密级：　　保密期限：

北京郵電大學

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。

本文首先研究了如何进行数据分析平台的搭建，实现了基于 Ambari 的 Hadoop 分析平台；在此基础上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，本文基于 APP 的流量消耗和流行度建立 APP 推荐模型，并结合用户相关的使用偏好研究 APP 推荐模型。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。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。

最后，本文依据移动互联网用户相关的数据集，为上述 APP 推荐模型进行了验证，结果表明在满足用户的使用偏好情况下，为用户推荐的 APP 能够比用户原来使用的 APP 花费更少的流量或者推荐的 APP 具有更高的流行度。这样一来达到了提高用户使用体验的目的。

关键词 用户使用偏好 APP 推荐 移动互联网

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 experiences in massive data. Based on those information, this pager will start with the traffic usage, in combination with user preferences and then recommend the applications which will consume less traffic and are more popular.

First, this paper studies how to build the Hadoop platform, then implements the platform based on Ambari. With the platform, we could do some classification and then process and analyze the dataset.

Second, based on user ‘preferences and APP pattern we build the recommend model. According to the model, we get users ‘preferences, and then recommend them with those applications which will consume less traffic and are more popular. Finally we aim to enhance users’ experience. One more thing, we have to study the using period of either the user or the APP.

In the end, we verify the model with the mobile internet dataset, the result shows that under the condition that we satisfy users ‘preferences, the applications that this platform recommends will consume less traffic or are more popular. Finally their experiences get enhanced.

KEY WORDS user preferences APP recommend mobile internet

目 录

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。报告中指出，截至到 2015 年第三季度，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[1]；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。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[2]。

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?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。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[11],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。2015 年 3 月，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》[3]，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。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。

当今时代海量的 APP 存在的情况下，利用 Hadoop 大数据分析平台，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。

2013 年潘宇斌在文献[9]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。2014 年，祝恒书在文献[10]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。

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[4]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[7]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。

2014 年，Tekin, C.和 Zhang, S.和 van der Schaar 在文献[25]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[27]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使用偏好，为用户推荐使用流量最少，并且符合用户使用偏好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偏好为用户进行推荐。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偏好，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偏好。将使用时间和使用频度结合就能够确立用户的 APP 使用偏好。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：着重介绍了搭建本平台用的 Ambari 工具以及平台的核心 Hadoop 大数据分析平台，包括本文数据处理所用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。

第三章：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。

第四章：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。

第五章：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 智能应用数据分析平台

2.1 平台信息说明

本文的数据分析平台基于 Ambari 搭建。Apache Ambari 现在是一个 Apache 基金会的顶级项目，随着时代的发展，现在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[12]。

Apache Ambari 是一种基于 Web 的工具。 Ambari 取得了很多显著的成果：

- 通过图形化的安装界面向导极大的简化了集群的安装步骤。
- 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。

本文处理数据用到的 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。

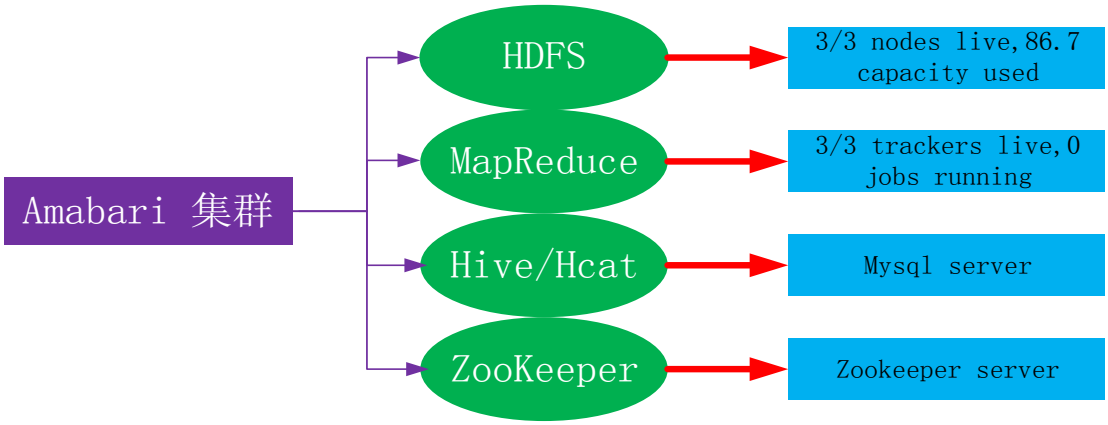

图 2-1 Ambari 集群功能图

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[13]，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：

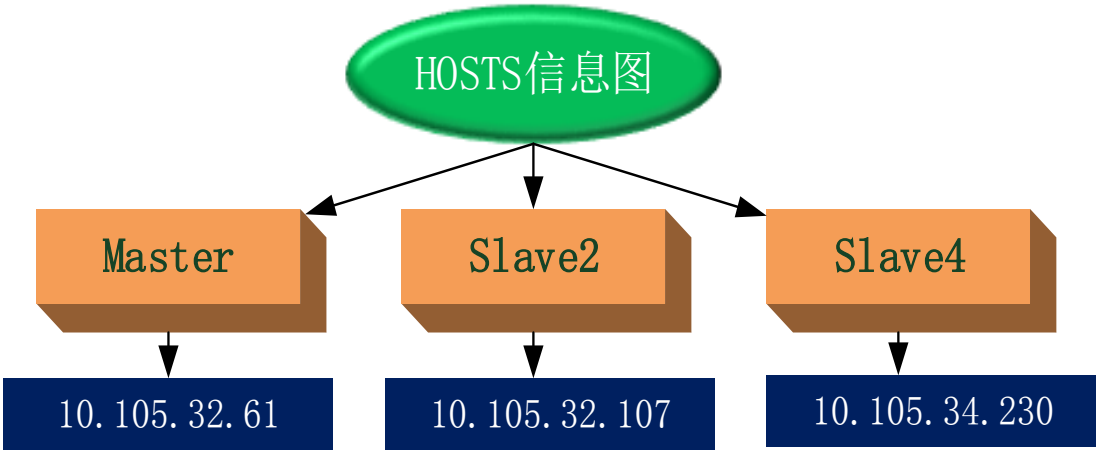

图 2 - 2 集群节点信息图

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 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 平台框架简介

本集群的核心组件便是 Hadoop。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非关系型数据库。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 平台的数据存储方式

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。

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 也有其局限性：

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：

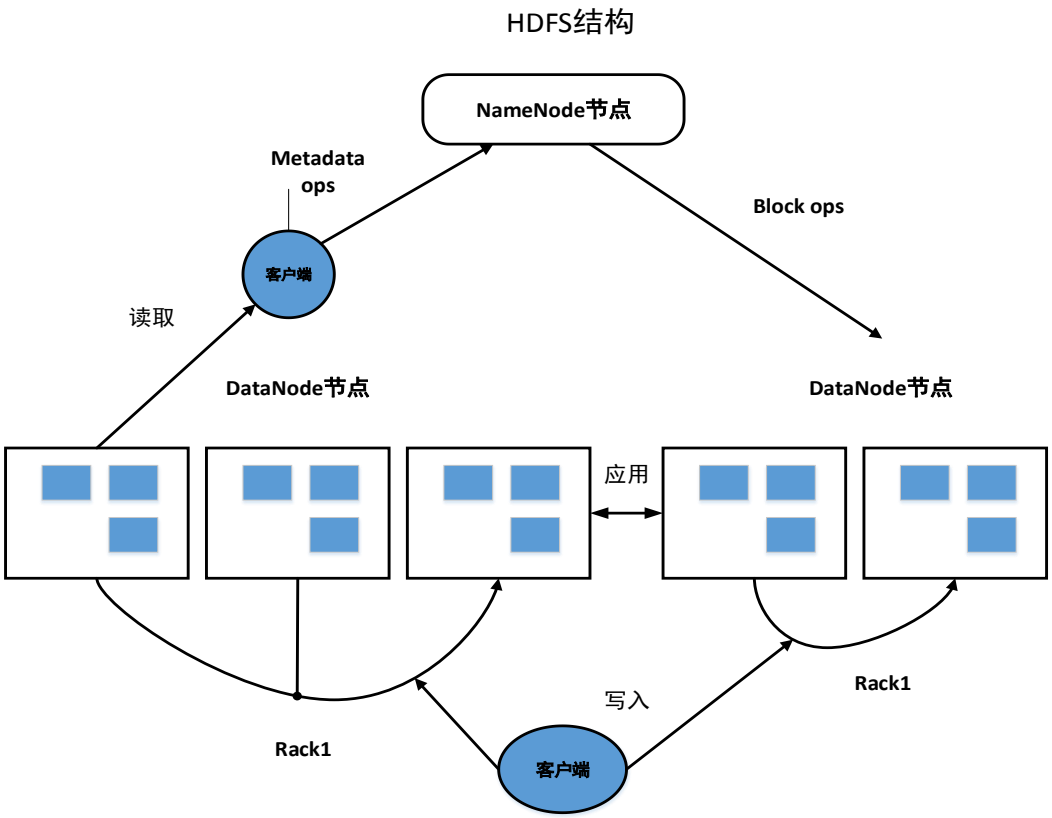

图 2 - 3 HDFS 结构

一个 HDFS 集群是由一个 NameNode 节点和若干个 DataNode 节点组成的。

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。

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，HDFS 系统里的文件会被切割存放，默认情况下是按照 64MB（可以设置）被切分成不同的数据块，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，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 数据处理流程

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。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。就像上一节里文章讲到的一样，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。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，同时进行一定的排序，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$[1,2,3,4]$进行乘以 2 的映射就变成了$[2,4,6,8]$。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$[2,4,6,8]$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：

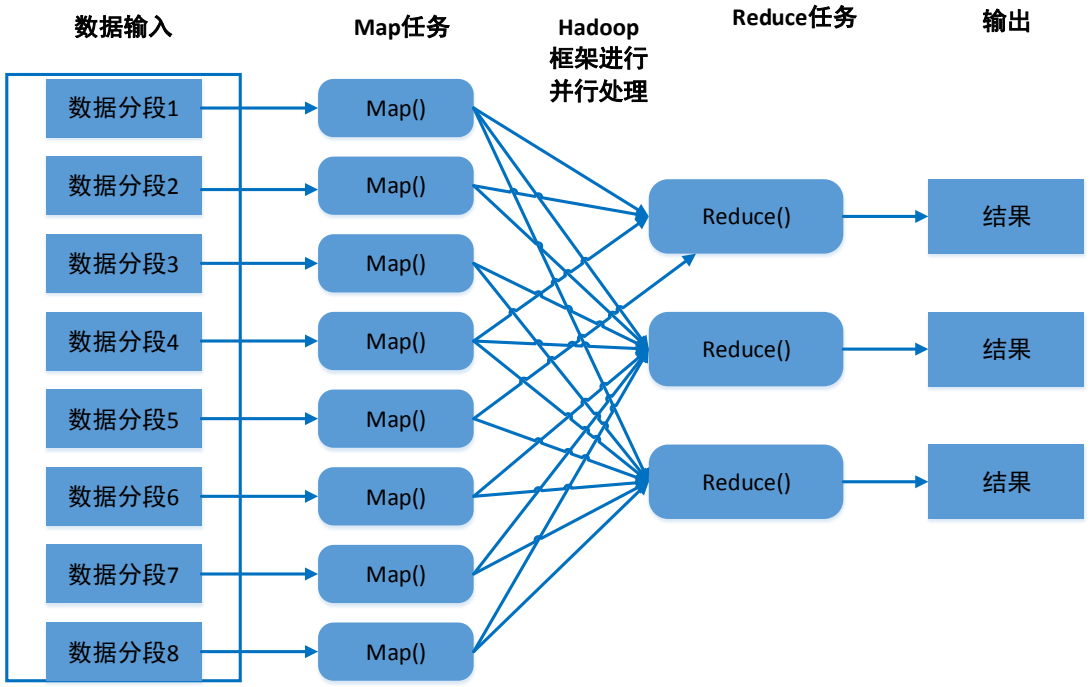

图 2-4 MapReduce 的处理流程图

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 文章数据集说明

2.5.1 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 2 - 5 所示

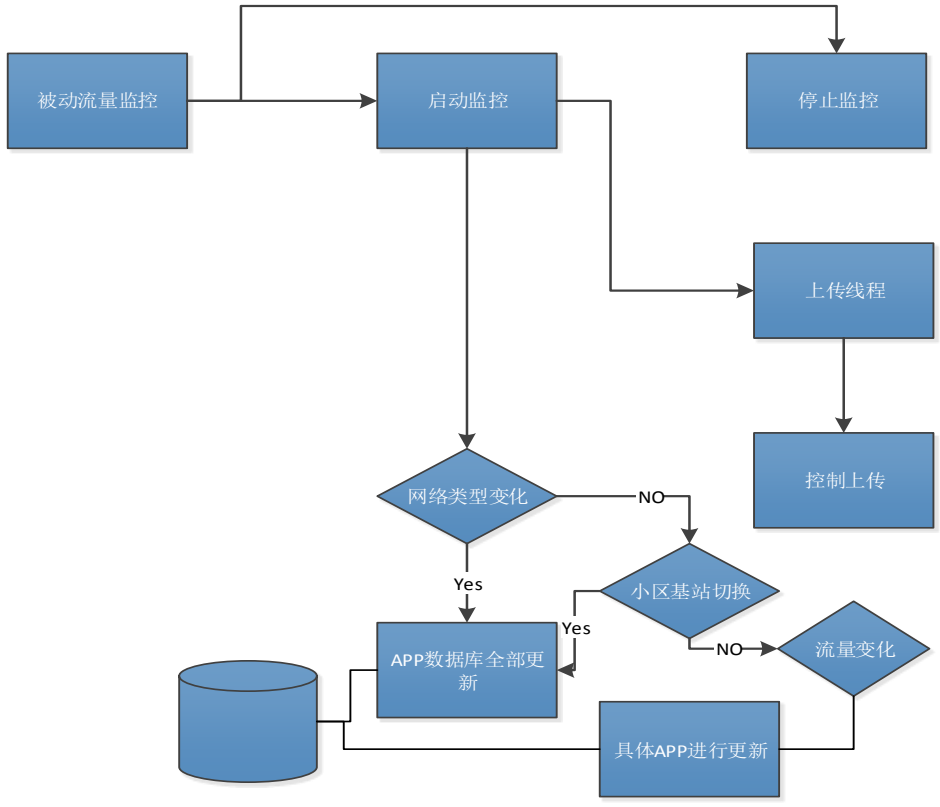

图 2 - 5 数据收集流量图

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 2 - 6 所示：

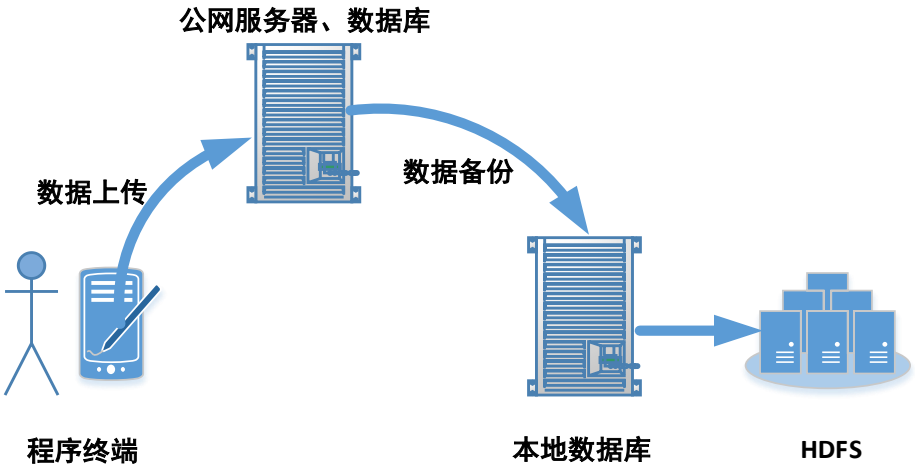

图 2 - 6 系统数据存储过程

2.5.2 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 2 - 4 所示：

表 2 - 4 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

（续上表）

表项	数据类型	解释
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 2-5：

表 2-5 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

（续上表）

表项	数据类型	解释
upload_traffic	varchar(20)	上行测试流量
location_detail	text	定位详细信息
download_traffic	varchar(20)	下行测试流量

2.6 基于平台的数据分析

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 2 - 7 所示

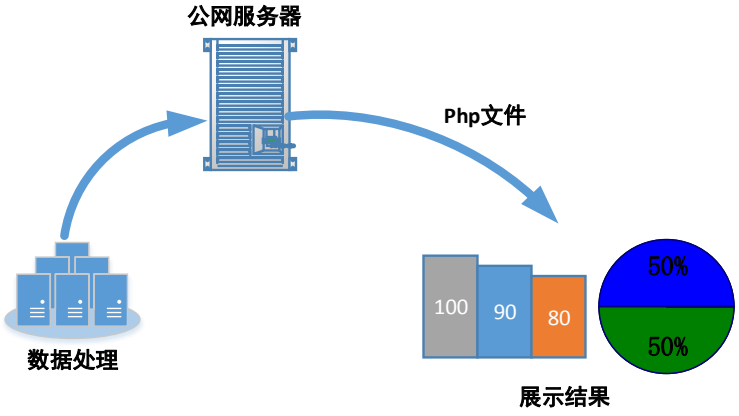

图 2 - 7 数据处理展示流程图

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。

2.6.1 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 2 - 8 是统计结果：

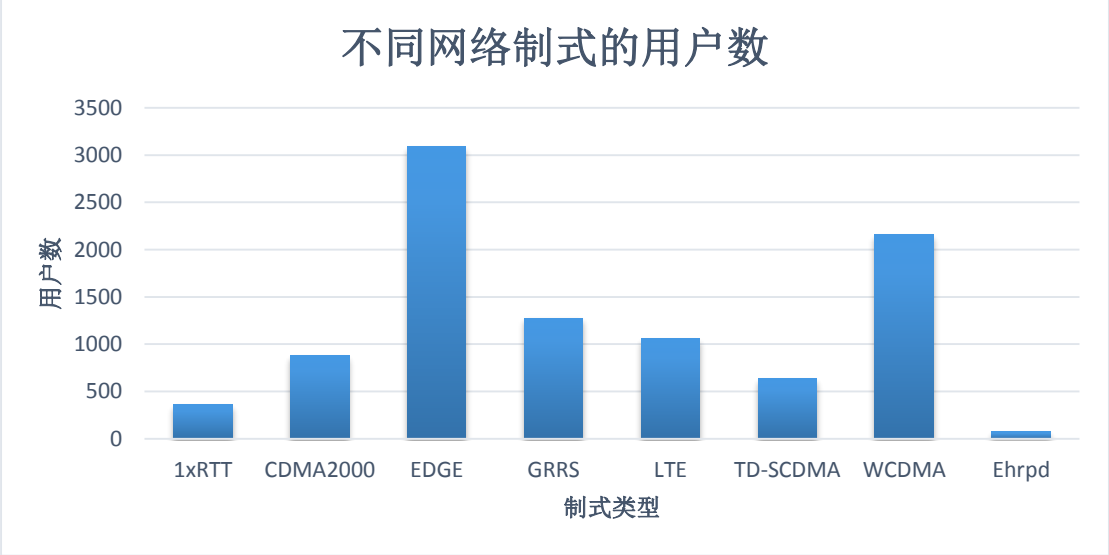

图 2 - 8 不同网络制式的用户数

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 2 - 8 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 2 - 8 里可以看到，电信用户在三大运营商里还处于少数。

图 2 - 9 统计了每天活跃的用户数

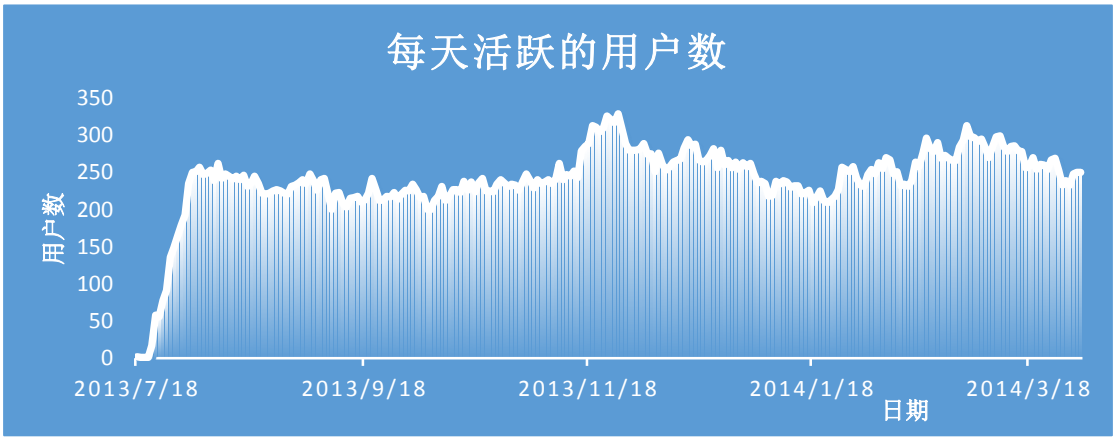

图 2 - 9 每天活跃的用户数

从图 2 - 9 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 2 - 10 统计了，每天监控到的其他软件的总的流量信息

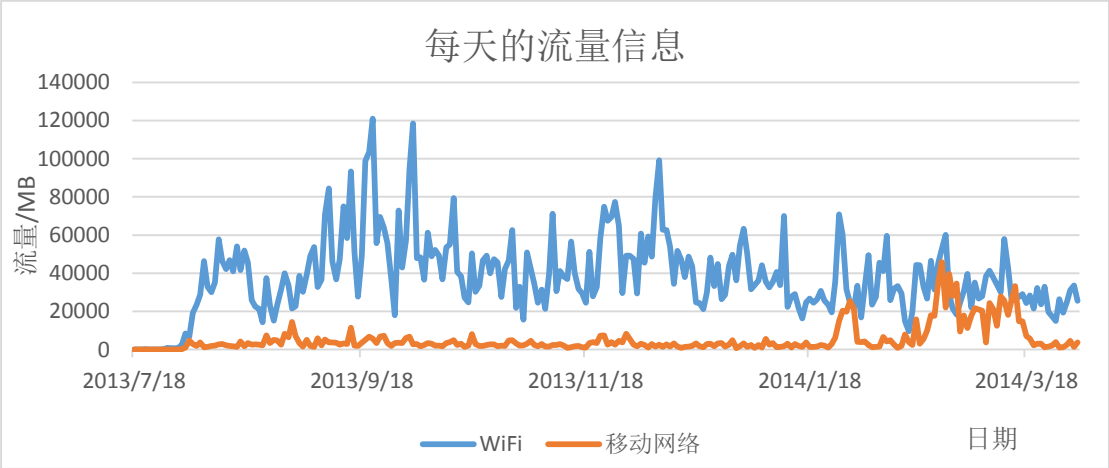

图 2 - 10 每天的流量信息

从图 2 - 10 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 2 - 11 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

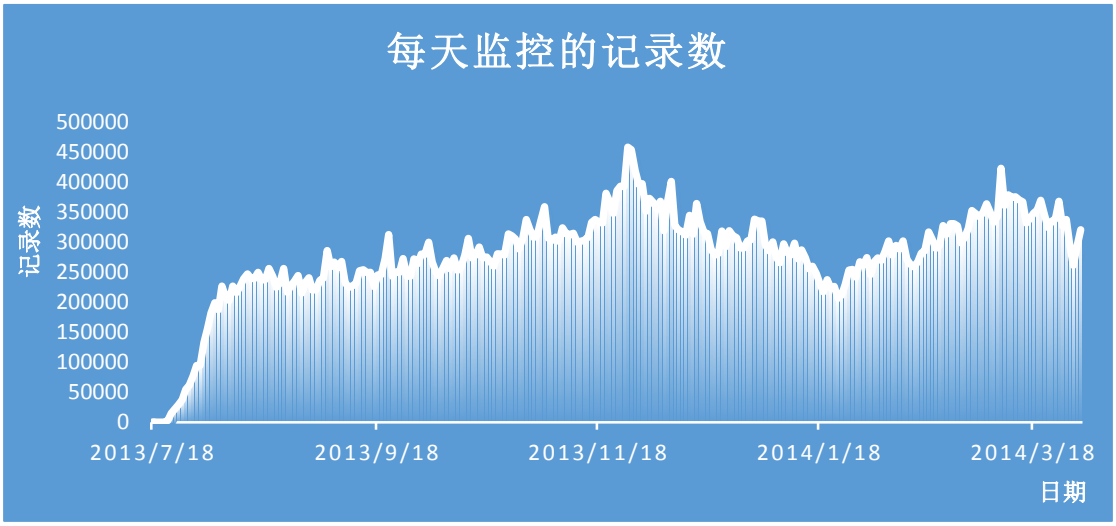

图 2-11 每天监控的记录数

图 2-12 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。

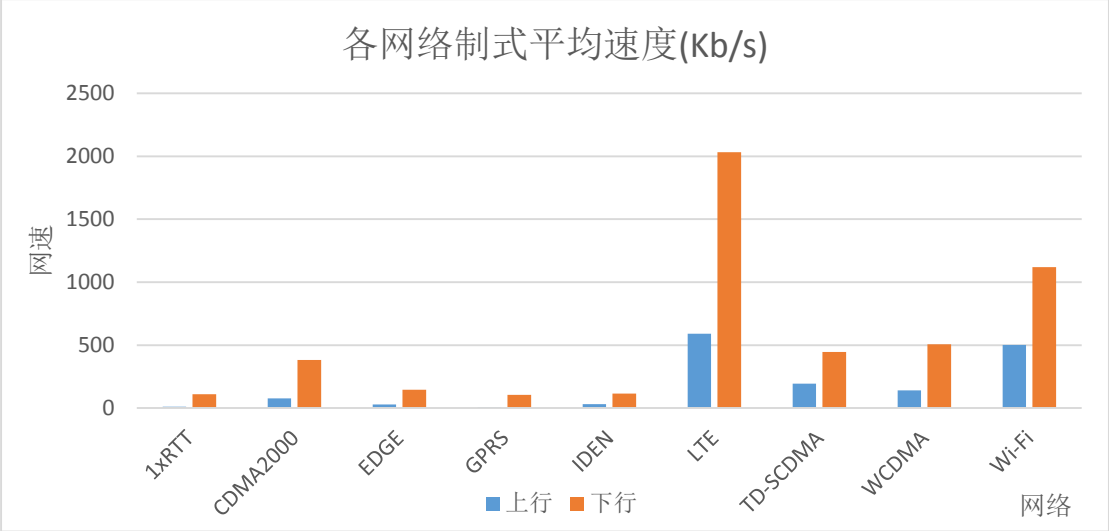

图 2 - 12 各制式网络速度

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。

2.6.2 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 2 - 13 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 2 - 13 的分析结果还是能说明一定的问题。

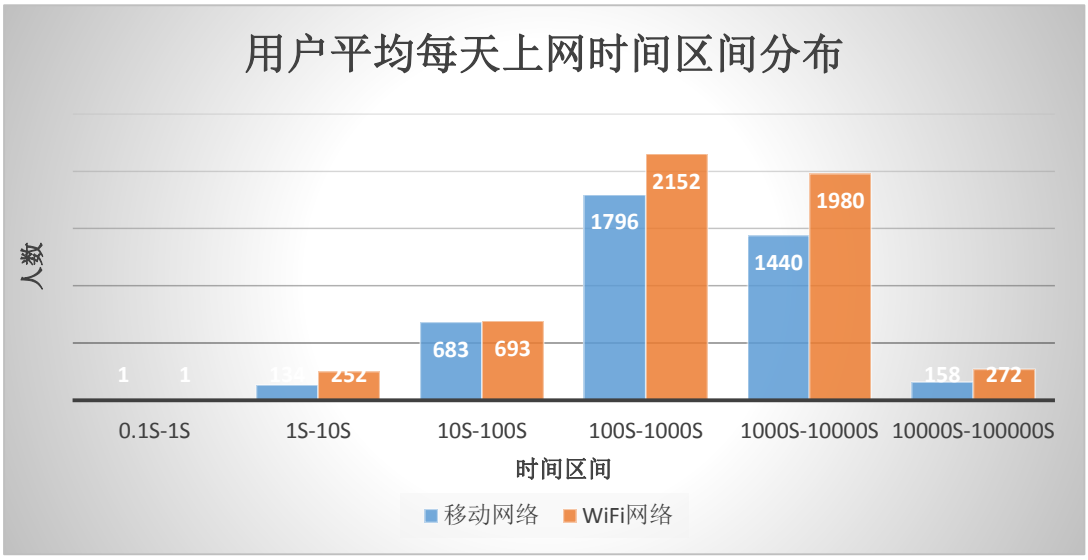

图 2 - 13 用户上网时间分布图

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 2 - 14 所示：

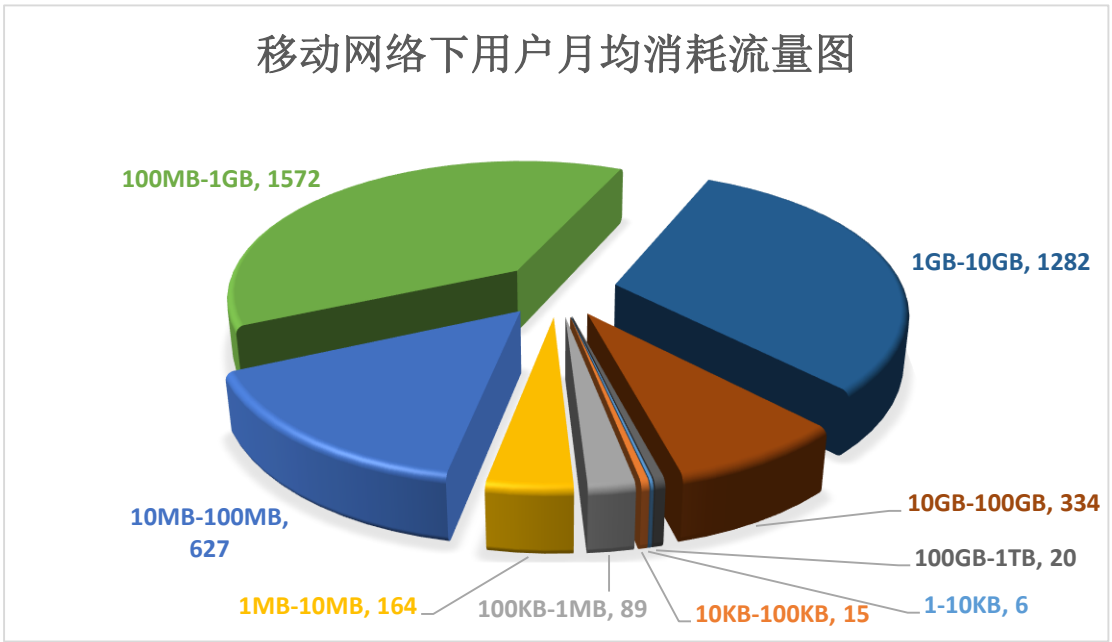

图 2 - 14 移动网络月均流量消耗图

图 2 - 15 是 WiFi 情况下的：

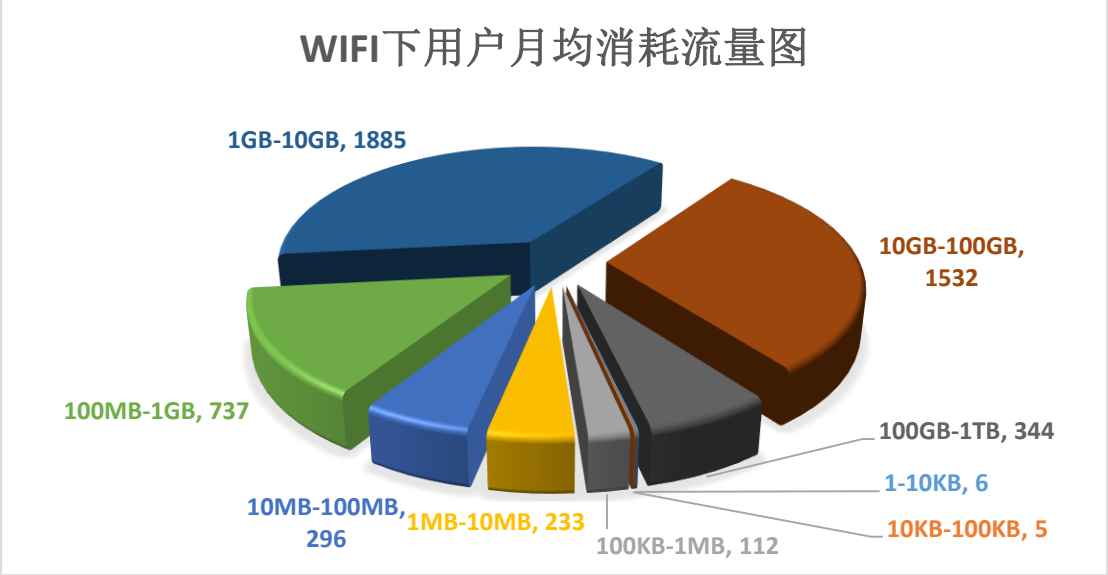

图 2 - 15 wifi 月均流量消耗图

图 2 - 14 和图 2 - 15 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。

2.6.3 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 2 - 16 对不同的 APP 每天的进行流量传输的时间进行了分析。从图 2 - 16 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。

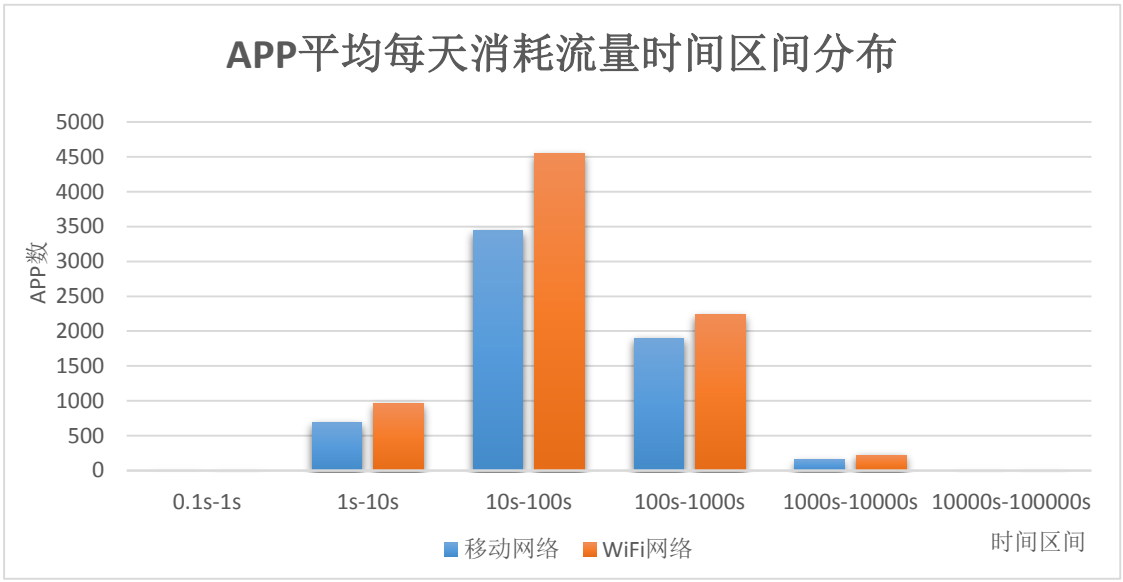

图 2 - 16 APP 平均每天消耗流量区间分布

图 2 - 17 和图 2 - 18 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。

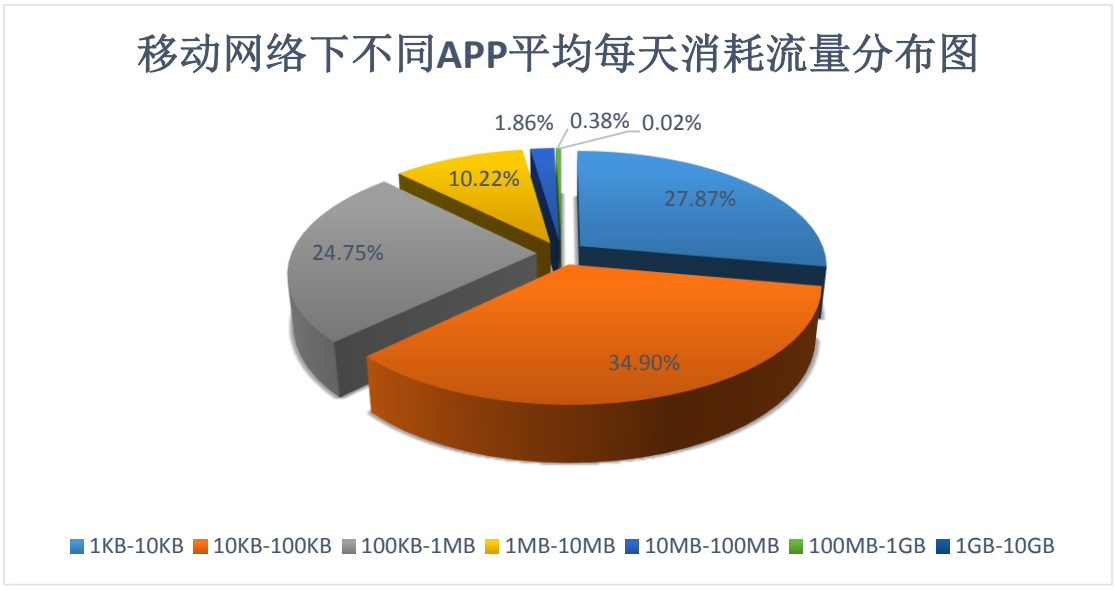

图 2 - 17 移动网络不同 APP 平均每天消耗流量分布图

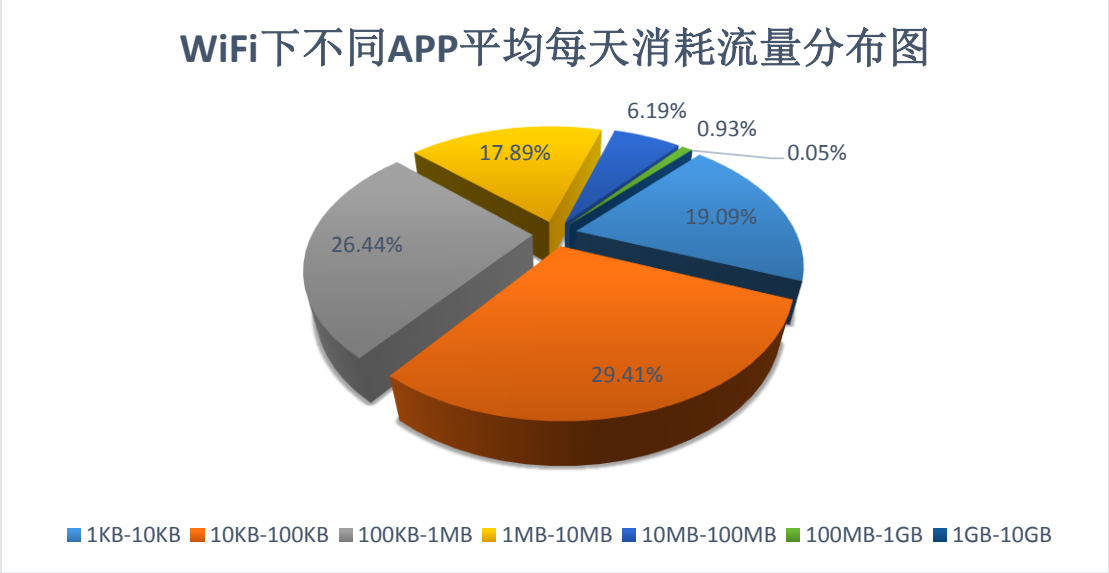

图 2 - 18 WiFi 下不同 APP 平均每天消耗流量分布图

从图 2 - 17 和图 2 - 18 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。

2.7 本章小结

本章内容首先介绍了数据分析平台的基本的信息，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。

其次基于现有的数据，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。

第三章 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 3 - 1 所述：

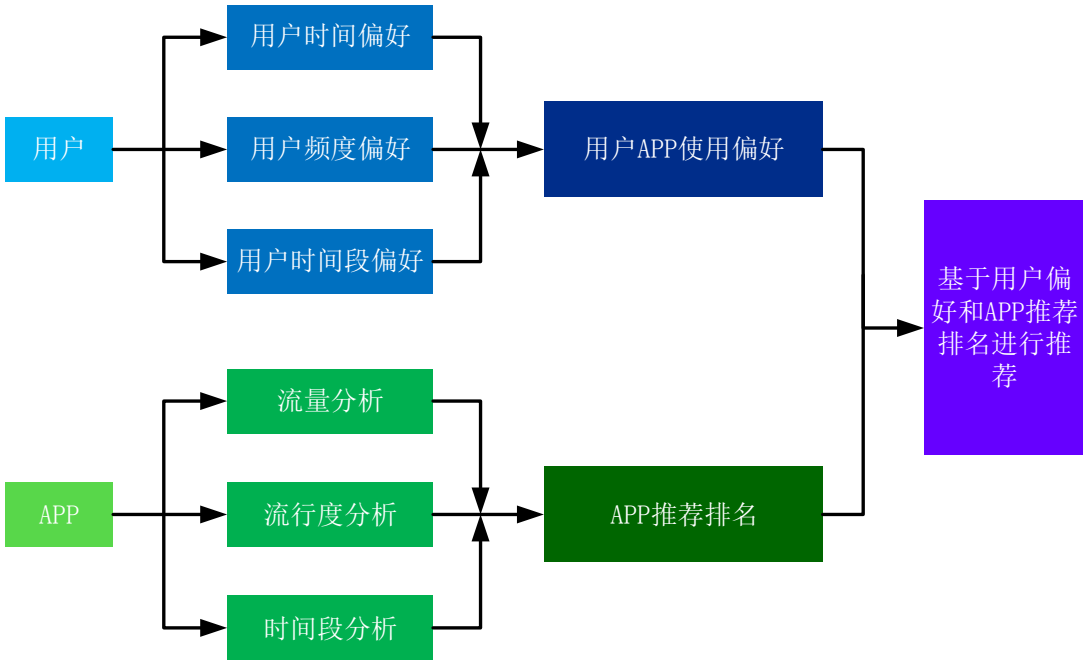

图 3 - 1 推荐流程图

3.1 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 $prefer$ 来表示，则 $prefer = P(time) * P(frequency)$，其中 $P(time)$ 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。$P(frequency)$ 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。

3.1.1 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t_u，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$\bar{t}_{um}$，其中

$$\bar{t}_{um}=\frac{1}{n}\sum_{i=1}^{n}t_{umi} \tag{3-1}$$

n用户使用的天数，m为用户编号，如表 3 - 1 所示

表 3 - 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	t_{u11}	t_{u12}	t_{u13}	……	t_{u1n}	$\bar{t}_{u1}$
用户 2	t_{u21}	t_{u22}	t_{u23}	……	t_{u2n}	$\bar{t}_{u2}$
用户 3	t_{u31}	t_{u32}	t_{u33}	……	t_{u3n}	$\bar{t}_{u3}$
⋮				……		
用户 m	t_{um1}	t_{um2}	t_{um3}	……	t_{umn}	$\bar{t}_{um}$
平均时间						$\bar{t}$

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$\bar{t}$，其中

$$\bar{t} = \frac{1}{m}\sum_{i=1}^{m}\bar{t}_{um} \tag{3-2}$$

m 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 $\bar{t}_{um}$ 去减去平均每个用户在视频类软件上平均每天的使用时间 $\bar{t}$，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 r，其中 $r = \bar{t}_{um} - \bar{t}$，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 r，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

$$p = \frac{1}{1+e^{-r}}, \quad p \in (0,1) \tag{3-3}$$

r 为上面计算出来的差值，其图 3 - 2 所示

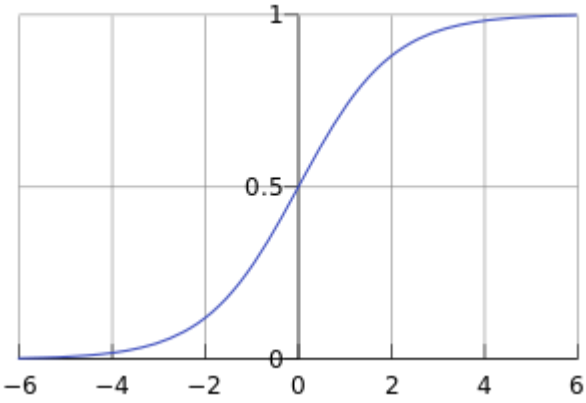

图 3 - 2 逻辑回归图

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$(0,1)$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r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r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

$$P(time)=\frac{1}{1+e^{-(\bar{t}_{um}-\bar{t})}} \tag{3-4}$$

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$P(time)$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。

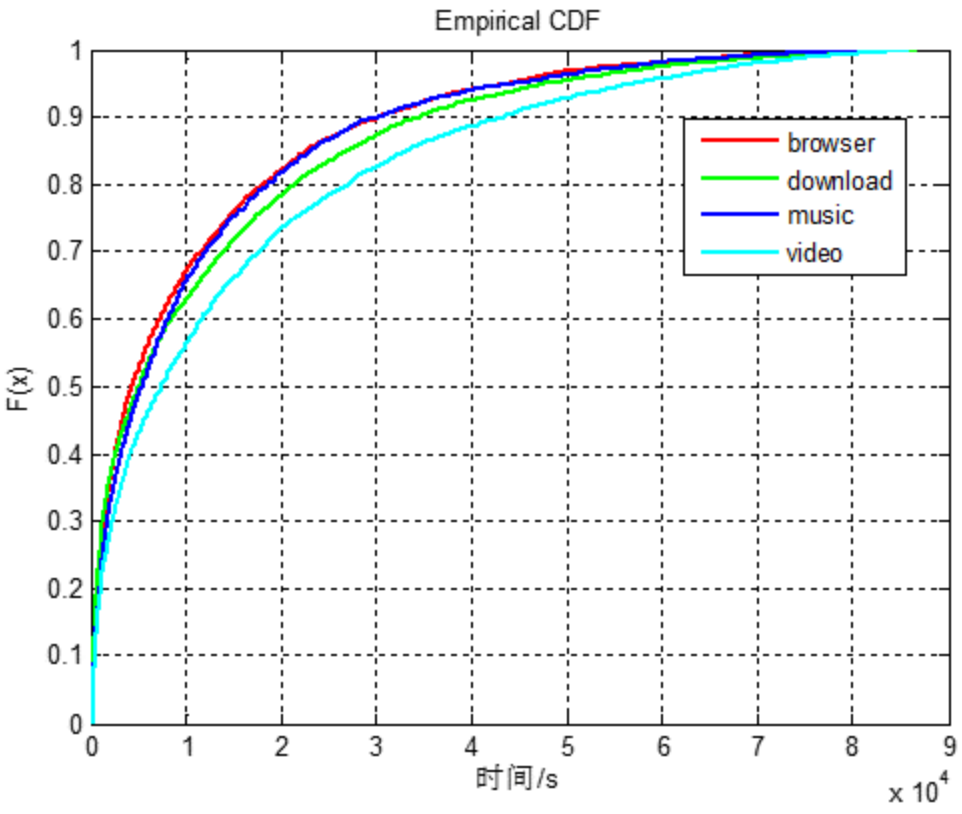

图 3-3 用户花费在各类软件上的时间分布图

图 3 - 3 是用户平均每天花费在各类软件上的时间分布 cdf 图，从中可以看出来，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。

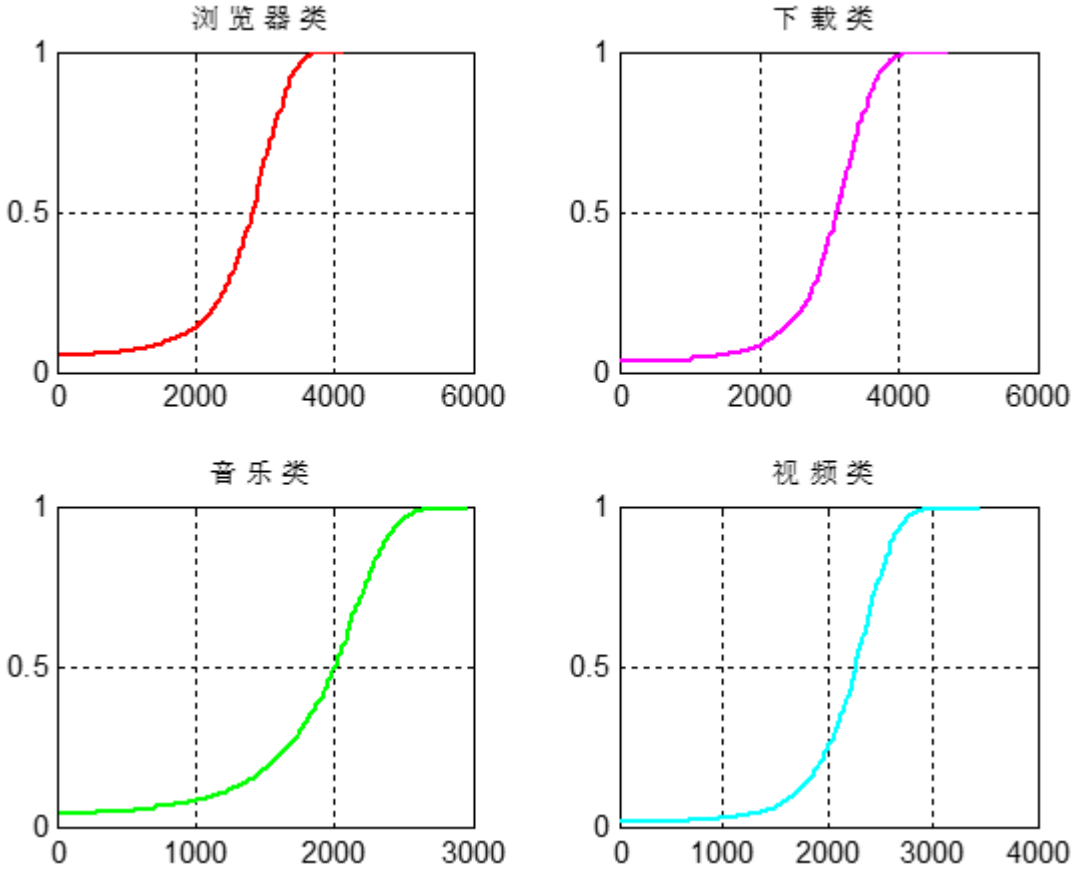

图 3 - 4 用户平均每天花费在各类软件上的时间分布

图 3 - 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。

3.1.2 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 t_{all} 来表示统计到的用户记录的总的天数，t_{use} 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

$$P(frequency)=\frac{t_{use}}{t_{all}} \tag{3-5}$$

其中 t_{use} 表示用户使用的天数，t_{all} 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 3 - 2 所示：

表 3 - 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

（续上表）

用户	浏览器类偏好	下载类偏好	音乐类偏好	视频类偏好
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 3 - 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 $P(time)$ 和用户使用的频度 $P(frequency)$，我们确定用户对某一类软件最终的使用偏好 $prefer$，

$$prefer = P(time) * P(frequency) \tag{3-6}$$

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。

3.1.3 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 3 - 3 所示：

表 3 - 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

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。

3.2 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。

3.2.1 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 h 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 $traffic(times, apps)$，其中 $times$ 为统计的小时 h 出现的次数，$traffic(times, apps)$ 的意思是指应用 APPs 在 $times$ 个小时内的总流量，则平均每个小时的流量 $traffic(h)$ 表示总流量除以小时个数。

则其计算方式如下所示

$$traffic(h) = \frac{traffic(times, apps)}{times} \tag{3-7}$$

根据这个公式就可以得到每个 APP 平均每小时消耗的流量。

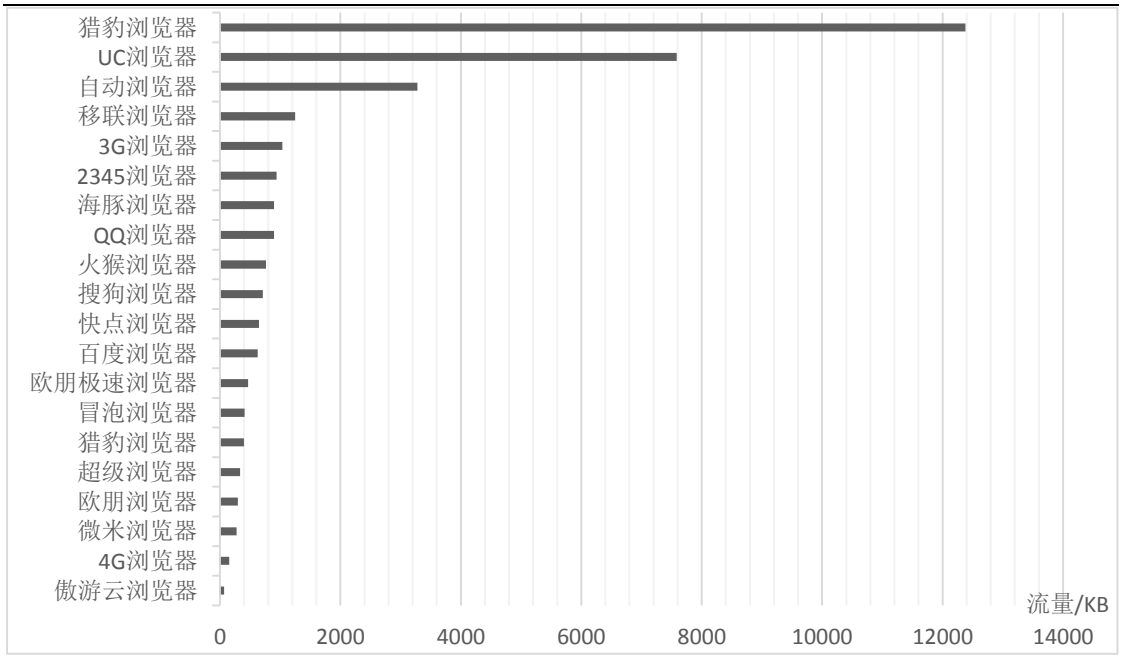

图 3 - 5 浏览器使用流量排行

图 3 - 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。

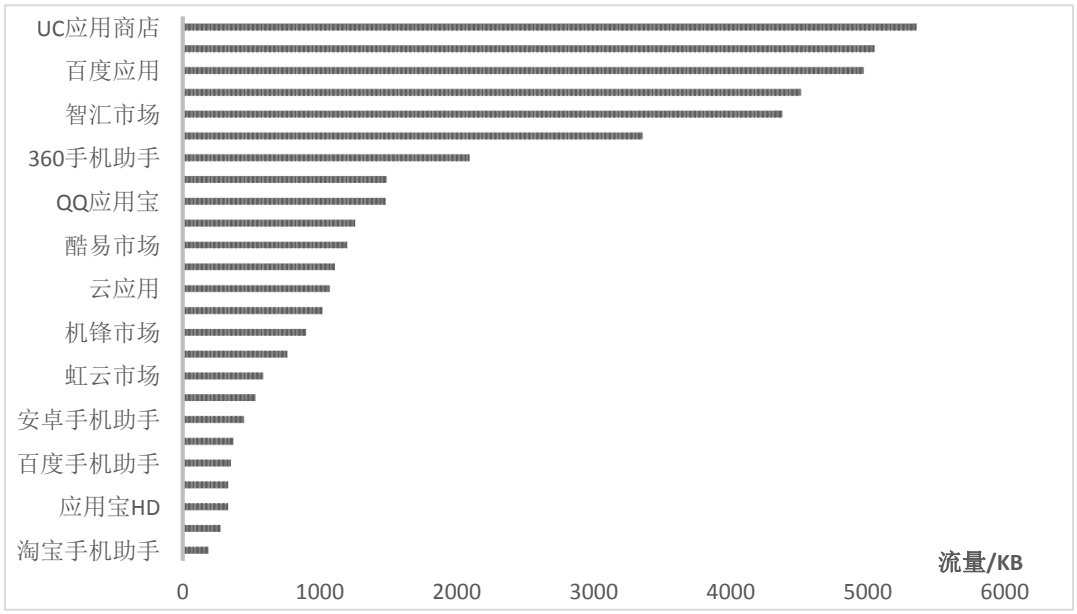

图 3 - 6 下载类软件流量消耗情况

图 3 - 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。

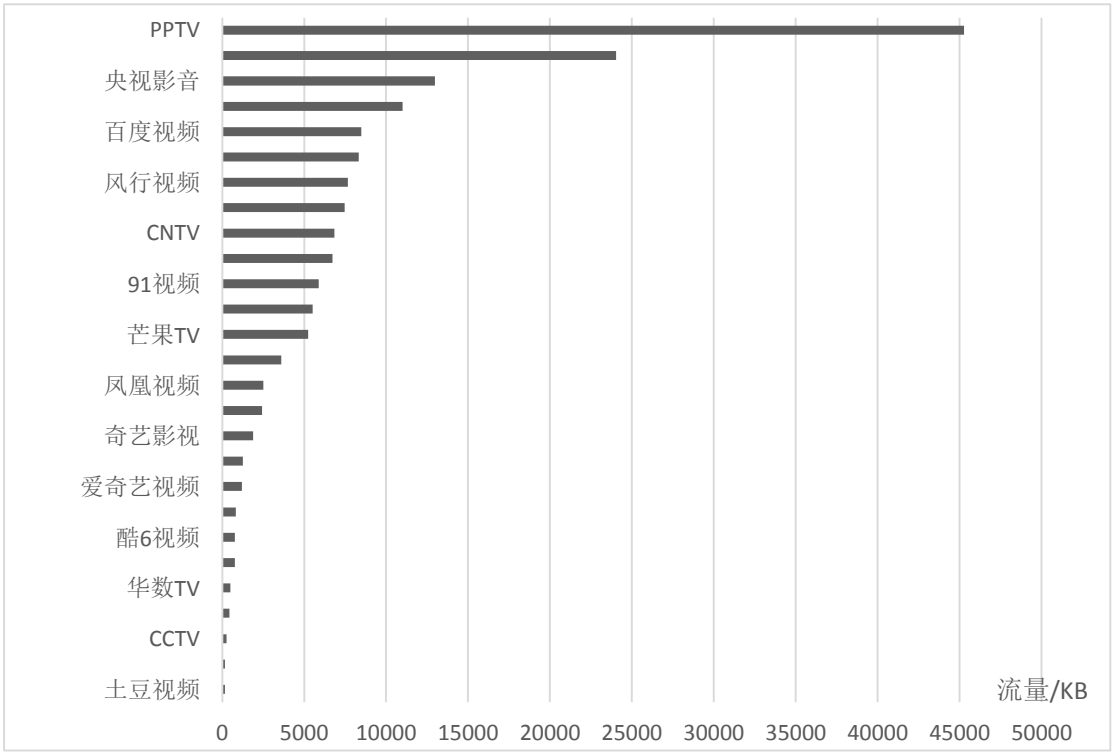

图 3 - 7 视频类软件流量消耗

图 3 - 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。

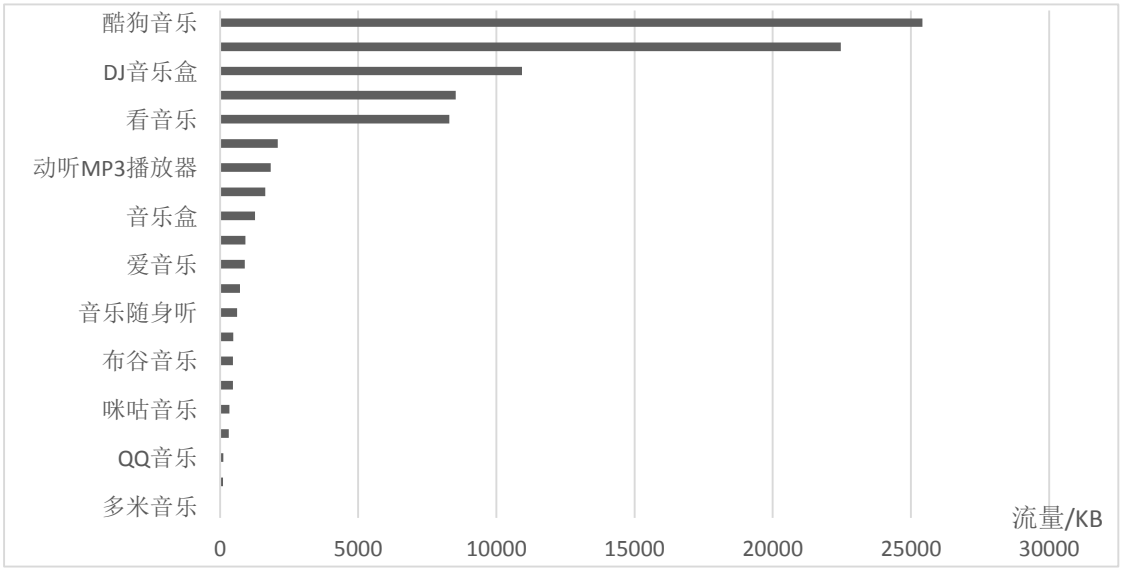

图 3 - 8 音乐类软件流量消耗

图 3 - 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。

3.2.2 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，国内约 80%安卓应用均是从百度助手、91 助手、360 手机助手、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 $app-users$。

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 3 - 4 所示：

表 3 - 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 2013	
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 3 - 4 来进行。

3.2.3 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 3.2.1 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[10]。

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[10]。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 Υ 是由 n 个 APP 以及各个 APP 所分配的权值 w_i（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：

$$\Upsilon = \{(a_i, w_i)\} \tag{3-8}$$

其中 a_i 表示 APP，$\sum_i w_i = 1$。

根据文献[21]中的讨论，权值 w_i 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$\mathrm{E}[\Upsilon]$，其可以计算如下：

$$\mathrm{E}[\Upsilon]=\sum_{i}^{n} w_i \bullet \Delta_i^{-1} \tag{3-9}$$

其中Δ_i表示在基于流行度的排序列表即表 3 - 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$\mathbb{R}[\Upsilon]$，其计算方式如下

$$\mathbb{R}[\Upsilon]=\sum_{i}^{n} w_i \bullet (1-\Delta_i^{-1}) \tag{3-10}$$

其中Δ_i表示在基于流量的排序列表即 3.2.1 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w_i，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[10]。所以具体的优化问题定义如下：

$$\arg\max \mathrm{E}[\Upsilon]-\mathbb{R}[\Upsilon] \tag{3-11}$$

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w_i，然后再根据这个权值，我们就可以得到 APP 推荐的排序顺序。

3.2.4 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 3 - 9 所示：

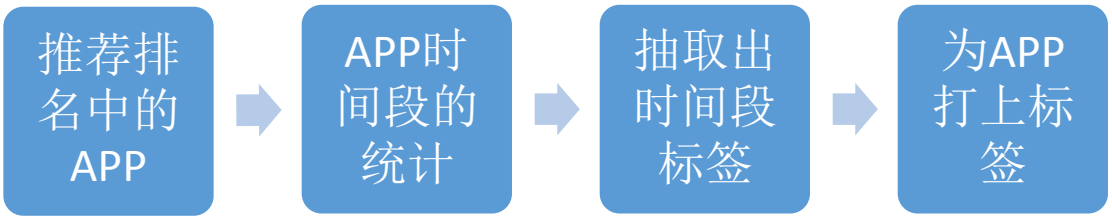

图 3-9 活跃时间段流程图

3.3 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偏好，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。

第四章 推荐结果分析

本节测试推荐模型所选用的数据，来自第二章中分析过的数据集，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或者提高用户 APP 流行度的目的。

4.1 视频类软件推荐结果分析

在第二章介绍的数据集中，首先根据前面介绍的用户使用偏好的相关分析，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 3.1.3 章节里的表 3-3 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 4-1 所示。

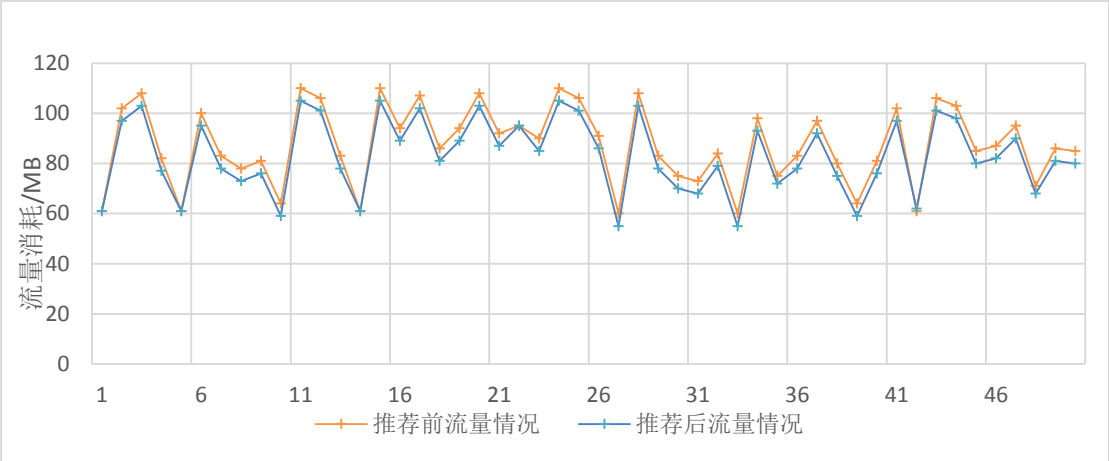

图 4 - 1 视频类软件推荐结果对比图

图 4 - 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。

4.2 下载类软件推荐结果分析

首先根据前面介绍的用户使用偏好的相关分析，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 3.1.3 章节里的表 3 - 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，所以这个推荐结虽然让用户的流量消耗有些许的增加，但是用户所使用的 APP 的流行度却得到了显著的增加。具体结果如图 4 - 2 所示：

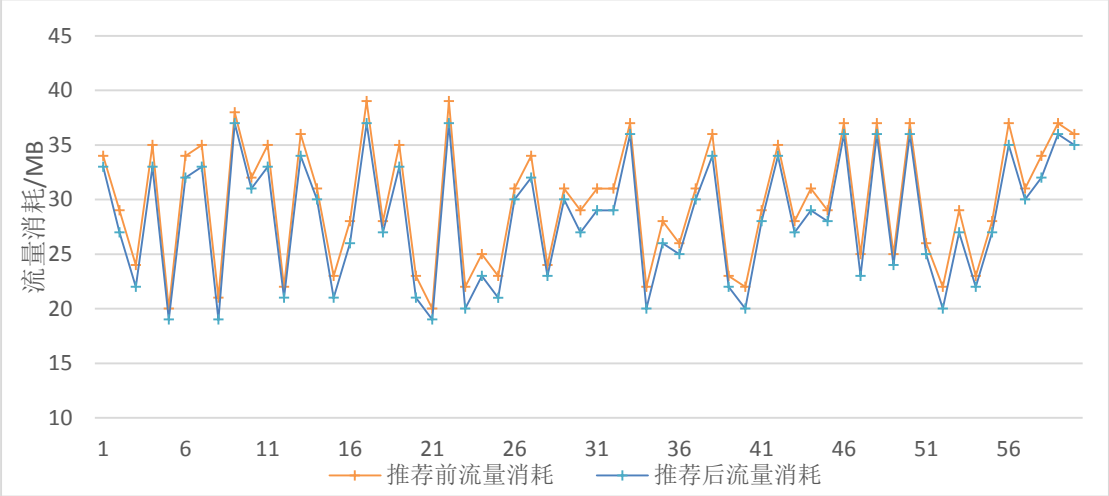

图 4 - 2 下载类软件推荐结果对比图

图 4 - 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。

4.3 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 3.1.3 章节里的表 3 - 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 4 - 3 所示：

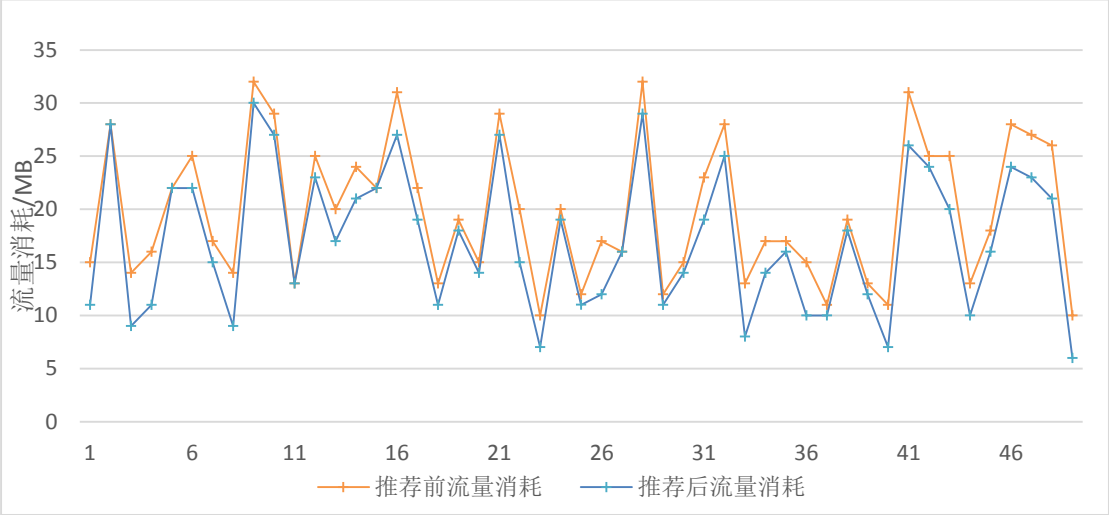

图 4 - 3 音乐类软件推荐结果对比图

图 4 - 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。

4.4 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 3.1.3 章节里的表 3 - 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 4 - 4。

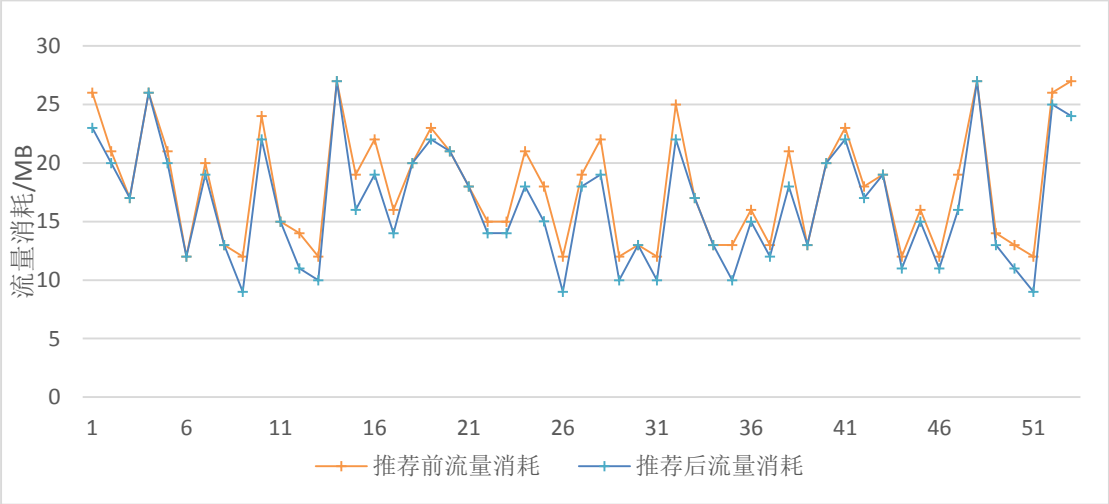

图 4 - 4 浏览器类软件推荐结果对比图

图 4 - 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。

4.5 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。

第五章 总结及展望

5.1 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 APP 的流量使用和流行度这一部分。

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，这就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。

最终的推荐结果表明，推荐后的 APP 能够在满足用户使用偏好的情况下减少流量的消耗，或者能够提高使用 APP 的流行度信息，最终提升了用户的使用体验。

5.2 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：数据规模需要是海量的（vast）、数据的流转性应该快速

和数据体系应该是动态的（velocity）、数据类型应该是多样的（variety）和数据价值非常大（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。

希望在未来的工作中，能够解决上面提到的不足，设计出一套更精确的移动 APP 推荐系统。

参考文献

[1] 中国互联网络信息中心. 第 36 次中国互联网络发展状况统计报告[EB/OL]. 2015.http://www.cnnic.net.cn/hlwfzyj/hlwxzbg/hlwtjbg/201507/P020150723549500667087.pdf

[2] 36 氪网. Google Play 应用数量首次超越 APP Store 2015 [EB/OL]. 2015. http://36kr.com/p/218796.html

[3] 199IT 中文互联网数据咨询中心. 2015 年中国大数据发展调查[EB/OL]. 2015. http://www.199it.com/archives/381589.html?from=timeline&isappinstalled=1

[4] 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.

[5] 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.

[6] 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.

[7] 杨艳. 下一代网络业务用户行为研究[D].西南交通大学,2012.

[8] 王璐. 移动互联网用户行为分析[D].重庆邮电大学,2012.

[9] 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.

[10] 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.

[11] 李威. 移动互联网用户行为分析研究[D].北京邮电大学,2013.

[12] 赵志勇. 移动 Hadoop 集群监控系统的设计与实现[D].北京交通大学,2015.

[13] 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.

[14] 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.

[15] 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.

[16] 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.

[17] 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.

[18] 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.

[19] 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.

[20] 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.

[21] 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.

[22] 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.

[23] 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.

[24] 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.

[25] 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

[26] 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

[27] 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

[28] 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，裘庚师兄、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，张小奕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