中山大学硕士学位论文

基于 Hadoop 的教育资源推荐系统的设计与实现

Design and Implementation of Education Resources Recommender System Based on Hadoop

学位申请人：于凯丰

指导教师：郑贵锋 讲师

专业名称：工程硕士（软件工程）

答辩委员会主席（签名）：

答辩委员会委员（签名）：

二〇一七 年 五 月 二十一 日

论文原创性声明

本人郑重声明：所呈交的学位论文，是本人在导师的指导下，独立进行研究工作所取得的成果。除文中已经注明引用的内容外，本论文不包含任何其他个人或集体已经发表或撰写过的作品成果。对本文的研究作出重要贡献的个人和集体，均已在文中以明确方式标明。本人完全意识到本声明的法律结果由本人承担。

学位论文作者签名：

日期：　　年　　月　　日

学位论文使用授权声明

本人完全了解中山大学有关保留、使用学位论文的规定，即：学校有权保留学位论文并向国家主管部门或其指定机构送交论文的电子版和纸质版；有权将学位论文用于非赢利目的的少量复制并允许论文进入学校图书馆、院系资料室被查阅；有权将学位论文的内容编入有关数据库进行检索；可以采用复印、缩印或其他方法保存学位论文；可以为建立了馆际合作关系的兄弟高校用户提供文献传递服务和交换服务。

保密论文保密期满后，适用本声明。

学位论文作者签名：　　导师签名：

日期：　　年　　月　　日　　日期：　　年　　月　日

论文题目：基于 Hadoop 的教育资源推荐系统的设计与实现

专　　业：工程硕士（软件工程）

硕 士 生：于凯丰

指导教师：郑贵锋 讲师

摘 要

互联网在中国的迅速发展和普及，为在线教育的发展提供了前提条件。在线教育资源利用互联网技术进行传播，使得人们可以随时随地获得优质的教育资源，解决了传统教育方式的资源分配不均、投入产出失衡等问题。但随着在线教育资源迅速增多，人们越来越难以找到自己需要的资源，产生了信息过载的问题。推荐系统是解决信息过载的有效方法，让推荐系统与在线教育平台进行结合，设计一个教育资源推荐系统是很多在线教育平台当前的首要任务。

本文源于国家自然科学基金支持的云教育平台构建若干关键技术创新研究项目。旨在针对云教育平台的需求和特点，对推荐系统相关理论进行研究，设计并实现一个适用于云教育平台的教育资源推荐系统，主要工作有以下三点：

（1）设计并实现了一个海量日志收集系统。该系统可以收集云教育平台上用户多个维度的行为日志，包括点击、浏览、评分和收藏等。系统使用了分布式消息队列和负载均衡等技术，以保证其可以应对高并发、高负载的日志收集需求。使用了 Flume 技术，负责对消息队列中的日志进行收集，传输并存储到 HDFS 中。

（2）本文对推荐系统相关理论和技术进行了研究，并基于协同过滤推荐算法实现了一个推荐引擎。该推荐引擎通过对用户行为日志分析建模，结合相应的算法，计算并获取推荐结果。该推荐引擎对基于用户的推荐和 Slope One 推荐两个算法进行了改进和优化，使其能够在 Hadoop 平台上并行化计算。该引擎还实现了基于教育资源热门度推荐和基于用户信息推荐等多种推荐算法。另外，该引擎还使用了 Mahout 框架中的基于物品的推荐和 ALS-WR 推荐等推荐算法。多种推荐算法的使用，使得本引擎可以处理不同类型的用户行为日志，并且解决了推荐系统冷启动的问题。本文还对不同推荐算法在不同配置参数下的推荐效果进行了测试，以找到适合本系统的推荐算法和配置参数。

（3）设计并实现了一个推荐服务器，提出了一个适合教育资源的混合推荐

方法。推荐服务器可以对推荐引擎计算得到的结果进行转存和处理，通过提供服务接口的形式，与云教育平台进行整合。另外，本文针对教育资源的特点，提出了一个适合教育资源的混合推荐方法，云教育平台可以通过混合推荐获得多个推荐算法的组合推荐。

本文最终实现了一个基于 Hadoop 的教育资源推荐系统，通过提供服务接口实现了对云教育平台资源的推荐。

关键词：推荐系统，协同过滤，Hadoop，MapReduce

Title： Design and Implementation of Education Resources Recommender System Based on Hadoop

Major：Software Engineering

Name：Kaifeng Yu

Supervisor： Lecturer Guifeng Zheng

Abstract

The rapid development and popularization of the Internet in China provides a prerequisite for the development of online education. Online education resources use Internet technology to spread, so that people can get quality education resources at any time and place, and online education solves the problem of uneven distribution of resources and imbalance of input and output which traditional education mode has. But with the rapid increase of online education resources, it is more and more difficult to find the resources we need, it causes information overload problem. Recommender system is an effective way to solve the problem of information overload. Combing the recommender system with the online education platform and designing an education resources recommender system become the most important task of many online education platforms.

This paper comes from Research on Key Enabling Technologies for Cloud Based Education, NSFC. According to the requirements and characteristics of cloud based education platform, studied the theories of recommender system, this paper has designed and implemented a recommender system which is suitable for cloud based education platform. The main works of this paper consist three parts:

(1) This paper has designed and implemented a mass log collection system. It can collect a variety of types of user's behavior logs, including click, browse, grade, favorite, etc. It uses some technologies including distributed message queue and load balancing, to ensure that it can deal with high concurrent, high load log collection requirements. It uses Flume to collect logs which are stored in distributed message queue and save them into HDFS.

(2) This paper has studied the theories and technologies of recommender system, and has implemented a recommendation engine based on collaborative filtering recommendation algorithms. This engine can analyze and model user's behavior logs, and can use recommendation algorithms to get recommendation results. This recommendation engine has improved and optimized the user based and the Slope One recommendation algorithms, to make them can implement parallel computing on Hadoop. This engine has also implemented some recommendation algorithms which are based on the popularity of education resources or user information. In addition, the engine also uses the item based and ALS-WR recommendation algorithms which are implemented by Mahout. The use of a variety of recommendation algorithms, makes this engine can deal with different types of user's behavior logs, and it solves the cold boot problem of recommender system. This paper also tested the recommendation accuracy of each algorithm when they get different configuration parameters, to find the best algorithm and configuration parameters which are suitable for this system.

(3) This paper has designed and implemented a recommendation server, and proposed a new hybrid recommendation method which is suitable for education resources. The recommendation server can store and process the results of recommendation engine, and it provides service interfaces to integrate with the cloud based education platform. In addition, according to the characteristics of education resources, this paper has proposed a hybrid recommendation method for education resources, which can combine multiple recommendation algorithms.

Finally, this paper has implemented an education resources recommender system based on Hadoop. It provides the recommendation service for cloud based education platform to recommend education resources by service interfaces.

Keywords: Recommender System, Collaborative filtering, Hadoop, MapReduce

目录

第 1 章 绪论

1.1 研究背景及意义

近年来，随着经济的快速发展，互联网在中国得到了普及，教育资源以一种新的形式呈现在人们面前——在线教育资源。在很长的一段时间内，中国的教育资源一直存在“资源分配不均、投入产出失衡、素质教育水平较低”等问题[1]，矛盾非常突出。而在线教育有着非常显著的优点，可突破时空限制，传播速度快，容易获取，教学形式多样，学习形式可交互，任何人只要有一台联网的电脑就可以获得优质的教育资源，越来越受到人们的喜爱。国务院也专门提到了，要“积极发展‘互联网+教育’”[2]。在线教育的“春天”正在来临。

但是随着在线教育的发展，教育资源迅速增多，呈现一种爆炸性的增长的趋势，从浩瀚的教育资源海洋中找到自己需要的资源已经越来越难，造成了信息过载[3]。按照传统解决信息过载的方法，网站可以提供分类目录和搜索引擎，但这仅仅解决了用户在需求明确情况下的信息过载问题[4]。推荐系统是一种比较新的方法，它可以根据用户的历史行为记录，推测用户兴趣，从而进行个性化推荐，并不需要用户事先已经明确自己的需求。

本文来源于国家自然科学基金支持的云教育平台构建若干关键技术创新研究项目。该项目最终实现了一个云教育平台，同所有在线教育网站一样，云教育平台存储了大量的教育资源，包括文档、视频、音频等，并且该平台已经实现了分类目录和搜索引擎的功能。在此基础上我们在云教育平台中加入了教育资源推荐系统，它能够给每个用户推荐不同的教育资源，使得每个用户都有一所“专属学校”，从而解决了信息过载的问题，能够让优质的教育资源更多地被需要的人发现，极大的提高用户体验。

1.2 国内外研究现状

1.2.1 推荐系统

推荐系统是一种解决信息过载的工具，它通过对用户的历史行为分析建模，推测用户的兴趣偏好，并给他们推荐与其兴趣相关的物品或信息。对于推荐系统

的研究，要追溯到 20 世纪 90 年代，纵观其发展历程可以分为以下三个阶段[5]：

（1）理论准备阶段。本阶段产生了推荐系统的几个重要理论，为推荐系统发展奠定了基础。1992 年，Glodberg 等人发表了关于 Tapestry 邮件过滤系统的论文[6]，论文中使用了用户对邮件的显式反馈信息对邮件进行过滤，首次提出了“协同过滤”的概念。1994 年 Resnick 等人发表了一篇论文，介绍了 GroupLens 系统对新闻进行过滤并帮助用户获取他们感兴趣新闻所使用的协同过滤方法[7]。1997 年 Resnick 发表论文[8]，并在论文中首次提到了“推荐系统”的概念。

（2）商用阶段。推荐系统与商业应用相结合，使得推荐系统从单纯的理论研究走向应用，产生了巨大的经济价值。电子商务网站亚马逊（Amazon）使用了基于物品的协同过滤推荐技术[9]，亚马逊前首席科学家 Andreas Weigend 透露，亚马逊有 20%~30%的销售来自于推荐系统[4]。不仅是电商网站，推荐在各个领域都有很广泛的应用，包括影视、音乐、社交网络等。表 1-1 列出了推荐系统在国内外各个商业领域的应用实例。

表 1-1 推荐系统应用实例

	国内	国外
电子商务	京东、当当、淘宝、天猫	Amazon、ebay
影视	优酷、爱奇艺、腾讯视频	Netflix、YouTube
音乐	网易云音乐、豆瓣电台	Pandora、Last.fm
社交网络	新浪微博、QQ	Facebook、Twitter、LinkedIn
阅读	今日头条	Google Reader、Digg
广告	微信、今日头条	Facebook、Google

（3）大规模发展阶段。推荐系统得到社会各界的重视，不断涌现出新的优秀推荐算法。2000 年 Miyahara 和 Pazzani 提出了一种使用简单贝叶斯分类器的协同过滤技术[10]。2000 年，Sarwar 等人将奇异值分解（SVD）引入了推荐系统[11]，用于发现数据中的潜在关系。同年，Burke 提出了一种新的推荐方法，基于知识的推荐[12]。2001 年 Sarwar 等人发表论文[13]提出了基于物品的推荐，解决了传统基于用户的协同过滤技术在注册用户过多情况下计算效率低下的问题。同年，Goldberg 发表论文，将主成分分析技术引入了推荐系统[14]。2004 年 Hofmann 提出了隐语义模型[15]。2005 年 Lemire 和 Maclachlan 提出了 Slope One 推荐算法[16]，这种算法与其他推荐算法相比更加简单和高效。2006~2009 年 Netflix 公司举办了

推荐系统大赛，旨在希望研究人员能够提升 Netflix 推荐算法的准确率，期间许多优秀的推荐系统研究人员参与到比赛中来，他们根据不同推荐算法的特点对其进行优化改进，并把它们混合，产生了很多优秀的推荐算法[17][18][19]。 近年来，随着大数据技术的发展，推荐系统与大数据技术产生了紧密结合[20],提出了适合并行化计算的 SGD[21]和 ALS[22]等算法。在应用领域也产生了很多提供推荐功能的软件系统，如 Apache Mahout 和 Spark Mllib 等。在学术方面，美国 ACM 协会从 2007 年开始，每年都会举行推荐系统大会，到 2016 年已经举行了 10 届，会议主要交流推荐系统的一些新的研究方法和成果。

1.2.2 在线教育

互联网相关技术的发展和普及，为在线教育的发展创造了前提条件，特别是近几年移动互联网的爆发，使得人们可以随时随地获取各种优秀的教育资源。根据对在线教育教学模式的研究和分析，它的教学模式可分为如下几个[23]：

（1）UGC 模式。本模式中的教学资源是由互联网用户原创的，其他用户可以根据兴趣爱好选取需要的资源。互联网在这里只是提供了一个资源传播的工具。目前国内直播平台上的教育课堂很多都采用了这种模式。

（2）MOOC 模式。本模式中的教学资源大多数来自于世界著名高校的课堂录像，使得世界上每个人都有机会接触到顶尖的教学资源。后来产生了 Coursera、Udacity、edX 等课程提供商，可以为学生提供更多的优质教学资源。

（3）翻转课堂模式。本模式区别于传统教学模式，学生的知识学习过程是自己完成的，而老师则负责对学生进行引导和解惑[24]。

在线教育刚刚兴起，会存在教学质量良莠不齐、课堂缺乏互动性和学生自控力差等缺点，这些问题亟需解决。目前在线教育正逐步与云计算和大数据等新兴技术结合[25][26]，改善了在线教育网站的数据处理方式。未来在线教育会朝着移动化、免费化、互动化和个性化等方向发展[27]。

1.3 本文主要工作

本文的主要工作包括以下三点：

一是设计并实现了一个海量日志收集系统，可以对云教育平台用户多个维度的行为日志进行收集，包括点击、浏览、评分、评论和收藏等行为，并把日志存

储到 Hadoop 的 HDFS 中，以作为推荐引擎的输入。它采用了 Nginx、Kafka 和 Flume 等技术，以保证本系统能够应对高并发、高负载的日志收集需求。

二是研究设计并实现了基于协同过滤算法的推荐引擎。主要负责对用户行为日志建模分析，并计算生成推荐结果。它采用两种计算方式，离线计算和在线计算。其中离线计算使用的是 Hadoop 的 MapReduce 计算模型，负责数据量大、耗时长的计算任务，使用了基于用户的推荐、基于物品的推荐、Slope One 推荐和 ALS-WR 推荐等协同过滤推荐算法。本文对 Slope One 推荐算法和基于用户的推荐算法做了改进和优化，使其能够在 Hadoop 集群上并行计算，即 MapReduce 化。基于物品的推荐和 ALS-WR 推荐则使用了 Mahout 框架提供的实现。在线计算负责获取基于教育资源热门度和基于用户信息推荐的推荐结果。文中还测试了不同推荐算法在不同配置参数下的推荐效果，以找到适合本系统的推荐算法和配置参数。

三是设计并实现了一个推荐服务器并且提出了适合教育资源的混合推荐方法。推荐服务器对推荐引擎所得到的推荐结果进行转存过滤和处理，设计并封装了一系列接口供云教育平台前端调用，以获得推荐结果，方便教育资源推荐系统与研究项目中其他的子系统进行整合。针对不同的推荐场景，设计了根据用户兴趣推荐、根据教育资源相关性推荐、根据用户信息推荐、根据教育资源热门度推荐等多种推荐方式。并且，本文提出了适合教育资源的混合推荐方法，云教育平台可以通过混合推荐获得多个推荐算法的组合推荐。

1.4 论文结构

本文共分为 7 章，各章具体内容如下：

第 1 章，绪论。介绍了本文的研究背景，阐释了本文的研究内容及意义，并且讨论了推荐系统和在线教育在国内外的研究现状，最后说明了本文的结构。

第 2 章，推荐系统理论研究及实现技术。简单介绍了推荐系统领域基本理论。阐释了相似度度量和推荐系统的各种评测方法。详细讨论了协同过滤推荐中基于用户的推荐、基于物品的推荐、Slope One 推荐和 ALS-WR 推荐等算法。最后，介绍了推荐系统在实现过程中使用到的相关技术。

第 3 章，推荐系统需求分析。对推荐系统进行了需求分析，主要从业务需求、

功能需求和非功能需求三个方面展开讨论。

第 4 章，推荐系统概要设计。对推荐系统进行了概要设计，主要从总体设计、功能模块划分、技术架构、数据库设计和日志埋点规范设计五个方面展开讨论。

第 5 章，推荐系统详细设计。对推荐系统进行了详细设计，根据概要设计对推荐系统的模块划分，详细实现了各模块的算法和功能。

第 6 章，推荐系统运行与测试。对推荐系统部署、运行，对推荐系统相关功能和提供的服务接口进行测试，并对部分推荐算法进行实验分析。

第 7 章，总结与展望。对本文的研究工作进行总结，归纳其中的优缺点，并对本文可改进的地方进行探讨，展望推荐系统未来的发展方向。

第 2 章 推荐系统理论研究及实现技术

本章主要介绍推荐系统的相关理论，包括三种主要的推荐算法、相似度度量和推荐系统评测标准，并详细介绍协同过滤领域中的四种重要算法，最后对推荐系统在实现过程中使用到的关键工具和技术进行介绍。

2.1 推荐算法

2.1.1 协同过滤推荐

协同过滤推荐是最早出现的推荐算法。该算法的主要思想是，将用户的历史行为进行建模，比如对某物品的评分、购买、收藏等行为，然后通过相应的算法获得用户的兴趣偏好，最后根据用户兴趣偏好进行推荐[28][29]。协同过滤推荐最大的优点就是不需要了解用户或商品的大量信息，只需要用户对物品的行为数据。

协同过滤推荐算法自产生起就得到了迅猛的发展，在业界得到了广泛应用。它包括了很多算法，比如基于用户的推荐、基于物品的推荐、Slope One 推荐、隐语义模型等。

2.1.2 基于内容的推荐

算法的主要思想是，将物品内容抽象成物品的属性特征，然后根据用户的兴趣偏好，推荐给他与其偏好相似的物品。一个典型的例子是，某用户喜欢体育新闻，并且特别关注足球和篮球，那么推荐系统就可以把有关足球和篮球的体育新闻推荐给他。从该算法的原理来看，基于内容的推荐需要两种类型数据，一类是物品的属性特征，另一类是用户的兴趣偏好。用户的兴趣偏好可以通过对用户的历史行为分析，或对用户在线调查获得。而获取物品的属性特征就比较难，因为基于内容的推荐算法希望能够自动从物品内容中抽象并提取出相关属性特征，这就用到了很多信息检索和信息过滤领域的技术，比如 TF-IDF 算法[30]、向量空间模型[31]和朴素贝叶斯文本分类[32]等。

在应用方面，基于内容的推荐常用于各大新闻网站和音乐视频网站，并且该算法与协同过滤推荐结合后推荐效果更好。

2.1.3 基于知识的推荐

上文介绍的协同过滤推荐和基于内容推荐的前提条件是用户要产生大量的行为数据，通过分析这些数据了解到用户的兴趣偏好进行推荐。现在考虑一种特殊情况，对于某一类物品，用户并不会频繁的对其产生行为，比如，用户不会在短时间内多次购买汽车、手机和电脑等物品，并且用户的兴趣会随着时间改的变和社会地位的改变而发生变化，所以对于这类物品单纯地使用协同过滤推荐，效果是非常不好的。基于知识的推荐解决了这个问题[33]，它是通过对用户兴趣的调查，明确用户需求，从物品集中找到与用户需求匹配的物品，进行推荐。比如，某用户想要购买一辆国产、黑色、价位在 15-20 万之间的 SUV 汽车，那么基于知识的推荐系统就可以从所有汽车中筛选出符合条件的汽车推荐给用户。

基于知识的推荐可以分为两类，分别为基于实例推荐[34]和基于约束推荐[35]，二者的原理大体相同，不同之处在于基于实例推荐是根据用户的需求找到与其相似的物品推荐给他，而基于约束的推荐则严格按照用户的需求约束，找到严格符合用户需求的物品并推荐给他。

2.2 相似度度量

在进行推荐计算时，需要计算用户之间或物品之间的相似度，这就会用到相似度度量。下面介绍几种在推荐系统中常用的相似度度量[36]。

设在 n 维空间中有两个向量，分别为 $\mathbf{a} = (a_1, a_2, \dots, a_n)$ 和 $\mathbf{b} = (b_1, b_2, \dots, b_n)$，向量 $\mathbf{a}$和$\mathbf{b}$之间的相似度为$\mathrm{sim}(\mathbf{a}, \mathbf{b})$。

2.2.1 欧氏距离相似度

欧氏距离相似度把用户对物品的评分向量视为空间中的点，通过计算两个点之间距离大小来衡量向量间的相似度。

设向量 $\mathbf{a}$和$\mathbf{b}$之间的欧氏距离为$\mathrm{dist}(\mathbf{a}, \mathbf{b})$，公式可以定义如下。

$$\mathrm{dist}(\mathbf{a}, \mathbf{b}) = \sqrt{\sum_{i=1}^{n}(a_i - b_i)^2} \tag{2-1}$$

对于向量 $\mathbf{a}$和$\mathbf{b}$，二者之间的距离越小，说明它们越相似。实际应用中，使用欧氏距离计算向量间相似度的公式如下。

$$\mathrm{sim}(\mathbf{a}, \mathbf{b}) = \frac{1}{1+\mathrm{dist}(\mathbf{a},\mathbf{b})} \tag{2-2}$$

公式的取值区间为[0,1]，dist(**a**, **b**)越大，sim(**a**, **b**)越趋近于 0，向量 **a**和**b**越不相似，反之亦然。

2.2.2 曼哈顿距离相似度

曼哈顿距离同欧式距离计算向量相似度的原理类似，只是二者计算距离的方式不同，曼哈顿距离通过计算两点在标准坐标系上的绝对轴距离总和来计算向量间距离。

设向量**a**和**b**之间的曼哈顿距离为d(**a**, **b**)，公式可以定义如下。

$$\mathrm{d}(\mathbf{a},\mathbf{b}) = \sqrt{\sum_{i=1}^{n}|a_i - b_i|} \tag{2-3}$$

同欧氏距离一样，向量**a**和**b**之间的距离越小，向量越相似。实际应用中，使用曼哈顿距离计算相似度的公式如下。

$$\mathrm{sim}(\mathbf{a},\mathbf{b}) = \frac{1}{1+d(\boldsymbol{a},\boldsymbol{b})} \tag{2-4}$$

取值区间为[0,1]，d(**a**, **b**)越大，sim(**a**, **b**)越趋近于 0，向量 **a**和**b**越不相似，反之亦然。

2.2.3 余弦相似度

余弦相似度通过计算两个向量之间夹角的余弦值来计算相似度，其公式可以定义如下。

$$\mathrm{sim}(\boldsymbol{a},\boldsymbol{b}) = \frac{\boldsymbol{a}\cdot\boldsymbol{b}}{\boldsymbol{a}*\boldsymbol{b}} = \frac{\sum_1^n(a_i \times b_i)}{\sqrt{\sum_1^n {a_i}^2}\times\sqrt{\sum_1^n {b_i}^2}} \tag{2-5}$$

取值区间为[-1,1]，向量之间的夹角越小，余弦相似度值越接近于 1，此时两个向量接近重合，两个向量越相似。反之亦然。

基本的余弦相似度是存在缺陷的，它着重考虑两个向量在方向上的相似性，而没有考虑每个向量内部各维度数值的大小。可能会发生一种极端情况，两个向量方向基本相同，但向量的模却相差很大，在这种情况下，使用基本余弦相似度所得到的结果是不准确的。在推荐系统中，修正的余弦相似度针对上述问题进行了改进，在计算过程中向量内部每个维度都需要减去用户对物品评分的平均值。其公式可以定义如下，其中$\bar{u}$表示用户对物品评分的平均值。

$$\mathrm{sim}(\mathrm{a},\mathrm{b}) = \frac{\sum_{i=1}^{n}(a_i-\bar{u})(b_i-\bar{u})}{\sqrt{\sum_{i=1}^{n}(a_i-\bar{u})^2}\sqrt{\sum_{i=1}^{n}(b_i-\bar{u})^2}} \tag{2-6}$$

2.2.4 皮尔逊相关系数

皮尔逊相关系数表征的是两个向量之间的相关程度，其公式可以定义如下。

$$\mathrm{sim(a,b)} = \frac{\sum_{i=1}^{n}(a_i-\bar{a})(b_i-\bar{b})}{\sqrt{\sum_{i=1}^{n}(a_i-\bar{a})^2}\sqrt{\sum_{i=1}^{n}(b_i-\bar{b})^2}} \tag{2-7}$$

取值区间为[-1,1]。当取值趋近于 1 时，两个向量强正相关，二者也越相似。当取值趋近于-1 时，两个向量强负相关，二者完全不同。

2.2.5 谷本系数

谷本系数是用来计算布尔型向量之间相似度的，它不关心用户评分是多少，只关心用户是否表达过偏好。通过计算两个用户各自表达过偏好物品的交集大小与它们并集大小的比值来获取二者的相似度。公式可以定义如下，其中 N 代表集合的大小。

$$\mathrm{sim(A,B)} = \frac{N(A\cap B)}{N(A\cup B)} \tag{2-8}$$

2.3 推荐系统评测

衡量一个推荐系统的好坏，需要有明确的评测指标。下面将简要介绍推荐系统领域中常用的几种评测指标。

2.3.1 用户满意度

用户对系统推荐给他们的物品是否满意，是推荐系统评测的重要指标之一。对于用户满意度，只能通过在线调查或者在线实验获取。

另外，除了以上两种方式，网站还可以通过用户的行为数据间接统计用户满意度。比如对于电商网站，如果推荐给用户的物品，用户点击并且购买了，那么就说明用户对本次推荐是满意的。

2.3.2 预测准确度

预测准确度用来衡量推荐系统预测用户行为的能力，通过离线计算获取。是目前应用最广泛的推荐系统评测指标。在计算时把用户评分数据集分成训练集和测试集，推荐系统使用训练集分析并给用户推荐相关物品，最后通过计算给用户推荐的物品在测试集中的重合率来衡量预测准确度。预测准确度有以下两个分类。

（1）评分预测

评分预测的原理很简单，就是根据用户的历史行为数据（给物品的评分），预测用户在见到未评分的物品时会给该物品打多少分。

评分预测准确度有两个计算方法，一个是均方根误差（RMSE），另一个是平均绝对误差（MAE）。均方根误差公式可以定义如下。

$$\mathrm{RMSE} = \sqrt{\frac{\sum_{u,i\in T}(r_{ui} - \widehat{r_{ui}})^2}{|T|}} \tag{2-9}$$

平均绝对误差公式可以定义如下。

$$\mathrm{MAE} = \frac{\sum_{u,i\in T}|r_{ui} - \widehat{r_{ui}}|}{|T|} \tag{2-10}$$

其中，u 指的是一个用户，i 指的是一个物品，r_{ui}指的是用户 u 给物品 i 的评分，$\widehat{r_{ui}}$指的是推荐系统预测的用户 u 给物品 i 的评分。

（2）TopN 预测

现在的大部分网站在给用户做推荐时，都会生成一个推荐列表，比如，有的电商网站会给用户生成“购买过此商品的用户还会购买”、“与此商品相似的商品”这样的推荐列表，类似这样的推荐叫做 TopN 预测。它的准确度是通过计算准确率（Precision）与召回率（Recall）来获取的。准确率的公式可以定义如下。

$$\mathrm{Precision} = \frac{\sum_{u\in U}|R(u)\cap T(u)|}{\sum_{u\in U}|R(u)|} \tag{2-11}$$

召回率的公式可以定义如下。

$$\mathrm{Recall} = \frac{\sum_{u\in U}|R(u)\cap T(u)|}{\sum_{u\in U}|T(u)|} \tag{2-12}$$

对于准确率和召回率，在推荐系统中由于物品的总数庞大，所以如果 TopN 很小，那么给用户推荐的物品列表跟测试集的重合也会比较小，所以在推荐系统中，准确率和召回率只是一个小的参考维度，目前更多使用 MAE 和 RMSE。

2.3.3 覆盖率

目前不管是电商网站还是音乐视频网站，它们都希望推荐系统有发现长尾物品的能力。覆盖率就是衡量这种能力的评测标准。最常见的计算方式是，用推荐的物品数除以物品的总数。公式可以定义如下。

$$\mathrm{Coverage} = \frac{|U_{u\in U}R(u)|}{|I|} \tag{2-13}$$

其中，U 表示所有用户，$R(u)$表示给某个用户的推荐列表。

通过公式可以看出，如果只给用户推荐热门物品，那么覆盖率将会很低，因为热门物品只会在所有物品中占据比较小的比例。如果系统的覆盖率为 100%，那意味着所有的物品都会被推荐给特定的用户。

2.3.4 多样性

每个用户的兴趣都是多种多样的，推荐系统的多样性是指，系统可以根据用户兴趣的不同，推荐给用户不同类别的物品，增加用户的满意度。对于单个用户 u，给他推荐物品的多样性可以定义如下。

$$\text{Diversity}\big(\text{R(u)}\big) = 1 - \frac{\sum_{i,j\in R(u),i\neq j} s(i,j)}{\frac{1}{2}|R(u)|(|R(u)|-1)} \tag{2-14}$$

其中，$s(i,j)$表示物品 i 和 j 之间的相似度，值域为[0,1]。

对于整个系统而言，整体多样性可以通过计算所有用户推荐列表多样性的平均值来计算，公式可以定义如下。

$$\text{Diversity} = \frac{1}{|U|}\sum_{u\in U} Diversity(R(u)) \tag{2-15}$$

2.3.5 新颖性

新颖性用来描述推荐系统给用户推荐物品的新颖程度。举例说明，在一个电商网站，如果一个用户在几个月内频繁查看某个商品，但用户并没有购买这个商品，此时如果推荐系统把这个商品推荐给该用户，那么这属于一次失败的推荐，因为该用户已经知道并且有可能喜欢此商品，此时推荐给他是没有意义的。推荐系统应该给用户推荐他们以前没见过的、不熟悉的物品。网站可以通过去除推荐列表中用户以前浏览过或表达过喜好的物品来提高推荐的新颖性。

计算推荐新颖性最简单的方法是，计算推荐列表中被推荐物品的流行程度的平均值，因为物品越不流行，用户就会觉得越新颖。

2.3.6 惊喜性

如果系统给用户推荐的物品中，并不是所有的都根据用户的历史兴趣推荐，有一部分物品用户从来没有表达过兴趣，但用户看了之后却非常满意，这样用户会感到惊喜，这就是惊喜性。

对于惊喜性，目前学术界并没有确定的计算方式，但网站可以通过在线实验或用户调查获取。

2.4 协同过滤推荐算法

2.4.1 基于用户的推荐

基于用户的推荐是一种经典的协同过滤推荐算法。它的核心思想就是，给定一个用户和一个用户对物品的评分矩阵，根据本用户的历史行为和其他用户的历史行为找到与此用户兴趣偏好相似的用户，即寻找用户邻域，找到用户邻域后，根据邻域中用户的历史行为，给此用户没见过的物品预测评分，然后根据一定规则进行推荐。即推荐给用户与其爱好相似用户喜欢的物品。基本原理如图 2-1 所示。

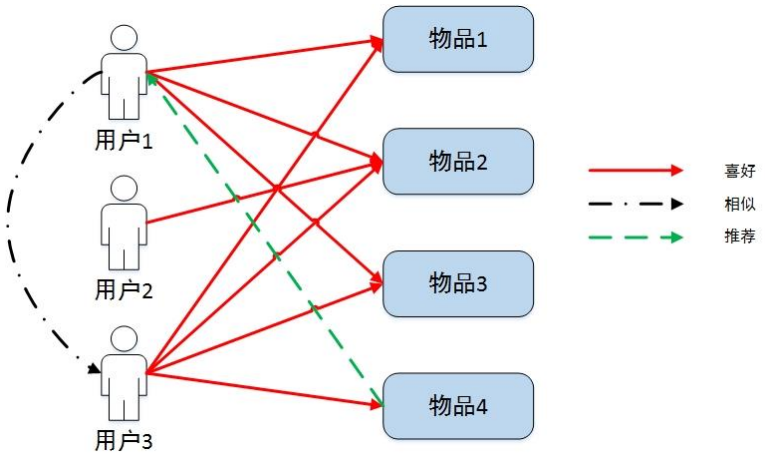


图 2-1 基于用户的推荐

在图 2-1 中，用户 1 和 3 同时对物品 1、2 和 3 表达过喜好，所以此处得到用户 1 的一个相似用户，用户 3。而用户 3 对物品 4 表达过喜好，用户 1 没有见过物品 4，此时根据基于用户推荐的规则，系统就可以把物品 4 推荐给用户 1。

下面通过一个的例子解释该算法的原理，该例子该算法预测用户 1 对物品 107 的评分。表 2-1 列出了用户物品评分矩阵。计算过程可分为以下几个步骤。

表 2-1 用户物品评分矩阵

物品 / 用户	101	102	103	104	105	106	107
1	5.0	1.0	3.5	5.0	4.5	2.0	?
2	4.5	2.5	3.5	4.0	4.5	3.0	4.5
3	4.0	1.5	2.5	4.5	5.0	1.0	4.5
4	3.0	3.5	3.5	5.0	4.5	5.0	1.5
5	1.0	4.5	2.5	1.0	2.0	4.0	1.5

（1）计算用户间相似度，构造相似度矩阵

使用提到皮尔逊相关系数，即公式(2-7)来计算用户间的相似度。经过计算，得到如表 2-2 的用户相似度矩阵。

表 2-2 用户相似度矩阵

	1	2	3	4	5
1		0.952	0.902	0.035	-0.987
2	0.952		0.708	0.235	-0.743
3	0.902	0.708		-0.222	-0.885
4	0.035	0.235	-0.222		0.233
5	-0.987	-0.743	-0.885	0.233	

从表 2-2 中可以看出，用户 1 跟用户 2 和 3 最相似，相关系数达到 0.9 以上，可以推断三者有着相同的喜好。用户 1 跟用户 4 的相关系数为 0.035，基本无关，跟用户 5 的相关系数为-0.987，二者强负相关，可以推断二者的喜好完全不同。

（2）寻找用户邻域

寻找用户邻域，即找到一定数量跟被计算用户相似的用户，这个数量是不确定的，用户邻域选取的大小直接影响到推荐算法的准确度。以下列出了两种寻找用户邻域的方法。

1）固定大小的领域

首先确定一个整数 n，然后从用户相似度矩阵中找到 n 个与被计算用户最相似的用户，构建邻域。根据用户相似度矩阵，以用户 1 为中心作图，如图 2-2。

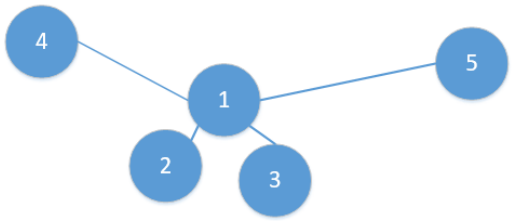


图 2-2 固定大小的领域

从图中可以看出，当 n 取 2 时，用户 2 和 3 都会被选进用户 1 的邻域。但是当 n 取 3 时，用户 4 也会被选进邻域，而用户 4 跟用户 1 几乎无关，如果选进邻域会导致推荐准确度降低。所以，实际应用中要通过实验选择推荐算法适合的邻域。

2）基于阈值的邻域

首先确定一个阈值，当用户间相似度超过这个阈值时才可以被选进邻域，这样就会避免邻域中被选入与用户毫不相关或完全负相关的用户。本例中当阈值取 0.7 时，用户 2 和用户 3 都可以被选入用户 1 的邻域，如图 2-3 所示。

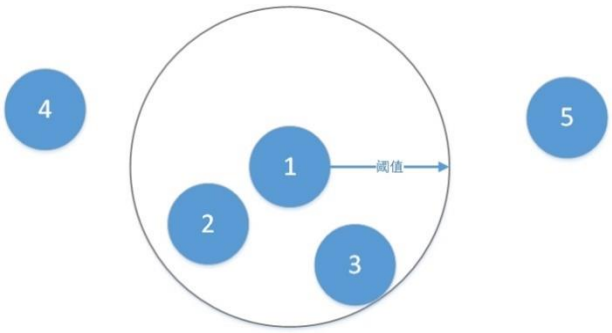


图 2-3 基于阈值的邻域

（3）预测评分

根据用户邻域和用户对物品的评分计算用户对未评价物品的预测评分，下面公式给出了计算用户 a 对物品 p 预测评分的计算方法，其中，N 代表用户邻域。

$$\mathrm{pred(a,p)} = \bar{r_a} + \frac{\sum_{b \in N} sim(a,b) * (r_{b,p} - \bar{r_b})}{\sum_{b \in N} sim(a,b)} \tag{2-16}$$

本例中，选取用户 2 和 3 作为用户 1 的邻域，则根据计算公式，用户 1 对物品 107 的评分为：

$$3.5 + \frac{0.952*(4.5-3.786)+0.902*(4.5-3.286)}{0.952+0.902} = 4.457$$

（4）进行推荐

根据计算得到的预测评分，经过排序过滤后，进行推荐。比如，当预测评分大于 3.0 时，就可以把该物品推荐给用户，本例中用户 1 对物品 107 的评分为 4.457，可以推荐给用户 1。

2.4.2 基于物品的推荐

基于物品的推荐也是应用非常广泛的一个推荐算法。它的核心思想是，给定一个用户和一个用户对物品的评分矩阵，计算获取物品间相似度，然后通过物品间相似度计算得到该用户对未评分物品的预测评分，最后根据预测评分进行推荐。即推荐给用户与其喜欢物品相似的物品。基本原理如图 2-4 所示。

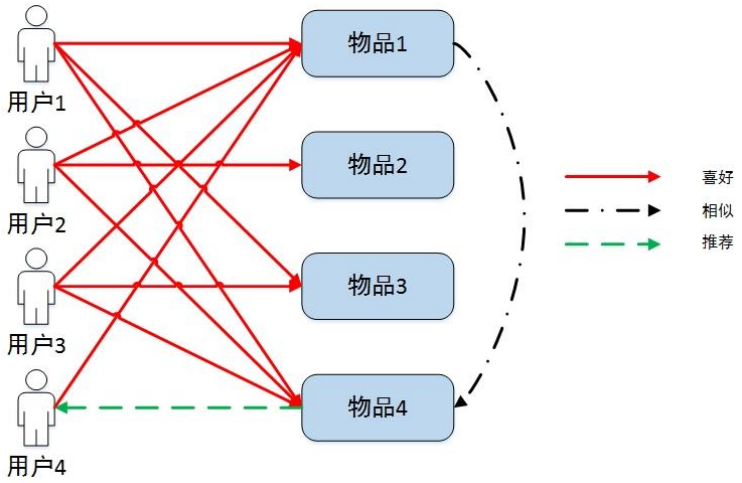


图 2-4 基于物品的推荐

在图 2-4 中，物品 1 和 4 同时被 3 个用户表达过喜好，此处判断物品 1 和 4 为相似物品。而用户 4 又对物品 1 表达过喜欢好，物品 1 和 4 为相似物品，此时根据基于物品推荐的规则，系统可以把物品 4 推荐给用户 4。

下面通过一个的例子解释该算法的原理，该例子通过基于物品的推荐算法预测用户 1 对物品 107 的评分。计算数据使用表 2-1 提供的用户物品评分矩阵。计算过程可分为以下几个步骤。

（1）计算物品间相似度，构造相似度矩阵

使用余弦相似度，即公式(2-5)计算物品间相似度并构造相似度矩阵。经过计算，得到如下的物品相似度矩阵。

表 2-3 物品相似度矩阵

	101	102	103	104	105	106	107
101		0.681	0.95	0.977	0.979	0.743	0.97
102	0.681		0.868	0.723	0.782	0.967	0.698
103	0.95	0.868		0.957	0.974	0.913	0.882
104	0.977	0.723	0.957		0.989	0.801	0.893
105	0.979	0.782	0.974	0.989		0.826	0.939
106	0.743	0.967	0.913	0.801	0.826		0.658
107	0.97	0.698	0.882	0.893	0.939	0.658	

（2）预测评分

通过计算用户 1 对所有与物品 7 相似物品的加权平均值来预测用户 1 对物品 7 的评分，权值即是物品相似度。下面公式给出了计算用户 a 对物品 p 预测评分的计算方法，其中，N 为用户 1 评分过的物品集合。

$$\mathrm{pred(a,p)} = \frac{\sum_{i \in N} sim(i,p) * r_{a,i}}{\sum_{i \in N} sim(i,p)} \quad (2\text{-}17)$$

根据以上公式，计算得到用户 1 对物品 7 的评分为 3.7

（3）进行推荐

根据计算得到的预测评分，经过排序过滤后，进行推荐。本例中预测用户 1 对物品 107 的评分为 3.7，可以推荐给用户 1。

2.4.3 Slope One 推荐

Slope One 算法是一种新的推荐算法，与其他的协同过滤推荐算法相比，Slope One 更为简单，实现更容易，效率更高，并且推荐准确率一点也不逊色于其他算法。如今，Slope One 推荐算法在电子商务网站、视频网站和音乐网站等都有广泛的应用。

Slope One 推荐算法的核心思想是，利用用户对不同物品的评分差异来预测用户对某一物品的偏好程度。下面通过一个的例子解释 Slope One 推荐算法的原理，该例子使用 Slope One 算法预测用于 1 对物品 103 的评分。表 2-4 列出了用户物品评分矩阵。该算法计算过程可分为如下 3 个步骤。

表 2-4 用户物品评分矩阵

物品 / 用户	101	102	103
1	1.0	5.0	?
2	2.5	4.0	5.0
3	5.0	?	3.0

（1）计算物品间评分的平均差值

计算公式为：

$$dev_{j,i} = \sum_{(u_j,u_i \in S_{j,i}(R))} \frac{u_j - u_i}{|S_{j,i}(R)|} \quad (2\text{-}18)$$

其中，R 代表用户所有的评分数据，u 代表用户对物品的评分，j 和 i 代表物品，$S_{j,i}(R)$代表同时包含用户对物品 j 和 i 的评分集合。

从表 2-4 得知，同时对物品 1 和 3 评分的用户有两个，二者之间评分的平均差值为((5.0-2.5)+(3.0-5.0))/2 = 0.25。同理，同时对物品 2 和 3 评分的用户只有一个，二者之间评分的平均差值为(5.0-4.0) = 1.0。

（2）预测评分

根据用户对物品的评分和物品间评分的平均差值，计算得到用户对未评价物品的评分。计算公式如下，其中S(u)表示用户对物品有评分的元素集合。

$$\mathrm{pred(u,j)} = \frac{\sum_{i\in \mathrm{S(u)}-\{\mathrm{j}\}}(dev_{j,i}+u_i)*|S_{j,i}(R)|}{\sum_{i\in \mathrm{S(u)}-\{\mathrm{j}\}}|S_{j,i}(R)|} \tag{2-19}$$

用户 1 对物品 1 和物品 2 的评分分别为 1.0 和 5.0，则根据公式，计算得到用户 1 对物品 3 的评分为如下。

$$\frac{2\times(1.0+0.25)+1\times(5.0+1.0)}{2+1} = 2.83$$

（3）进行推荐

将计算得到的预测评分进行排序过滤，根据需求将物品推荐给用户。本例中，用户 1 对物品 3 的预测评分为 2.83，可以推荐。

2.4.4 ALS-WR 推荐

本小节简要介绍 ALS-WR 推荐算法的原理，ALS-WR 是对 ALS 算法的改进和优化。ALS 算法常用于基于矩阵分解的推荐系统中，它的主要原理就是把用户物品评分矩阵分解为两个矩阵，在矩阵的分解过程中，补全了用户未评分物品的评分，推荐系统即可根据这个评分给用户推荐。

2.5 实现技术

本节将简要介绍构建教育资源推荐系统中使用到的相关技术，探讨它们的原理，为后面实现系统打下坚实的基础。主要包括 Hadoop、Mahout、Kafka、Flume、Spring MVC 等。

2.5.1 Hadoop

Hadoop 是一个分布式计算框架，它基于 Java 语言，是由 Apache 基金会负责开发和维护的。它具有高可靠性、高扩展性、高容错性、低成本等特点。

Hadoop 项目开始于 2005 年，是开源网页搜索引擎 Nutch 项目下的一个子项目。Hadoop 的诞生要归功于 Google 发表的两篇论文，它们分别介绍了 GFS[37]和 Google MapReduce 计算模型[38]。Hadoop 的创始人 Doug Cutting 根据这两篇论文，开发出了 Hadoop。Hadoop 中的 HDFS 和 MapReduce 分别是 GFS 和 Google

MapReduce 的开源实现。下面对它进行简单介绍[39]。

（1）HDFS，即 Hadoop 分布式文件系统。它使用数据块存储文件，每个文件都会被分割成多个数据块存储，它默认大小为 64MB。使用数据块存储文件有着很多好处，比如，任何大小的文件都可以被分块存储到 HDFS 中，块存储适合于数据备份并且能够提高文件系统的容错能力。

HDFS 中存在两类主要节点，分别为一个 namenode 和多个 datanode，它们采用主从模式架构，即 Master/Slaver 模式，namenode 在充当 Master，datanode 则充当 Slaver。datanode 在 HDFS 中是用来存储数据块的，并且提供了冗余备份的功能。namenode 有两个功能，一个是提供一个统一的文件系统命名空间，用户可以像访问普通文件系统一样来访问 HDFS，所有文件和目录的元数据都存储在 namenode 上（包括文件或目录的层级关系、权限信息、各个数据块的信息等）。另一个功能是管理所有的文件操作，包括对文件的新建、读取、移动等操作。客户端在操作文件的时候，首先需要访问 namenode，获取文件的元数据，然后根据元数据直接操作 datanode，不需要经过 namenode，这样就避免了 HDFS 因为 namenode 性能有限而产生系统瓶颈。HDFS 的架构如图 2-5 所示。

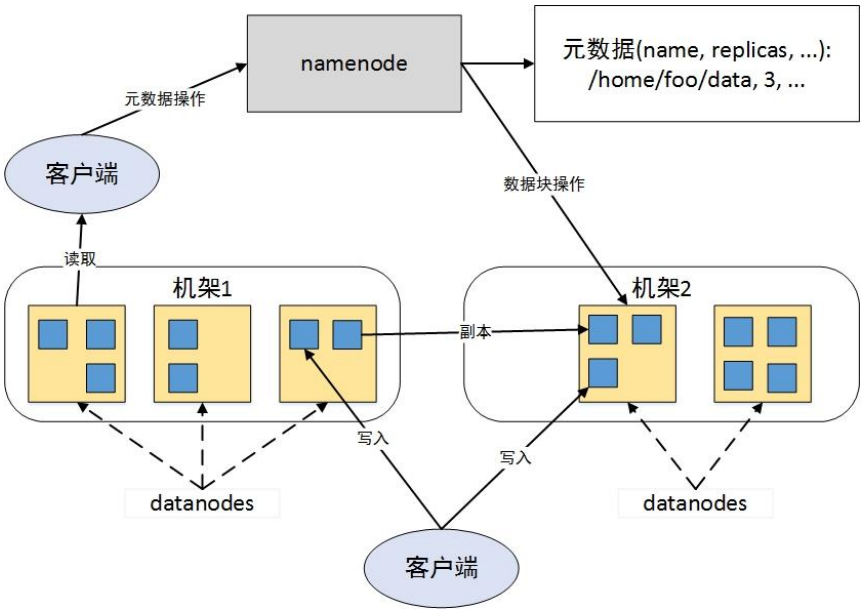


图 2-5 HDFS 架构

（2）MapReduce，是一个计算模型，用于在 Hadoop 集群上进行分布式计算。它分为 map 和 reduce 两个重要阶段，中间还穿插着 shuffle 过程，下面对它们的计算过程做一个简要的介绍。图 2-6 列出了一个典型 MapReduce 计算过程[40]。

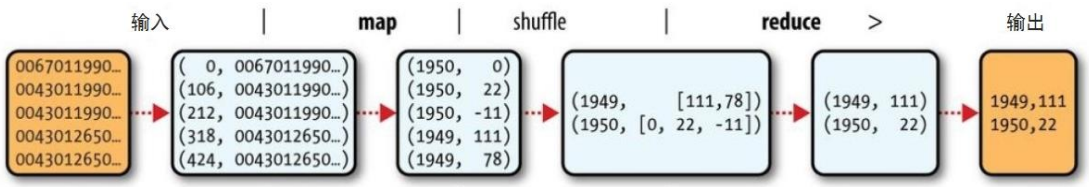


图 2-6 一个典型的 MapReduce 过程

map，以键值对的形式读取原始数据，并对数据进行转换和处理，然后再以键值对的形式输出。Hadoop 集群中的任何节点都可以运行 map 任务，多个 map 任务可以并行运行。

shuffle，在 map 任务即将结束前，即 map 输出前，将 map 任务将要输出的所有键值对按照键的大小进行排序，并输出，属于一个中间过程。

reduce，对 map 传来的数据进行处理，如求和、求均值、筛选等。多个 reduce 任务可以并行运行。

2.5.2 Mahout

Mahout 是一个开源的机器学习库，由 Apache 基金会负责管理。它基于 Java 语言开发，最初是 2008 年作为 Apache Lucene 的子项目出现的。2010 年，Mahout 从 Lucene 项目中脱离，成为了 Apache 的一个独立的顶级项目。Mahout 主要实现了机器学习领域的推荐、聚类和分类等算法，其中推荐算法使用的是协同过滤技术。Mahout 的开发者以高效和可扩展的方式实现了这些算法，并且还对其中的一些算法进行了分布式实现，以能够处理更大规模的数据。

2.5.3 Kafka

Kafka 是一个分布式的消息系统。最初由 LinkedIn 公司开发，基于 Scala 语言，并于 2011 年开源，后加入 Apache 基金会并由其负责维护。它有着高扩展、高吞吐、低延迟等特点。

Kafka 作为一个消息系统，使用了发布/订阅的模式实现消息队列，使消息的生产者（Producer）和消费者（Consumer）在交换消息的时候并不需要知道对方是谁、处于什么位置，达到解耦合的目的。

Kafka 集群可以由一台或多台服务器组成，每一台服务器被称为 broker，broker 没有主从关系，可以随时添加或删除 broker，实现了 Kafka 集群的高扩展

性。所有发布到 Kafka 集群的消息都需要有一个主题（Topic），消费者可以根据主题订阅消息。在 Kafka 集群中，主题支持分区操作，为了负载均衡，一个主题可能被分成多个分区，存储到不同的 broker 上。一个或多个消费者可以组成一个消费者组（Consumer Group），每个消息只能被一个组中的一个消费者消费，不同组的多个消费者可以消费同一个消息。图 2-7 描述了 Kafka 的工作过程。

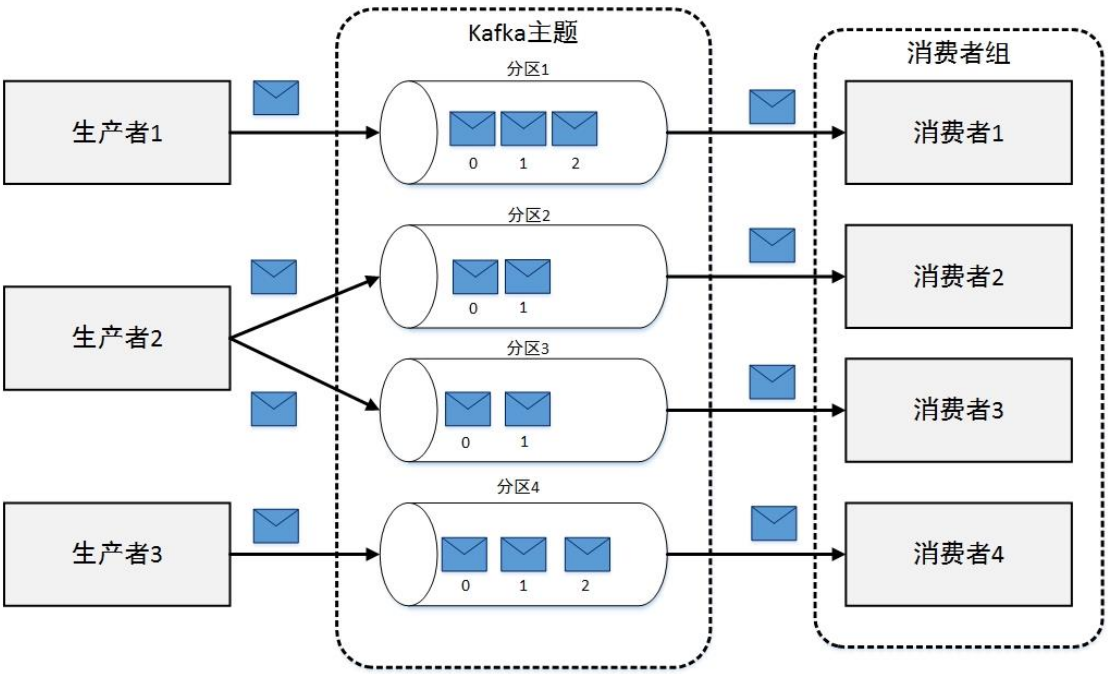


图 2-7 Kafka 工作过程

2.5.4 Flume

Flume 常用于日志收集，它采用分布式架构，能够实现对海量日志的收集、聚集和传送。具有高可靠性、高稳定性和高可用性等特点。它最早是由 Cloudera 公司开发，用于收集日志并存储于 HDFS 中。2011 年 10 月，Cloudera 对 Flume 进行了大规模重构，并把其纳入了 Apache 基金会。

Flume 可以收集各种类型的数据，并且进行简单的过滤和处理，然后写入各种类型的数据接收方中。一个典型的 Flume 日志收集节点是由三大部分组成的，分别为 Source、Channel 和 Sink，它们是 Flume 的核心组件。Source 用于收集日志，Channel 连接 Source 和 Sink，提供传输通道，Sink 则用于把日志写入不同的数据接收方（如 HDFS）。三者的关系如图 2-8 所示。

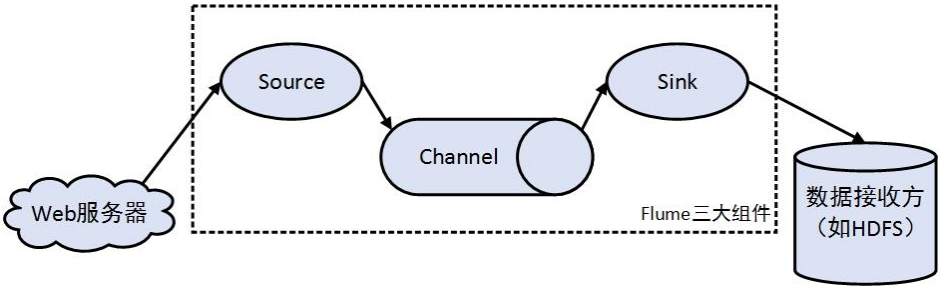


图 2-8 Flume 三大组件

2.5.5 Spring MVC

Spring MVC 是 Spring 框架提供的一个全功能 MVC 模块，它基于 Java 开发，是目前最流行、最优秀的 MVC 框架之一。它有很多优点，比如，它可以实现与 Spring 的完美融合，不会出现框架不兼容问题；实现了通过注解进行系统配置，在开发过程中可以大大减少开发难度；完美支持 Restful 风格接口等。

Spring MVC 底层基于 Servlet 实现了一个 DispatcherServlet，即前端控制器，前端所有的请求都经由前端控制器处理。Spring MVC 处理请求的具体流程如图 2-9 所示。

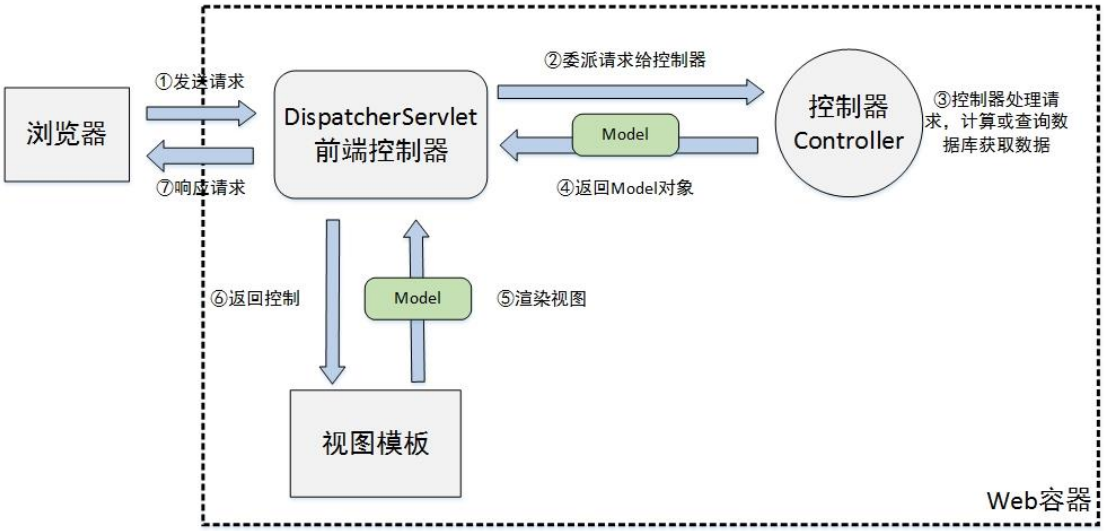


图 2-9 Spring MVC 处理请求流程

2.6 本章小结

本章首先对推荐系统领域中三类重要算法进行了简要介绍，然后对推荐系统相似度度量和推荐系统评测指标做了介绍。接下来详细介绍了协同过滤推荐的四种常用算法。最后介绍了推荐系统在实现过程中使用到的关键工具和技术。

第 3 章 推荐系统需求分析

本章将对推荐系统进行需求分析，主要从业务需求、功能需求和非功能需求三个方面进行讨论。其中，对功能需求的分析过程中需要使用用例分析和领域建模等软件工程方法。

3.1 业务需求分析

本文实现的教育资源推荐系统是云教育平台的一个子系统，它给云教育平台提供资源的推荐服务。关于业务，云教育平台提出了如下要求：

（1）云教育平台是一个大型的在线教育网站，目前已上线运行，它不允许在开发推荐系统时修改原系统代码，这样有可能导致原系统代码被入侵，甚至导致原系统崩溃。所以，推荐系统应该被设计成一个独立运行的子系统，通过接口的形式为云教育平台提供推荐服务。

（2）为了能够获取更加精确地推荐结果，云教育平台要求推荐系统能够全面详尽地分析平台上用户的行为。因此，推荐系统需要从云教育平台收集用户行为日志，包括用户的点击行为、浏览行为、评分行为和收藏行为等。

（3）为了应对不同类型用户的需求，云教育平台要求推荐系统可以提供针对网站不同场景的推荐服务。按照要求，经过归纳总结，推荐系统需要提供根据用户兴趣推荐、根据教育资源相关性推荐、根据用户信息推荐和根据教育资源热门度推荐等多种推荐方式。其中前三者属于个性化推荐，最后一个属于非个性化推荐。

（4）在对用户进行推荐过程中，推荐系统需要提供对特别用户和特殊教育资源进行过滤和特别推荐的功能。

（5）推荐系统在提供资源推荐服务时，需要能够应对云教育平台大量的高并发请求。所以本系统需要实现负载均衡的功能，能够做到给云教育平台提供低延迟、高效率、高可靠的资源推荐服务。

3.2 功能需求分析

功能需求要求明确软件系统应该提供的功能和服务[41]。本节首先将通过用例分析和领域建模来对其进行讨论。

3.2.1 功能概述

根据本文上一节中列出的云教育平台对推荐系统的业务需求，可以归纳出推荐系统的三个主要功能，分别为日志收集、推荐计算和推荐接口服务。图 3-1 列出了这三个功能与云教育平台的关系。

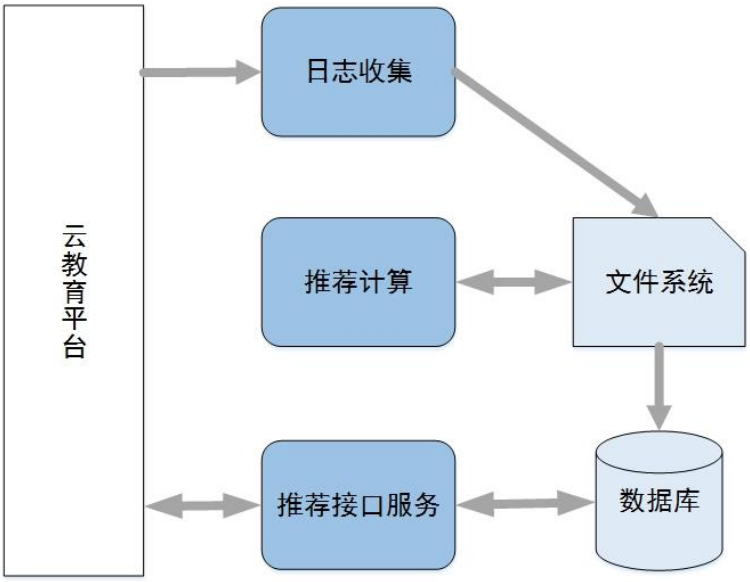


图 3-1 教育资源推荐系统主要功能

从图 3-1 可以了解到，本推荐系统首先需要从云教育平台收集到用户行为日志，并存储到文件系统中。推荐计算功能读取用户行为日志，使用推荐算法进行建模、计算，得到推荐结果，并把推荐结果存储到文件系统中。最后，系统把文件系统中的推荐结果转存到数据库中，然后给云教育平台设计推荐服务接口，供其获取推荐结果。

3.2.2 用例分析

基于上一小节中对教育资源推荐系统功能的概述，得到系统的三个主要功能，本小节将对这三个功能进行用例分析。

（1）日志收集

用户行为日志是教育资源推荐系统在计算过程中所必需的数据。为避免推荐系统对云教育平台代码做出大量修改，本推荐系统在收集用户行为日志时，应采用在云教育平台前端埋点的方式进行收集，这样基本上不会对原系统代码造成入侵。用户在前端的所有行为都需要被收集，比如，点击、浏览、评分等行为。用例图如图 3-2 所示。

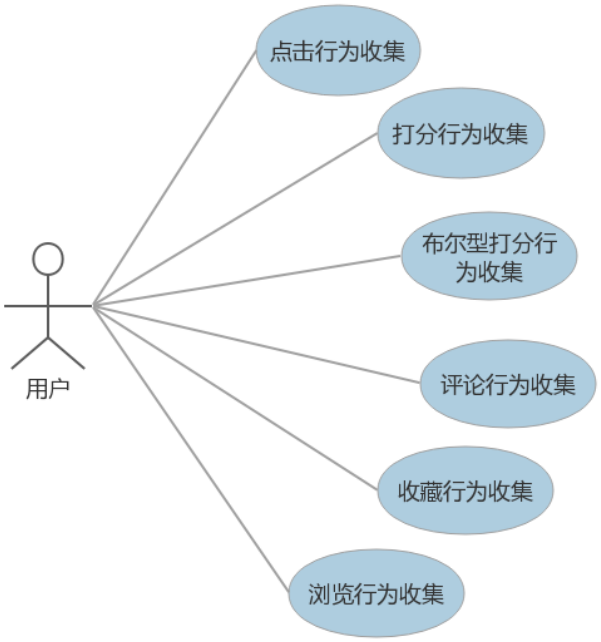


图 3-2 日志收集功能用例图

从图 3-2 可以得到，本功能总共有一个参与者和 6 个用例。参与者为云教育平台的用户。表 3-1 对本功能的 6 个用例进行了详细描述。

表 3-1 日志收集功能用例描述

用例	用例描述
点击行为收集	收集用户在前端对教育资源的点击行为
打分行为收集	收集用户对教育资源的打分行为
布尔型打分行为收集	区别于打分行为，这里收集的是用户对教育资源的布尔型打分，如喜欢或不喜欢等
评论行为收集	收集用户对教育资源的评论行为
收藏行为收集	收集用户在前端对教育资源的收藏行为
浏览行为收集	收集用户在前端对教育资源的浏览行为

（2）推荐计算

这是本推荐系统的核心功能，利用所收集到的用户行为日志，构建用户评分向量，通过相应的推荐算法给每一个用户计算出个性化的推荐结果。当然，在计算前，需要对用户行为日志进行预处理，使得日志符合本功能所要求的格式。本推荐系统离线计算使用四种推荐方法，分别是基于用户的推荐、基于物品的推荐、Slope One 推荐和 ALS-WR 推荐，在线计算使用两种计算方法。用例图如图 3-3 所示。

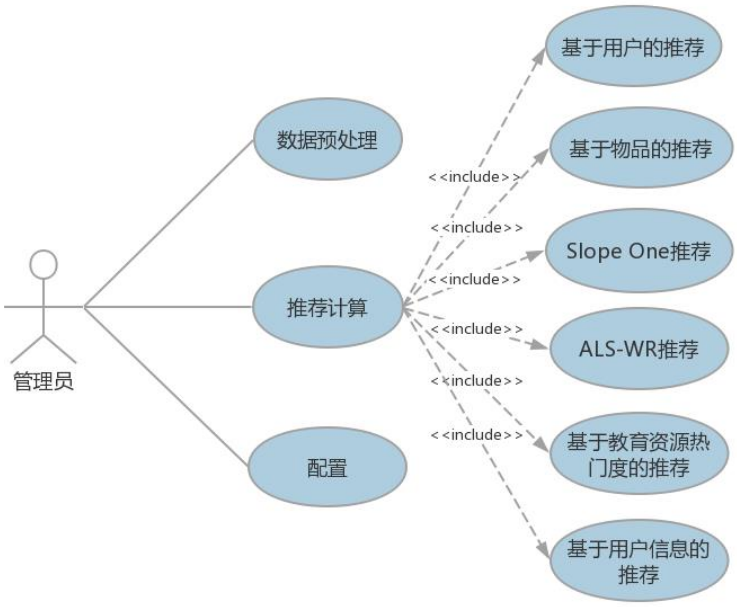


图 3-3 推荐计算功能用例图

从图 3-3 可以得到，本功能总共有一个参与者和 9 个用例。参与者为推荐系统管理员，可以对推荐系统进行配置、对数据预处理和进行推荐计算。表 3-2 对本功能的 9 个用例进行了详细描述。

表 3-2 推荐计算功能用例描述

用例	用例描述
数据预处理	对收集到的用户行为日志进行预处理，得到规范格式的日志文件，供后面计算使用
推荐计算	通过对用户行为日志进行建模，计算，获得针对每个用户的个性化推荐结果
配置	系统管理员对系统进行相关配置
基于用户的推荐	通过基于用户的推荐算法计算获得推荐结果
基于物品的推荐	通过基于物品的推荐算法计算获得推荐结果
Slope One 推荐	通过 Slope One 推荐算法计算获得推荐结果
ALS-WR 推荐	通过 ALS-WR 推荐算法计算获得推荐结果
基于教育资源热门度的推荐	通过计算教育资源热门度获得推荐结果
基于用户信息的推荐	通过用户信息获得推荐结果

（3）推荐接口服务

上述两个功能收集了用户行为日志，并通过计算生成了推荐结果。本功能设计相关接口，提供推荐接口服务，云教育平台可以通过调用接口获得推荐结果。

由于上面计算生成的推荐结果存储在文件系统中，而文件系统的查询和检索速度比较慢，所以本系统需要把文件系统中的推荐结果转存到数据库中。然后搭建一个 B/S 服务器，提供接口供云教育平台使用。本推荐系统提供了四个推荐场景，分别是，根据用户兴趣推荐、根据教育资源相关性推荐、根据用户信息推荐和根据教育资源热门度推荐，除此之外云教育平台还可以获得由不同推荐算法组成的混合推荐结果。用例图如图 3-4 所示。

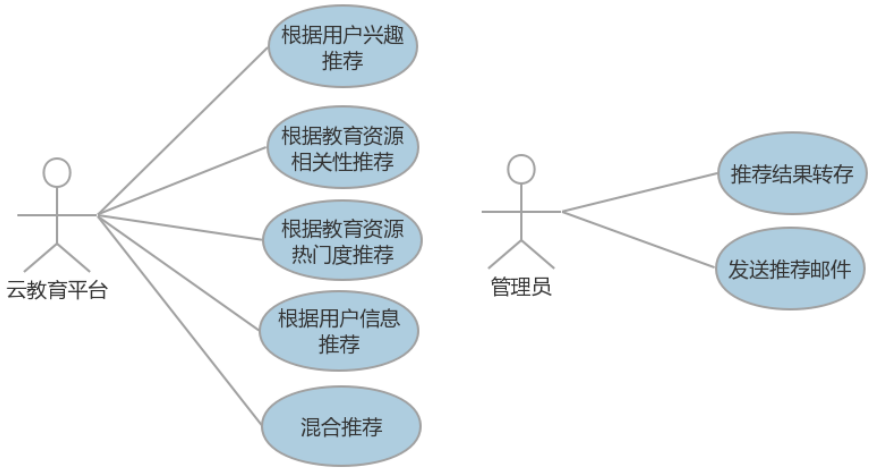


图 3-4 推荐接口服务功能用例图

从图 3-4 可以得到，本功能总共有 2 个参与者和 7 个用例。2 个参与者分别为系统管理员和云教育平台。表 3-3 对本功能的 7 个用例进行了详细描述。

表 3-3 推荐接口服务功能用例描述

用例	用例描述
根据用户兴趣推荐	推荐给与被推荐用户相似用户所喜好的教育资源，属于个性化推荐
根据教育资源相关性推荐	推荐给用户与他原本喜欢资源类似的教育资源，属于个性化推荐
根据教育资源热门度推荐	根据教育资源的热门排行榜进行推荐，属于非个性化推荐
根据用户信息推荐	根据用户的个人注册信息进行推荐，属于个性化推荐
混合推荐	混合不同推荐方法的推荐结果，进行混合推荐
推荐结果转存	把文件系统中的推荐结果转存到数据库中
发送推荐邮件	给用户发送推荐邮件

3.2.3 领域模型

上文中对推荐系统的业务需求和功能需求进行了分析，本小节对推荐系统进行领域建模，领域模型是一种常用的需求分析工具。如图 3-5 所示，推荐结果是整个系统的核心，整个业务都是围绕推荐结果展开的。

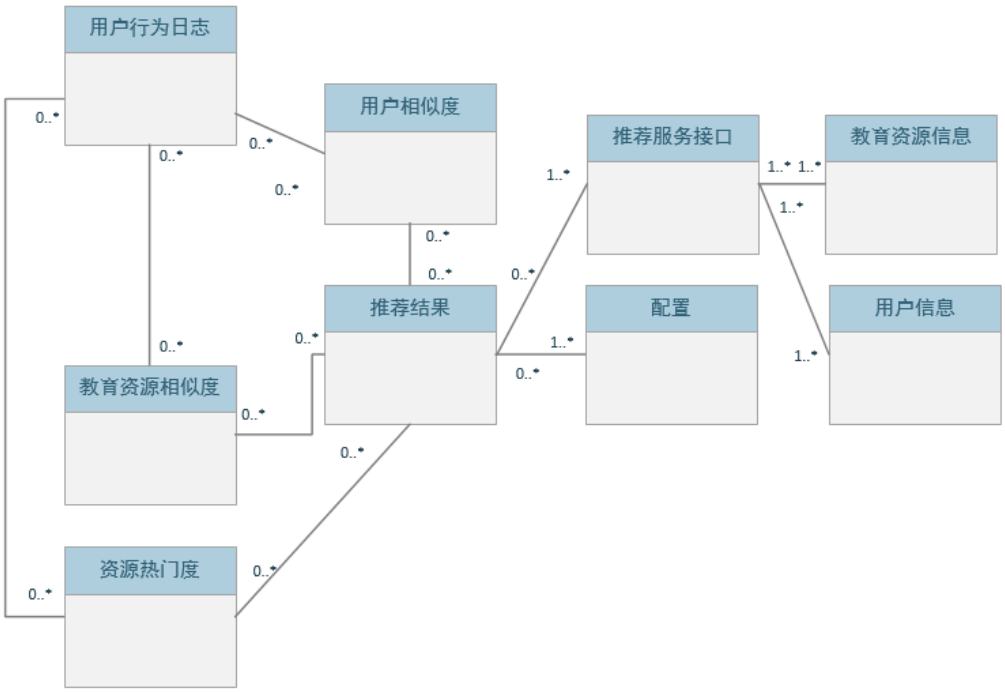


图 3-5 教育资源推荐系统领域模型

3.3 非功能需求分析

上文中对推荐系统的功能需求进行了讨论，并通过用例分析和领域模型等方法对系统的三个主要功能进行了详细分析。本小节将讨论推荐系统的非功能需求。非功能需求是对功能需求的补充，包括对系统的一些限制和要求等。经过分析，教育资源推荐系统需要满足如下的非功能需求，如表 3-4 所示。

表 3-4 推荐系统非功能需求

非功能需求	描述
实时性	系统推荐服务所提供的接口需要做到低延迟、快速响应
高并发	云教育平台在高峰期会产生大量的用户访问，本推荐系统中无论是日志收集功能还是推荐接口服务功能都需要能够应对大量的高并发用户请求
可靠性	系统需要有较强的可靠性，保持较低的故障率
数据安全性	系统在发生故障后，能够快速恢复，保证数据不丢失

3.4 本章小结

本章首先对推荐系统进行了业务需求分析，然后根据客户对业务的需求，归纳总结出了系统的三大功能，分别为日志收集、推荐计算和推荐接口服务。通过用例图和领域模型对这三大功能进行了详细讨论和分析。最后本章对系统的非功能需求进行了讨论。

第 4 章 推荐系统概要设计

本章对推荐系统进行概要设计，为详细设计做准备，会从总体设计、功能模块划分、技术架构、数据库设计和日志埋点规范设计五个方面展开讨论。

4.1 总体设计

本小节对推荐系统进行了总体设计，如图 4-1 所示。从图中可以看出，推荐系统的日志收集功能可以从云教育平台前端收集日志并通过数据服务存储到 HDFS。推荐引擎读取日志并计算获取推荐结果。推荐服务器把 HDFS 中的推荐结果转存到数据库，并设计封装服务接口，提供推荐服务。云教育平台前端 UI 可以通过调用这些接口获取推荐结果。

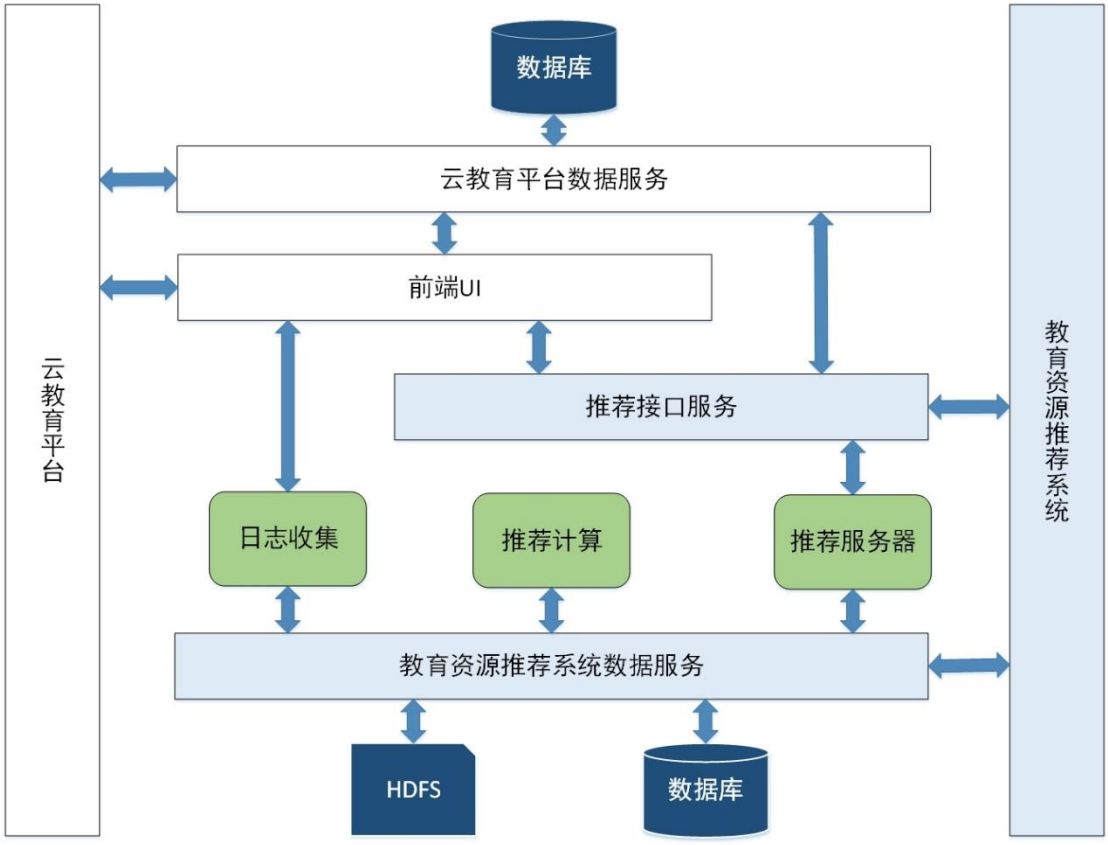


图 4-1 教育资源推荐系统总体设计

4.2 功能模块划分

根据本文对教育资源推荐系统需求分析和对系统的总体设计，结合相关技术

条件，本推荐系统可以被划分为三大功能模块，如图 4-2 所示。

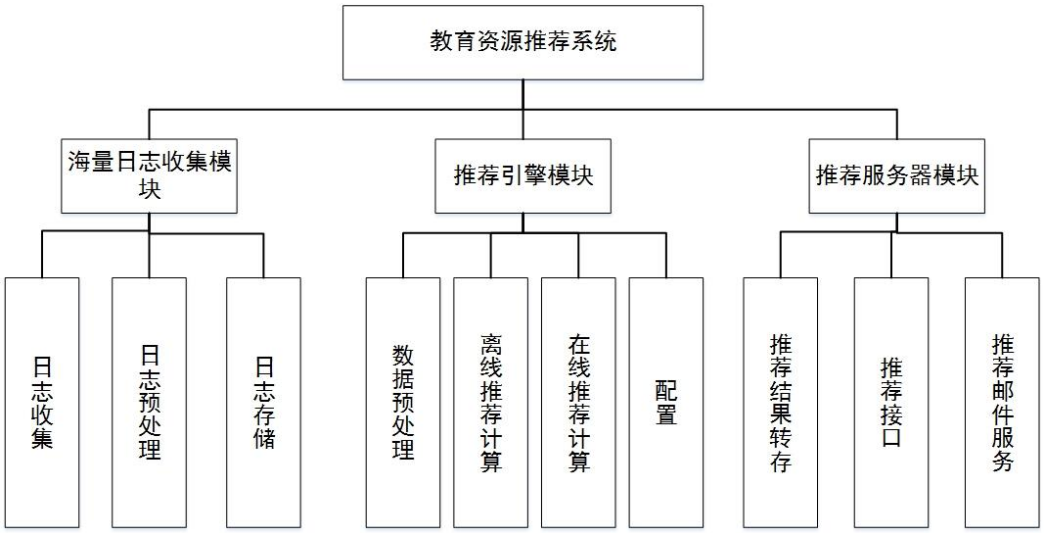


图 4-2 教育资源推荐系统功能模块

4.2.1 海量日志收集模块

海量日志收集模块是整个教育资源推荐系统的基础。它收集云教育平台用户的行为日志，存储到 HDFS 中，供推荐引擎模块使用。

本文在第 3 章需求分析中已经说明，推荐系统在设计实现过程中是不能修改云教育平台系统源代码的，不能对原系统造成入侵。所以海量日志收集模块采用前端埋点的方式收集用户行为日志。

海量日志收集模块在设计过程中，需要实现一个基于 B/S 模式的日志服务器，用于对前端收集来的数据进行初步过滤、清洗和预处理，然后存储到 HDFS 中。

用户在前端的所有行为都需要被收集，比如打分、点击、浏览、评论、收藏等行为。

由于云教育平台在高峰期会产生大量的用户行为，而所有的用户行为都会通过前端埋点的方式发送到日志服务器，会产生大规模、高并发的请求，所以日志服务器需要加入负载均衡的功能，避免出现延迟响应、服务崩溃等问题。

4.2.2 推荐引擎模块

推荐引擎模块是教育资源推荐系统中最重要的模块。它是由一系列的推荐算法组成。它将海量日志收集模块收集到的用户行为日志作为输入，经过推荐引擎算法的计算，生成推荐结果。

推荐引擎的计算过程可分为离线计算和在线计算，由于大部分推荐算法都是基于海量数据的，所以本推荐引擎主要是基于离线计算。另外，推荐系统还提供对热门教育资源推荐和基于用户注册信息推荐的功能，它们是基于在线计算完成的。离线计算生成的推荐结果需要存储到分布式文件系统中，在线计算的推荐结果可以直接存储到数据库中。由于离线计算需要耗费大量的时间，占用大量系统资源，所以本推荐系统需要在云教育平台的访问低峰时间进行离线计算。

由于推荐系统的诞生时间比较短，社会各界对其的研究热度一直很高，未来可能会出现更优化、更准确、效率更高的推荐算法，所以本推荐引擎模块采用可扩展的方式设计，如果有新算法产生，推荐引擎中可以随时加入该算法。

由于不同的推荐算法对于不同的推荐场景、输入数据和配置参数会产生不同的推荐效果，所以本模块还提供了配置功能。系统管理员可以根据用户体验和反馈对推荐引擎进行配置，以获得最优的推荐体验。

推荐引擎模块还需要实现对收集到的用户行为日志进一步清洗、过滤和预处理。以获得符合推荐算法计算的数据格式。

以上对推荐引擎功能模块做了详细介绍，图 4-3 列出了本模块的架构。

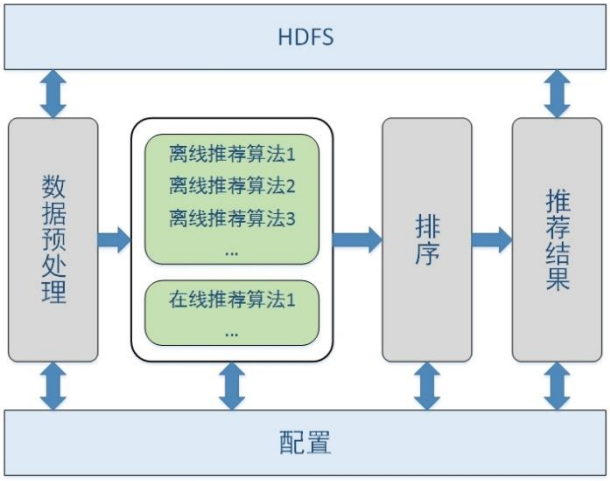


图 4-3 推荐引擎模块架构

4.2.3 推荐服务器模块

推荐服务器模块封装了一系列接口，为云教育平台提供推荐服务。

用户行为日志在经过推荐引擎模块的计算后，得到相关的推荐结果并存储在 HDFS 中，由于 HDFS 中的文件不是结构化的，所以推荐服务器第一步要做的就是从 HDFS 中读取推荐结果，并存储到数据库中。

推荐服务器为云教育平台提供四种推荐场景，分别为根据用户兴趣推荐、根据教育资源相关性推荐、根据用户信息推荐和根据教育资源热门度推荐。其中，根据用户兴趣推荐用到了基于用户的推荐算法的计算结果。根据教育资源相关性推荐用到了基于物品的推荐或 Slope One 推荐或 ALS-WR 推荐的计算结果，即给用户推荐与他看过的教育资源类似的资源，实际应用中选择哪种算法是由系统管理员决定的。前两者为个性化推荐，根据教育资源热门度推荐是非个性化推荐，根据用户信息推荐是专门适用于教育资源的推荐方式，因为系统可以根据用户注册时所提供的年龄和地区信息，为用户提供特定的教育资源，比如山东地区 16 周岁的用户，系统就可以给他们推荐适合使用鲁教版教材的高一学生的教育资源。

推荐服务器还需要提供混合推荐服务，这样可以结合各种推荐算法的优缺点，获得更加优化的推荐结果。

在给用户返回推荐结果前，推荐服务器需要对推荐结果进行处理，按照设置对特别用户过滤掉一些物品，也可以对特别用户特别推荐一些物品。

另外，当推荐系统处于冷启动阶段时，系统还没有获取到大量的用户行为日志，无法进行推荐计算。推荐服务器可以使用根据用户信息推荐和根据教育资源热门度推荐这两种方式解决推荐系统冷启动问题。

同海量日志收集模块一样，推荐服务器也需要应对云教育平台高峰期发来的大规模、高并发的请求，需要加入负载均衡和数据缓存功能。

为了让用户更方便地了解和获取教育资源，推荐服务器会提供为用户发送推荐邮件的功能。此功能需要管理员配合完成。

4.3 技术架构

本章在 4.2 节中对教育资源推荐系统功能进行了划分，分成了三个主要的模块，并对其进行了详细介绍。本小节将会对这三个模块进行技术架构，下面对其进行详细介绍，首先表 4-1 列出了各模块用到的技术。

表 4-1 推荐系统功能模块主要技术

模块	技术
海量日志收集模块	Ajax、Spring MVC、Nginx、Kafka、Flume、Tomcat
推荐引擎	Hadoop、Mahout
推荐服务器	Spring MVC、MySQL、Nginx、Sqoop、Redis、Tomcat

（1）海量日志收集模块。

海量日志收集模块需要使用 Ajax、Spring MVC、Nginx、Kafka、Flume 和 Tomcat 等技术。Ajax 用于在云教育平台前端埋点。Spring MVC 是目前最优秀的 MVC 框架，本模块使用它搭建日志服务器，用于处理 Ajax 从前端发送的用户行为日志。Nginx 是一个高性能的 HTTP 服务器，在本模块中使用 Nginx 实现对日志服务器的负载均衡。Kafka 是一个优秀的分布式消息系统，日志服务器接收到前端发送的日志后，并不会直接把日志写入 HDFS，而是写入 Kafka 集群，Kafka 集群具有高吞吐量，可以应对大规模高并发请求，这是实现本模块功能最重要的技术。Flume 从 Kafka 集群中获取日志，当积累到一定数量的时候再把日志写入到 HDFS 中，这样可避免对 HDFS 的重复写入，造成系统瓶颈。Tomcat 是提供一个 Web 容器，供 Spring MVC 运行。图 4-4 显示了本模块的技术架构。

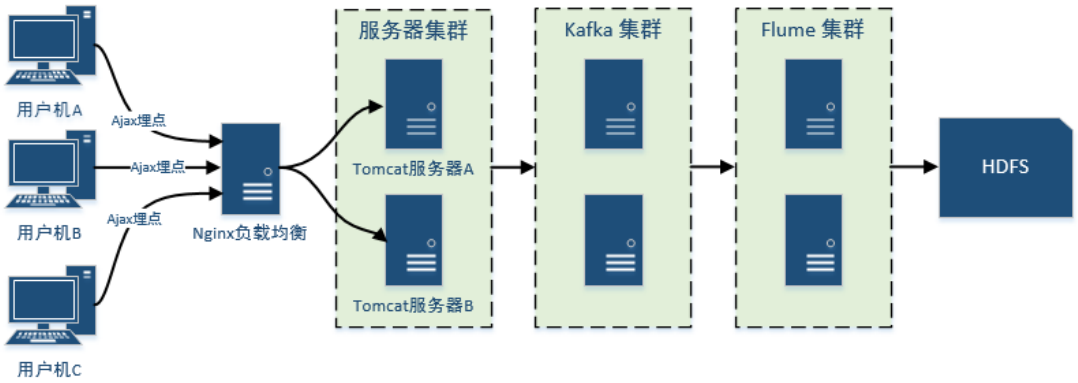


图 4-4 海量日志收集模块技术架构

（2）推荐引擎模块

推荐引擎模块需要使用 Hadoop 和 Mahout 等技术。其中 Hadoop 提供了 HDFS 和 MapReduce，HDFS 用于存储用户行为日志和离线计算得到的推荐结果。本推荐系统使用了 4 个推荐算法，其中基于用户的推荐和 Slope One 推荐需要本文基于 MapReduce 技术自行实现。基于物品的推荐和 ALS-WR 推荐使用的是 Mahout 框架的实现。四个算法需要运行在 Hadoop 集群上进行并行计算。

（3）推荐服务器模块

推荐服务器模块需要使用 Spring MVC、MySQL、Nginx、Sqoop、Redis 和 Tomcat 等技术。MySQL 是一款优秀的开源数据库，在本模块中它用于存储推荐结果。Sqoop 技术用于把存储在 HDFS 中的推荐结果转存到 MySQL 数据库中。同日志收

集模块一样，本模块也需要搭建一个 Web 服务器，即推荐服务器。也需要使用 Nginx 技术实现负载均衡。另外，本推荐服务器模块还用到了 Redis 数据缓存，Redis 可以把云教育平台经常请求的数据存储到内存中，以减少服务器对数据库的访问，提高服务器性能。图 4-5 显示了本模块的技术架构。

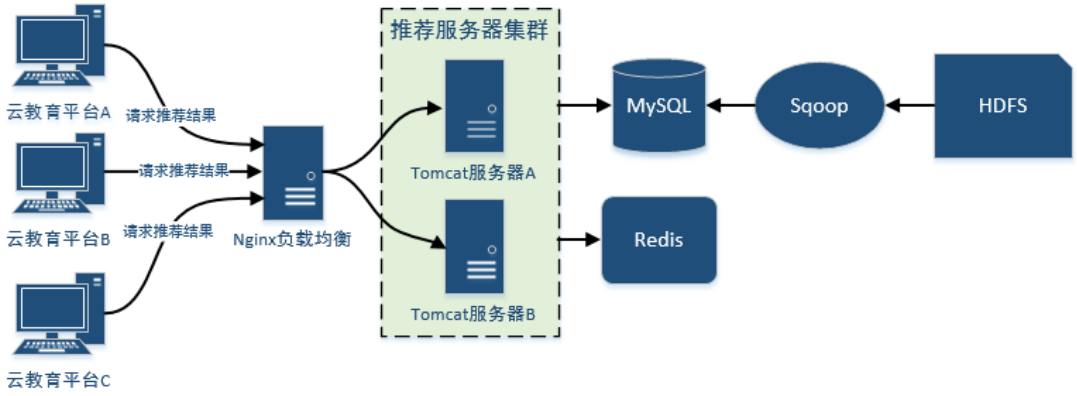


图 4-5 推荐服务器模块技术架构

4.4 数据库设计

根据第 3 章中对教育资源推荐系统的需求分析，本小节对系统的数据库进行了设计，主要包括如下数据表，如图 4-6 所示。其中大部分表都需要依赖 user_info 和 file_info。

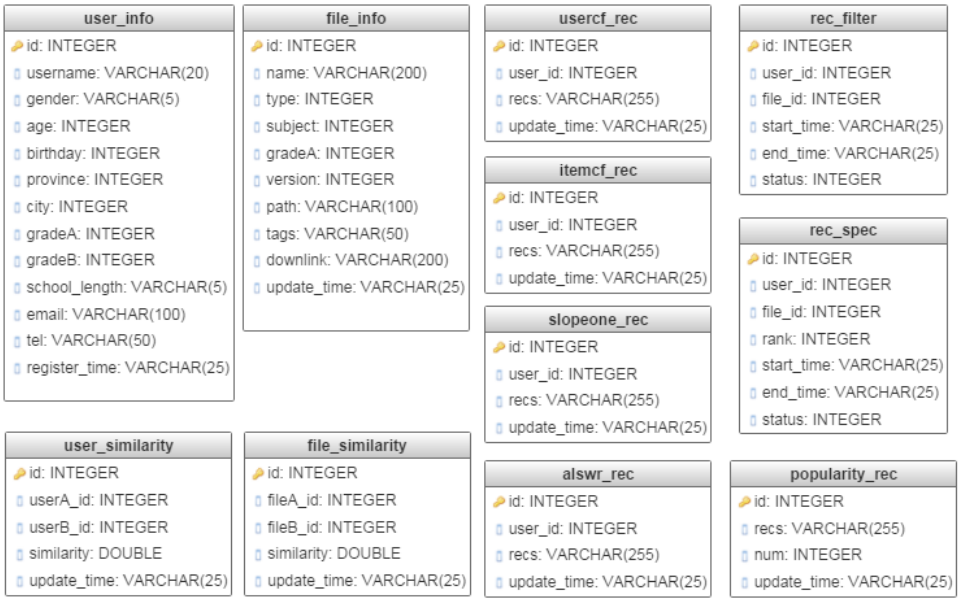


图 4-6 推荐系统数据库表

根据系统功能，数据库表可以分为四大类，第一类用于存储用户或教育资源的属性信息，包括 user_info 和 file_info，其他表格都需要参考这两个表格的信息；第二类用于存储用户之间或教育资源之间的相似度，包括 user_similarity 和 file_similarity；第三类用于存储各种推荐算法得到的推荐结果，包括 usercf_rec、itemcf_rec、slopeone_rec、alswr_rec 和 popularity_rec；第四类用于存储对原始推荐结果的处理数据，包括 rec_spec、rec_filter。下面对数据库的表结构进行详细介绍。

（1）用户信息表 user_info

用户信息表记录了用户的详细信息。是推荐系统中基于用户信息推荐的重要数据来源。比如，推荐系统可以根据用户所在的省份和年龄，判断用户所处的教育阶段，然后推荐给用户适合他的教育资源。由于用户的年龄和所处的教育阶段会随着时间的变化而变化，所以系统会根据用户的注册时间和出生日期定期更新这部分信息。表结构如表 4-2 所示。

表 4-2 用户信息表 user_info

属性名	数据类型	长度	约束	描述
id	INTEGER	20	主键，非空	用户 id
username	VARCHAR	50	非空	用户名
gender	VARCHAR	5	非空	用户性别
age	INTEGER	3	非空	用户年龄
birthday	VARCHAR	25	非空	用户出生日期
province	INTEGER	2		用户所在省份
city	INTEGER	2		用户所在城市
gradeA	INTEGER	2		用户教育阶段，如小学、初中、高中
gradeB	INTEGER	2		用户所在年级
school_length	VARCHAR	5		用户所处地区九年义务教育的学制，如 63 制、54 制
email	VARCHAR	100	非空	用户邮箱
tel	VARCHAR	50	非空	用户电话
register_time	VARCHAR	25	非空	用户注册时间

（2）教育资源信息表 file_info

教育资源信息表记录了教育资源的详细信息。它与用户信息表 user_info 共同为推荐系统中基于用户信息推荐提供数据。表结构如表 4-3 所示。

表 4-3 教育资源信息表 file_info

属性名	数据类型	长度	约束	描述
id	INTEGER	20	主键，非空	教育资源 id
name	VARCHAR	200	非空	资源名称
type	INTEGER	2	非空	资源类型，如文档、视频、图片
subject	INTEGER	2	非空	资源所属科目，如语文、数学
gradeA	INTEGER	2		资源适配教学阶段，如小学、初中、高中
gradeB	INTEGER	2		资源适配的年级
version	INTEGER	2		资源版本，如人教版、鲁教版
path	VARCHAR	100	非空	资源在服务器的存储位置
tags	VARCHAR	50		资源的标签
downlink	VARCHAR	200	非空	资源的下载链接
update_time	VARCHAR	25	非空	资源更新时间

（3）用户相似度表 user_similarity

用户相似度表记录了用户之间的相似度，需要定时更新。表结构如表 4-4 所示。

表 4-4 用户相似度表 user_similarity

属性名	数据类型	长度	约束	描述
id	INTEGER	20	主键，非空	相似度 id
userA_id	INTEGER	20	非空	用户 A 的 id
userB_id	INTEGER	20	非空	用户 B 的 id
similarity	DOUBLE	15	非空	用户相似度
update_time	VARCHAR	25	非空	相似度更新时间

（4）教育资源相似度表 file_similarity

教育资源相似度表记录了教育资源之间的相似度，需要定时更新。表结构如表 4-5 所示。

表 4-5 教育资源相似度表 file_similarity

属性名	数据类型	长度	约束	描述
id	INTEGER	20	主键，非空	相似度 id
fileA_id	INTEGER	20	非空	教育资源 A 的 id
fileB_id	INTEGER	20	非空	教育资源 B 的 id
similarity	DOUBLE	15	非空	教育资源相似度
update_time	VARCHAR	25	非空	相似度更新时间

（5）推荐过滤表 rec_filter

推荐过滤表记录了在推荐过程中需要给用户过滤掉的物品。其中，当 user_id 为 0 时为全局过滤，即在所有给用户的推荐结果中过滤掉所列出的物品。表结构如表 4-6 所示。

表 4-6 推荐过滤表 rec_filter

属性名	数据类型	长度	约束	描述
id	INTEGER	20	主键，非空	推荐过滤 id
user_id	INTEGER	20	非空	用户 id
file_id	INTEGER	20	非空	教育资源 id
start_time	VARCHAR	25	非空	过滤开始时间
end_time	VARCHAR	25	非空	过滤结束时间
status	INTEGER	2	非空	过滤有效标志

（6）特别推荐表 rec_spec

特别推荐表记录了需要特别推荐给用户的物品。同样，当 user_id 为 0 时为全局特别推荐，即在所有给用户的推荐结果中插入所列出的物品。表结构如表 4-7 所示。

表 4-7 特别推荐表 rec_spec

属性名	数据类型	长度	约束	描述
id	INTEGER	20	主键，非空	特别推荐
user_id	INTEGER	20	非空	用户 id
file_id	INTEGER	20	非空	教育资源 id
rank	INTEGER	2	非空	建议排名
start_time	VARCHAR	25	非空	特别推荐开始时间
end_time	VARCHAR	25	非空	特别推荐结束时间
status	INTEGER	2	非空	特别推荐有效标志

（7）推荐算法结果表

推荐算法结果表用于存储推荐系统中 4 种推荐算法的推荐结果，分别为 usercf_rec、itemcf_rec、slopeone_rec 和 alswr_rec。这 4 个表的结构类似，如表 4-8 所示。

表 4-8 推荐算法结果表

属性名	数据类型	长度	约束	描述
id	INTEGER	20	主键，非空	推荐结果
user_id	INTEGER	20	非空	用户 id
recommendation	VARCHAR	255	非空	推荐结果
update_time	VARCHAR	25	非空	更新时间

（8）教育资源热门度推荐表 popularity_rec

教育资源热门度推荐表用于存储当前最热门的教育资源。需要定时更新。

表 4-9 教育资源热门度推荐表 popularity_rec

属性名	数据类型	长度	约束	描述
id	INTEGER	20	主键，非空	推荐结果 id
recs	VARCHAR	255	非空	推荐结果
num	INTEGER	5	非空	推荐数量
update_time	VARCHAR	25	非空	更新时间

4.5 日志埋点规范设计

本文在介绍海量日志收集模块功能时已经说明，推荐系统采用前端埋点的方式收集日志。前端日志埋点需要借助 Ajax 技术实现。前端在需要埋点的链接处埋点，当用户点击埋点的链接后，Ajax 会发送一个 JSON 数据给日志服务器，日志服务器会做相应的处理。

根据用户在前端可能产生的行为，结合推荐引擎需要的日志类型，本文设计了如下的日志埋点规范：

（1）点击行为

点击行为日志用来记录用户在网站中所有的点击操作，包括点击文档、视频、音频等。此类型日志可以作为布尔型日志供推荐引擎使用。日志具体规范如表 4-10 所示：

表 4-10 点击行为日志埋点规范

参数	类型	描述
behavior_type	String	用户行为类型，此处为点击，取值为“click”
user_id	Long	用户 id
file_id	Long	教育资源 id
behavior_time	String	行为产生时间

（2）打分行为

打分行为日志用来记录用户在网站中对教育资源的打分操作，打分的分值范围为[1.0, 5.0]。日志具体规范如表 4-11 所示：

表 4-11 打分行为日志埋点规范

参数	类型	描述
behavior_type	String	用户行为类型，此处为打分，取值为“grade”
user_id	Long	用户 id
file_id	Long	教育资源 id
value	Float	用户所打分值
behavior_time	String	行为产生时间

（3）布尔型打分行为

布尔型打分行为日志用来记录用户在网站中对教育资源的布尔型打分操作，取值为 0 或 1，当用户喜欢某一资源时可以打分为 1，当用户不喜欢某一资源时可以打分为 0。日志具体规范如表 4-12 所示：

表 4-12 布尔型行为打分日志埋点规范

参数	类型	描述
behavior_type	String	用户行为类型，此处为点击，取值为“bool_grade”
user_id	Long	用户 id
file_id	Long	教育资源 id
value	Integer	用户所打分值，取值为 0 或 1
behavior_time	String	行为产生时间

表 4-13 评论行为日志埋点规范

参数	类型	描述
behavior_type	String	用户行为类型，此处为点击，取值为“comment”
user_id	Long	用户 id
file_id	Long	教育资源 id
grade_value	Float	用户给物品的打分
content	String	评论内容
behavior_time	String	行为产生时间

（4）评论行为

评论行为日志用来记录用户在网站中对教育资源的评论操作。在这里考虑到

一个问题，用户的评论一般是文字内容，而本系统使用的是协同过滤的推荐算法，不适用于文字。所以在这里，系统要求当用户在进行文字评论时，需要对教育资源进行打分，这样才可以进行推荐计算。日志具体规范如表 4-13 所示。

（5）收藏行为

收藏行为日志用来记录用户在网站中对教育资源的收藏操作。用户如果喜欢某一资源，可以点击收藏按钮进行收藏。此类型日志可以作为布尔型日志供推荐引擎使用。日志具体规范如表 4-14 所示：

表 4-14 收藏行为日志埋点规范

参数	类型	描述
behavior_type	String	用户行为类型，此处为点击，取值为“favorite”
user_id	Long	用户 id
file_id	Long	教育资源 id
behavior_time	String	行为产生时间

（6）浏览行为

浏览行为日志用来记录用户在网站中对教育资源的浏览操作。此类型日志可以作为布尔型日志供推荐引擎使用。日志具体规范如表 4-15 所示：

表 4-15 浏览行为日志埋点规范

参数	类型	描述
behavior_type	String	用户行为类型，此处为点击，取值为“browse”
user_id	Long	用户 id
file_id	Long	教育资源 id
behavior_time	String	行为产生时间

4.6 本章小结

本章首先对教育资源推荐系统进行了总体设计，然后结合需求分析，对推荐系统进行了功能模块划分，共划分出三大功能模块，分别为海量日志收集模块、推荐引擎模块和推荐服务器模块。接下来，本章对实现三大模块功能的技术架构进行了设计。最后本章对数据库和日志埋点规范做了设计。

第 5 章 推荐系统详细设计

本章根据概要设计对教育推荐系统的功能模块的划分，详细设计并实现推荐系统中各模块的算法和功能。

5.1 海量日志收集模块详细设计

海量日志收集模块实现对云教育平台用户行为日志的收集。本模块在详细设计过程中将分为两个子模块进行设计。

5.1.1 日志服务器子模块

（1）类图设计

日志服务器是海量日志收集模块的核心，云教育平台前端通过埋点发送用户行为日志到日志服务器，日志服务器对日志的清洗、过滤和预处理，将其发送到 Kafka 集群。图 5-1 显示了该模块的类图。日志服务器是一个使用 Spring MVC 技术搭建的 B/S 服务器，Controller 负责处理前端发来的请求，具体业务由 Service 实现。从图中可以看出，LogController 类是整个服务器的核心，前端发来的所有日志都是经过它处理的。Click、Grade、Favorite、Browse、BoolGrade、Comment 类分别是用户点击、评分、收藏、浏览、布尔型评分、评论等行为日志的实体类。KafkaProducer 提供向 Kafka 集群生产日志的功能。接口 KafkaProperties 中定义了一系列的常量，用于对 KafkaProducer 进行配置。类 LogService 实现了接口 ILogService，通过 KafkaProducer 向 Kafka 集群发送日志。类 CleanLogService 实现了接口 ICleanLogService，用于对各种类型的日志进行清洗、过滤和预处理。

下面简要介绍 LogController、LogService 和 CleanLogService 类中的主要方法。

LogController 提供了诸如 sendClickLog(Click)、sendGradeLog(Grade)等方法，用于接收云教育平台通过前端埋点发送来的数据，即用户行为日志，经过处理后，发送到 Kafka 集群。

LogService 提供了诸如 produceClickLog(Click)、produceGradeLog(Grade)等方法，用于向 Kafka 集群生产日志。

CleanLogService 提供了诸如 cleanClickLog(Click)、cleanGradeLog(Grade)等方法，用于对收集到的日志进行清洗、过滤和预处理。

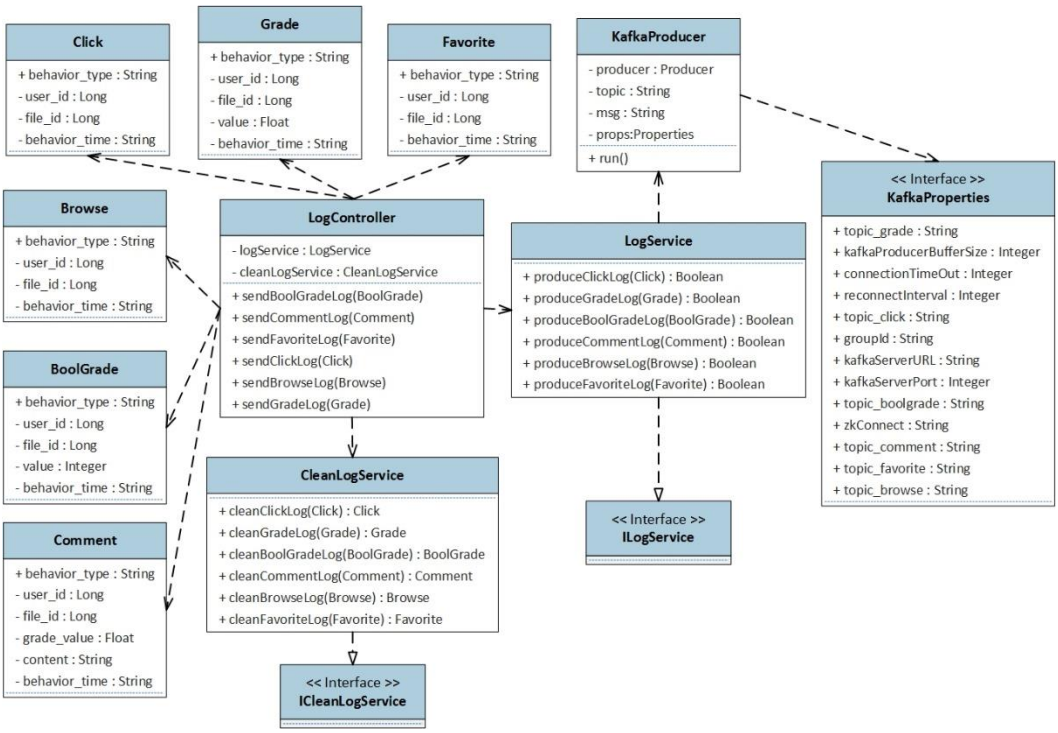


图 5-1 日志服务器类图

（2）时序图

以收集用户点击行为日志（Click）为例，作出时序图，如图 5-2 所示。云教育平台在前端首先收集用户的点击行为，Ajax 向日志服务器中的 LogController 发送 Click 日志，调用 sendClickLog()方法，该方法首先调用 CleanService 中的 cleanClickLog()方法，对日志进行预处理，然后返回一个标准的 Click 对象。sendClickLog()方法再调用 LogService 中的 produceClickLog()方法启动一个 KafkaProducer 线程向 Kafka 集群写入日志，以供 Flume 进行收集。

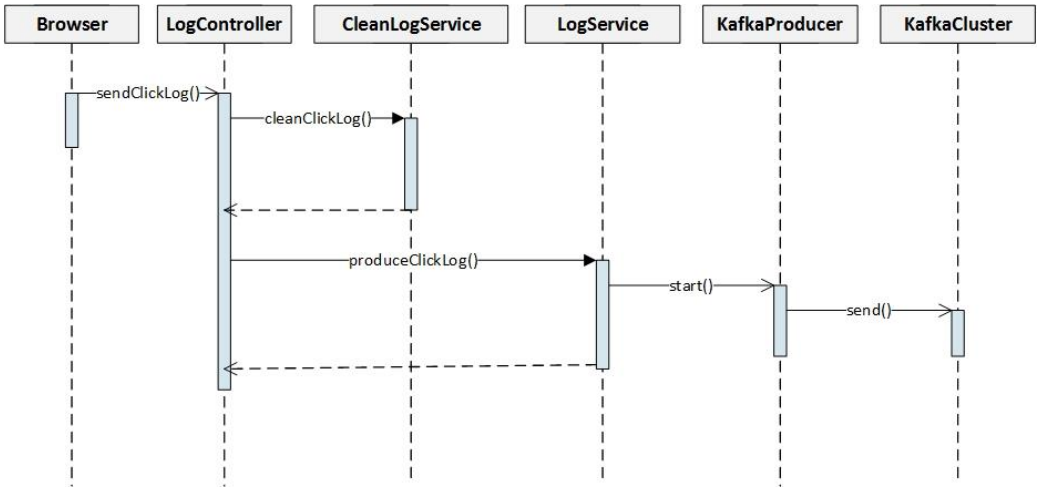


图 5-2 用户点击行为日志收集时序图

5.1.2 Flume 日志收集子模块

Kafka 和 Flume 是实现海量日志收集模块高吞吐和高并发的关键技术，常用的日志组件 log4j 并不能很好的满足本系统的要求。在对 Kafka 集群的设计过程中，根据用户行为日志的类型和特点，在 Kafka 中创建了 6 个 Topic，如 Click Topic、Grade Topic 等，日志服务器在获得前端发送来的用户行为日志后，根据其种类发送到 Kafka 里与其对应的 Topic 中。Flume 则负责从 Kafka 中获取日志，并最终存储到 HDFS 中。整个过程如图 5-3 所示

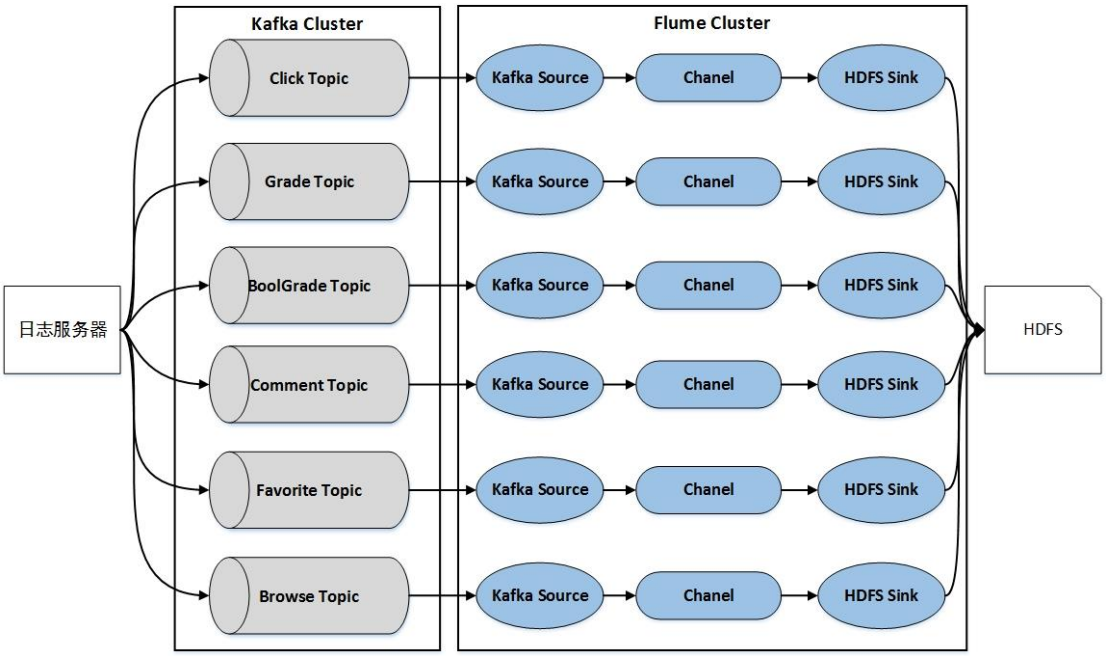


图 5-3 Flume 收集 Kafka 中用户行为日志

5.2 推荐引擎模块详细设计

推荐引擎模块实现了两种计算方式，分别为离线计算和在线计算。离线计算部分需要使用 4 个协同过滤推荐算法，本文设计并实现了 Slope One 推荐算法和基于用户的推荐算法。基于物品的推荐算法和 ALS-WR 推荐算法则使用了 Mahout 的实现。下文将会对 Slope One 推荐算法的实现和基于用户推荐算法的实现进行详细介绍。

5.2.1 Slope One 推荐算法的实现

本文在第 2 章中已经详细介绍了 Slope One 推荐算法的基本原理。为了使

Slope One 推荐算法能够在 Hadoop 集群上并行计算，本文依据该算法的基本原理，结合并行化计算的知识，将该算法进行了 MapReduce 实现。图 5-4 显示了 Slope One 推荐算法的类图。

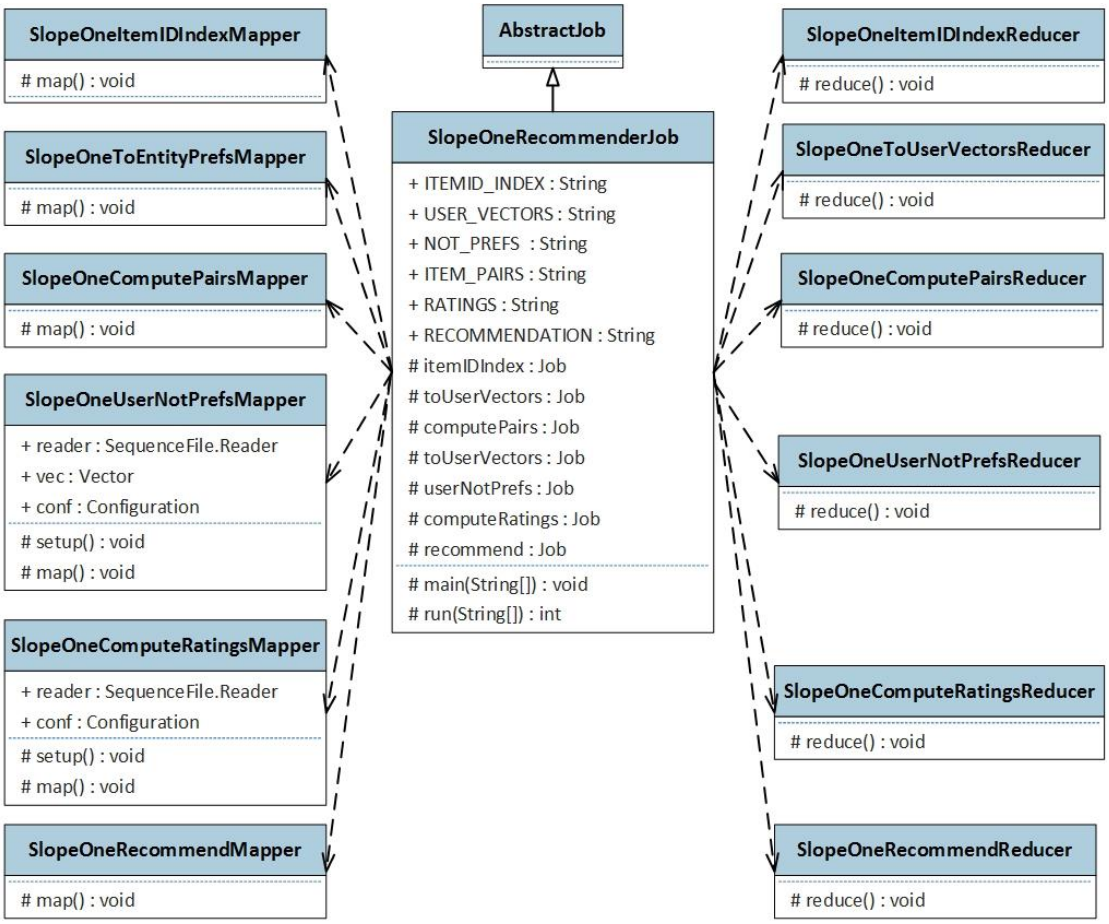


图 5-4 Slope One 推荐算法类图

从图 5-4 中可以看出，对 Slope One 推荐算法的 MapReduce 实现需要 6 对 MapReduce 过程， 13 个类，其中 SlopeOneRecommenderJob 负责有序调用这 12 个类，最终生成推荐结果。图 5-5 描述了该推荐算法的计算过程。下面对该算法的 MapReduce 计算过程进行详细介绍。

（1）SlopeOneItemIDIndexMapper 和 SlopeOneItemIDIndexReducer 负责将物品 ID（itemID）转换为内部索引。输入数据为用户物品评分矩阵，即收集到的用户行为日志，格式为“userID, itemID, pref”。最终生成 itemIDIndex 文件，并存储到 HDFS 中。

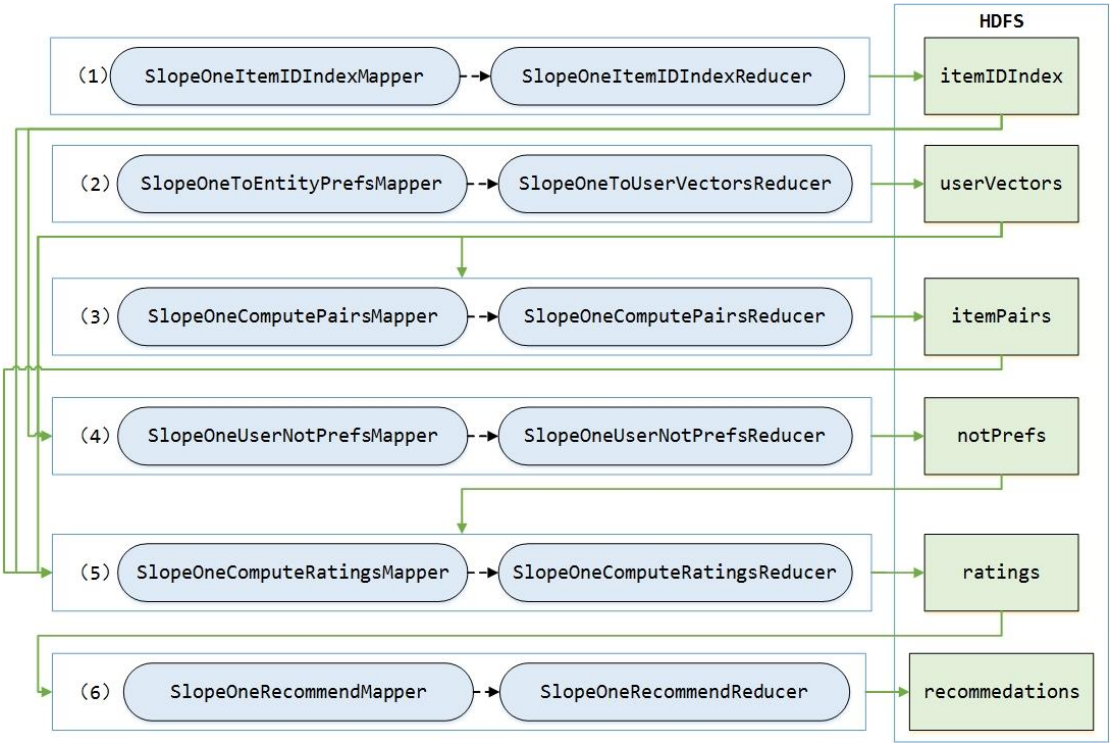


图 5-5 Slope One 推荐算法计算过程

（2）SlopeOneToEntityPrefsMapper 和 SlopeOneToUserVectorsReducer 负责将用户物品评分矩阵转换为用户对物品的评分向量。输入数据同样为用户行为日志。map 过程读取每一行日志，获得 userID、itemID 和 pref，并以 userID 为键，“itemID pref”为值输出。reduce 过程把同一个用户对所有物品的评分存储到一个向量中，然后输出。最终生成 userVectors 文件。过程如图 5-6 所示。

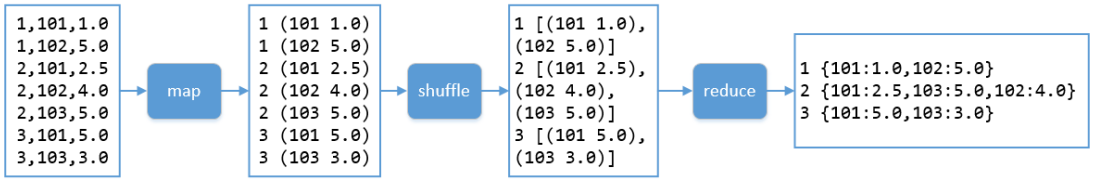


图 5-6 Slope One 推荐第（2）步计算过程

（3）SlopeOneComputePairsMapper 和 SlopeOneComputePairsReducer 负责计算物品间评分的平均差值。输入数据为 userVectors。map 过程读取用户对物品的评分向量后，首先根据物品 ID 将评分排序并存储到数组中，然后把数组内的评分两两相减，以相减两个物品的 itemID 对为键，格式为“itemB itemA”，以二者评分差为值输出。reduce 过程把所有键相同的用户对物品的评分差相加，并同时

统计相加的个数。同样，以“itemB itemA”为键，以“sum diff”为值输出，其中，sum 为同一键值的评分差相加的个数，diff 为评分差。最终生成 itemPairs 文件。过程如图 5-7 所示。

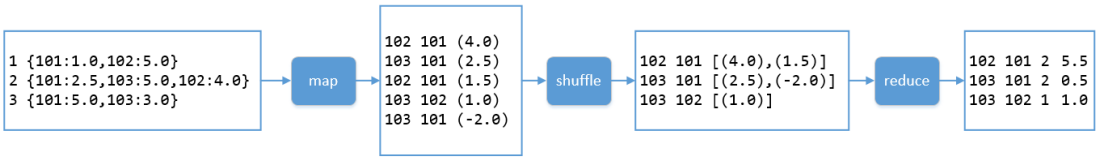


图 5-7 Slope One 推荐第（3）步计算过程

（4）SlopeOneUserNotPrefsMapper 和 SlopeOneUserNotPrefsReducer 负责计算获取用户未评价的物品列表。输入数据为 userVectors。另外在 map 过程前的 setup 阶段，需要加载第（1）步的输出文件 itemIDindex，以得到所有物品 ID。map 过程读取用户对物品的评分向量后，对用户评价过的物品做标记，最终可得到该用户未评价物品的 itemID，经过 shuffle 和 reduce 之后输出，输出格式为“userID itemID”。最终生成 notPrefs 文件。过程如图 5-8 所示。

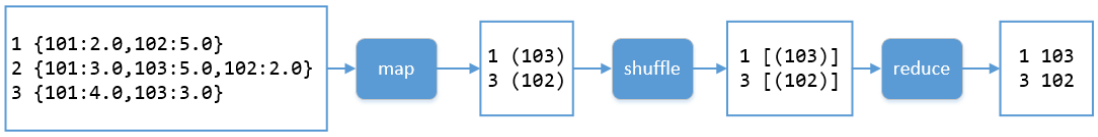


图 5-8 Slope One 推荐第（4）步计算过程

（5）SlopeOneComputeRatingsMapper 和 SlopeOneComputeRatingsReducer 负责计算获得用户对未评价物品的预测评分。输入数据 userVectors。在 setup 阶段，需要加载上文中得到的物品间评分的平均差值 itemPairs 文件和存有所有物品 ID 的 itemIDIndex 文件。在 map 阶段，读取用户对物品的评分向量，根据用户对每个物品的评分和物品间评分的平均差值，以公式(2-19)计算得到用户对未评价物品的评分，然后以 userID 为键，“itemID predPref”为值输出，其中 predPref 代表用户对未评价物品的预测评分。reduce 阶段把同一用户对所有未评价物品预测评分存储到向量中并输出。最终生成 ratings 文件。过程如图 5-9 所示。

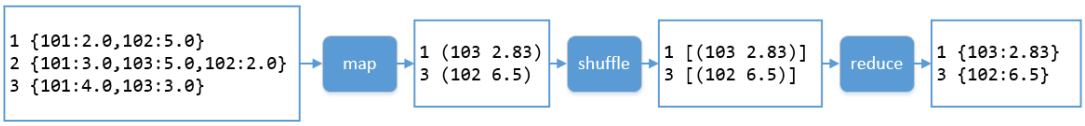


图 5-9 Slope One 推荐第（5）步计算过程

（6）SlopeOneRecommendMapper 和 SlopeOneRecommendReducer 负责计算获得最终推荐结果。输入数据为 ratings。在 map 阶段，根据 TopN 推荐方式的规则，首先把用户对未评价物品的预测评分排序，然后根据调用算法时输入的 numofRecommendation 参数，即算法给每个用户推荐的数量，获得 TopN 推荐，并以 userID 为键，以“itemID predPref”为值输出。reduce 阶段把对每个用户的推荐结果存储到向量中并输出。最终生成 recommendations 文件。过程如图 5-10 所示。

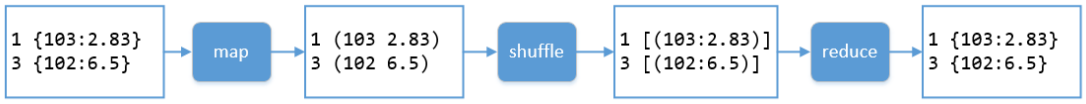


图 5-10 Slope One 推荐第（6）步计算过程

5.2.2 基于用户的推荐算法的实现

同 Slope One 推荐算法一样，本文在第 2 章中也详细介绍了基于用户的推荐算法的基本原理。本小节将该算法进行了 MapReduce 实现。图 5-11 显示了基于用户的推荐算法的类图。

从图 5-11 中可以看出，基于用户的推荐算法的 MapReduce 实现总共需要 11 个 Map 过程和 8 个 Reduce 过程，加上 UserCFRecommenderJob 总共需要 20 个类。UserCFRecommenderJob 负责有序地执行 MapReduce 过程，最终生成推荐结果。

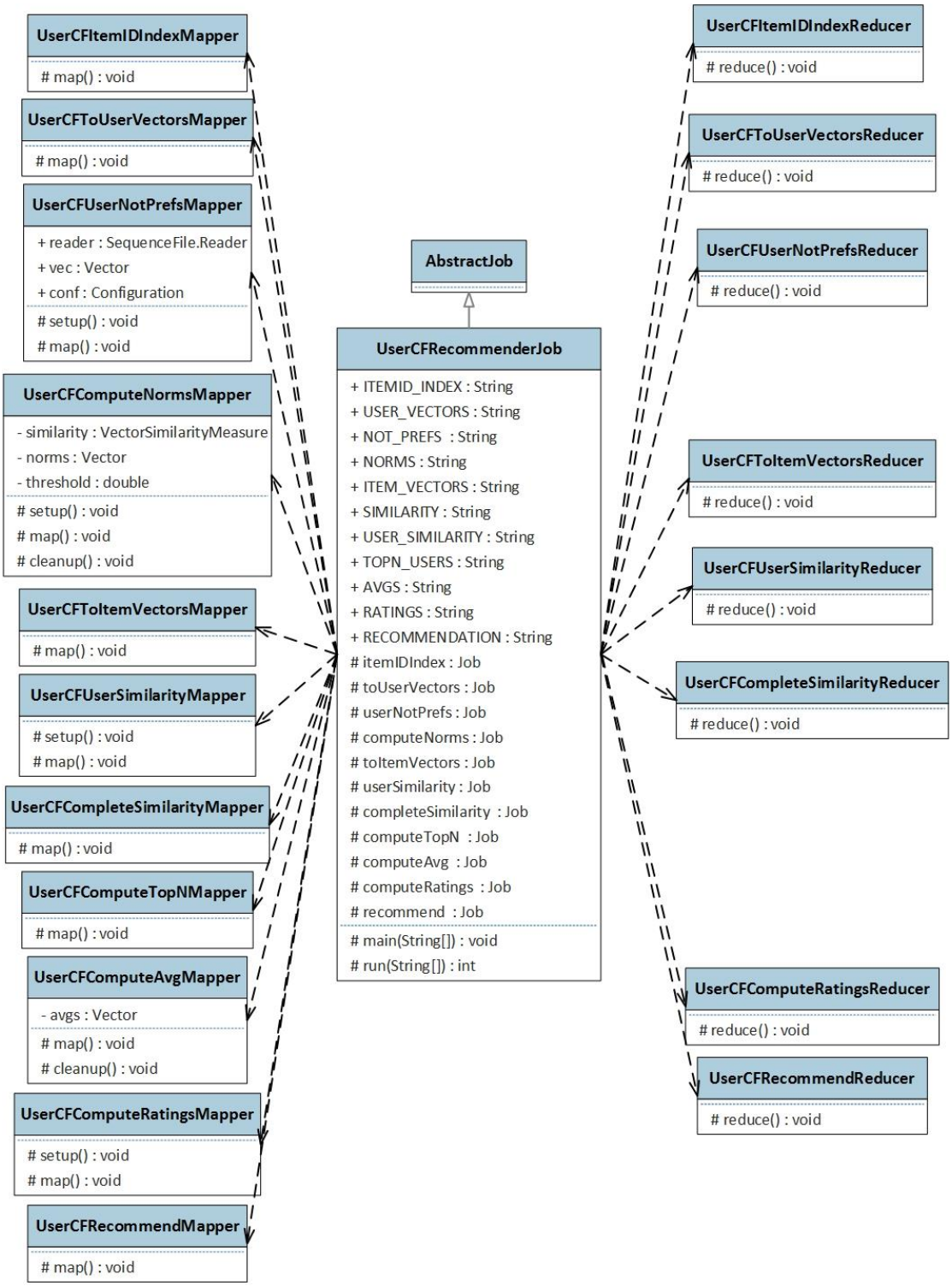


图 5-11 基于用户推荐算法类图

图 5-12 描述了基于用户推荐算法的整个计算过程。下面对其进行详细介绍。

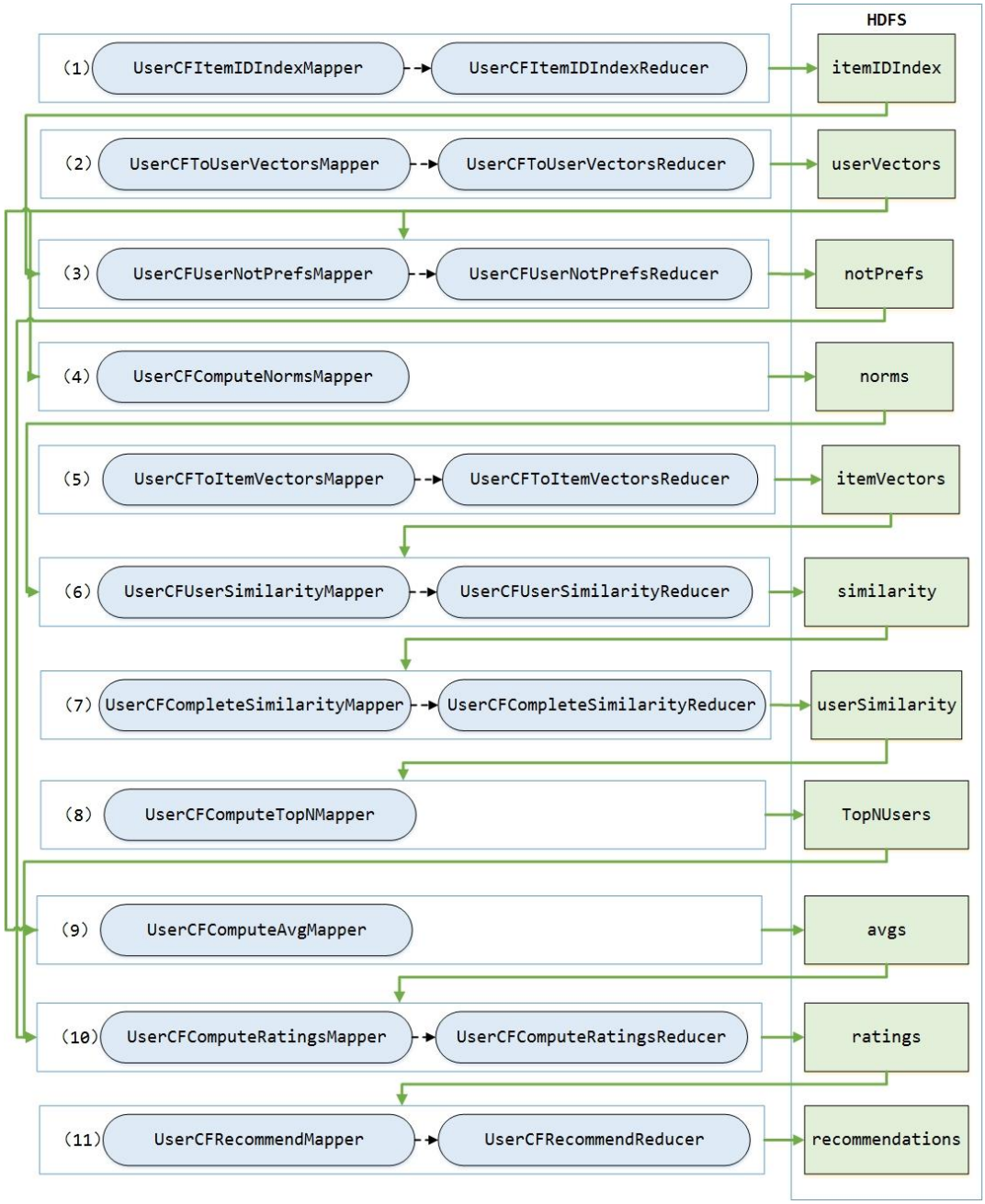


图 5-12 基于用户推荐算法计算过程

（1）UserCFItemIDIndexMapper 和 UserCFItemIDIndexReducer 负责将物品 ID（itemID）转换为内部索引。输入数据为用户物品评分矩阵，即收集到的用户行为日志，格式为“userID, itemID, pref”。最终生成 itemIDIndex 文件，并存储在 HDFS 中。

（2）UserCFToUserVectorsMapper 和 UserCFToUserVectorsReducer 负责将用户物品评分矩阵转换为用户对物品的评分向量。本过程与 Slope One 推荐算法的第（2）步的计算过程类似，输入数据为收集到的用户行为日志，输出文件

userVectors。此处只列出它们的输入和输出，如图 5-13 所示。

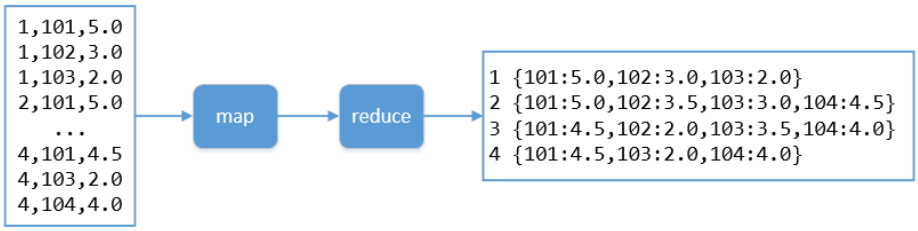


图 5-13 基于用户的推荐第（2）步计算过程

（3）UserCFUserNotPrefsMapper 和 UserCFUserNotPrefsReducer 负责获取用户未评价的物品列表。本过程与 Slope One 推荐算法的第（4）步的计算过程类似，输入数据 userVectors，输出文件为 notPrefs。此处只列出它们的输入和输出，如图 5-14 所示。

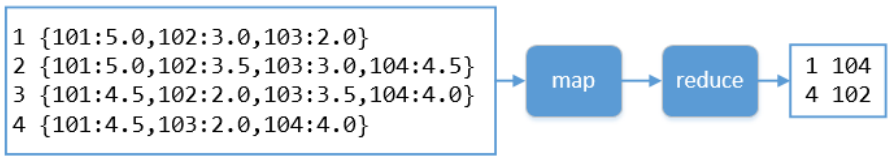


图 5-14 基于用户的推荐第（3）步计算过程

（4）UserCFComputeNormsMapper 负责计算 norms，此处 norms 是指每个用户对物品评分的平方和。此过程只需要一个 Mapper 即可完成。输入数据 userVectors。map 读取用户对物品的评分向量，计算得出用户对物品评分的平方和存储到向量中并输出。最终生成 norms 文件。过程如图 5-15 所示。

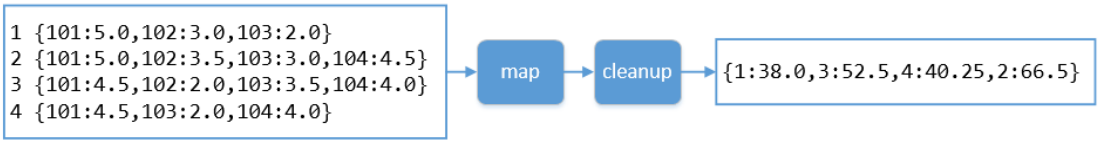


图 5-15 基于用户的推荐第（4）步计算过程

（5）UserCFToItemVectorsMapper 和 UserCFToItemVectorsReducer 负责将用户物品评分矩阵转换为物品的用户评分向量。输入数据为收集到的用户行为日志。map 过程读取每一行日志，获得 userID、itemID 和 pref，以 itemID 为键，“userID pref”为值输出。reduce 过程将所有用户对同一物品的评分存储到一个向量中输出。最终生成 itemVectors 文件。过程如图 5-16 所示。

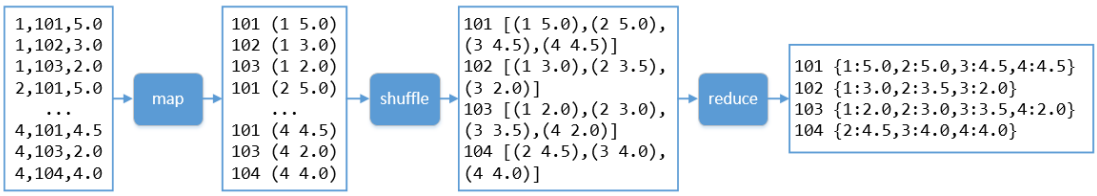


图 5-16 基于用户的推荐第（5）步计算过程

（6）UserCFUserSimilarityMapper 和 UserCFUserSimilarityReducer 负责计算获取用户间的相似度。输入数据为 itemVectors。此处选择欧氏距离相似度解释此计算过程，即公式(2-2)。对欧式距离的计算公式进行分解，如下所示：

$$d(\mathbf{a},\mathbf{b}) = \sqrt{\sum_{i=1}^{n}(a_i - b_i)^2}$$

$$= \sqrt{({a_1}^2 + {b_1}^2 - 2a_1b_1) + ({a_2}^2 + {b_2}^2 - 2a_2b_2) + \cdots + ({a_n}^2 + {b_n}^2 - 2a_nb_n)}$$

从计算公式中可以看出，欧式距离的计算过程中需要获得用户对物品评分的平方和与两个用户对物品评分的乘积 agg。map 阶段读取物品的用户评分向量，并把向量中用户对物品的评分两两相乘并输出。reduce 阶段把两个相同用户对物品评分的乘积相加，得到最终的 agg。本过程 setup 阶段需要读取第（4）步计算得到的用户对物品评分的平方和，即 norms 文件。最后本过程根据上面计算获得的数据和欧氏距离的计算公式，得到用户间的相似度并存储到向量中。最终生成 similarity 文件。不同的相似度计算方法计算 norms 和 agg 的方法是不同的，此处只是使用欧氏距离相似度举例。过程如图 5-17 所示。

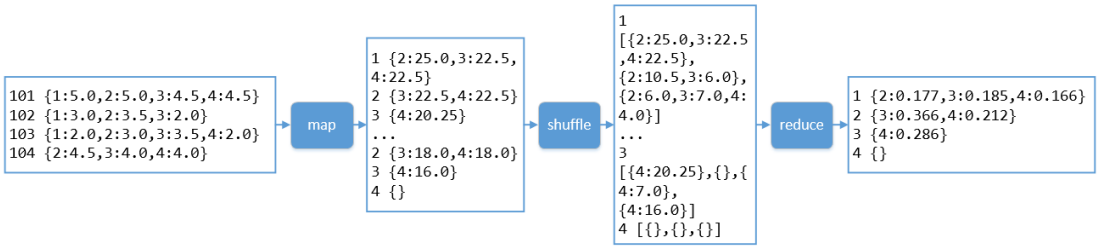


图 5-17 基于用户的推荐第（6）步计算过程

（7）UserCFCompleteSimilarityMapper 和 UserCFCompleteSimilarityReducer 负责补全用户间的相似度，生成相似度矩阵。输入数据为 similarity。map 过程读取每一个用户与其他用户的相似度向量，通过交换 userID 补全用户间的相似度，比如，map 从原文件中读取到用户(1,2)之间的相似度为 0.177，但原文件中并没有

存储用户(2,1)的相似度，此时只需要交换(1,2)便可以得到(2,1)的相似度，同样为0.177。map 过程以 userAID 作为键，以“userBID similarity”作为值输出。reduce 过程即把一个用户对其他所有用户的相似度存储到向量中输出。最终生成 userSimilarity 文件。过程如图 5-18 所示。

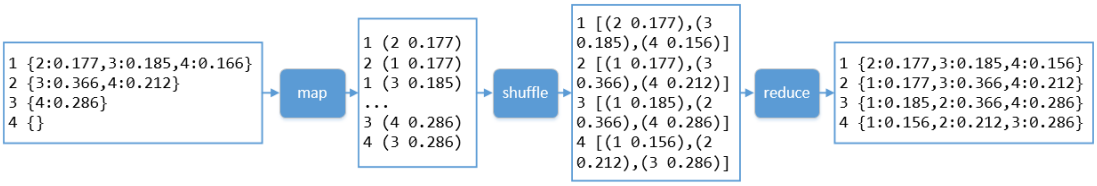


图 5-18 基于用户的推荐第（7）步计算过程

（8）UserCFComputeTopNMapper 负责获取用户的邻域。此过程只需要一个 Mapper 即可完成。输入数据为 userSimilarity。此过程很简单，根据用户间的相似度大小将用户排序，获得用户邻域，邻域的大小是由用户调用算法时设置的。最终生成 topNUsers 文件。过程如图 5-19 所示。

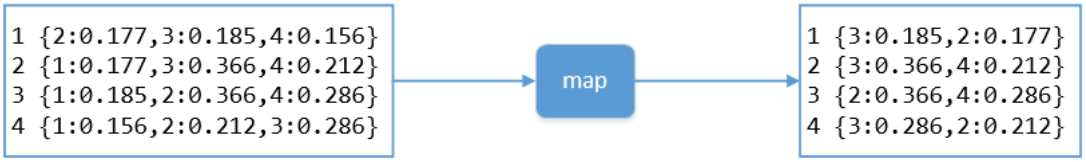


图 5-19 基于用户的推荐第（8）步计算过程

（9）UserCFComputeAvgMapper 负责计算每个用户对物品打分的平均值。此过程只需要一个 Mapper 即可完成。输入数据为 userVectors。map 读取用户对物品的评分向量，计算出向量中用户对物品评分的平均值，存储到另一个向量中并输出。最终生成 avgs 文件。过程如图 5-20 所示。

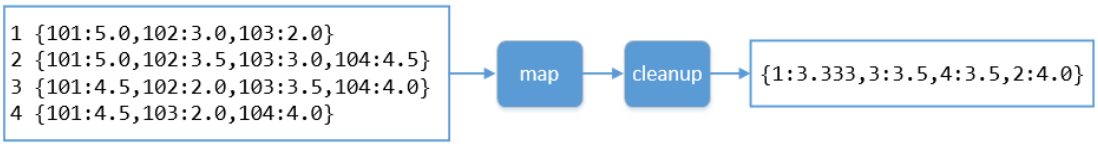


图 5-20 基于用户的推荐第（9）步计算过程

（10）UserCFComputeRatingsMapper 和 UserCFComputeRatingsReducer 负责计算得到用户对未评价物品的预测评分。输入数据为 notPrefs。在 setup 阶段需要读取 avgs、topNUsers 和 userVectors 文件，并加载到相关的容器中，供 map 阶

段调用。map 阶段获取每个未评价的物品 ID，根据公式(2-17)，计算得到用户对该物品的预测评分并输出。reduce 阶段得到每个用户对所有未评价物品的预测评分，写入向量中并输出。最终生成 ratings 文件。过程如图 5-21 所示。

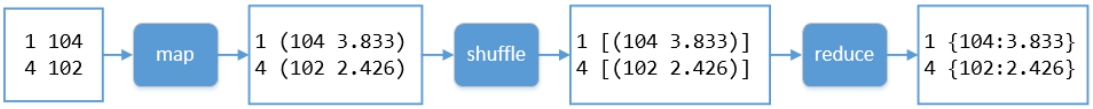


图 5-21 基于用户的推荐第（10）步计算过程

（11）UserCFRecommendMapper 和 UserCFRecommendReducer 负责计算获得最终推荐结果。输入数据为 ratings 文件。在 map 阶段，首先把用户对物品的预测评分排序，然后根据调用算法时输入的 numofRecommendation 参数，即算法给每个用户推荐的数量，获得 TopN 推荐。最后经过 shuffle 和 reduce 阶段，生成为每个用户推荐的推荐向量。最终生成 recommendations 文件。过程如图 5-22 所示。

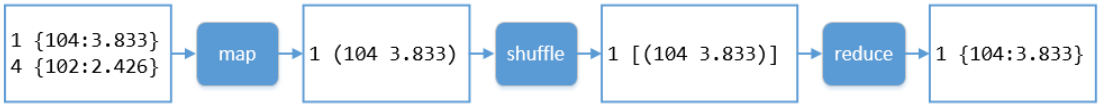


图 5-22 基于用户的推荐第（11）步计算过程

5.2.3 其他推荐方式的实现

（1）基于教育资源热门度的推荐。系统可以通过在线计算的方式获取教育资源热门度，计算原理非常简单，通过读取近几天的用户行为日志，包括点击、评论、收藏等行为，统计每个资源所产生的用户行为数量，用户行为数量越多，教育资源也就越热门。对于教育资源热门度的计算结果，系统可以直接存储到数据库中。

（2）基于用户信息的推荐。本推荐方式也是通过在线计算获取推荐结果的。推荐系统可以通过查询数据库，获取与用户相关信息匹配的教育资源并推荐给他。比如，如果用户的注册信息显示他目前正处于初中三年级，那么推荐系统就可以推荐适配初中三年级的教育资源给他。当然，有部分用户在注册的时候并没有提供足够的信息，比如他们只填写了出生年月，此时推荐系统可以通过简单的计算，推断用户所处的教育阶段，并推荐适当的教育资源给他。

5.3 推荐服务器模块详细设计

推荐服务器模块是连接云教育平台和教育资源推荐系统的桥梁，它通过封装设计一系列接口，供云教育平台前端调用，获取推荐结果。

5.3.1 推荐服务器

（1）类图设计

推荐服务器是一个使用 Spring MVC 技术搭建的 B/S 服务器，Controller 负责处理前端发来的请求，具体业务是由 Service 实现的。图 5-23 显示了推荐服务器的类图，本推荐服务器为了实现相关业务共有几十个类和接口，无法全部列出，故此处只列出与推荐系统有关的重要类。其中，业务类 RecommendService 用于获取各种推荐算法和推荐场景的推荐结果。RecommendController 用于提供接口以处理前端的请求，通过 RecommendService 获取推荐结果，并把结果返回到前端。业务类 TreatRecsService 用于对推荐结果进行处理，如过滤、特别推荐等。业务类 SimilarityService 用于处理和获取用户间相似度和教育资源间相似度。SimilarityController 用于提供接口供前端获取用户或教育资源间的相似度。HybridRecommendService 和 HybridRecommendController 用于混合推荐，下一小节将着重介绍。MailService 和 MailController 用于给用户发送推荐邮件。Rec、RecItem、PopRec 和 Sim 分别是推荐结果、推荐结果具体项目、热门推荐结果和相似度结果的实体类。另外，由于推荐服务器涉及到的表非常多，涉及到的操作类也非常多，无法全部列出，故在本图中是用 DAO 类代表对数据库的操作。下面详细介绍 RecommendService、TreatRecsService 和 SimilarityService 类的一些关键方法。

表 5-1 RecommendService 类的关键方法

方法名	功能描述
getRecsByUserCF(Long,Integer)	获取基于用户的推荐算法的推荐结果
getRecsByItemCF(Long,Integer)	获取基于物品的推荐算法的推荐结果
getRecsByALSWR (Long,Integer)	获取 ALS-WR 推荐算法的推荐结果
getRecsBySlopeOne (Long,Integer)	获取 Slope One 推荐算法的推荐结果
getRecsByPopularity(Integer)	获取基于资源热门度的推荐结果
getRecsByUserInfo(Long,Integer)	获取基于用户信息的推荐结果

表 5-2 TreatRecsService 类的关键方法

方法名	功能描述
convertRecs (String)	解析并转化推荐结果，存储到容器中
sortRecs (List)	对推荐结果进行排序
filterRecs (List)	对推荐结果进行过滤
specRecs (List)	在推荐结果中加入特别推荐的教育资源

表 5-3 SimilarityService 类的关键方法

方法名	功能描述
treatUserSim()	处理用户间相似度
treatFileSim()	处理教育资源间的相似度
getUserSim(Long,Long)	获取用户间的相似度
getFileSim(Long,Long)	获取教育资源间的相似度

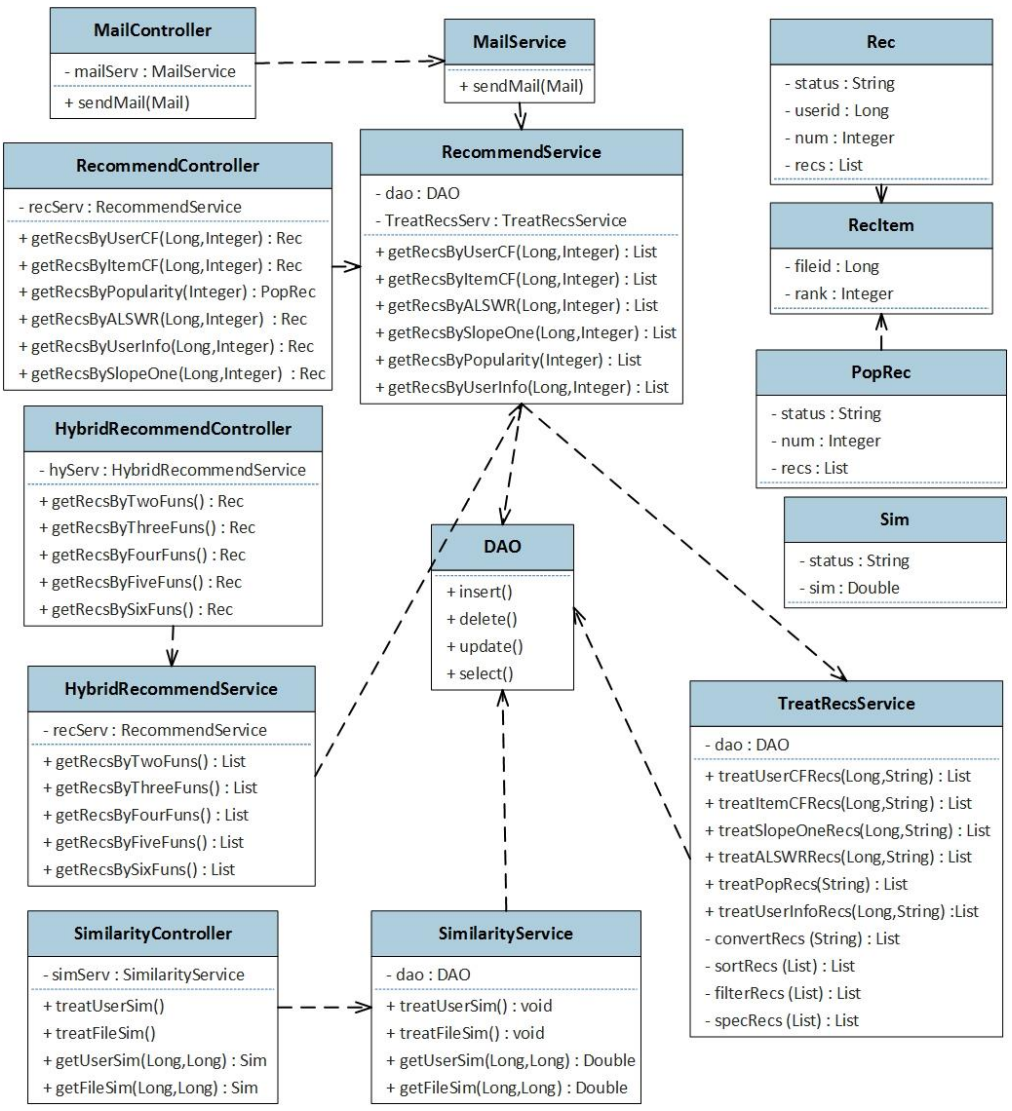


图 5-23 推荐服务器类图

（2）时序图

以获取基于用户推荐的算法的结果为例，作出时序图，如图 5-24 所示。云教育平台首先通过调用相关接口，执行 RecommendController 中的 getRecsByUserCF() 方法，该方法调用业务类 RecommendService 中的 getRecsByUserCF()方法，业务类中的 getRecsByUserCF()通过 DAO 类对数据库进行查询，获取对某一用户的推荐结果，推荐结果是以字符串的形式存储的。接下来业务类中的 getRecsByUserCF()调用 TreatRecsService 类中的 treatUserCFRecs()方法对推荐结果进行处理，treatUserCFRecs()接下来依次调用 convertRecs()、sortRecs()、filterRecs()、specRecs()，分别对推荐结果进行格式转换、排序、过滤和特别推荐，最后返回一个最终的推荐结果。

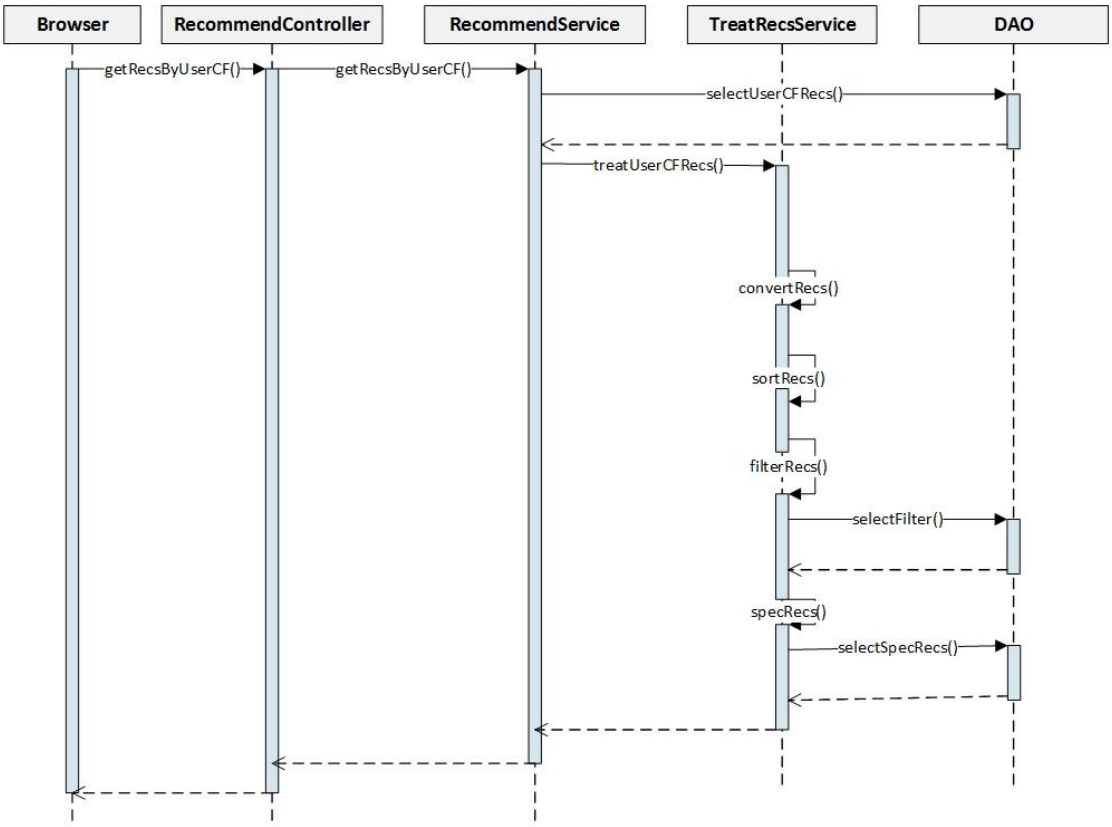


图 5-24 获取基于用户推荐算法推荐结果的时序图

5.3.2 混合推荐服务

本推荐系统设计并实现了 6 种推荐方法，这 6 种推荐方法存在不同的优缺点，如果把它们组合起来，就可以取长补短，提高推荐效果。

经过对混合推荐相关文献的研究[42]，本文提出并设计了一个适合教育资源的

混合推荐方法。该方法可以对上文中 6 种推荐方法的任意混合，并产生推荐结果。它的实现原理为，首先获取每种推荐方法的推荐结果，然后根据它们所占的权重，把推荐结果重新排序，获取新的推荐结果。注意，此处的权重需要推荐系统根据云教育平台用户对推荐效果的反馈，动态调整。

举例说明，云教育平台想要获得某用户的 5 个基于用户推荐方式和基于物品推荐方式的混合推荐结果，且二者所占的权重分别为 0.7 和 0.3，计算过程可分为如下几个步骤：

（1）分别获取两种推荐方法的推荐结果。如表 5-4 所示，其中 rank 代表本物品在本推荐方法中的排名，排名越小，越优先推荐。

表 5-4 两种推荐方法的推荐结果

基于用户的推荐结果	fileid	rank		基于物品的推荐结果	fileid	rank
	101	1			106	1
	102	2			107	2
	103	3			108	3
	104	4			109	4
	105	5			110	5

（2）根据两种推荐方法所占的权重，结合 rank，重新计算并排序。计算公式可定义如下，本公式计算了推荐方式 j 中物品 i 的排序参数。

$$\mathrm{nr}(\mathrm{i},\mathrm{j}) = (\mathrm{n} - r_i + 1) * w_j \tag{5-1}$$

其中， n 表示本次推荐所请求的推荐结果数量，r_i表示物品 i 在原推荐方式中的排名，即 rank，w_j表示 j 推荐方式所占的权重。

经过计算，获取以下新的排名。如表 5-5 所示。

表 5-5 混合两种推荐方法后的新排名

排名	1	2	3	4	5	6	7	8	9	10
fileid	101	102	103	106	104	107	108	105	109	110
nr	3.5	2.8	2.1	1.5	1.4	1.2	0.9	0.7	0.6	0.3

（3）根据新排名，获取新的推荐结果。在本例子中，可以取物品 101、102、103、106 和 104 作为混合推荐的推荐结果，返回给云教育平台。

实现本功能的业务类 HybridRecommendService 中提供了五个方法，如 getRecsByTwoFuns()、getRecsByThreeFuns()等，它们分别用来对不同数量的推荐方

法进行混合推荐，如 getRecsByTwoFuns()就提供对两种推荐方法的混合推荐，需要依次传入的参数有 funa、funb、weighta、weightb、userid、num，它们分别表示第一个方法名、第二个方法名、第一个方法所占权重、第二个方法所占权重、用户 ID 和需要推荐的数目。其他方法与该方法类似。

5.4 本章小结

本章对教育资源推荐系统的三大功能模块进行了详细设计，实现了它们的算法和功能，并结合了类图和时序图对其进行说明。重点介绍了 Slope One 推荐算法和基于用户推荐算法的 MapReduce 实现。

第 6 章 推荐系统运行与测试

本章对教育资源推荐系统进行运行与测试，首先介绍系统开发环境和部署环境，然后对系统功能进行演示，最后通过实验测试不同推荐算法的推荐效果。

6.1 系统开发和部署环境

6.1.1 系统开发环境

此处列出了教育资源推荐系统各功能模块在开发过程中所用到的软件，以及它们的版本和作用，如表 6-1 所示。

表 6-1 系统开发环境

软件名称	版本	作用
Windows	Windows 10	系统开发操作系统
JDK	1.7.0_79	Java 软件开发工具包
Eclipse	Mars.2 Release (4.5.2)	开发 IDE
Kafka	2.10	分布式消息系统
Flume	1.6.0	日志收集
Hadoop	2.6.4	分布式存储和计算
Mahout	0.10.1	机器学习框架
MySQL	5.7.10	数据库
Nginx	1.11.10	负载均衡服务器
Redis	2.8.17	数据缓存
Spring MVC	4.0.2.RELEASE	Web 应用框架
Tomcat	7.0.64	服务器

6.1.2 系统部署环境

本推荐系统分为 3 个模块，它们部署在不同的服务器中，服务器的配置如表 6-2 所示。

6-2 服务器配置

名称	描述
CPU	2.93GHz Intel(R)Core(TM)i3
内存	4GB 1600MHz DDR3
硬盘	350G
操作系统	CentOS-7-x86_64-Minimal-1511

6.2 系统功能演示

6.2.1 日志收集

推荐系统中的海量日志收集模块，从云教育平台前端收集到日志后，写入到 HDFS 中，图 6-1 列出了 Hadoop 的管理页面。本 Hadoop 集群共有 3 个 datanode 和 1 个 namenode。

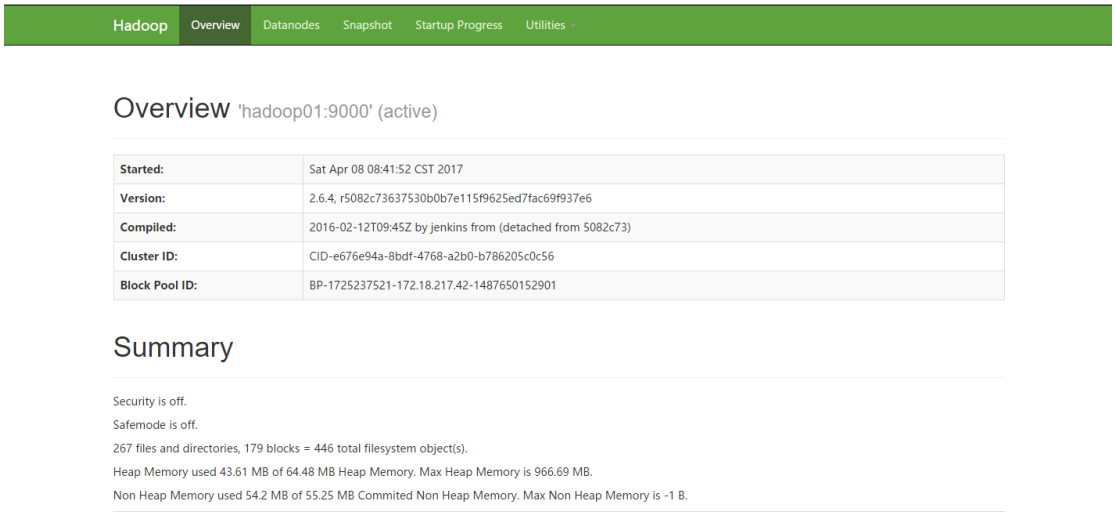
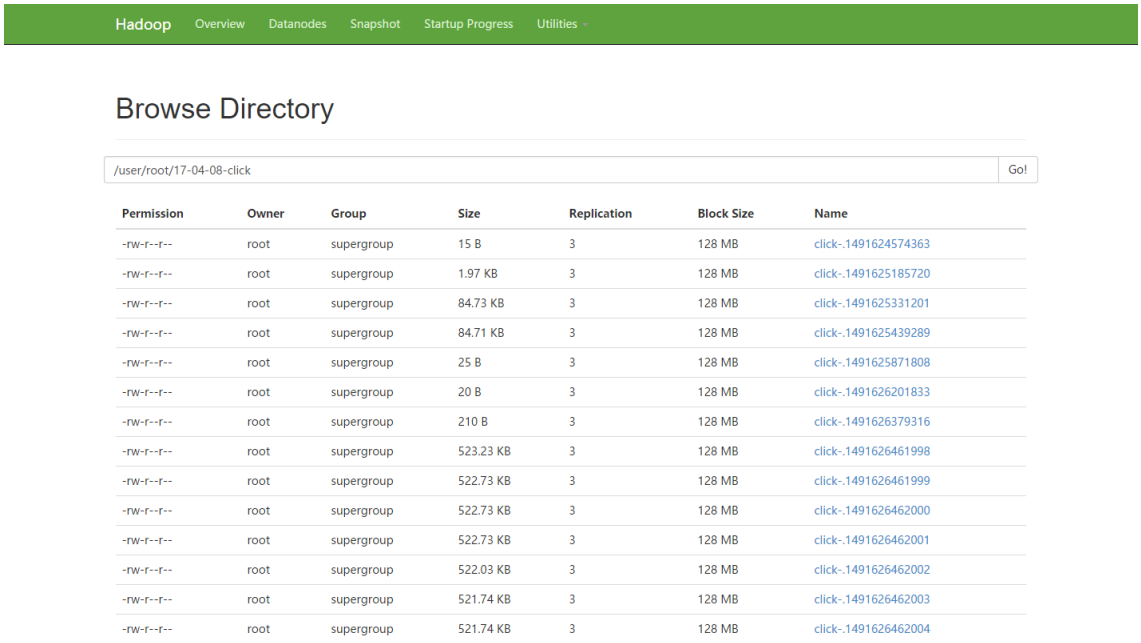


图 6-1 Hadoop 管理界面

为了测试在本部署环境中海量日志收集系统在能否应对大数据高并发日志的收集需求，本文使用软件模拟用户操作，在 60 秒内向系统发送了 25 万条 Click 日志。经过测试，证明了海量日志收集系统能够应对大数据高并发的日志收集需求。图 6-2 列出了采集到 Click 日志文件的部分内容，图 6-3 列出了存储在 HDFS 中的 Click 日志文件。

```
1 257124 20170408124156
1 257125 20170408124156
1 257126 20170408124156
1 257127 20170408124156
1 257128 20170408124156
1 257129 20170408124156
1 257130 20170408124156
1 257131 20170408124156
1 257132 20170408124156
1 257133 20170408124156
1 257134 20170408124156
1 257135 20170408124156
1 257136 20170408124156
1 257137 20170408124156
```

图 6-2 Click 日志文件部分内容



Permission	Owner	Group	Size	Replication	Block Size	Name
-rw-r--r--	root	supergroup	15 B	3	128 MB	click-.1491624574363
-rw-r--r--	root	supergroup	1.97 KB	3	128 MB	click-.1491625185720
-rw-r--r--	root	supergroup	84.73 KB	3	128 MB	click-.1491625331201
-rw-r--r--	root	supergroup	84.71 KB	3	128 MB	click-.1491625439289
-rw-r--r--	root	supergroup	25 B	3	128 MB	click-.1491625871808
-rw-r--r--	root	supergroup	20 B	3	128 MB	click-.1491626201833
-rw-r--r--	root	supergroup	210 B	3	128 MB	click-.1491626379316
-rw-r--r--	root	supergroup	523.23 KB	3	128 MB	click-.1491626461998
-rw-r--r--	root	supergroup	522.73 KB	3	128 MB	click-.1491626461999
-rw-r--r--	root	supergroup	522.73 KB	3	128 MB	click-.1491626462000
-rw-r--r--	root	supergroup	522.73 KB	3	128 MB	click-.1491626462001
-rw-r--r--	root	supergroup	522.03 KB	3	128 MB	click-.1491626462002
-rw-r--r--	root	supergroup	521.74 KB	3	128 MB	click-.1491626462003
-rw-r--r--	root	supergroup	521.74 KB	3	128 MB	click-.1491626462004

图 6-3 存储在 HDFS 中的 Click 日志文件

6.2.2 推荐服务接口

教育资源推荐系统的主要功能是通过对云教育平台的用户行为日志进行计算并获得对每个用户的个性化教育资源推荐，云教育平台跟推荐系统之间是通过相关服务接口交换数据的。本推荐系统使用了 MovieLens-100k 数据集做系统测试，该数据集是由 GroupLens 网站提供，包含了 1000 个用户对 1700 部电影的 100000 条评分数据。

下面对本推荐系统所提供的的接口进行说明和演示：

（1）常规推荐服务接口

常规推荐服务提供 6 种推荐方法，云教育平台前端调用服务接口，推荐系统返回一个 JSON 格式的推荐结果给前端，本文采用软件 Fiddler 模拟对接口的请求，对推荐服务器接口进行测试，如图 6-4 所示。下面几个表格列出了这些接口的调用方法和返回数据。

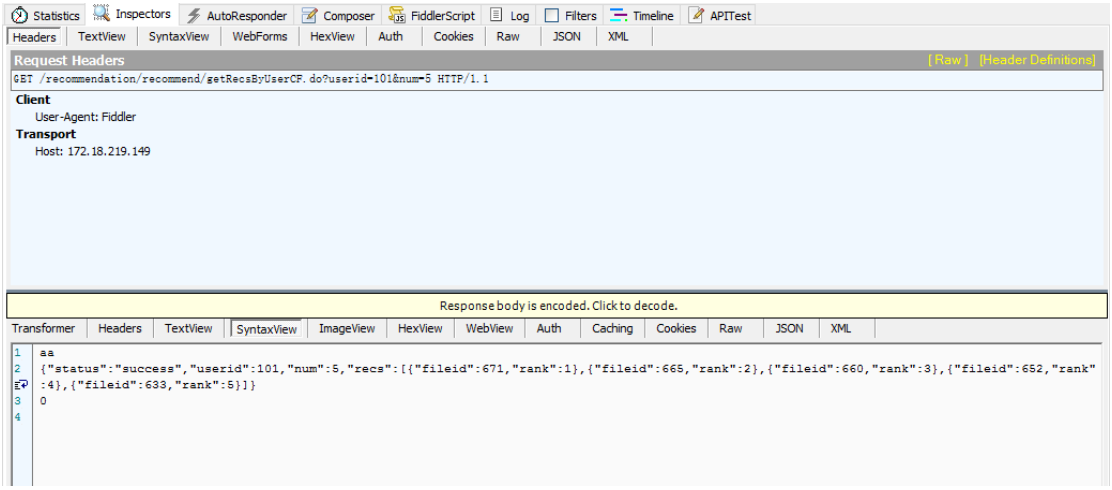


图 6-4 Fiddler 测试推荐服务器接口

表 6-3 基于用户的推荐算法服务接口

接口调用实例	recommend/getRecsByUserCF.do?userid=101&num=5
功能	获取基于用户推荐算法的推荐结果
调用参数	userid 表示用户 ID；num 表示需要获取的推荐数量
返回结果	{"status":"success","userid":101,"num":5, "recs":[{"fileid":898,"rank":1}, {"fileid":324,"rank":2},{"fileid":329,"rank":3}, {"fileid":302,"rank":4},{"fileid":315,"rank":5}]}
返回参数	status 表示接口调用状态；userid 表示用户 ID；num 表示返回的推荐数量；recs 中包含了推荐结果，每个推荐结果包含两个属性，fileid 表示教育资源 ID，rank 表示本教育资源在本次推荐中的排名

表 6-4 基于物品的推荐算法服务接口

接口调用实例	recommend/getRecsByItemCF.do?userid=211&num=5
作用	获取基于物品推荐算法的推荐结果
返回结果	{"status":"success","userid":211,"num":5, "recs":[{"fileid":229,"rank":1}, {"fileid":404,"rank":2},{"fileid":73,"rank":3}, {"fileid":77,"rank":4},{"fileid":732,"rank":5}]}

表 6-5 Slope One 推荐算法的服务接口

接口调用实例	recommend/getRecsBySlopeOne.do?userid=335&num=5
作用	获取 Slope One 推荐算法的推荐结果
返回结果	{"status":"success","userid":335,"num":5, "recs":[{"fileid":1500,"rank":1}, {"fileid":1639,"rank":2},{"fileid":1541,"rank":3}, {"fileid":1189,"rank":4},{"fileid":1449,"rank":5}]}

表 6-6 ALS-WR 推荐算法的服务接口

接口调用实例	recommend/getRecsByALSWR.do?userid=389&num=5
作用	获取 ALS-WR 推荐算法的推荐结果
返回结果	{"status":"success","userid":389,"num":5, "recs":[{"fileid":156,"rank":1}, {"fileid":237,"rank":2},{"fileid":608,"rank":3}, {"fileid":1123,"rank":4},{"fileid":582,"rank":5}]}

表 6-7 基于教育资源热门程度推荐的服务接口

接口调用实例	recommend/getRecsByPopularity.do?num=6
作用	获取基于教育资源热门度推荐的推荐结果
返回结果	{"status":"success","num":6, "recs":[{"fileid":288,"rank":1},{"fileid":1,"rank":2}, {"fileid":100,"rank":3},{"fileid":286,"rank":4}, {"fileid":181,"rank":5},{"fileid":121,"rank":6}]}

表 6-8 基于用户信息推荐的服务接口

接口调用实例	recommend/getRecsByUserInfo.do?userid=131&num=5
作用	获取基于用户信息推荐的推荐结果
返回结果	{"status":"success","userid":131,"num":5, "recs":[{"fileid":7,"rank":1},{"fileid":159,"rank":2}, {"fileid":386,"rank":3},{"fileid":27,"rank":4} ,{"fileid":871,"rank":5}]}

（2）混合推荐服务接口

本推荐系统提供混合推荐服务，云教育平台可以根据具体需求，对上文中提到的 6 种推荐方法任意组合，生成混合推荐结果。在调用混合推荐服务接口时，需要提供推荐方法的名称和它们所占的权重。表 6-8 列出了混合两个推荐方法的服务接口，其他的混合推荐服务接口与此类似。

表 6-9 混合推荐服务接口

接口调用实例	hybridRecommend/getRecsByByTwoFuns.do?funa=usercf&funb= itemcf&weighta=0.3&weightb=0.7&userid=101&num=7
作用	获取混合两种推荐方法的推荐结果
调用参数	funa 表示一个方法名；funb 表示另一个方法名；weighta 表示 funa 推荐结果所占权重；weightb 表示 funb 方法推荐结果所占权重；userid 表示用户 ID；num 表示需要获取的推荐数量

返回结果	{"status":"success","userid":101,"num":7," recs":[{"fileid":13,"rank":1},{"fileid":58,"rank":2}, {"fileid":124,"rank":3},{"fileid":692,"rank":4}, {"fileid":286,"rank":5},{"fileid":898,"rank":6}, {"fileid":324,"rank":7}]}

（3）相似度服务接口

本系统还提供对用户之间和教育资源之间的相似度查询服务接口。

表 6-10 用户相似度服务接口

接口调用实例	similarity/getUserSim.do?useraid=101&userbid=102
作用	获取用户间的相似度
调用参数	useraid 表示用户 A 的 ID；userbid 表示用户 B 的 ID
返回结果	{"status":"success","sim":0.9661}
返回参数	status 表示接口调用状态；sim 表示用户间的相似度

表 6-11 教育资源相似度服务接口

接口调用实例	similarity/getFileSim.do?fileaid=101&filebid=102
作用	获取教育资源间的相似度
调用参数	fileaid 表示资源 A 的 ID；filebid 表示资源 B 的 ID
返回结果	{"status":"success","sim":0.2469}
返回参数	status 表示接口调用状态；sim 表示教育资源的相似度

6.3 推荐效果分析

6.3.1 使用固定邻域的基于用户的推荐算法的效果分析

本次实验评测了不同的相似度度量在基于用户的推荐算法中，所选取用户邻域大小对推荐结果的影响，本实验中使用了准确率（Precision）、召回率（Recall）、均方根误差（RMSE）和平均绝对误差（MAE）进行评测。所评测的相似度度量为曼哈顿距离相似度、欧氏距离相似度、皮尔逊相关系数、余弦相似度和谷本系数等。

数据集为 MovieLens-100k 数据集，邻域取值范围为 10~110，训练集:测试集 = 8:2。实验结果如图 6-5 和图 6-6 所示所示。从图中可以看出，对于皮尔逊相关系数和余弦相似度，二者在用户邻域为 70 时推荐效果最好，准确率和召回率达到了最高。对于欧氏距离相似度、曼哈顿距离相似度和谷本系数，用户邻域为 10 时推荐效果最好，准确率和召回率达到了最高。

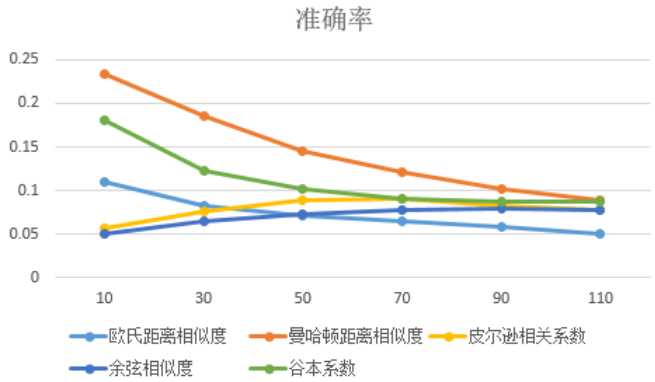


图 6-5 使用固定邻域的基于用户的推荐算法的准确率

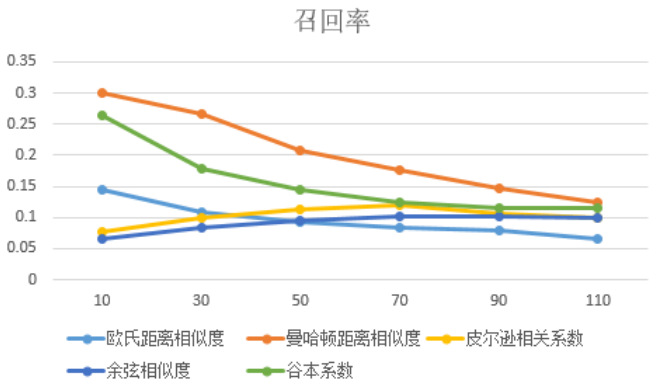


图 6-6 使用固定邻域的基于用户的推荐算法的召回率

本文在第 2 章已经说明，准确率和召回率在推荐系统中，当在物品数量非常多的情况下，对用户进行的 TopN 推荐，推荐列表中的物品跟测试集的重合率会比较小，所以准确率和召回率只能作为推荐系统评测的一个小的维度。目前常用的对推荐系统的评测指标为平均绝对误差（MAE）和均方根误差（RMSE），下面本文使用二者评测基于用户的推荐算法在不同相似度度量下使用最佳用户邻域大小的推荐效果。实验结果如图 6-7 所示。

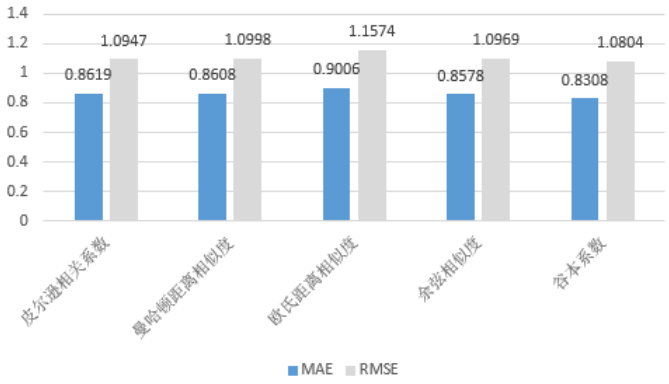


图 6-7 使用固定邻域的基于用户的推荐算法的推荐效果

从图中可以看出，不同的相似度度量在使用最佳用户邻域大小的基于用户推荐的算法中都有比较稳定的推荐效果，MAE 稳定在 0.86 左右，RMSE 稳定在 1.11 左右。

6.3.2 使用阈值邻域的基于用户的推荐算法的效果分析

本次实验评测不同的相似度度量在基于用户的推荐算法中，所选邻域阈值大小对推荐结果的影响，本实验中使用了准确率（Precision）和召回率（Recall）、均方根误差（RMSE）和平均绝对误差（MAE）进行评测。所评测的相似度度量有欧氏距离相似度、皮尔逊相关系数、余弦相似度等。

数据集为 MovieLens-100k 数据集，阈值选取范围为 0.5~0.99，训练集:测试集=8:2。得到试验结果如图 6-8 和图 6-9 所示。

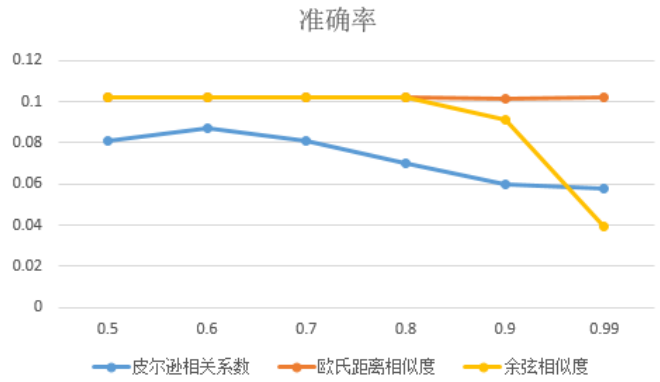


图 6-8 使用阈值邻域的基于用户的推荐算法的准确率

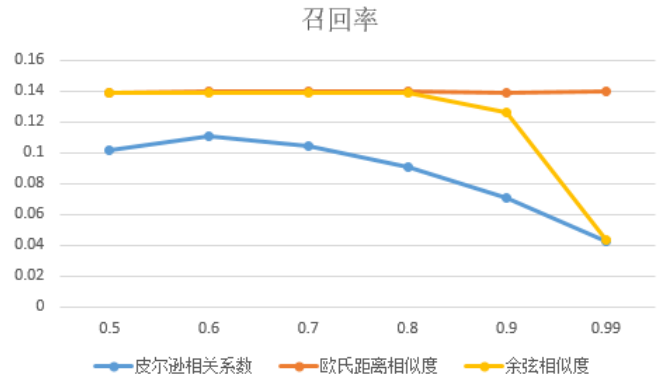


图 6-9 使用阈值邻域的基于用户的推荐算法的召回率

从图 6-8 和图 6-9 中可以看出，当阈值选取为 0.6 时，各个相似度度量的效果最好，准确率和召回率达到了最高。下面使用平均绝对误差（MAE）和均方根

误差（RMSE）评测基于用户的推荐算法在不同相似度度量下使用最佳用户邻域阈值的推荐效果。实验结果如图 6-10 所示。

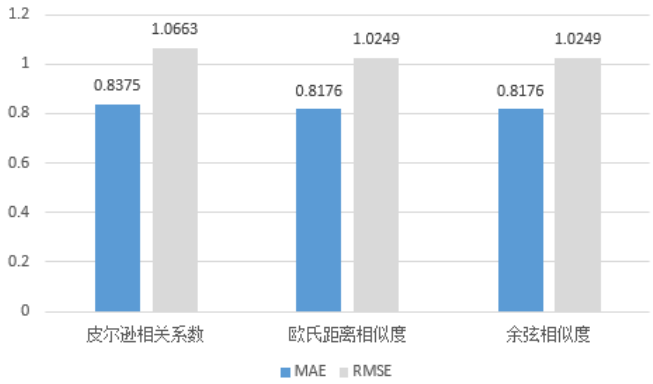


图 6-10 使用阈值邻域的基于用户的推荐算法的推荐效果

从图 6-10 中可以看出，不同的相似度度量在使用最佳阈值的基于用户推荐的算法中都有比较稳定的推荐效果，MAE 稳定在 0.82 左右，RMSE 稳定在 1.04 左右。

6.3.3 基于物品的推荐算法的效果分析

本实验评测了常用的相似度度量在基于物品的推荐算法中的推荐效果，包括皮尔逊相关系数、曼哈顿距离、欧氏距离、余弦相似度和谷本系数等，使用了平均绝对误差（MAE）和均方根误差（RMSE）作为评测指标。数据集为 MovieLens-100k 数据集，训练集:测试集= 8 : 2。实验结果如图 6-11 所示。

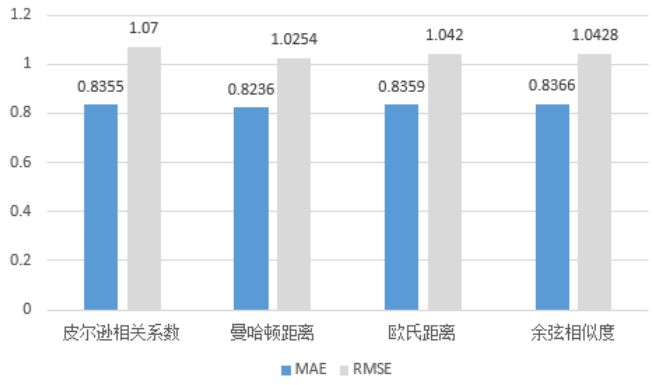


图 6-11 基于物品的推荐算法的推荐效果

从图 6-11 中可以看出，不同的相似度度量在基于物品的推荐算法中都有比较稳定的推荐效果，MAE 稳定在 0.83 左右，RMSE 稳定在 1.05 左右。

6.3.4 Slope One 推荐算法的效果分析

本实验评测了 Slope One 推荐算法的推荐效果，使用了平均绝对误差（MAE）和均方根误差（RMSE）作为评价指标。数据集为 MovieLens-100k 数据集，训练集:测试集=8 : 2。评测结果如图 6-12 所示。

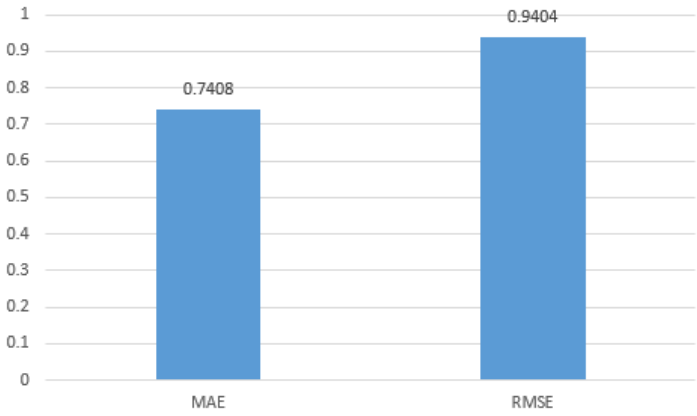


图 6-12 Slope One 推荐算法的推荐效果

从图 6-12 中可以看出，Slope One 推荐的 MAE 在 0.74 左右，RMSE 在 0.94 左右，它们都小于基于用户推荐算法和基于物品推荐算法的 MAE 和 RMSE 值，所以推荐效果要优于以上两个推荐算法。

6.3.5 ALS-WR 推荐算法的效果分析

对 ALS-WR 推荐算法进行效果分析，使用均方根误差（RMSE）作为评价指标，数据集为 MovieLens-100k 数据集，当训练集:测试集=8:2 时，RMSE 大小为 3.2693，误差比较大，这是因为 MovieLens-100k 数据集过于稀疏，在教育资源推荐系统的具体应用中需要注意。

6.4 本章小结

本章首先介绍了推荐系统的开发和部署环境，然后对推荐系统中重要功能进行了演示，包括日志收集和推荐服务接口，最后对不同推荐算法在不同的配置参数下的推荐效果进行了实验测试。

第 7 章 总结与展望

本章将对项目进行总结，总结项目研究成果和系统的实现情况，分析项目所存在的不足之处，并对未来工作进行展望。

7.1 项目总结

本文的研究项目源自于国家自然科学基金支持的关于云教育平台构建若干关键技术创新研究项目，为了解决云教育平台信息过载的问题，设计并实现了一个基于 Hadoop 的教育资源推荐系统。

本文实现的系统依据推荐系统的相关理论，将其分为了三个模块，分别为海量日志收集模块、推荐引擎模块和推荐服务器模块。

为了避免对云教育平台造成代码入侵，海量日志收集模块通过日志埋点的方式收集用户行为日志，可以对平台上多个维度的用户行为进行收集，包括点击、浏览、评分、评论和收藏等，并存储到 Hadoop 的 HDFS 中，并采用了 Nginx、Kafka 和 Flume 等技术，以保证本系统能够应对高负载、高并发、高吞吐的日志收集需求。

推荐引擎模块对用户行为日志进行建模计算，并获得推荐结果。共实现了两种计算方式，分别为离线计算和在线计算，离线计算使用的是 Hadoop 的 MapReduce 计算模型，负责数据量大、耗时长的计算任务，使用了协同过滤的推荐方法，用到了基于用户的推荐、基于物品的推荐、Slope One 推荐和 ALS-WR 推荐等四种推荐算法，本文对 Slope One 推荐和基于用户的推荐两个算法做了改进和优化，使其能够在 Hadoop 平台上并行计算，即 MapReduce 化。而基于物品的推荐和 ALS-WR 推荐则使用了 Mahout 框架的实现。另外，推荐引擎还提供了在线计算的功能，以获取基于教育资源热门度和基于用户信息推荐的推荐结果。本推荐引擎采用了可扩展的方式设计，可以随时加入新的推荐算法。文中还评测了各个算法在遇到不同配置参数情况下的推荐效果，以找到适合本系统的推荐算法和配置参数。

推荐服务器模块对推荐引擎各个算法计算得到的推荐结果进行转存、过滤和处理，设计并封装了一系列接口供云教育平台前端调用，以获得推荐结果，方便推荐系统与研究项目中其他的子系统进行整合。针对不同的推荐场景，设计了根

据用户兴趣推荐、根据教育资源相关性推荐、根据用户信息推荐、根据教育资源热门度推荐等多种推荐方式，不同的推荐方式对应了不同的推荐算法。并且，本文提出了适合教育资源的混合推荐方法，云教育平台可以混合多个推荐算法的结果形成组合推荐。最后，推荐服务器还提供了给用户发送推荐邮件的功能。

7.2 未来展望

本文对推荐系统相关理论进行了深入研究和探讨，结合相关工具和技术，实现了基于 Hadoop 的教育资源推荐系统，经过实验和测试本系统基本上能够满足云教育平台对推荐系统的需求，但由于时间不足条件有限等原因，该系统也存在很多不足之处，需要改进和优化，这也是本文未来工作的重点之处。

（1）本系统主要使用了协同过滤领域的相关算法，此类算法的特点是根据用户的行为日志建模计算并进行推荐，不需要用户或物品的大量信息，可以迅速搭建和应用。而对于教育资源，无论是文档还是视频，其内容中都包含了大量的信息，如果可以充分利用这些信息，可以极大的提高推荐准确度。在未来的工作中，本文将会研究在推荐系统中加入语义分析、图片内容识别、视频内容识别等机器学习领域的技术，结合基于内容的推荐方法，提高推荐准确度。

（2）本系统大部分推荐计算任务都是离线状态下完成的，无法提供实时推荐的功能。因此，本文在未来的工作中，将结合相关技术，给推荐系统加入实时推荐的功能，争取做到用户实时反馈，系统实时推荐。

（3）本文提出了适用于教育资源的混合推荐方法。但由于时间的原因，并没有对推荐算法组合在不同配置参数下的推荐效果进行测试。本文在未来的工作中，将会对其进行详细的测试，以找到最优的算法组合和配置方案。

参考文献

[1] 前瞻网. 2016 年中国在线教育行业市场现状及发展趋势分析[EB/OL]. www.qianzhan.com, 2016-05-25

[2] 国务院. 国务院关于印发国家教育事业发展“十三五”规划的通知[EB/OL]. http://www.gov.cn, 2017-01-19

[3] 蔺丰奇,刘益. 信息过载问题研究述评[J]. 情报理论与实践,2007,(05):710-714.

[4] 项亮. 推荐系统实践[M]. 北京:人民邮电出版社, 2012.

[5] 朱扬勇,孙婧. 推荐系统研究进展[J]. 计算机科学与探索,2015,(05):513-525.

[6] Goldberg D. Using collaborative filtering to weave an information tapestry[J]. Communications of the Acm, 1992, 35(12):61-70.

[7] Resnick P, Iacovou N, Suchak M, et al. GroupLens: an open architecture for collaborative filtering of netnews[C]// ACM Conference on Computer Supported Cooperative Work. ACM, 1994:175-186.

[8] Resnick P, Varian H R. Recommender systems[J]. Communications of the Acm, 1997, 40(3):56–58.

[9] Linden G, Smith B, York J. Amazon.com Recommendations: Item-to-Item Collaborative Filtering[J]. IEEE Internet Computing, 2003, 7(1):76-80.

[10] Miyahara K, Pazzani M J. Collaborative Filtering with the Simple Bayesian Classifier[M]// PRICAI 2000 Topics in Artificial Intelligence. Springer Berlin Heidelberg, 2000:679-689.

[11] Sarwar B, Karypis G, Konstan J, et al. Application of Dimensionality Reduction in Recommender Systems[J]. In Acm Webkdd Workshop, 2000.

[12] Burke R. Knowledge-Based Recommender Systems[J]. 2000.

[13] Sarwar B, Karypis G, Konstan J, et al. Item-based collaborative filtering recommendation algorithms[C]// International Conference on World Wide Web. ACM, 2001:285-295.

[14] Goldberg K, Roeder T, Gupta D, et al. Eigentaste: A Constant Time Collaborative Filtering Algorithm[J]. Information Retrieval Journal, 2001, 4(2):133-151.

[15] Hofmann T. Latent semantic models for collaborative filtering[J]. Acm

Transactions on Information Systems, 2004, 22(1):89-115.

[16] Lemire D, Maclachlan A. Slope One Predictors for Online Rating-Based Collaborative Filtering[J]. Computer Science, 2007:21--23.

[17] Bell R M, Koren Y, ! Y, et al. The bellkor 2008 solution to the netflix prize[J]. Korbell Teams Report to Netflix, 2008.

[18] Koren Y. The bellkor solution to the netflix grand prize[J]. Netflix Prize Documentation, 2009.

[19] Bell R M, Koren Y, Volinsky C. All together now: A perspective on the NETFLIX PRIZE[J]. CHANCE, 2010, 23(1):24-24.

[20] 孟祥武, 纪威宇, 张玉洁. 大数据环境下的推荐系统[J]. 北京邮电大学学报, 2015, 38(2):1-15.

[21] Takács G, Pilászy I, Németh B, et al. Scalable Collaborative Filtering Approaches for Large Recommender Systems.[J]. Journal of Machine Learning Research, 2009, 10(10):623-656.

[22] Gábor Takács, Tikk D. Alternating least squares for personalized ranking[C]// ACM Conference on Recommender Systems. ACM, 2012:83-90.

[23] 潘雪峰,张宇晴,毛敏,崔鹤. 在线教育产业发展现状及产品设计研究[J]. 科技和产业,2013,(08):13-16.

[24] 何克抗. 从“翻转课堂”的本质,看“翻转课堂”在我国的未来发展[J]. 电化教育研究,2014,(07):5-16.

[25] 陈池,王宇鹏,李超,张勇,邢春晓. 面向在线教育领域的大数据研究及应用[J]. 计算机研究与发展,2014,(S1):67-74.

[26] 邢丘丹,焦晶,杜占河. 基于云计算和大数据的在线教育交互应用研究[J]. 现代教育技术,2014,(04):88-95.

[27] 管佳,李奇涛. 中国在线教育发展现状、趋势及经验借鉴[J]. 中国电化教育,2014,(08):62-66.

[28] Dietmar Jannach, Alexander Felfernig, Gerhard Friedrich, 等. 推荐系统[M]. 北京:人民邮电出版社, 2013.

[29] 冷亚军,陆青,梁昌勇. 协同过滤推荐技术综述[J]. 模式识别与人工智

能,2014,(08):720-734.

[30] 施聪莺，徐朝军，杨晓江. TFIDF 算法研究综述[J]. 计算机应用，2009, 29(s1):167-170.

[31] 杨小平，丁浩，黄都培. 基于向量空间模型的中文信息检索技术研究[J]. 计算机工程与应用, 2003, 39(15):109-111.

[32] 高洁，吉根林. 文本分类技术研究[J]. 计算机应用研究, 2004, 21(7):28-30.

[33] Francesco Ricci, Lior Rokach, 等. 推荐系统：技术、评估及高效算法[M]. 北京:机械工业出版社, 2015.

[34] Bridge D, Göker M H, Mcginty L, et al. Case-based recommender systems[J]. Knowledge Engineering Review, 2005, 20(3):315-320.

[35] Felfernig A, Burke R. Constraint-based recommender systems: technologies and research issues[M]. 2008.

[36] Sean Owen, Robin Anil, Ted Dunning, Ellen Friedman. Mahout 实战[M]. 北京:人民邮电出版社, 2014.

[37] Ghemawat S, Gobioff H, Leung S T. The Google file system[J]. Acm Sigops Operating Systems Review, 2003, 37(5):29-43.

[38] Dean J, Ghemawat S. MapReduce: Simplified Data Processing on Large Clusters.[C]// Conference on Symposium on Opearting Systems Design & Implementation. DBLP, 2004:137-150.

[39] Sameer Wadkar, Madhu Siddalingaiah, Jason Venner, 等. 深入理解 Hadoop（原书第 2 版）[M]. 北京:机械工业出版社, 2016.

[40] Tom White. Hadoop 权威指南(第 3 版)(修订版)[M]. 北京:清华大学出版社, 2015.

[41] 殷人昆，郑人杰，马素霞，等. 实用软件工程(第 3 版)[M]. 北京:清华大学出版社, 2010.

[42] Burke R. Hybrid Recommender Systems: Survey and Experiments[J]. User Modeling and User-Adapted Interaction, 2002, 12(4):331-370.

致谢

首先我要非常衷心地感谢我的导师郑贵锋老师。感谢导师在我两年的研究生生涯里对本人的耐心栽培和真心帮助，特别是导师对本人学业的指导，使得本人的学习能力和科研能力都有了很大的提高。本论文是在导师的悉心指导之下顺利完成的，论文从选题到完成，每一步都有着导师的指导和鞭策，倾注了导师大量的心血，在此我要向我的导师郑贵锋老师表示深切的谢意与祝福!

我要感谢李师兄、胡师姐和黄师兄，感谢他们在我整个论文撰写期间的对我的帮助和支持。

我要感谢同实验室的同学们，他们在我整个论文的研究过程中给了我很多的帮助和鼓励，也提出了很多建设性的意见。

我要感谢一直默默支持我的父母和亲人，感谢他们对我的理解、包容和鼓励，使得我能够在追求梦想的道路上一直前行。

最后，我要感谢在这两年的研究生生涯里每一个曾经给予过我帮助的人，正是因为有他们，才使得我的研究生生涯变得无比有意义，无比精彩，无比充实。祝他们，一切顺利，永远健康，永远快乐。