\_traffic\_db中的监控记录已经达到了将近**12000000**多条。下面，本文将对数据集中的一些情况做简要的分析。

首先本文分析了各网络制式下的用户数，也就是说，使用不同网络制式的用户各有多少，这里的网络制式，是本文前面数据表里提到过的具体的网络信息，统计方式，就是以网络制式为分组然后计算每个分组下不同的IMEI个数，每个IMEI对应着每一个单独的用户。图 3 - 4是统计结果：

图 3 - 4 不同网络制式的用户数

截止到2015年6月份的数据显示，中国移动6月份新增用户数86万多，而所有的用户总数累计达到8.17多亿户，2015年这一年累计净增客户数一千多万；另外，由于4G网络的逐渐普及，截止到6月份，移动公司3G用户减少了786.8万户，3G用户总数还剩下大约不到2.15亿户；但是，移动4G用户净增1932.7万户，总数达到了将近1.9亿户，这说明大量的3G用户正在变成4G用户。

中国联通公司6月的运营数据表明，联通公司移动电话用户减少了将近100万，总用户数将近2.9亿；其中，3G和4G用户净增500多万，用户总数接近1.58亿；另外，联通公司的2G用户数正在大幅减少，已经减少了大约600万人，所有的2G用户还剩下1.3亿人多一点；并且随着智能手机的普及，使用固定电话的用户越来越少，联通公司的固定电话用户数净减40多万，累计降至7800多万户；但是联通公司宽带用户数净增28万多户，累计已达到7059万户。

中国电信6月份运营数据表明，电信公司6月份，移动电话用户净增76万人，移动电话总用户达到1.9亿多，逼近2亿大关。其中，电信公司的3G和4G用户在这个月增加了182万人，3G/4G用户总数为达到了1.31亿，和联通的用户比较的话，少不了多少。电信公司的固定电话用户也在减少，固定电话用户数净减59万，但是还剩下将近1.4亿；电信宽带用户数当月净增38万，累计将近1.1亿户，比联通的宽带用户要多很多。

图 3 - 4中1xRTT属于中国电信网络的2.5G网络，对应中国联通，中国移动的2.5G网络是EDGE。Ehrpd网络是电信3G向4G演进过程中的一个过渡。还有一点需要说明的是，同一用户的网络可能会跳转，比如一个中国移动4G网络的用户，在网络不好的时候可能会变为TD-SCDMA的3G网络，甚至会变为EDGE网络，这都是有可能的。从这个图 3 - 4里可以看到，电信用户在三大运营商里还处于少数。

图 3 - 5统计了每天活跃的用户数

图 3 - 5 每天活跃的用户数

从图 3 - 5中可以看出，用户数大致趋于变少，这是由于这款测速软件并没有进行相关的宣传以及后期并没有足够的时间和精力对这款软件进行维护，其次最重要的一点，测速软件并没有很大的用户粘性，大部分用户在进行测速之后，都会选择删除测速软件，种种原因都导致用户数越来越少。在做相关的数据分析时，本文会对这一特点进行评估，以保证相关推荐的合理性和正确性。从图中可以看出在2014年11月24号和2014年12月22号两天，数据有一个很大的低谷，从后来的分析中了解到，产生低谷的原因是，公网数据库数据已经满了，而本地备份后并没有及时的删除已经备份的数据。这样一来，用户产生的数据就不能存到数据块里，导致数据丢失。

图 3 - 6统计了，每天监控到的其他软件的总的流量信息

图 3 - 6 每天的流量信息

从图 3 - 6可以看出，用户在使用网络上网时，大部分用户还是选择使用WiFi网络，这都是因为WiFi网络对手机来说是免费的，并且WiFi的速度要比普通的移动网络速度要快。尽管现在的各大运营商都在声称调整相关的资费，并且提升网速，但是实际上还是换汤不换药，本质上根本没有什么改变。毕竟用户每个月的流量是有限的，并且如果使用流量看视频，用户每个月的包月流量几乎在一天之内就会消耗完毕。移动网络流量始终还是一个限制用户上网的关键因素，所以在这种情况下，为用户推荐流量少的应用是很必要的，毕竟WiFi网络，还没有移动网络那么普及。

图 3 - 7统计了，每天上传的监控的记录数量的情况，这个记录和每天活跃的用户目基本上成正比，虽然用户不多，但是用户产生的记录数还是有一定的数量的，这也很好的为我们的分析提供了数据支撑

图 3 - 7 每天监控的记录数

图 3 - 8统计了个网络制式下的一个实际使用中的平均速度，从表格中可以看出，4G（LTE）网络的网速是最快的。自从2013年12月4日下午，工信部正式向中国的三大移动通信运营商颁发了TD-LTE制式的4G牌照以来，中国电子通信行业慢慢进入了4G时代，对普通用户而言，4G可以带来更快的网速，其网速是3G网络的10倍以上，在4G时代，快速的网络传输能支持很多高清视频和更多应用。4G的理论速度达到了100Mbps甚至以上，但是实际使用中，速度远未达到官方生成的速度。也可能4G网络覆盖不全，信号不好可能会导致某些地方4G网速很低，多以我们统计到的4G网络平均速度不是很高。但也已经超出了3G网络的速度，并且比WiFi网络的速度也高出不少。而除了4G网络外，统计到的WiFi网络的速度是最快的，比平均的三个运营商的3G速度快了不少，这在一方面也说明了用户更喜欢使用wifi网络来上网，毕竟WiFi网络网速快，还不计流量。除此之外三个运营商的3G网络是用户上网的主力网络。虽然已经进入4G代，但从网络覆盖以及用户手机对4G网络的支持情况来看，一部分的用户手机还不支持4G网络，其次4G网络的覆盖情况不如3G网络好。所以3G网络依旧是用户上网的主力。

图 3 - 8 各制式网络速度

以上几个图表，从数据集的大小方面对数据做了一个统计意义上的分析，通过这些分析，我们可以对数据的规模情况有一个基本的了解，接下来的几个小节，出发点会更深一步。

### 3.2.2 用户方面的情况分析

本文的主题在于为用户推荐相关的应用情况，所以针对这个数据集，对数据集中一些基本的用户情况进行分析还是很有必要的。从下面的一些分析中我们可以对用户在流量使用方面有一个基本的了解，除此之外，我们还可以在用户的流量使用模式上有一个基本的认识。下面是分析的具体情况。

图 3 - 9对用户平均每天在移动网络和WiFi网络下的上网时间做了统计，然后把统计的时间归在了如下所示的6个区间内，从该图分析结果可以看出，大部分用户每天上网的时间段分布在100-1000s和1000s-10000s这两个区间内，考虑到实际的数据集的矩阵的稀疏性。部分用户的数据收集时间比较短。具体来说就是用户安装上软件之后，进行了一次测速就把软件删除了。这样的结果就是用户上网的时间可能比记录到的时间要长。不过，图 3 - 9的分析结果还是能说明一定的问题。

图 3 - 9 用户上网时间分布图

其次我们分析了，月均流量这个概念，对用户平均一个月消耗的流量做了一个基本的统计，结果如图 3 - 10所示：

图 3 - 10 移动网络月均流量消耗图

图 3 - 11是WiFi情况下的：

图 3 - 11 wifi月均流量消耗图

图 3 - 10和图 3 - 11两幅图里可以看出，移动网络情况下，大多数用户月均流量消耗集中在10M-100M，100M-1G，以及1G-10GB之间，并且100M-1G之间的用户数量最多，与之对应的WiFi情况下，用户平均每月在使用WiFi网络时花费的流量大约集中在100MB-1GB，1GB-10GB，以及10GB-100GB之间，由此可知，用户在WiFi上花费的流量比在移动网络上花费的流量整整高了一个数量级。所以，用户在移动网络上使用流量还是有很大的限制的，如果我们能为用户进行相关的推荐，比如说在移动网络每月100MB-1GB流量的情况下，使用原来类别下其他APP能够体验到，使用原来APP每月1G-10G流量的体验。这样一来，就能够让用户花费较少的流量体验相同的使用感受，这也就达到了本文的目的。

### 3.2.3 APP方面的情况分析

由于在后面我们需要对app相关情况进行分析，在这里先大体对app的总体情况了解一下。

图 3 - 12对不同的app每天的进行流量传输的时间进行了分析。从图 3 - 12中可以看出，绝大多数需要联网的app每天联网的时间聚集在10-100s和100-1000s这两个区间内。这个结果并不奇怪。试想一下，我们每天经常使用的app的类别是很少的，数目也不会很多，大多数app用户是不经常使用的。而图中使用时间在1000s以上的app的数目虽然站的比例不多，但也有好几百的。

图 3 - 12 app平均每天消耗流量区间分布

图 3 - 13和图 3 - 14两幅图是本文统计的，分别在移动网络和WiFi情况下，不同app平均每天消耗的流量分布图情况。通过这两幅图我们可以对不同的app每天的在不同的网络下的流量消耗情况有一个基本的了解。

图 3 - 13 移动网络不同app平均每天消耗流量分布图

图 3 - 14 WiFi下不同app平均每天消耗流量分布图

从图 3 - 13和图 3 - 14两幅图可以看出，移动网络情况下，每天消耗流量1M的app种类占到了大约85%。这也可以理解，一方面，由于某些app的使用频率不高，每天可能就打开一次，对于大部分应用程序来说都不需要传递很多数据，所以消耗的流量也不是很多，这一部分应用程序，对用户的使用习惯影响可能也不大，我们不需要密切关注。而消耗流量大于1M，小于10M，以及日均消耗流量在10M到100M之间的app，这一部分app的种类大约占得比例在百分之十几，从用户的使用情况来看，这一部分app是我们需要密切关注的app。与之对应的wifi情况下app的分布情况，与移动网络下稍有不同，日均销量流量在1K-1M之间的app占得比例大约在75%，较移动网络有所下滑，在日均消耗流量在1M到10M以及10M到100M之间的app比例攀升到了接近25%。这和移动网络的情况区别还是很大的。

由上面的分析，我们了解到用户在移动网络下和WiFi网络下使用app的一个变化，那就是在WiFi网络下，用户会毫无顾虑的去使用各种各样的app，而从来不需要去考虑流量的问题，这也是很正常的，毕竟在wifi网络下，流量是不需要额外花钱的。有的用户可能是用家里的宽带搭的无线路由器，这样的话，上网是不计流量费的。相对来说很划算。而在移动网络下，流量是用有时候就会显得捉襟见肘了，毕竟用户在使用移动网络时，每个月的流量都是固定的，超过了就得再花钱买，并且很多用户每个月的流量包不一定够用，这就导致了用户在使用移动网络上网的时候，使用app时格外的小心。花费流量大的app是不敢使用的。由于这种情况的存在，如何让用户在移动网络下能够放心的使用各种app，而不必太过于担心流量问题，正是本文需要考虑的问题。本文接下来的内容将围绕着这个问题继续展开讨论。

## 3.3 本章小结

本章内容主要是基于现有的数据，对数据的基本情况做一下介绍，然后从用户的角度，以及app的角度，分析了WiFi网络和移动网络情况下，用户和app方面的一下区别，WiFi网络下，用户不用考虑流量的问题，而能够放心的使用各种app，移动网络下，用户流量使用情况就会有所收敛。基于这种情况，也正论证了本文提出的问题，如何能让用户在使用移动网络的时候，能花费更少的流量而又不丧失原来的app使用体验。最好能让用户在移动网络下体验到WiFi网络下的使用体验，这是本文努力研究的目标。

# 第四章 推荐模型的建立

在已知用户所使用的各类app的情况下，其中包括用户的app使用时间，使用频度等等，研究用户的使用模式及偏好。关于偏好模型的建立，首先需要对app进行分类，本文分类的原则很简单：第一，数据集中有软件名称这一项，第二，参考各大应用市场对软件的分类情况。将相应的软件属于的类别打上标签，以便在以后使用。本文对app分的类主要包括视频类，软件下载类，音乐类以及浏览器类这四大类。对于社交类软件，用户本身对社交类的软件依赖较大，并且使用频繁，用户的朋友关系大多维系在社交类软件上，用户很难改变社交类软件的使用习惯，所以本文暂且不对社交类软件相关app进行推荐。而其他几类软件都有很多类似的软件可供用户选择，具有很高的推荐价值。

推荐模型的建立过程为，第一、分析用户使用偏好模型，这一步主要分为三个部分，首先是基于使用时间的用户使用偏好，这一部分是根据用户对某一类app使用时间的长短来量化用户在时间长短方面对某一类app的喜好程度，具体的量化方法采用的是逻辑回归（下面会具体介绍逻辑回归）。其次是分析用户使用这款软件的频度，文章中频度的定义是使用某款软件的天数和总的天数的比值。最后一部分是用户使用软件的时间段的分析，在前面两项分析的基础上可以分析出用户app的使用偏好，但是为了个性化更强一点的推荐，特意加上使用时间段这一参数，为喜欢在不同时间段使用app的用户推荐相同时间段下的类似的app。

第二、分析app的相关特征，这一步也是分为三个部分，首先是app消耗流量的计算，这也是本文着重推荐的点，只有弄清楚了app的流量值，才能进一步推荐适当的app。其次，分析app的流行度，只考虑app的流量的话，推荐结果太单一，并且不实际，没有说服力。只有将app的流行度考虑在内，才能达到理想的效果。但是如何将app的流量和app的流行度结合起来，也就是如何为用户推荐流量少并且流行度高的app是一个难题，这一部分用到了资产投资组合理论（文章稍后有详细介绍）。基于这两部分会得出一个app的推荐列表，推荐工作将会基于这个列表进行。除此之外，前面说过对用户有一个时间段的分析，这一部分对app也需要有个时间段的分析，只有分析了app的活跃时间段，才能对用户在相应的时间段进行相关的app推荐。整个推荐过程的流程图如图 4 - 1所述：

图 4 - 1 推荐流程图



## 4.1 用户使用偏好模型的建立

使用偏好旨在分析用户是否对某一类软件的使用喜好超出了大众的平均使用情况，并根据超出的情况来选择性的为用户进行推荐。本文的使用偏好模型和用户对某一类软件使用的时间，使用的频度有关。我们把使用偏好用来表示，则，其中是指用户在对某一类软件使用时间方面的评估，得出用户在某一类软件的时间方面的偏好程度。是指在某一类软件上使用频率方面的评估，得出用户在某一类软件的频度方面的偏好程度。这两者之间并不存在干扰，使用时间长并不影响使用频度，反之亦然。偏好模型的建立是本文分析的一个基础。

### 4.1.1 使用时间的分析

衡量一个用户对一款软件的使用喜好程度，最直接的一个体现就是，用户在一款软件上花费的时间，时间能够说明一切问题。用户在一款软件上花费的时间越长，则能说明用户比较喜欢使用这款软件。但是这个时间的长度该怎么去划分是一个问题，也就是说，用户一天内在一款软件或者说一类软件上使用多长时间算是喜好这款软件，使用多长时间又算是对这款软件使用的还算可以，这个度还需要去权衡。在实际情况中，只有比较才能得出一个用户是不是比另一个用户更喜欢使用某一款或者某一类软件。在得到每一个用户在某一款软件或者某一类软件平均每天花费的时间后，就可以进一步处理这个问题。首先，统计出平均每一个用户在某一类软件上平均每天的使用时间。以视频类软件为例，先统计每一个用户每天在在视频类软件上使用的时间，得到一个时间集，这个时间集就是一个用户每天花费在某一类软件上的时间，然后统计每天在视频类软件上使用的平均时间，也就是对这个时间集内的每个时间求平均，得到，其中

 (4-1)

用户使用的天数，为用户编号，如表 4 - 1所示

表 4 - 1 用户平均时间统计

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 天数  时间  用户 | 第1天 | 第2天 | 第3天 | ……… | 第n天 | 用户平均时间 |
| 用户1 |  |  |  | ……… |  |  |
| 用户2 |  |  |  | ……… |  |  |
| 用户3 |  |  |  | ……… |  |  |
|  |  |  |  | ……… |  |  |
| 用户m |  |  |  | ……… |  |  |
| 平均时间 |  |  |  |  |  |  |

这样就得到了每个用户在视频类软件平均每天的使用时间，有了这个时间集，就可以比较每个用户在这一类软件上使用时间的异同，就能够进一步区分这个不同。把所有用户在视频类软件每天的平均使用时间做一个平均，就得到了平均每个用户在视频类软件上平均每天花费的时间，其中

 (4-2)

为用户的人数。有了这个最终的平均值，就可以将不同的用户对这一类软件的使用程度进行区分了。用户每天在视频类软件平均每天的使用时间去减去平均每个用户在视频类软件上平均每天的的使用时间，就得了用户平均每天在视频类软件使用的时间和平均值之间的一个差值，这里记为，其中

，这里有了这个时间差值，只能将不同用户在这一类软件上的使用时间区分出来，只能说明一个用户比另一个用户在这一类软件上花费的时间更多或者更少。上面得到的差值，可能很大，也就是说用户在这一类软件上每天花费的时间远远超过了平均值；也可能为0，这时意味着用户在这类软件上花费的时间处于一个平均水平；更有可能为负值，也就说这类用户在这类软件上花费的时间很少，这类用户也不是我们将要关心的对象，可以忽略掉。这个差值无法定量的表现问题，如何将这个差值，这个不同通过概率或者什么方式表现出来是一个问题。经过大量的调查研究，发现逻辑回归是在这种情况下比较常用的一种方法，逻辑回归用于估计某种事物的可能性。比如某用户购买某商品的可能性，某病人患有某种疾病的可能性，以及某广告被用户点击的可能性等。着这里逻辑回归表示用户使用某一类软件的可能性。

下面引入逻辑回归

 (4-3)

为上面计算出来的差值，其图 4 - 2所示

图 4 - 2 逻辑回归图

通过逻辑回归，将用户平均每天的使用时间和平均每个用户平均每天的时间映射到这个区间内，从而判断出用户使用这一类软件的可能性。从图中可以看出差值越大，则通过逻辑回归计算出来的值就会越大，也就是用户使用这类软件的可能性越大，反之亦然，而当差值为0的时候，也就是之前文章讨论过的那样，这时说明用户在这一类软件的使用时间处于平均值，通过逻辑回归计算出来的值是0.5，这也是符合常理的，差值为0，相比差值比较大的用户和差值比较小的用户，0.5的可能性，还是正确的。从上面的分析我们得出用户在时间使用方面的偏好程度为

 (4-4)

根据这个公式，就能够将用户在某一类软件上的使用时间，映射到一个概率值上上，使用时间比平均值越大，计算出来的值越大，反之相同。这个计算出来的值能够反映用户在某一类软件上使用的时间，进而能够部分反映用户在这类软件上的使用偏好程度。

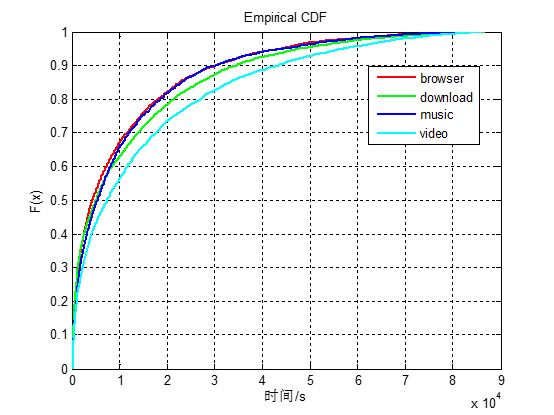
图 4 - 3 用户平均每天花费在各类软件上的时间分布

图 4 - 3是用户平均每天花费在各类软件上的时间分布cdf图，从图 4 - 3中可以看出来，大部分的用户，每天花费在应用上的时间，无论是哪一类应用，人数比较多的是10000-20000秒这个区间段，除了这个区间段，一直到50000s区间，都有一定的用户，只是用户相对于前面的区间正在变少。也有一些用户每天花费在某一类软件上的时间很少或者很多。就不同类别的APP来说，从图片可以看出，用户分布情况差不多，但存在轻微的差别，从图中可以看出，浏览器类和音乐类用户的时间分布情况基本吻合，只是存在极少的差别。下载类和视频类和前两种的分布情况相差大一点，主要的区别在于用户在10000s-50000s区间内的分布更平均一些。

图 4 - 4 用户平均每天花费在各类软件上的时间分布

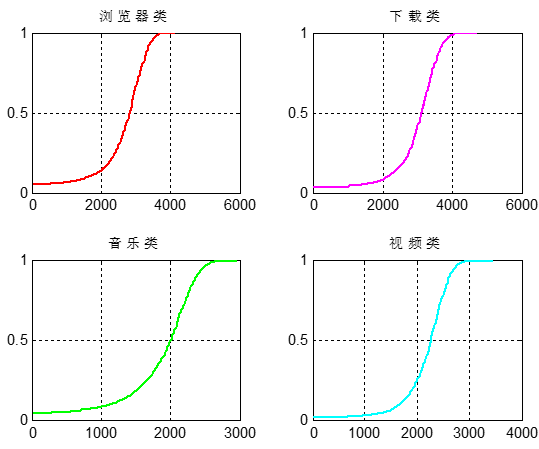


图 4 - 4列出了各类app下，用户在每类app下消耗时间的偏好图。其中横坐标是用户的编号值，其中用户你的编号值是根据用户的花费时间排序得出来的，编号值越大则说明用户使用某一类app的时间越多，则对应的偏好值越大，纵坐标是各个用户对当前类别app的使用时间偏好值。计算方式是先把用户在各个类别下的平均app使用时间计算出来，然后再和所有人的总的平均时间作比较。得出的结果再经过逻辑回归将用户对某一类app的使用时间偏好值映射到0~1内，就得到了上面的图。根据上图的结果可以找到对某一类app的使用偏好情况。

### 4.1.2 使用频度的分析

除了使用时间能够说明用户对某一类软件的使用偏好外，使用频度也能在一定程度上说明这个问题。但是使用频度的分析，相对来说比较简单，由于很难确切的定义一天使用多少次某一类软件就说明使用该类软件很频繁。也很难说一天内使用某一类软件次数比较多，而使用另外一类软件次数少，我们就设定用户对于使用次数多的软件更喜欢用，这种假定也不是很合理。比如用户一天内只使用了一次视频软件看了2个小时的电视剧，在此期间用QQ进行了几次聊天，这种情况无法判断用户更喜欢用哪一款软件。根据本文使用的数据集情况，由于对于一些用户的软件使用情况的搜集信息表明，从开始收集信息的日期，到最后用户卸载这款可以收集用户信息的软件，在这一段时间里，从收集到的信息可以看出，用户每天会使用多款软件，但是有些软件，用户并不是每天都在使用。也就是说，用户使用某些软件的时候有可能每天都在用，也有可能用过一次就不在使用了。基于这种情况本文将使用频度定义在每一天，也就是说我们统计的时间天数里，统计用户使用某一类软件的天数，然后计算使用某一类软件的天数和统计的总的天数的比值，用这个比值来表明使用一类软件的频率。我们用来表示统计到的用户记录的总的天数，来表示用户使用某一类软件的天数，这样的话频度方面的偏好程度

 (4-5)

其中表示用户使用的天数，表示观察到的所有天数。根据这个比值，就可以确定用户在某一类软件的使用频度情况。具体情况如表 4 - 2所示：

表 4 - 2 用户偏好表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 用户 | 浏览器类偏好 | 下载类偏好 | 音乐类偏好 | 视频类偏好 |
| 35451605358714 | 0.835821 | 0.537313 | 0.761194 | 0.850746 |
| 91990050401072 | 0.5625 | 0.6875 | 0.3125 | 0.6875 |
| 99000084137365 | 0.861111 | 0.638889 | 0.388889 | 0.097222 |
| 99000310518357 | 0.545455 | 0.5 | 0.454545 | 0.363636 |
| 99000310681745 | 0.354839 | 0.645161 | 0.677419 | 0.419355 |
| 99000310748469 | 0.26087 | 0.565217 | 0.26087 | 0.347826 |
| 99000311352186 | 0.333333 | 0.444444 | 0.277778 | 0.388889 |
| 99000311928648 | 0.25 | 0.642857 | 0.107143 | 0.642857 |
| 99000314243090 | 0.5 | 0.5 | 0.2 | 0.35 |
| 99000316757395 | 0.586207 | 0.655172 | 0.206897 | 0.586207 |
| 99000346392873 | 0.586957 | 0.673913 | 0.369565 | 0.782609 |
| 99000455162310 | 0.706767 | 0.924812 | 0.030075 | 0.225564 |
| 99000455366895 | 0.285714 | 0.47619 | 0.238095 | 0.52381 |
| 99000472062730 | 0.746032 | 0.84127 | 0.555556 | 0.777778 |
| 99000477475035 | 0.484848 | 0.69697 | 0.393939 | 0.090909 |
| 99000477639845 | 0.807692 | 0.807692 | 0.096154 | 0.692308 |
| 99000478840453 | 0.84 | 0.64 | 0.106667 | 0.866667 |
| 99000505049514 | 0.583333 | 0.333333 | 0.25 | 0.541667 |
| 99000519996397 | 0.434783 | 0.565217 | 0.347826 | 0.217391 |
| 99000535404382 | 0.560976 | 0.756098 | 0.585366 | 0.317073 |
| 99000535555981 | 0.466667 | 0.833333 | 0.666667 | 0.283333 |
| 99000536705736 | 0.388889 | 0.444444 | 0.388889 | 0.222222 |
| 99000537317820 | 0.210526 | 0.473684 | 0.421053 | 0.315789 |
| 99000550302332 | 0.555556 | 0.555556 | 0.62963 | 0.592593 |
| 99000554242668 | 0.5 | 0.4 | 0.45 | 0.5 |
| 351585050517296 | 0.409091 | 0.272727 | 0.272727 | 0.545455 |
| 351867050367645 | 0.911602 | 0.944751 | 0.143646 | 0.618785 |
| 351867053698368 | 0.553571 | 0.821429 | 0.660714 | 0.821429 |
| 351910054307694 | 0.774648 | 0.859155 | 0.788732 | 0.84507 |
| 351941068941848 | 0.366197 | 0.661972 | 0.43662 | 0.859155 |
| 352110052493070 | 0.642857 | 0.642857 | 0.642857 | 0.642857 |
| 352123054993799 | 0.482759 | 0.655172 | 0.655172 | 0.655172 |
| 352166057174529 | 0.473684 | 0.421053 | 0.473684 | 0.421053 |
| 352178050304300 | 0.444444 | 0.62963 | 0.222222 | 0.296296 |
| 352203061951608 | 0.352941 | 0.411765 | 0.352941 | 0.411765 |
| 352265054753556 | 0.919355 | 0.653226 | 0.879032 | 0.919355 |
| 352265059261407 | 0.521739 | 0.478261 | 0.173913 | 0.565217 |
| 352315055917344 | 0.677419 | 0.677419 | 0.677419 | 0.677419 |
| 352315056736651 | 0.666667 | 0.755556 | 0.688889 | 0.777778 |
| 352315056803253 | 0.911504 | 0.867257 | 0.265487 | 0.539823 |
| 352315057376622 | 0.166667 | 0.458333 | 0.458333 | 0.583333 |
| 352315058122678 | 0.333333 | 0.222222 | 0.444444 | 0.444444 |
| 352315059660676 | 0.848485 | 0.80303 | 0.727273 | 0.151515 |
| 352317051430041 | 0.592593 | 0.481481 | 0.111111 | 0.62963 |
| 352343051301719 | 0.717391 | 0.782609 | 0.456522 | 0.478261 |
| 352343055439978 | 0.319149 | 0.87234 | 0.457447 | 0.893617 |
| 352343059719243 | 0.297872 | 0.765957 | 0.765957 | 0.787234 |
| 352345050443210 | 0.388889 | 0.388889 | 0.388889 | 0.444444 |
| 352443060428014 | 0.592593 | 0.185185 | 0.407407 | 0.62963 |

表 4 - 2给出类数据量较多的50个用户在不同app类别下的使用频度数据。

根据用户使用某一类软件的可能性和用户使用的频度，我们确定用户对某一类软件最终的使用偏好,

 (4-6)

这个值是本文分析的一个基本，文章后面的推荐将基于这个值进行相关的推荐。

### 4.1.3 用户使用时间段的分析

这一小节旨在，将用户的上网时间段纳入用户使用app偏好的范畴。以视频类软件来说，不同的用户可能会在不同的时间段选择看视频。比如说，有的用户可能喜欢在上下班的路上看视频，有的用户则喜欢在中午休息的时候看视频，更有可能有的用户喜欢在晚上躺在床上看视频。鉴于这种情况，在为用户推荐相关app的时候，就需要考虑时间段的问题。从另外一个方面来讲，对于前面讲到的使用偏好相同的用户，由于数据集设计到的用户相关的参数信息比较少，则推荐的时候可能推荐的app是相同的。这就达不到个性化推荐。为此，在前面讨论的用户使用时间长短的偏好和频度方面，再加上用户使用时间段的分析，也就是说，如果用户偏好使用视频类软件，则分析该用户在什么时间段使用视频类的软件就能更好的为用户推荐，具体的时间段分类如表 4 - 3所示：

表 4 - 3 时间段分类表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 用户 | App类别 | 时间 | 时间段分类 | 时间段分类标记 |
| 用户1 | 视频类 | 7:00-10:00 | 早晨 | A |
| 用户2 | 视频类 | 10:00-12:00 | 上午 | B |
| 用户3 | 视频类 | 12:00-14:00 | 中午 | C |
| 用户4 | 视频类 | 14:00-18:00 | 下午 | D |
| 用户5 | 视频类 | 18:00-20:00 | 晚上 | E |
| 用户6 | 视频类 | 20:00-00:00 | 晚间 | F |
| 用户7 | 视频类 | 00:00-7:00 | 深夜 | G |

再为用户确定了使用偏好之后，在给对应的使用偏好打上时间段分类标签，这样就可以对应用户的偏好以及在该偏好下的使用时间段来为用户推荐相关的app。如果该时间段内没有相应的app可用于推荐，则选取近似时间段内标签的app进行推荐。

## 4.2 APP画像的分析

在对用户相关软件使用偏好做了相关的分析之后，了解了用户在使用app时的一些行为特征，接下来对app的一些特征做一些分析。对app的分析主要分为以下几个方面：首先分析的是最重要的一点，那就是app的平均流量消耗，这也是本文分析的一个根本问题，本文需要找出消耗流量最少的app并且符合用户行为特征，并对用户进行相关app推荐。在找出消耗流量最少的app之后，还需要了解app其他的信息，以完善app画像的分析。其次，需要分析app的使用时间，对一个用户来说，在什么时候用什么样的app是很不同的，以视频类app来讨论，有的用户可能在中午喜欢观看视频，有的用户可能在晚上喜欢观看视频，这时就需要将各个时间段中不同视频类app的使用情况加以分析，然后对在不同时间段内观看视频的用户推荐相同时间段内同类别其他消耗流量少的app。最后，需要分析app的流行度，本文中分析使用app的人数，本节的重点在于将同一个类别下app消耗的流量情况和app流行度的关联耦合，如何将消耗流量少但是流行度高的app推荐给用户，是本节将要讨论的话题。

### 4.2.1 APP流量分析

本小结对一款app的流量情况的衡量方法和之前分析讨论的用户使用时间的分析类似，具体方法就是统计出平均在某一类软件中的每一款软件的平均每个小时的消耗流量，这里采用小时作为分析的周期，是考虑到一些app可能在不同的时间段内访问的资源不尽相同，并且不同的用户每天使用app的时间长短并不一样，无法将所有用户的在一天内的使用情况同一分析处理，但是将分析维度限制在一个小时以内的话，可以有效的减少这种差异。分析方法如下，以小时为维度，将数据集中所有出现过的某一类下的相关app的流量信息统计起来，将总流量记为，其中为统计的小时出现的次数，的意思是指应用apps在个小时内的总流量，则平均每个小时的流量表示总流量除以小时个数。

则其计算方式如下所示

 (4-7)

根据这个公式就可以得到每个app平均每小时消耗的流量。

图 4 - 5 浏览器使用流量排行

图 4 - 5计算了浏览器类app每小时流量消耗情况，并对流量消耗进行了排名，从图中可以看出，浏览器的流量消耗差别还是很大的，每小时消耗从十几兆到几兆不等。从计算结果来看，浏览器的推荐还是很有价值的，毕竟不同的浏览器功能基本上都是一样的，找出他们的流量差别至关重要。基于这个流量差别就可以为用户进行相关的推荐。

图 4 - 6 下载类软件流量消耗情况

图 4 - 6计算了应用下载类软件每小时的流量消耗情况，下载类软件的流量的消耗和浏览器类相比，流量消耗差距就没有那么大了，但是还是存在一些差别。

图 4 - 7 视频类软件流量消耗

图 4 - 7计算的是视频类软件的流量消耗排名，大部分软件的每个小时的流量消耗集中在10MB这个级别。就推荐来讲有一定的可推荐性。

图 4 - 8 音乐类软件流量消耗

图 4 - 8计算的是音乐类软件的每小时流量排行，从计算结果可以看出有几款软件的流量消耗很大，在10M左右，但是其他大部分软件的流量都很低，这个差别还是挺大的。

### 4.2.2 APP流行度分析

一直以来，应用商店被视为移动互联网的重要入口，是新增用户手机应用的最主要来源。安卓平台比其他平台更为开放，用户下载应用软件的渠道也更为普遍。但据相关数据显示，约80%安卓应用均是从百度、91、360、豌豆荚、应用汇、安卓市场等第三方商店下载。

所有的应用商店的功能其实都很相似，比如所有的应用商店都会对应用进行排名，这些排名又包括下载排名，又允许用户根据自己的使用感受对软件进行评分，这又有了评分排名等等，其次所有的app都允许用户对其进行使用感受的评论。实际上,这些不同的排名信息、评分动作以及评论信息都和app的流行度有关系。软件流行度是移动互联网时代app相关服务中的重要角色，比如app服务提供商甚至app开发商能从app流行度中看到市场发展的趋势以及用户喜好的改变等等。随着移动互联网的发展,部分研究人员已经开始研究app流行度在移动应用服务中的应用 [41,56,58],但是这方面的相关信息还是比较松散，没有一套完整的理论在确切的分析相关方面的信息。其实围绕app流行度建模的问题面临着诸多的挑战，最明显的案例就是flappy bird这款软件，毫无征兆的突然就火了，流行度突然就非常高，但是没过多久这股流行之风就过去了，这款app就像昙花一现，短暂的占据排行榜榜首一段时间就再也进入不了排行榜了。这说明app的流行度信息变化的非常快，并且具有很强的时间依赖性。

根据本文所使用的数据集的情况，分析app的流行度只能从使用的人数入手。毕竟两款不同的软件做比较，如果其中一款软件的使用人数比另一款的多则可以认为使用人数多的软件更流行。接下来要统计每个app使用的用户数记为。

给出了各个类型下app的流行度排名情况，排名的依据主要是每个APP对应的用户数，并且只是取出了各个类型使用人数比较多，排名比较靠前的APP情况，如表 4 - 4所示：

表 4 - 4 各类软件流行度排名

|  |  |  |  |
| --- | --- | --- | --- |
| 下载类 | 视频类 | 音乐类 | 浏览器类 |
| 360手机助手 | 爱奇艺视频 | 酷狗音乐 | UC浏览器 |
| 应用商店 | 腾讯视频 | QQ音乐 | QQ浏览器 |
| 应用宝 | 暴风影音 | 天天动听 | 360浏览器 |
| 百度手机助手 | 百度视频 | 酷我音乐 | 百度浏览器 |
| 安智市场 | PPTV聚力 | 百度音乐 | 欧朋浏览器 |
| 安卓市场 | 搜狐视频 | 多米音乐 | 猎豹浏览器 |
| 应用市场 | PPS影音 | 虾米音乐 | 搜狗浏览器 |
| 搜狗手机助手 | 乐视视频 | 咪咕音乐 | 傲游浏览器 |
| 淘宝手机助手 | 360影视大全 | 爱音乐 | 2345浏览器 |
| 安卓市场 | 土豆视频 | 网易云音乐 | 海豚浏览器 |
| 搜狗市场 | 风云直播 | 音乐+ | 4G浏览器 |
| 应用汇 | 凤凰视频 | 音乐圈 | 欧朋浏览器 |
| 安全市场 | 风行视频 | 音乐随身听 | 微米浏览器 |
| 卓易市场 | 乐视影视 | 千千静听 | UC浏览器 |
| 机锋市场 | 奇艺影视 | DJ音乐盒 | 欧朋极速浏览器 |
| 安卓应用市场 | 芒果TV | 在线音乐 | 天天浏览器 |
| 易悠市场 | 56视频 | 音乐播放器 | 3G浏览器 |
| 木蚂蚁市场 | 迅雷影音 | 酷狗音乐HD |  |
| 安卓手机助手 | QQ影音 | 酷我音乐HD |  |
| QQ应用宝 | 百度影音 | 布谷音乐 |  |
| 智汇市场 | 爱奇艺影视 | 百度音乐2014 |  |
| 巨人手机助手 | PPTV |  |  |
| 淘应用 | 搜狐视频HD |  |  |
| 迅雷手机助手 | CCTV |  |  |
| N多市场 | 千寻影视 |  |  |

下面将要介绍的，app推荐排名的建立中的流行度的分析，主要基于表格 四‑4来进行。

### 4.2.3 APP推荐排名的建立

本文的目的在于为用户推荐符合用户使用偏好，且使用流量少的App。然而考虑到实际情况，单纯考虑App的流量消耗问题并非符合用户的需要。相同类别内，有些App即使花费的流量比较少，可能由于面向的市场不同，使用的场景可能不尽相同，使用的用户可能只局限于某些领域。所以在考虑推荐app的时候不仅要在流量方面进行考虑，还得从流行度方面进行考虑。

根据4.2.1讨论的app流量分析，在得到每个app每个小时平均花费的流量后，可对同一类别下的app按照流量消耗进行升序排列。然后为用户进行推荐。进一步，如果有些app具有相同的流量消耗值，则再将这些app按照流行度进行降序排序。总结起来就是

流量大小的原则:首先将等待推荐的App按照每小时流量消耗的大小进行升序排列，,然后消耗相同流量的App则进一步按照其流行度值得大小进行降序排列。

流行度的原则:首先将等待推荐的App按照其流行度值得大小进行降序排列,然后对于具有相同流行度的App则需要按照流量消耗进行升序排列。

可能流行度排名中的一些app并未出现在流量排名中，或者流量排名中的一些app并未出现在流行度排名中，在这种情况下，对于未出现在另外一个排名中的app则默认该app在另外一个排名表的排名为最后一位。

接下来的任务，是需要将流量大小和app的流行度折中。因此，本文采取经济学上著名的资产投资组合理论来进行混合推荐。投资组合理论原意是，若干种证券组成的投资组合，其收益是这些证券收益的加权平均数，但是其风险不是这些证券风险的加权平均风险，投资组合能降低非系统性风险。举例来说就是，你想投资n支股票，你希望这n个投资组合能够给你带来最大收益，并且能够最小化期望风险。在本文中待推荐的app可以当做股票，而股票的收益和风险则可以认为是app的流量大小和流行度。

一个app的投资组合是由个app以及各个app所分配的权值（类似于投资股票中的每一股股票的投资比例的意义）。根据这些分析可知：

 (4-8)

其中表示app，

根据文献[12]中的讨论，权值可以被认为是推荐系统希望用户对相应的关注度。因此，可以用这个权值来对待推荐的app进行排序，即据此权值对降序排列并进行推荐。在介绍如何学习这些权值之前，我们首先定义一个投资组合的期望收益为，其可以计算如下：

 (4-9)

其中表示在基于流行度的排序列表即表 4 - 4各类软件流行度排名中的排名。同时，我们也可以定义投资组合的期望风险为，其计算方式如下

 (4-10)

其中表示在基于流量的排序列表即4.2.1 APP流量分析所涉及到的内容。

在我们的app推荐排名的建立过程中，目标是学习合适的权值，使得投资组合（即候选APP）的期望收益能达到最大化同时期望风险要求最小化。所以具体的优化问题定义如下：

 (4-11)

在设计了这个推荐模型之后，对于一款app，找出它在流量排行中的排名和流行度中的排名，根据上面的公式可以得到app最合适的权值，然后再根据这个权值，我们就可以得到app推荐的排序顺序。

### 4.2.4 APP活跃时间段的分析

和用户使用app偏好的时间段相对应，本文需要对app的活跃时间段进行分析，以便为不用时间段内的用户偏好使用某一类的app进行相应时间段内的推荐。具体计算方式为，根据App消耗流量排名表中的app以及流行度排名中的app，计算统计app所有出现的时间段，然后取出活跃度最高的两个时间段，为该app打上时间段标签，这样就完成了app活跃时间段的分析。流程图如图 4 - 9所示：

图 4 - 9 活跃时间段流程图

## 4.3 本章小结

本章基于Hadoop大数据分析平台，从移动互联网产生的用户数据和用户产生的app相关的数据出发，根据用户对某一类app的使用时间和使用频度来分析确立用户的使用偏好，然后根据app的流量消耗和流行度来确立待推荐的app排名顺序，最后将这两部分结合起来，为用户推荐符合他的使用模式，并且消耗较少流量，具有较高流行度的app，最终提高了用户的流量使用率，改善了在移动网络情况下用户的使用体验。

# 第五章 推荐结果分析

本节测试推荐模型所选用的额数据，来自第三章中分析过得数据集，这个数据集中的用户有一万多人，但是每个用户的数据记录却是千差万别，有的用户的数据记录能够持续一年多，并且每天都会产生很多数据，但是大部分用户的数据记录都在几十天甚至几天的范围，还有一部分用户数据记录只存在了一天不到。本文将选取一个用户，对其使用偏好进行分析，并且分析其推荐前的流量使用情况，然后再基于推荐列表，对其进行app推荐，再对推荐后的流量使用情况进行分析。然后综合比较两次情况，看有没有达到为用户节省流量的目的。

## 5.1 视频类软件推荐结果分析

在第三章介绍的数据集中，首先根据前面介绍的用户使用模式的相关分析，计算出偏好使用视频类软件的用户群体，然后随机挑出使用时间比较长的一个用户，然后计算该用户每天流量的使用情况。然后计算出该用户在什么时间段使用视频类软件比较多，在根据4.1.3章节里的表 4 - 3将用户打上时间段标签，这样就完成了对用户日常流量使用情况以及用户偏好使用情况的统计。除此之外，还需要将该用户每天使用视频类软件的时间统计出来，并根据这个时间来计算应用推荐后评价每个小时减少的流量消耗。接下来根据用户的视频类使用偏好情况，在app推荐排名中选择排名第一的app来代替用户原来使用的视频软件，在本例用户中，用户原来使用的视频软件是百度视频，百度视频在流行度排名中是在第四名，而百度视频在流量消耗中的排名是第五，并且用户使用该视频软件的时间段是早晨，根据规则时间段标签就是A。根据前一章计算得出的app推荐排名榜，在时间段标签为A的推荐排名榜单中排名第一的视频软件为爱奇艺视频，根据流量消耗榜单和app流行度榜单可知爱奇艺视频的流行度排名是第一，而爱奇艺视频的流量消耗排名是第十九。由此可知流量消耗爱奇艺视频更少，但是爱奇艺视频的流行度更高，所以符合我们的推荐原则，具体结果如图 5 - 1所示。

图 5 - 1 视频类软件推荐结果对比图

图 5 - 1给出了视频类软件推荐结果的分析。从结果对比图来看，再进行了应用推荐之后，该用户每天的流量消耗比推荐前有了一些下降。其中有些点是重合的，这说明用户在那一天并没有使用视频类软件，还有一些点的差距相比其他的数据点的差距更小，这些点的存在说明在那一天用户使用视频类软件的时间比较少，消耗的流量也比较少。则根据每个小时流量消耗的排名榜

## 5.2 下载类软件推荐结果分析

首先根据前面介绍的用户使用模式的相关分析，计算出偏好使用下载类软件的用户群体，然后随机挑出使用时间比较长的一个用户，本例用户的记录时间是六十天，然后计算出该用户每天流量的使用情况。然后计算出该用户在什么时间段使用下载类软件比较多，在根据4.1.3章节里的表 4 - 3将用户打上时间段标签，这样就完成了对用户日常流量使用情况以及用户偏好使用情况的统计。与视频类软件的分析类似，同样需要计算该用户评价每天使用下载类软件的时间，并根据使用时间来对比每天减少的流量使用。接下来根据用户的下载类应用使用偏好情况，在下载类软件app推荐排名中选择相同时间段内排名第一的app来代替用户原来使用的下载类软件，如果该时间段内没有，则选择下一时间段内的。在本例用户中，用户原来使用的下载类软件是QQ应用宝，QQ应用宝在流行度排名中是在第二十名，排名比较靠后，而QQ应用宝在流量消耗中的排名是第九名，并且用户使用该下载类软件的时间段是上午，根据规则时间段标签就是B。根据前一章计算得出的app推荐排名榜，在时间段标签为B的推荐排名榜单中排名第一的下载类软件为360手机助手，根据流量消耗榜单和app流行度榜单可知360手机助手的流行度排名是第一，但是360手机助手的流量消耗排名是第八。由此可知360手机助手的流量消耗稍微多一点，但是360手机助手的流行度却是比QQ应用宝的高很多，虽然有点不太符合推荐的原则，但是由推荐结果来看，差别不是很明显。具体结果如图 5 - 2所示：

图 5 - 2 下载类软件推荐结果对比图

图 5 - 2给出类下载类app推荐结果的对比分析图。由图可以看出，推荐后的流量较推荐之前比差别不是很大。但是从app流行度来讲，用户可能会得到更好的用户体验和应用服务。

## 5.3 音乐类软件推荐结果分析

与前面介绍的方法一样，计算出偏好使用音乐类软件的用户群体，然后随机挑出使用时间比较长的一个用户，本例用户的记录时间大概在六十天，然后计算出该用户每天流量的使用情况。然后计算出该用户在什么时间段使用下载类软件比较多，在根据4.1.3章节里的表 4 - 3将用户打上时间段标签，这样就完成了对用户日常流量使用情况以及用户偏好使用情况的统计。与视频类软件的分析类似，同样需要计算该用户评价每天使用音乐类软件的时间，并根据使用时间来对比每天减少的流量使用。接下来根据用户的音乐类应用使用偏好情况，在音乐类软件app推荐排名中选择相同时间段内排名第一的音乐类app来代替用户原来使用的音乐类软件，同样的，如果该时间段内没有，则选择下一时间段内的。在本例用户中，用户原来使用的音乐类软件是酷狗音乐，酷狗音乐在流行度排名中是在第一名，排名非常靠前，而酷狗音乐在流量消耗中的排名也是第一名，说明流量消耗也是非常大。并且用户使用该音乐类软件的时间段是晚上，根据规则时间段标签就是E。根据前一章计算得出的app推荐排名榜，在时间段标签为E的推荐排名榜单中并没有其他的音乐类软件，只能在其他的时间段中找，找到时间段为F的榜单中排名最高的音乐类app是天天动听，根据流量消耗榜单和app流行度榜单可知天天动听的流行度排名是第三，比酷狗音乐略低，但是天天动听的流量消耗排名是第六，并且根据榜单来看，流量差距很大，流量消耗比酷狗音乐小很多。由此可知天天动听的流行度虽然较酷狗音乐稍微低一点，但是天天动听的流量消耗却是比酷狗音乐的低很多，虽然牺牲了一点流行度但是换来的低流量消耗却是非常值得的。这也非常符合本文的推荐原则。具体情况如图 5 - 3所示：

图 5 - 3 音乐类软件推荐结果对比图

图 5 - 3给出了音乐类软件的推荐流量消耗对比图。由图可以看出，总体上来说流量消耗有所下降。但是和视频类软件对比图类似，每一天的流量消耗减少的是不一样的，有的天用户消耗流量基本没有变化。

## 5.4 浏览器类软件推荐结果分析

和前面三个介绍的类似，首先计算出偏好使用浏览器类软件的用户群体，然后随机挑出使用时间比较长的一个用户，本例用户的记录时间大概在五十三天，然后计算出该用户每天流量的使用情况。然后计算出该用户在什么时间段使用浏览器类软件比较多，在根据4.1.3章节里的表 4 - 3将用户打上时间段标签，这样就完成了对用户日常流量使用情况以及用户偏好使用情况的统计。与视频类软件的分析类似，同样需要计算该用户评价每天使用浏览器类软件的时间，并根据使用时间来对比每天减少的流量使用。接下来根据用户的浏览器类应用使用偏好情况，在浏览器类软件app推荐排名中选择相同时间段内排名第一的浏览器类app来代替用户原来使用的浏览器类软件，同样的，如果该时间段内没有，则选择下一时间段内的。在本例用户中，用户原来使用的浏览器类软件是UC浏览器，UC浏览器在流行度排名中是在第一名，排名非常靠前，而UC浏览器在流量消耗中的排名是第二名，说明流量消耗也是非常大。并且用户使用该浏览器类软件的时间段是下午，根据规则时间段标签就是D。根据第四章计算得出的app推荐排名榜，在时间段标签为D的推荐排名榜单排名最高的浏览器类app是百度浏览器，根据流量消耗榜单和app流行度榜单可知百度浏览器的流行度排名是第四，比UC浏览器略低，但是百度浏览器的流量消耗排名是第十，并且根据榜单来看，流量差距很大，流量消耗比UC浏览器小很多。由此可知百度浏览器的流行度虽然较UC浏览器稍微低一点，但是百度浏览器的流量消耗却是比UC浏览器的低很多，和上一节的音乐类类app推荐类似，虽然牺牲了流行度但是换来的低流量消耗却是非常值得的。这个推荐结果也比较合适。根据这个结果画出了图 5 - 4。

图 5 - 4 浏览器类软件推荐结果对比图

图 5 - 4给出了浏览器类app推荐的结果分析，其中推荐前流量消耗是指用户目前的流量使用情况，推荐后流量消耗是指。总体来讲，流量消耗相比推荐之前有了减少。

## 5.5 本章小结

本章对根据第四章的推荐模型，分别对视频类、浏览器类、下载类、音乐类四类app推荐的结果进行对比。根据随机选取的对四个类别有使用偏好的用户进行分析，并进行相关的应用推荐，验证了推荐结果。在进行推荐后，能够减少用户的流量消耗，并且满足推荐的app具有一定的流行度这个目标。最大程度上满足了用户的使用偏好，改善了用户的使用体验。

# 第六章 总结及展望

## 6.1 总结

本文依托Hadoop大数据平台和移动互联网数据进行了面向用户体验的智能应用使用模式分析与优化研究，主要的研究方向是针对流量使用这一部分。随着移动互联网的盛行，随着智能手机的迅速崛起，各式各样的app进入人们的视野。Google Play Store 2015年 的应用和开发者数量首次双双超过苹果 App Store。Play Store 的 Android 应用总量达到 183 万款，如何在这些软件里找到适合自己使用的app是让很多用户头疼的一件事情。

本文首先介绍了数据分析所使用的Hadoop平台相关技术，文章所有的数据处理计算工作都是基于这个平台而进行的，然后利用这个数据处理平台对本所采用的数据集进行了总体方面、用户方面、app方面进行了分析处理。然后本文建立了一套应用推荐模型。推荐模型主要分为两部分，用户使用偏好和app相关画像。其中分析用户的使用偏好主要从用户对app的使用时间以及使用频度两个方面进行。在这两个方面确立了用户的使用偏好之后，再分析用户的使用时间段偏好，最终为该用户打上时间段标签，这几完成了用户使用偏好模型的建立。另一方面是app画像的分析，本文的目的是为用户推荐使用流量少的app并且不能影响用户的使用体验，所以app画像的分析主要分为两个方面。一方面是app流量消耗的分析，毕竟这是本文的一个根本出发点，另一个方面是app活跃度的分析。只考虑app的流量消耗带来的后果是推荐的结果比较单一。所以这个部分根据app的流量消耗以及app的流行度来确立最终的推荐排名。具体采用的方法是投资组合理论，最大化流行度同时最小化流量。除此之外，还需要对每个app’进行时间段分析，并为每个app打上时间段标签。有了推荐排名之后，就可根据用户的使用偏好对用户进行应用推荐。

## 6.2 展望

基于移动互联网的应用推荐有着巨大的前景，但是，本文的推荐系统还有很多不足，主要表现的一下方面：

1. 大数据的四大特征：海量的数据规模（vast）、快速的数据流转和动态的数据体系（velocity）、多样的数据类型（variety）和巨大的数据价值（value）。而本文的数据集只是规模大一点，可采用的信息不多。
2. 用户使用偏好模型，可使用的参数，要想更精确的分析用户的使用偏好，可能还得需要其他的参数。
3. App画像的分析，同使用偏好模型一样，需要更多的参数来精确分析app的画像。这在以后还有很大的提升空间。

# 参考文献

1. 余泓. 基于移动终端的移动互联网服务质量及用户行为分析研究[D].安徽大学,2014.
2. 郑桂凤. 移动互联网的用户行为分析系统的设计与实现[D].北京邮电大学,2010.
3. 宝腾飞. 面向移动用户数据的情境识别与挖掘[D].中国科学技术大学,2013.
4. 杨艳. 下一代网络业务用户行为研究[D].西南交通大学,2012.
5. 王璐. 移动互联网用户行为分析[D].重庆邮电大学,2012.
6. 潘宇彬. 基于个性化推荐的移动应用管理系统的设计与实现[D].西安电子科技大学,2013.
7. 祝恒书. 面向移动商务的数据挖掘方法及应用研究[D].中国科学技术大学,2014.
8. 李迎辰. 基于社交网络的移动应用推荐系统研究及应用[D].重庆大学,2014.
9. 曹嘉伦. 个性化推荐技术研究与系统设计[D].南京邮电大学,2014.
10. 李龙飞. 基于Hadoop+Mahout的智能终端云应用推荐引擎的研究与实现[D].电子科技大学,2013.
11. 鄢舒源. 移动个性化应用推荐系统的设计和实现[D].北京邮电大学,2015.
12. 冯铭,王保进,蔡建宇. 基于云计算的可重构移动互联网用户行为分析系统的设计[A]. 中国计算机学会(CCF).CCF NCSC 2011——第二届中国计算机学会服务计算学术会议论文集[C].中国计算机学会(CCF):,2011:4.
13. 赵勇. 移动互联网用户行为分析系统技术架构浅析[A]. 中国通信学会无线及移动通信委员会.2012全国无线及移动通信学术大会论文集（下）[C].中国通信学会无线及移动通信委员会:,2012:3.
14. 唐家琳. 移动互联网用户行为比较分析[J]. 西安邮电大学学报,2013,05:90-94+99.
15. 陈克寒,韩盼盼,吴健. 基于用户聚类的异构社交网络推荐算法[J]. 计算机学报,2013,02:349-359.
16. 孟祥武,胡勋,王立才,张玉洁. 移动推荐系统及其应用[J]. 软件学报,2013,01:91-108.
17. Lei X,Wu S,Ge L,et al. Clustering and overlapping modules detection in PPI network based on IBFO[J]. Proteomics, 2013,13(2): 278-290.
18. Ostermann S, Iosup A, Yigitbasi N, et al. A performance analysis of EC2 cloudcomputing services for scientific computing[M]//Cloud Computing. SpringerBerlin Heidelberg, 2010: 115-131.
19. Kuo-Wei Su,Chao-Hung Wang. Usability Testing on the Interface of the Location-Based M-tourism Application[A]. IEEE Beijing Section,China、Guangzhou University,China.Proceedings of 2013 IEEE International Conference on Computer Science and Automation Engineering VOL01[C].IEEE Beijing Section,China、Guangzhou University,China:,2013:4.
20. Xinye Lin,Xiao Xia,Shaohe Lv,Xiaodong Wang School of Computer Science,National University of Defense Technology,Changsha 410072,China. Research on the Predictability of Mobile App Usage[A].2011:13.
21. Jianlin Xu,Yifan Yu,Zhen Chen,Bin Cao,Wenyu Dong,Yu Guo,Junwei Cao. MobSafe:Cloud Computing Based Forensic Analysis for Massive Mobile Applications Using Data Mining[J]. Tsinghua Science and Technology,2013,04:418-427.
22. Tekin, C.; Zhang, S.; van der Schaar, M., "Distributed Online Learning in Social Recommender Systems," in Selected Topics in Signal Processing, IEEE Journal of , vol.8, no.4, pp.638-652, Aug. 2014 doi: 10.1109/JSTSP.2014.2299517
23. Jain, S.; Grover, A.; Thakur, P.S.; Choudhary, S.K., "Trends, problems and solutions of recommender system," in Computing, Communication & Automation (ICCCA), 2015 International Conference on , vol., no., pp.955-958, 15-16 May 2015

doi: 10.1109/CCAA.2015.7148534

1. Verbert, K.; Manouselis, N.; Ochoa, X.; Wolpers, M.; Drachsler, H.; Bosnic, I.; Duval, E., "Context-Aware Recommender Systems for Learning: A Survey and Future Challenges," in Learning Technologies, IEEE Transactions on , vol.5, no.4, pp.318-335, Oct.-Dec. 2012
2. Erdt, M.; Fernandez, A.; Rensing, C., "Evaluating Recommender Systems for Technology Enhanced Learning: A Quantitative Survey," in Learning Technologies, IEEE Transactions on , vol.PP, no.99, pp.1-1 doi: 10.1109/TLT.2015.2438867

# 致谢

转眼之间，两年半的研究生生活即将告一段落， 回想当时刚入学的情景，一切历历在目，北邮给我的第一感觉是很小，真的很小。但是北邮人给我的感觉是很牛，真的很牛。刚进实验室的时候，大家都在忙着做项目，瞬间觉得自己的大学生活都浪费了，但有很幸运能够来到这里继续读研。两年半下来，发现读研的生活并不轻松，代码、文档、开会成了我北邮生活的主要三件事，想想那段时间的每天开会，布置科研任务，写代码做研究，然后晚上再开会汇报成果，布置当天或者第二天任务，虽然有点枯燥乏味，但简单充实。

再说说我们实验室，首先要感谢的是张琳老师。张老师是我们的引路人，指引着我们不断前进、提高自己，保持在路上的心态，要求我们不要懈怠和停滞。在张老师的指导下，我慢慢接触了大数据这个时下热门的课题，并且参与了Android开发这个移动互联网时代必不可少的一门知识。在张老师的安排下，我慢慢进入了互联网这个圈子，也渐渐明确了自己未来的发展方向。其次感谢朱孔林老师对我毕业设计细心的指导，让我在迷茫的时候看到了毕设的曙光。其次刘雨老师、苏驷希老师、吴晓非老师、禹可老师、望育梅老师以及顾听钰老师每一位老师都身兼师长和朋友的身份，为实验室的科研和学生的学习发展倾注了自己的心血。没有这些老师的指导，就没有毕业的我们。

其次，我要感谢实验室那些可爱的小伙伴们，袭庚师兄、林剑辛师兄、李艺琳师姐他们在我刚进实验室的时候，都向我传达了一些学习的技巧，科研的技巧甚至一些生活的指导。向彬博士时刻的指导，让我在迷茫的时候找到了前进的方向，张小亦博士则不断地传授我一些需要学习的知识，失落的时候，任志远和刘岩则给了我很多的鼓励和支持，曲凯明博学，邓洁的认真，则是我学习的方向，陈池的社交和谈话技巧则一直是我学习的目标。还有陈雷师弟和赵瑾师妹，他们的问题则激励着我不断前进。成为他们中的一员让我很骄傲。

除了实验室的小伙伴，我还需要感谢，我本科那些和我一同考进北邮的好朋友，粘一龙、张立涛、郝佳伟、张伟、董舜源等等，不开心的时候始终有他们在身边。除了他们我还要感谢研究生期间宿舍的小伙伴们，赵晓飞、秦云博、卢宁、崔传金，宿舍生活让我们走到了一起，感谢你们的陪伴。

从山东大学到北京邮电大学，这一路，失去了原先的那份懵懂，失去了那份年少无知，失去了那份轻狂烦躁的心，却也收获了很多，收获了知识，收获了友谊，很多方面都得到了锻炼。

最后感谢父母默默的支持，感谢家庭默默的付出，这一切都让我常怀一颗感恩的心面对生活，面对困难。

希望北京邮电大学先进网络实验室在张琳教授的带领下越来越好！

# 攻读硕士期间发表的学位论文