密级：_____

浙江大学

硕 士 学 位 论 文

论文题目 基于多特征排序模型的网络课程推荐算法研究与应用

作者姓名 朱华

指导教师 陈珂

学科(专业) 计算机科学与技术

所在学院 计算机科学与技术学院

提交日期 2017 年 1 月

A Dissertation Submitted to Zhejiang University for the Degree of Master of Engineering

TITLE: Research and Implementation of Online Course Recommendation Based on Multi-Feature Ranking Model

Author: Zhu Hua

Supervisor: Chen Ke

Subject: Computer Science and Technology

College: Computer Science and Technology

Submitted Date: 2017-01

摘要

随着大规模网络开放课程的飞速发展，在线教育这一全新的学习形式开始被越来越多的人所接受。用户通过互联网可以学习到各种领域的知识和技能，但随着在线课程资源数量的增加以及种类的越来越多样化，用户在考虑想要学习的课程时经常会遇到选择难题。推荐算法的引入能够为用户的课程学习提供建议，但由于网络课程存在一些局限性，如文本信息较少、用户行为信息不够丰富、评价信息缺乏等，传统的推荐算法无法直接应用到网络课程的推荐中，需要基于网络课程的独特场景进行创新和改进。

本文中我们在对云课堂用户数据进行了充分分析的基础上，研究并实现了一种基于多特征排序模型的网络课程推荐算法。该算法结合了网络课程及用户相关的多个特征，包括基于主题的用户偏好、基于协同过滤的用户偏好、课程热门度、讲师影响力。通过排序学习的方法对这些特征进行线性组合，计算目标用户与网络课程间的匹配程度，从而为用户进行课程推荐。

为了验证本文算法的有效性，我们在云课堂真实数据集上进行了大量实验，实验证明我们的算法能够得到较好的推荐效果，与参照算法相比有一定提升。另外，我们设计实现了基于云课堂的课程推荐系统，其功能是在云课堂用户个人学习主页的基础上实现的。经测试系统运行良好，验证了本文算法的实用性。

关键词： 网络课程，特征提取，排序学习，推荐算法

Abstract

With the rapid development of Massive Open Online Courses, online education, as a new form of study, has been accepted by more and more people. Users can learn knowledge and skills from different domains through the Internet. But as the amount of online resources grows fast and the courses become more diverse, users will encounter selection problem when deciding which courses to learn. The use of recommendation algorithm can give advices for users' learning options. However, online course has its own limitation, such as the lack of text information, user behavior information and comments. Traditional recommendation algorithms cannot be applied directly to online courses. We need to innovate and improve the algorithm based on the unique scenario of online courses.

In our proposal, after sufficiently analyzing the data of Cloud Class, we research and implement an online course recommendation algorithm based on multi-feature ranking model. The algorithm considers several features related with online courses and users, including the preference based on topic, the preference based on collaborative filtering, the popularity of courses and the influence of teacher. The algorithm combines the features with linear function by using the method of learning to rank. Then it calculates the matching degree of target user and online courses to make recommendation to the user.

In order to verify the effectiveness of the proposed algorithm, we have done many experiments on the dataset of Cloud Class. The experiments prove that our algorithm can get great recommending result and outperform the compared algorithms. Furthermore, we design and implement the recommender system based on Cloud Class, which is mainly used on the personal learning page of users. The system works well and demonstrates the practicability of our algorithm.

Keywords：

Online Course, Feature Extraction, Learning to Rank, Recommendation Algorithms

目录

图目录

表目录

第1章 绪论

1.1 课题背景

随着互联网技术的飞速发展，人类知识的更新周期也开始加速，进入 21 世纪以来，人类知识的更新周期已经缩短为 2 至 3 年。在过去的十年时间里，网络教育资源的数量呈现出指数级增长的趋势，大规模网络开放课程（Massive Online Open Courses，简称 MOOCs）这一全新的教育方式也进入了大家的视野，以 Coursera、Udacity、edX 等为代表的许多机构应运而生。教育是社会关注的重要问题，在线教育的快速发展给高校和个人用户都带来了新的契机，近年来移动互联网技术的发展更促进了在线教育的普及。对高校来说，教育不再是局限于校园内的行为，所有的教育资源都可以分享到网络上，提供给所有渴望知识的用户，这给传统的教学方式带来了极大的影响。对用户来说，生活和学习方式也在悄然改变，教育资源不再有门槛，每个人足不出户就能够体验到世界各地的、各个领域的顶尖教学。而对整个社会来说，在线教育促进了知识的广泛传播，节约了大量的教学资源，可以说真正实现了全民教育。

在线教育的概念在我国也越来越受到广泛的关注，目前正处于快速发展的阶段。《2015 年中国在线教育平台研究报告》显示，2014 年中国在线教育用户规模达到 5999.2 万人，并将以年均将近 20%的速度增长，预计到 2018 年达到 13221.1 万人[1]。在线教育发展的势头正猛，众多的在线教育产品应运而生。其中，网易云课堂是网易公司倾力打造的在线教育平台，其于 2012 年 12 月底正式上线，虽然起步较晚，但其与多家权威教育、培训机构之间建立了合作关系，课程数量已达 10000+，课时总数超过 10 万[2]，涵盖了互联网、外语、金融等十余大门类，百余个细致分类。网易云课堂致力于为学习者提供海量、优质的课程，每一位想真正学到知识技能的用户都能在这里得到一站式的学习服务。

[1] http://www.iresearch.com.cn/report/2490.html
[2] http://study.163.com/about/aboutus.htm#/about?aboutType=1

在线教育的普及使得用户可以在网上学习到任何领域的课程，但如何找到想要的课程是用户面对的一大难题。网络课程的种类非常多，以网易云课堂为例，其部分课程分类情况如图 1.1 所示。云课堂的课程共有三级分类，图中左侧为 7 个一级分类，每个分类下又有 5 到 10 个二级分类，每个二级分类又被细分为若干个三级分类。图中右侧为 IT/互联网大类下的子分类示例，可以看到分类非常细，而且几乎覆盖了相关领域涉及到的大多内容。在目前的分类情况下，用户能够快速准确的找到自己想要的课程，但随着在线课程的进一步发展，分类也会越来越详细、越来越复杂，用户寻找课程的难度是在增加的。除此之外，也存在用户不确定自己的兴趣属于哪个分类的情况，因此单纯依靠分类并不能适应网络课程的发展，也无法满足用户日益增长的需求。

图 1.1 云课堂分类示例

网络上的课程除了种类繁多以外，同样的内容也会有很多不同的课。以网易云课堂为例，如图 1.2 所示，在前端开发的相关课程中，仅 JavaScript 一种语言就有 51 门不同的课程，而且课程的数量还在不断的增长中。查看这些课程的介绍可以发现，教学内容很多都是重复的，用户可能只需要学习其中的一至两门课程即可。面对如此庞大的课程数量，当用户想寻找自己感兴趣的课程时，他需要仔细阅读每一门课程的介绍、教学大纲，然后从中挑选最合适的内容。有时可能

还需要看其他用户对这门课程的评价，来选择质量最高的课。不仅仅是网络课程应用，在很多其他的应用场景中，用户往往习惯于这种主动的信息获取方式，这种方式能够保证用户获取到的信息的质量，但这一过程太过繁琐枯燥，并且会耗费用户大量的时间，在信息爆炸的互联网时代显得效率十分低下。

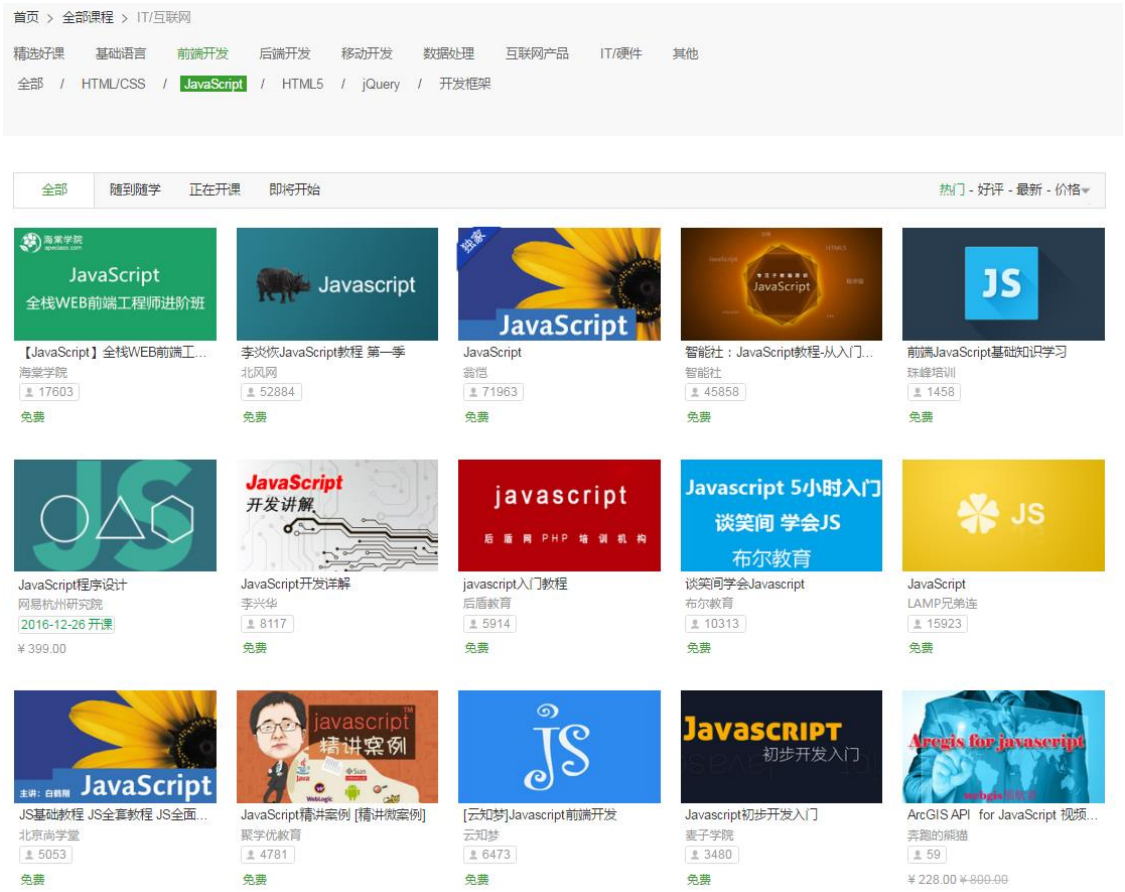

图 1.2 云课堂课程示例

除了课程数量巨大的问题之外，网络课程的内容通常会包括许多形式的多媒体资源，如视频、音频、文档等，这些内容的多样性也无形中增加了用户进行选择的难度。对用户来说多媒体内容往往是比较耗时的，如何从多媒体信息中概括出与课程相关的特征，向用户提供有用的信息，是网络课程搜索中重要的问题之一。信息过载是互联网时代常见的问题，这个问题暂时无法得到彻底的解决，但推荐系统的提出，能够有效地使这一状况有所改善。推荐系统能够根据课程基本信息、其他用户对课程的评价情况、用户曾上过的课程数据等诸多方面的信息，

将用户可能感兴趣的课程推荐给他，针对每一个用户进行个性化推荐，帮助用户进行选择。

推荐系统经过长期的发展已经比较成熟，被广泛地运用在了互联网的各个方面。传统的推荐系统应用场景包括新闻推荐、文章推荐、广告推荐等，这些场景中的待推荐物品大多包含丰富的文本信息，基于内容的推荐系统会根据这些文本信息提取出各个物品的特征，再根据用户过去的行为记录来提取用户的特征，进而根据用户与新物品的特征相似度情况来进行推荐。网络课程通常也包含文本信息，如课程介绍、教学目录等，因此传统的基于内容的推荐算法亦可使用。但与常见场景不同，课程的这些文本信息通常比较少。并且如果两门课程的内容高度相似，那它们的教学内容很可能是重复的，对用户来说他很大可能不会同时选择这两门课程。因此，如果仅依据课程内容此进行推荐，结果可能并不理想。基于协同过滤的算法也是一种常用的推荐方式，它主要考虑了用户与用户之间的相似程度，为用户推荐他的相似用户上过的课程。这种方式克服了基于内容的推荐算法过度专业化的问题，使推荐的结果并不局限于某一领域内，从而能够发掘用户的潜在兴趣。然而基于协同过滤的推荐也有一定的局限性，比如当用户行为数据较少时，无法得到很好的效果，对于没有行为数据的新用户则完全无法进行推荐。另外，在课程推荐的场景中，这种方式也没有考虑到用户主动提供的兴趣信息，如个人职业信息、自定义标签等。因此，为了得到较好的推荐结果，应当针对课程推荐的场景，将各种因素综合考虑来设计算法。

网络课程除了基本的介绍信息之外，还有许多与用户相关的信息，比如上过课的用户数量、评分评论等。这些信息反映了一门课程的质量和热门程度，而这些都会影响其他用户对课程的选择。另外，如今的网络课程应用的功能都非常全面，除了用户的历史行为记录以外，还有很多信息可以获知用户的兴趣偏好。比如对于新用户来说，虽然他尚未浏览、学习过任何课程，但如果在注册时提供了教育、职业、技能等信息，我们便能够对他可能感兴趣的课程进行预测。可以看出，这一特征的引入可以有效的避免传统推荐方法的冷启动问题。除此之外，在网络课程应用中用户还可以去关注其他用户，这些用户可能是普通的学习者，也

可能是课程的讲师。这种社交关系某种程度上也反映了用户的兴趣，可以将用户的好友们上过的课程推荐给他。上述这些影响用户选择的因素都是传统推荐系统中没有的，如何在网络课程推荐的特定场景下权衡不同的特征，得到更好的推荐结果，是相关研究所要解决的重点问题。

综上所述，在线教育在近几年迅速发展，网络课程的数量在飞速增长，并且正在吸引着越来越多的用户，推荐系统的引入将为用户搜索课程、学习课程提供建议和指导，为用户提供更高质量的学习体验。然而由于相关领域的发展，尤其是国内在线教育的发展还处于起步阶段，业界对于网络课程推荐相关的研究和实践工作都仍待完善，我们的研究将弥补这一领域的不足，对工业界和学术界都会有重要的意义。

1.2 本文的工作与贡献

在本文中，我们主要研究的是网络课程的推荐问题，并以提升现有算法的网络课程推荐效果为目标。具体而言，本文所要解决的问题是，给定目标用户和网络课程的集合，根据课程的相关信息，以及用户已有的课程学习情况，计算出用户与未参加过的课程之间的匹配程度，为其推荐合适的课程，并将推荐结果以直观的形式展示给用户。

传统的推荐算法多是以评分预测为目标，而忽视了推荐结果先后顺序的重要性，而在实际应用中，用户往往只会关注推荐列表中靠前的几个项目，因此推荐结果的排序是非常重要的。尤其是在网络课程推荐中，用户与课程之间是学习与未学习的关系，并没有明确的评分数据，因此网络课程的推荐更符合排序问题。另外，用户在选择网络课程进行学习时，可能会受到诸多因素的影响，网络课程本身也具有非常丰富的信息，如何将这些信息整合到一起，设计出更加全面的推荐方法，是相关研究工作的难点。

本文针对现有推荐算法的应用场景局限性进行了改进，设计了一个基于排序模型的网络课程推荐算法，该算法充分考虑了网络课程推荐过程中可能会对结果产生影响的各种重要因素，包括课程文本信息、用户潜在兴趣、课程热门程度、

讲师权威性等。根据课程的文本介绍信息，以及用户的课程历史记录，本文将计算用户对课程的偏好程度。根据每个用户的历史课程学习记录，本文将进一步计算基于协同过滤的用户偏好值。除此之外，本文还将课程的热门程度以及讲师的影响力作为特征进行量化，通过对课程学习人数、评分人数、平均打分的衡量，来加强课程推荐的合理性。在将上述特征量化提取后，本文将推荐问题转换为了排序学习问题，建立了排序模型来生成最终的推荐列表，以对各个特征的影响程度进行综合考量。

为了验证本文算法的有效性，我们使用云课堂的真实数据进行了对比实验，选取了若干传统推荐算法进行比较，并分析了它们在多种不同情况下的表现。最后，我们在云课堂用户个人学习主页的基础上，将课程推荐功能以及用户兴趣标签功能进行了实现，对算法的实用性进行了验证。

概括来说，本文工作的主要内容和贡献如下：

1. 对云课堂真实数据集进行了详细分析，通过从多个角度对数据进行的统计挖掘，发现网络课程及其学习用户的特点，以针对该应用场景设计特定的推荐算法。
2. 提出了基于多特征排序模型的网络课程推荐算法，算法分析了用户在选择网络课程时可能会影响其决策的各种因素，将这些特征量化，建立多特征网络课程推荐模型，将推荐问题概括为排序问题，对传统推荐算法以及现有单一特征的推荐算法进行了改进。
3. 将本文算法运用在云课堂的真实数据集上，并将其与传统推荐算法进行比较，通过对实验结果的分析比较，考量了算法在多种情况下的表现，验证了算法的推荐效果。
4. 设计实现了基于云课堂用户个人学习主页的网络课程推荐系统，系统中应用了本文提出的基于多特征排序模型的推荐算法，实现了包括网络课程推荐以及用户兴趣标签提取的功能。

1.3 本文的组织与结构

第一章说明了网络课程推荐相关研究的背景和意义，概括了本文的主要工作贡献，以及本文的基本框架和组织结构。

第二章介绍了与本文课题内容相关的一些研究工作，包括推荐系统常用算法的介绍，本文涉及到的排序算法的相关介绍，以及与在线教育及网络课程推荐相关的研究工作情况。

第三章针对本文实验使用到的云课堂真实数据集进行了详细分析，从多个角度阐述了网络课程及其用户的特点，为后续算法及功能的设计打下基础。

第四章将详细介绍本文提出的算法，包括算法的整体框架、算法的原理、各个特征的定义以及详细的计算过程。

第五章在云课堂用户学习主页的基础上，使用本文提出的算法设计了网络课程推荐系统，验证了算法的实用性。

第六章展示了本文算法与传统算法进行对比实验的过程，包括实验配置、实验步骤、实验结果，以及对结果的详细分析。

第七章对本文的工作进行了总结，分析了在工作过程中的体会与收获，也反思了工作的不足之处，同时还对今后的研究工作进行了展望。

1.4 本章小结

本章是论文的绪论部分，首先阐述了在线教育及网络课程的发展情况，提出了网络课程推荐相关研究的必要性，然后从整体上概括了本文的主要工作内容，以及本文的组织结构。

第2章 相关工作

本章主要介绍和本文工作相关的研究工作，首先对包括 MOOC 平台调研以及课程推荐算法在内的相关研究进行了概述，然后对传统推荐算法中基于协同过滤的算法以及基于内容的算法进行了简介，最后介绍了目前在推荐系统中常用的排序学习方法，这些研究的介绍是本文工作的基础。

2.1 网络课程相关研究

2.1.1 MOOC 平台调研

随着大规模网络课程（MOOC）的不断发展，涌现出来越来越多的 MOOC 平台，而随着平台中课程数据和用户数据的增多，如何快速寻找到自己感兴趣的课程，成为了用户在使用在线教育网站时面临的一大难题。网络课程推荐的出现为用户的课程发现提供了便利，也成为了在线课程网站必不可少的功能之一。我们将以 Coursera 为例，对在线课程网站的课程推荐功能进行调研。

Coursera[1]是由斯坦福大学计算机系的教授 Andrew Ng 和 Daphne Koller 于 2011 年创立的，创立 5 年以来，Coursera 已经成为了全球范围内规模最大、用户数最多的在线教育平台之一。网站中有来自超过 145 家合作高校的 1600 门课程，学习用户已经达到约 2200 万人次，已经成为了全世界用户进行在线学习的首选平台。Coursera 中的课程专业性较强，分类特征明显，对于用户来说可以比较方便的找到自己感兴趣的课程。但是每门课程的介绍内容都非常丰富，给用户的选择带来了一定的困难，因此课程推荐也是 Coursera 中的重要部分。我们在网站上可以看到，Coursera 课程推荐的主要依据的是用户自主选择的课程目标（如图 2.1 所示）以及课程兴趣（如图 2.2 所示）。课程目标反映了用户使用 Coursera 的目的，课程兴趣则通过细分标签的选择来获取用户的喜好信息。用户可以随时更改这两项的内容，Coursera 会给出不同的推荐结果。

[1] https://www.coursera.org/

图 2.1 Coursera 课程目标选择

图 2.2 Coursera 课程兴趣选择

用户选择课程目标和兴趣后，就可以在自己的个人主页中看到系统推荐的课程，如图 2.3 所示。可以看到结果是按照用户自己选择的分类标签进行区分的，每一个分类下会向用户推荐 5 门课程。这些课程基本上是近期即将开课，或刚刚开课不久的课程。Coursera 中的课程有开课时间限制，同一时间用户可选择的课

程数量并不大，同时网站的用户数量却非常大，因此在课程推荐时并没有使用太多的用户个性化数据，只选用了用户自定义的分类标签特征。而随着网站用户数和课程数的进一步增加，如何为每一位用户进行个性化推荐必将成为网站重点研究的功能之一。

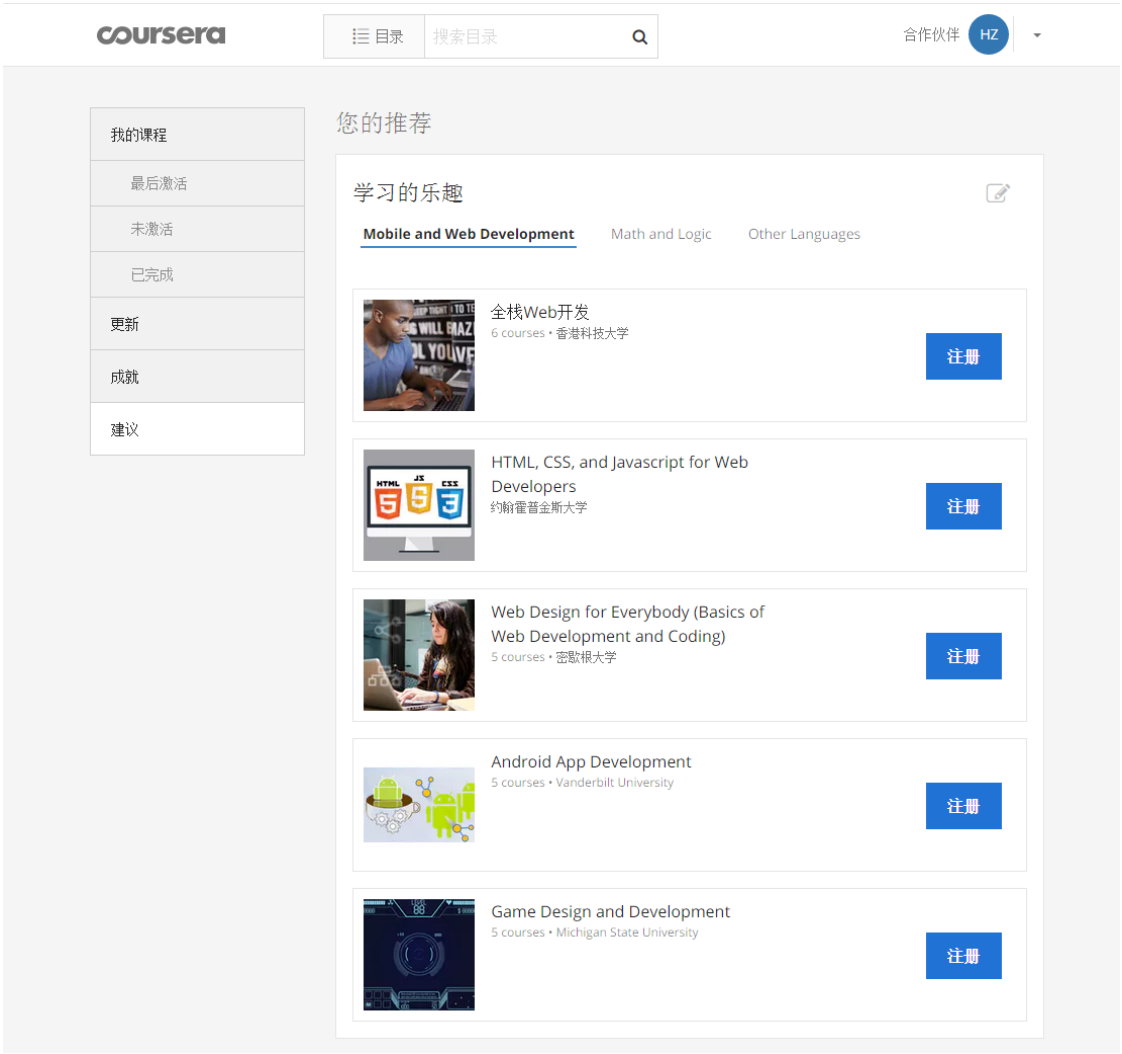

图 2.3 Coursera 课程推荐示例

2.1.2 课程推荐算法研究

与传统的推荐系统应用不同，网络课程推荐有其独有的特点及适用场景，最突出的特点就是内容形式的多样性。绝大多数网络课程都是视频的形式，对于学习者来说，视频形式给课程提供者和学习者都带来了方便，但对于初学者在进行

课程选择时，却很难完整的了解到课程具体的教学内容是什么。这一点也给网络课程的推荐工作造成了一定的困难，一般只能利用课程的标题、简介、分类等文本信息，或对视频内容进行人工标注。

Mozina 等人[18]提出了一种基于线性回归模型的混合课程推荐系统，该系统所做的主要工作是，利用用户学过的课程来给他未学过的课程打分。作者首先通过人工标注的方法将每个视频课程提供的文本信息（发布日期、课程类型、授课语言、课程分类、授课老师、标题中的关键词等）进行分类，打上特定的标签。然后利用这些标签和线性回归模型，可以计算出用户学习过的课程和没有学习过的课程之间的相关度，最后把所有这些相关度放到一个集合中进行排序，就能够得出该用户的推荐课程列表。在文章的最后作者表示，如果使用课程的文本属性的话，系统的推荐效果可以得到进一步的提升。

主题模型是推荐系统中常用的方法，除了较好的推荐效果外，它还能够有效的将高维的用户数据映射到低维空间，从而使得推荐系统的运行效率有明显的提升。Kuang 等人[19]提出了一种基于主题模型的在线课程推荐系统。文章指出，主题模型是文本挖掘中非常通用的模型，在实验及应用中已被证明，在准确性和效率上有很明显的优势。而当把主题模型应用到在线学习系统中时，它将能够把课程的文本信息转化到一个较好的向量模型上，使得高效可行的课程推荐任务成为可能。在主题模型的基础上，文章还表示可以利用系统中的用户访问行为和用户的学习记录建立起用户的兴趣模型，这样系统就可以使用兴趣挖掘技术和主题模型来自动得识别用户的学习兴趣，给用户推荐兴趣相关的课程。而实验结果也表明，基于主题模型的用户学习兴趣模型非常有效，在应用到推荐系统中时也推荐出了很好的相关课程。

MOOC 课程是各大高校将自己学校的教学视频公开，供网络用户共享的教育方式，对于在高校读书的学生来说，他们也可以利用课余时间，通过 MOOC 学习到自己没有选上的课程。高校课程与 MOOC 课程有极强的相关性，根据这一点，Farzan 等人[20]就提出了一种基于学生在学校里的课程成绩历史数据，来推荐相关的网络 MOOC 课程的方法。该方法主要利用了概率主题模型 LDA，首先分

别从学生的大学课程教学大纲，以及 MOOC 课程的教学大纲中建立模型，学习得到各个课程的主题分布情况，得到的课程主题分布会在后面的推荐系统中作为课程的特征向量来使用。论文中提出的推荐系统利用学生大学课程的成绩以及课程的特征向量，通过多元线性回归模型（Multilinear Regression Model）[21]给每个学生训练了一个分类器，通过这个分类器就可以给该学生所有未学的 MOOC 课程进行评分预测，然后优先向学生推荐打分高的课程。实验结果表明，该算法比之前仅基于余弦相似度的推荐算法有更好的表现。

2.2 传统推荐算法

2.2.1 协同过滤推荐算法

2.2.1.1 概述

在网络课程推荐场景中，传统的基于内容的推荐方式有时无法达到很好的效果。一方面是因为网络课程包含的信息大多数是多媒体形式的，如视频、幻灯片等，文本信息通常只有简短的课程介绍，以及教学大纲等，仅仅基于这些特征并不能很好的对课程间的关联进行比较。另一方面，一般用户在进行网络课程学习时，很少会学习两门内容非常接近的课程，而更多的是学习未知的课程。这两方面原因导致基于内容的推荐在课程推荐中并不适用，因此通常主要采用基于协同过滤的推荐方法。协同过滤推荐算法是现在推荐系统中最常用的算法，它的主要思想在于不以物品的内容来预测用户对它的喜好，而是以其他相似用户对物品的评价来进行比较，对于用户行为记录比较丰富的应用来说更加适用。

协同过滤推荐系统的任务通常有两种，评分预测和 Top-N 推荐[13]。其中评分预测指对用户未打分物品的评分进行预测，而 Top-N 推荐的重点则在于找到用户最感兴趣的 N 个物品。在实际应用中，Top-N 推荐是更常用的方式，因为随着网络信息量的爆炸式增长，用户能够获取到的信息越来越多，堆叠式的信息输出方式将不再适合用户的浏览习惯，以重要性排序展示的内容将能更好的满足用户的需求。在本文的课程推荐场景中，推荐结果也将主要以排序列表的形式展示给用户，因此主要研究 Top-N 推荐算法。

协同过滤算法通常会分为基于记忆的（Memory-based）和基于模型的（Model-based）两种[14]，前者直接使用用户和物品的关系数据，通过相似度衡量等方法在用户之间进行比较，而后者则是先根据历史数据离线训练出模型，然后再使用模型进行评分预测。两种方法分别适用于不同的场景，在实际应用中也经常混合使用，以适应更加丰富的条件。

协同过滤推荐算法在实际生活中应用的非常广泛，它总是能较好的捕获用户的兴趣偏好，并且随着用户行为数据的增加，推荐效果会越来越好。但是这种算法也有一定的局限性，主要有以下几点：

- 冷启动问题。当用新用户加入时，由于没有用户行为数据，协同过滤算法将无法为其进行推荐。同样的，当新的物品加入系统时，由于它还没有被任何用户评分过，将无法获得与已存在物品均等的被推荐机会。
- 用户行为数据较少时表现不佳。在课程推荐应用中，由于课程学习耗时较长，大多数用户学习过的课程数量是非常少的，所以推荐系统所能收集到的用户行为数据十分有限，而协同过滤推荐非常依赖用户数据，在这种情况下往往就无法达到很好的效果。
- 稀疏性问题。协同过滤推荐主要依赖用户-物品矩阵进行计算，然而在实际应用中用户数量和物品数量都是非常庞大的，矩阵维度在不断增加，与此同时矩阵的稀疏性也在扩大，从而导致计算的空间和时间复杂度越来越大，给推荐系统的性能带来极大的挑战。

2.2.1.2 相关算法

在协同过滤 Top-N 推荐中最常用的是邻居模型，它最早是由 Goldberg 等人[17]提出并在邮件过滤系统中使用。最常用的邻居模型是 K-最近邻（K-Nearest Neighbors）模型，主要指使用 K 个最相似物品或用户的评分进行加权，来预测用户对未知物品的评分，根据预测评分的排序来推荐前 N 个物品。邻居模型通常分为基于物品的（Item-based）和基于用户的（User-based）两类。

基于物品的协同过滤算法认为用户的兴趣偏好是比较固定的，与用户喜欢过

的物品比较相似的物品更容易被用户所喜欢。这一思想与基于内容的推荐方法类似，但在这里物品与物品之间的相似情况衡量并不是依据物品的内容，而是依据整个系统用户-物品矩阵来计算，也就是根据所有用户对物品的喜好情况来计算相似度。得到物品之间的相似度关系后，算法将选取与目标物品最相近的 K 个物品，通过目标用户对这些物品评分的加权来得到目标物品的预测评分，预测评分最高的 N 个物品就是最后的推荐结果。

基于用户的协同过滤算法与基于项目的类似，不同之处在于它选取了跟用户最相似的 K 个用户，以他们对目标物品的评分，以及他们与目标用户的相似度值来进行加权，最后得到目标用户对目标物品的预测评分，同样取预测评分最高的 N 个物品作为推荐结果。这种方法在用户行为数据较多的情况下，能够有非常出色的表现。

最近邻算法的关键步骤在于相似度的计算，以基于用户的近邻算法为例，相似度表示了目标用户与其他用户之间兴趣偏好的一致性，品味越相近的用户，其选择的物品也更容易被目标用户所接受。相似度的计算方法有很多种，其中最常用的是余弦相似度和皮尔森相似度，其计算公式分别如下：

$$\omega_{pearson}(\mathrm{u},\mathrm{v}) = \frac{\sum_{i\in I}(r_{u,i}-\bar{r}_u)(r_{v,i}-\bar{r}_v)}{\sqrt{\sum_{i\in I}(r_{u,i}-\bar{r}_u)^2}\sqrt{\sum_{i\in I}(r_{v,i}-\bar{r}_v)^2}} \tag{2.1}$$

$$\omega_{cosine}(\mathrm{u},\mathrm{v}) = \frac{r_u \cdot r_v}{\|r_u\|_2\|r_v\|_2} \tag{2.2}$$

在推荐问题中，若用户对物品的喜好是以打分来评判的，不同用户之间就可能出现打分范围的差异，部分用户的评分会普遍偏高或偏低。举例来说，假设有 4 个物品，两个用户对它们的评分分别为(3,3,4,2)和(4,4,5,3)，虽然分数的绝对数值完全不同，但其实用户对这些物品的相对偏好情况是一致的。在这种情况下，皮尔森相似度能够较好的消除评分噪音，正确反映出用户偏好的相似情况。而在没有用户评分的情况下，比如本文的网络课程推荐问题中只有学习和未学习两种状态，余弦相似度能够通过计算向量空间夹角余弦值来进行相似度评判。

2.2.2 基于内容的推荐算法

2.2.2.1 概述

基于内容的推荐算法最早起源于信息检索领域，其推荐过程通常包括三个步骤。首先对物品的特征进行提取，以结构化的形式对多样的物品原始信息进行表示，比如常用的以词向量进行文本表示，由于物品信息与用户行为无关，通常是保持不变的，因此这一步骤可以离线进行。算法的第二步是从目标用户的历史数据中，根据用户对物品的偏好程度建立用户的特征表示，在实际应用中用户行为在不断变化，其特征也需要实时更新。在得到了物品和用户的特征之后，将使用相似度计算方法得到与用户最相似的若干物品，作为结果推荐给用户。

在基于内容的推荐中最重要的步骤是物品特征以及用户特征的抽取，物品特征抽取常用到的方法有以单词权重进行文档表示的 TF-IDF 方法，以及隐含语义分析的主题模型方法。在用户特征表示中则通常采用机器学习的方法，从用户历史行为数据中抽象出偏好模型。用户特征的提取实际上是一个有监督的分类问题，可以用到机器学习中的分类算法，如最近邻算法、决策树算法、线性分类算法等。

在实际应用中，待推荐物品往往拥有丰富的属性信息，这些信息能够充分反映一个物品独有的特征，而将用户喜爱物品的特征综合起来，就能概括出用户的偏好特征。基于内容的推荐正是根据物品自身特征的相似度来进行的推荐，因此这种方法在用户行为数据较少的情况下也能够得到较好的效果。另外，基于内容的推荐算法具有较好的解释性，且方法对于新物品也比较友好，新物品能够得到与已有物品同等的被推荐机会。但是基于内容的推荐也有一定的局限性，主要有以下几点：

- 内容分析的局限性。内容推荐的技术主要对物品的文本内容进行分析，但随着待推荐物品的多媒体信息越来越丰富，对于这些特征的自动提取在技术上有一定的难度，人工标注则会消耗大量资源。
- 无法发掘用户的潜在兴趣。基于内容的推荐只能发现和用户已有兴趣相似的物品，其结果往往局限在某几个特定的领域，而无法为用户发现新的可能感兴趣的资源。

2.2.2.2 主题模型

主题模型（Topic Model）是在自然语言处理中使用的一种机器学习模型，主要用于发现目标文档中的潜在主题，并通过这些主题对文档进行标注。其基本思想认为在一篇文档中频繁出现的一些词语，能够代表文档的中心思想。主题模型就是使用数学框架对文档中出现的单词进行分析，判断文档包含哪些主题，以及各个主题分别占有的比例。主题模型是一种混合概率模型[11]，它把文档看作若干不同主题的概率分布，而每个主题又是词汇表中单词的条件概率分布组成。主题模型不仅能够挖掘出文档的潜在主题，更能对表示文档的维度进行压缩，它最初主要用于自然语言处理方面，现在随着不断的发展被延伸到了很多领域，比如图像处理、生物信息学等。

主题模型最早是 Christos 等人[15]于 1998 年引入的隐式语义索引（Latent Semantic Indexing，简称 LSI），它主要通过对文档进行奇异值分解，把高维向量空间模型表示的文档映射到了低维的语义空间中，但是 LSI 其实并不是一种概率模型。1999 年 Hofmann[16]在 LSI 的基础上引入了概率信息，提出了概率隐式语义索引（probabilistic Latent Semantic Indexing，简称 pLSI）。pLSI 通常被认为是第一个真正意义上的主题模型，但它还存在参数规模增加以及主题比例无法确定等问题。针对这些问题，Blei 等人[12]在 pLSI 的基础上提出了隐含狄利克雷分布（Latent Dirichlet Allocation，简称 LDA），其建模表现出色且计算复杂度相对较低，近些年得到了广泛的研究和应用。

LDA 是一种非监督的机器学习算法，主要用来对规模较大的文档集进行主题提取，它是最常用的一种主题模型。LDA 的生成过程主要有三个步骤，首先针对每一篇文档，从给定主题分布中抽取一个主题，然后从上一步中被抽到的主题所对应的单词分布中抽取一个单词，最后重复以上两步直到遍历完文档中的每一个单词。LDA 建模时的核心问题是获取主题的关键词分布，以及获取目标文档的主题分布。有了文档的主题分布后，也可以进一步根据用户对文档的偏好情况，生成用户兴趣的主题分布。这种方法能够高效地概括出最具代表性的物品特征，从而在推荐系统中能够得到比较好的效果。

LDA 算法在课程推荐中也得到了应用，Apaza 等人[5]提出了一种基于 LDA 的在线课程推荐方法，使用隐藏主题来表示课程内容，并且将用户在学校课程中的得分看做其对课程的偏好评分，证明了 LDA 在课程推荐中的实用性。

2.3 排序学习

2.3.1 概述

在传统的推荐算法中，往往会得到若干个推荐结果，然而物品间的重要性对比并没有体现出来，这些算法虽然能够达到较好的评分准确率，但却并没有得到排序结果。在实际的应用中，用户看到的往往是一个推荐列表，而他们重点关注的很可能只有列表靠前的几项，因此推荐结果的排序是非常重要的。为了解决这一问题，研究人员开始试着在推荐算法中融入排序学习（Learning to Rank）的技术，用机器学习的方法对多个排序模型的权重参数进行学习，从而训练出最佳的模型组合，得到更好的推荐效果。

关于排序学习最早的研究工作是 Norbert[25]使用的最小二乘回归方法，他使用这种方法来学习评分函数，然后根据评分大小进行排序。到了 2000 年以后，相关研究工作开始快速发展，尤其是微软亚洲研究院的研究小组提出了几个排序学习算法，奠定了该领域发展的基础。算法包括基于支持向量机的 Ranking SVM[26]、使用梯度下降训练神经网络的 RankNet[27]、使用提升策略的 RankBoost[28,29]等。随着研究的进一步发展，出现了越来越多的排序算法，根据输入样例的不同，排序学习方法一般可分为以下三类[8]：

- 点级方法。传统的基于协同过滤的推荐方法就是基于点级（Point-wise）的方法，这种方法的推荐过程一般是先对每个物品进行评分预测，然后根据评分结果降序排列得到推荐列表。这种方法实际上把排序问题转换为了回归问题，系统根据训练集得到回归函数，算法的重点在提升单个物品的评分准确率上，并没有体现出排序学习的优势，因此在实际应用中较少使用。
- 对级方法。对级（Pair-wise）方法将排序问题转换为了二元分类问题，

考虑了物品对之间的偏序关系，其侧重点在于对物品的顺序关系进行合理判断。这种方法在训练过程中的目标是，判断任意两个物品对是否满足顺序关系，即物品 1 是否应该排在物品 2 前面。但是这种方法有时会丢失有关偏序程度的信息，比如强相关和不相关物品之间的相对关系是同等的，输出结果只考虑了相对关系而没有考虑在最后排序列表中的位置情况。这种方法通常可以输出较好的结果，且模型训练的复杂度比较适中，在实际应用中较常使用。

- 列表级方法。列表级（List-wise）方法直接对整个排序列表进行优化，主要有两种优化方式，直接对排序的评价指标进行优化，以及构造损失函数进行优化。这种方法没有将排序学习转化为分类或回归问题，且完整地保留了物品之间的排序结构，所以一般会得到比较好的结果。但是实际使用中模型训练的复杂度太高，会导致系统性能的降低，在效率和性能之间需要进行平衡。

2.3.2 相关算法研究

Ranking SVM[22,23]是一种常用的对级（Pointwise）排序算法，在给定查询条件 q 的前提下，假设有三篇文档的相似度关系为 d1>d2>d3，也就是文档 d1 比 d2 相关，文档 d2 又比 d3 相关。为了应用机器学习的方法来进行排序任务，可以考虑将排序问题转化为一个分类问题。假设 x1、x2、x3 分别是文档 d1、d2、d3 的特征向量，可以定义全新的训练样本，令 x1-x2、x1-x3、x2-x3 为正样本，令 x2-x1、x3-x1、x3-x2 为负样本，这样就可以通过训练一个二分类器（支持向量机）来对这些新的训练样本进行分类。

Ranking SVM 在文档对的分类任务中应用了 SVM 的方法，给定训练集中的 n 个查询$\{q_i\}_{i=1}^{n}$，对应的文档对$\left(x_u^{(i)}, x_v^{(i)}\right)$和分类结果$y_{u,v}^{(i)}$，Ranking SVM 的数学表达式如下，其中使用了线性评分函数$f(\mathrm{x}) = \omega^T$：

$$\begin{aligned} &\min \frac{1}{2}\|\omega\|^2 + \mathrm{C}\sum_{i=1}^{n}\sum_{u,v:y_{u,v}^{(i)}=1}\xi_{u,v}^{(i)} \\ &\text{s.t.} \quad \omega^T\left(x_u^{(i)} - x_v^{(i)}\right) \geq 1 - \xi_{u,v}^{(i)} \ \text{, if } y_{u,v}^{(i)} = 1, \\ &\xi_{u,v}^{(i)} \geq 0, \mathrm{i} = 1, \ldots, \mathrm{n}. \end{aligned} \tag{2.3}$$

从上述公式中我们可以看到，Ranking SVM 表达式中控制模型复杂度的边际项$\frac{1}{2}\|\omega\|^2$和 SVM 中的完全一样，而它和 SVM 的主要区别在于，Ranking SVM 中的限制条件是由文档对来提供的，其中的损失函数则是由文档对提供的合页损失（hinge loss）。举例来说，给定一个特定的查询 q，如果文档x_u被认为比文档x_v更相关（也就是$y_{u,v} = +1$），那么如果$\omega^T x_u$比$\omega^T x_v$大 1，就不会带来损失，否则将会带来$\xi_{u,v}$的损失。

由于 Ranking SVM 是在基于 SVM 的基础上提出的，所以它充分继承了 SVM 的很多优良特性，比如在最大化边际值的帮助下，Ranking SVM 可以有很好的概括性。另外在 SVM 中常用的一些技巧也可以应用到 Ranking SVM 中，以此来优化一些非线性的问题。

在 Ranking SVM 的基础上，Sun 等人[37]将这种方法融合到了混合推荐系统中，提出了一种用词袋模型对每个用户和物品分别提取关键词构造特征向量的 LRHR 模型。该模型引入了两种不同的打分机制，低频词的权重采用了词频 TF，而高频词的权重则采用词频取对数的变形，进而基于得到的特征向量来计算用户-物品评分。Liu 等人[38]提出的方法是根据用户对物品的排序来计算得到目标用户的近似用户，然后根据近似用户的偏好对目标用户进行推荐。作者选择用 Kendall 排序相关系数[39,40]来计算用户之间物品评分交集的近似度，从得到与目标用户有相似偏好的用户集合。除此之外，Rendle 等人[41]指出，如果一个用户 u 访问了物品 i，那么相较于其他没有被该用户访问的商品来说，可以认为该用户更加喜欢物品 i。假定用户共同浏览过的商品对之间不存在偏序关系，且假定用户都没有浏览过的物品对之间也不存在偏序关系，那么用户-物品访问记录矩阵就可以转换为物品对

之间的偏序关系矩阵。在此基础之上，作者提出了基于贝叶斯的个性化排序算法LEARNBPR，该算法的主要目标是在已知参数向量的基础上，获得商品排序的后验概率最大化。LEARNBPR 算法首先在学习阶段进行训练，以得到最优的参数向量，然后在测试阶段对目标用户推荐商品。

2.4 本章小结

本章主要介绍了与本文工作相关的一些算法及研究情况。首先作为参考，对网络课程相关的研究工作进行了介绍。然后介绍了传统推荐算法中基于协同过滤的推荐和基于内容的推荐，对推荐的过程、优缺点，以及应用情况进行了描述。最后介绍了排序学习的相关概念，以及其在推荐系统中的应用。本章介绍的这些内容是本文研究工作的基础。

第3章 网络课程相关数据分析

本章将对网易云课堂的真实用户数据进行分析，挖掘用户学习行为中的潜在规律，为更好的设计推荐算法、满足用户需求做铺垫。

3.1 数据集描述

我们在研究中使用了云课堂的部分真实数据，数据包含了用户信息、课程信息、学习记录、讲师信息，各项目的数据量统计如表 3.1 所示。

表 3.1 数据集信息统计

内容	数值
用户数	10023
课程数	21080
讲师数	22482
用户学习记录数	2470517
用户收藏记录数	110044

数据集包含课程基本信息、课程章节信息、用户学习记录、用户收藏记录、课程创建者信息、用户基本信息 6 张表，数据表之间的关系如图 3.1 所示。

其中课程信息表（course）中的 tags 字段是课程的分类标签，一般按顺序列出三级分类信息，未分类课程将给定默认值“其他”。rating 字段为课程的平均评分，分数范围为 0 到 5，保留一位小数。

在章节信息表（course_chapter）中，每一门课会对应若干个章节，而每个章节又会有若干个课时，id 字段为该表的主键，唯一标记每一个课时。

学习记录表（learn_record）中的 stamp 字段为用户开始学习该课程的时间，count 字段为学习过的课时数，对应章节信息表中的课时。

用户信息表(user_info)中的 skill 字段是用户自己在个人信息中填写的技能。

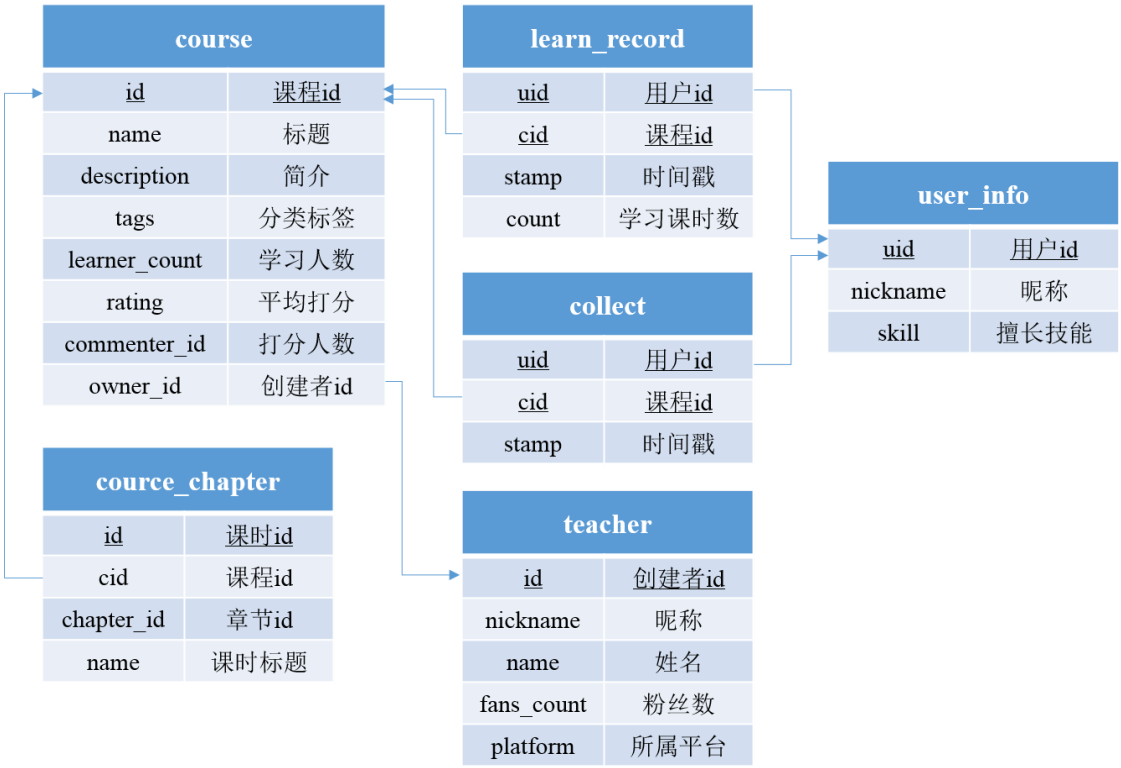

图 3.1 数据库表关系图

3.2 课程信息分析

云课堂的通用课程按标签被划分为 7 个一级分类，包括 IT/互联网、职场/金融、兴趣生活、考试认证、语言/留学、设计创作、中小学，以及默认的“其他”分类。每个分类下又会有若干个二级分类，以 IT/互联网为例，其二级分类包括基础语言、前端开发、后端开发、移动开发、数据处理、IT/硬件、互联网产品等。网络课程的分类是非常复杂的，这给用户寻找所需课程带来了一定的困难。

在所有的课程中，学习人数最多的一门课有 284933 名学生，我们按照学习人数进行排列，选取了学生较多的 5000 门课程来统计，其一级分类情况如图 3.2 所示。从图中可以看到，在网络课程学习中，更受欢迎的是与职场、互联网相关的课程。而对这些课程的二级分类进一步统计发现，涉及课程最多的分类有设计软件、办公软件、职场技能等，二级分类中排名靠前的几类如图 3.3 所示。通过以上分析我们可以发现，大多数网络课程用户更多的是通过在线课程学习来拓宽自己的技能树，更倾向于实用性技能的学习。

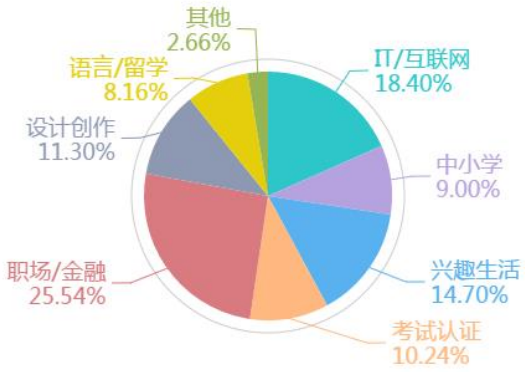

图 3.2 热门课程一级分类情况

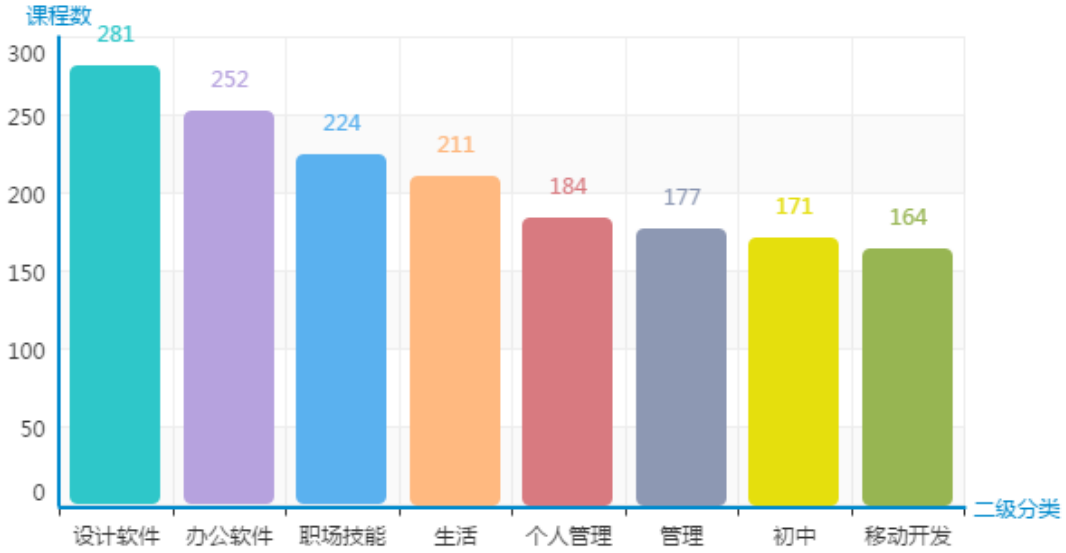

图 3.3 热门课程二级分类情况

不同分类下的学习人数有所不同，我们对 7 个一级分类下课程的学习人数进行了统计，结果如表 3.2 所示。可以看到，不同分类下课程的学习人数有较大差别，用户在选择课程时可能会倾向于选择热门的课程，但他们更多的会关注同类课程学习人数相对的多少，而不是某门课程人数绝对的数值，不同类别下课程人数的比较是没有意义的。

表 3.2 课程一级分类学习人数统计

一级分类	课程总数	课程平均学习人数	课程最大学习人数
IT/互联网	2279	3029	155755
中小学	1670	788	30285
兴趣生活	1729	5327	252713
考试认证	2435	614	67539
职场/金融	4083	2675	166528
设计创作	1913	2361	217773
语言/留学	1181	4189	284933

除了学习人数之外，与课程相关的重要信息还包括课程的评分。云课堂的评分采用 5 分制，我们在统计中发现，大多数用户并不是很喜欢对课程情况给出反馈，据统计平均只有约 5%的学习者会对课程进行评分。而在这其中还包括了一些刷分行为的存在，由于云课堂并没有对评分权限做限制，所有人都可以对课程进行评分，导致很多课程没有人学习过，但却打了很多 5 分，这种情况给普通用户对课程质量的正确判断造成了干扰，真正有效的评分应该是建立在足够的学习人数和打分人数之上的。

3.3 课程创建者分析

云课堂课程的创建者通常为机构或个人，机构大多是一些网络教育机构，如沪江、新航道、oeasy 等，个人可以是独立的用户，也可以是大学里的教师。据统计，在云课堂通用课的创建者中，拥有课程数最多的有 529 门课，最少的有 1 门课，平均每个创建者拥有 7 门课程。在用户选择学习课程的过程中，除了考虑课程本身的热门程度及质量以外，对于这门课程创建者的权威性也会有所考虑，因此我们选取了开课数大于 3 的创建者们，对其拥有的课程的学习人数以及评分情况进行了分析，部分统计结果如表 3.3 和表 3.4 所示。

表 3.3 课程平均学习人数最高的创建者

创建者用户名	课程数	总人数	平均人数	代表课程
钟百迪摄影课堂	4	241475	60368	循序渐进学摄影（初级篇）
翁恺	7	373306	53329	HTML5 入门
茶树网学院	7	279617	39945	CS5 CS6 CC 零基础 PS 教程 PS 学习
布尔教育	6	227897	37983	8 小时学会 HTML 网页开发
任杰 Jack	6	205322	34220	精看电影学英语【公开课】

表 3.4 课程平均评分最高的创建者

创建者用户名	课程数	平均分	平均打分人数	代表课程
茶禅一味教育	5	4.890	190	K 线图入门知识
李兴华	4	4.885	397	Oracle 从入门到精通
小黑老师 online	5	4.882	511	那些你不知道的 Word 高手技巧
李先知	6	4.880	913	项目管理 PMP 培训
妙味课堂	4	4.867	197	JS 基础课程

表 3.3 是课程平均学习人数最多的 5 名创建者，由于可能会有异常情况的存在，我们剔除了学习人数为 0 的课程。表中可以看到平均人数都在 3 万以上，最多的更是达到了 6 万人，说明这些用户创建的课程都是非常热门的。除了热门程度之外，课程质量也是用户很关心的内容，表 3.4 是课程平均评分最高的 5 名创建者，由于评分中有干扰项的存在，打分人数较少的评分不具代表性，我们在排序时限定了评分人数大于 100 的课程进行平均，可以看到表中的创建者课程评分都比较高。总体来说我们认为，平均学习人数较多、平均评分较高的创建者相对更具权威性，其课程更应该被推荐给用户。

我们对讲师数据进行进一步分析，以表 3.3 中的翁恺老师为例。我们知道他是浙江大学计算机学院的老师，所开课程多为计算机方向的基础课，其课程质量非常高。从表中可以看到他的课程学习人数非常多，并且他的课程平均打分也达

到了 4.84，可见用户对他的教学水平很信任。但是我们在表中也发现翁恺老师有一门课程学习人数尚不足 100，与他的其他课程有较大差距，经查证发现在获取数据集时该课程刚刚开设，因此学习人数还在快速增长中。对于这样的课程，虽然学习人数、评分人数、评分数据都较少，但因为课程讲师的权威性较高，对课程的质量有所保证，在推荐中也应该获得较高的权重。

3.4 用户信息分析

云课堂中有课程收藏的功能，用户收藏的课程也能从某种程度上反映用户的偏好。我们对用户收藏课程的数量进行了分析，结果如图 3.4 所示。有将近一半的用户收藏数在 5 以下，且数量越多的用户数量也越少。偏低的收藏数据会给我们的课程推荐带来一定的影响，尤其是基于相似用户进行的推荐，因此在实际推荐中应该考虑多方面因素，避免单一的推荐方法。

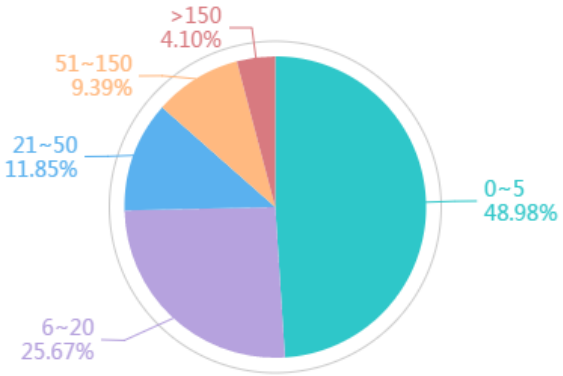

图 3.4 用户收藏课程数统计

在网络课程学习中，由于每门课程的学习花费时间较长，很多用户学习的课程数是比较少的。在推荐算法中需要充分考虑到这一问题，为这些用户制定合理的推荐方案。我们从数据集中选取了学习课程数小于 5 的用户，共 1693 名，这些用户所产生的学习记录有 3252 条，涉及到了 1549 门不同的课程，说明这些用户的选课有很多重叠的部分。在继续调查了重叠课程与热门课程（仅按学习人数多少排序）的关系后，我们得出了如图 3.5 的结果，其中横轴表示热门课程排名，纵轴表示这些课程覆盖的用户数（指前述课程数小于 5 的用户）。从图中可以看

到，对于学习数比较少的这些用户来说，很多人会倾向于选择热门课程进行学习，因此根据课程热门程度进行的推荐对他们将有较好的效果。

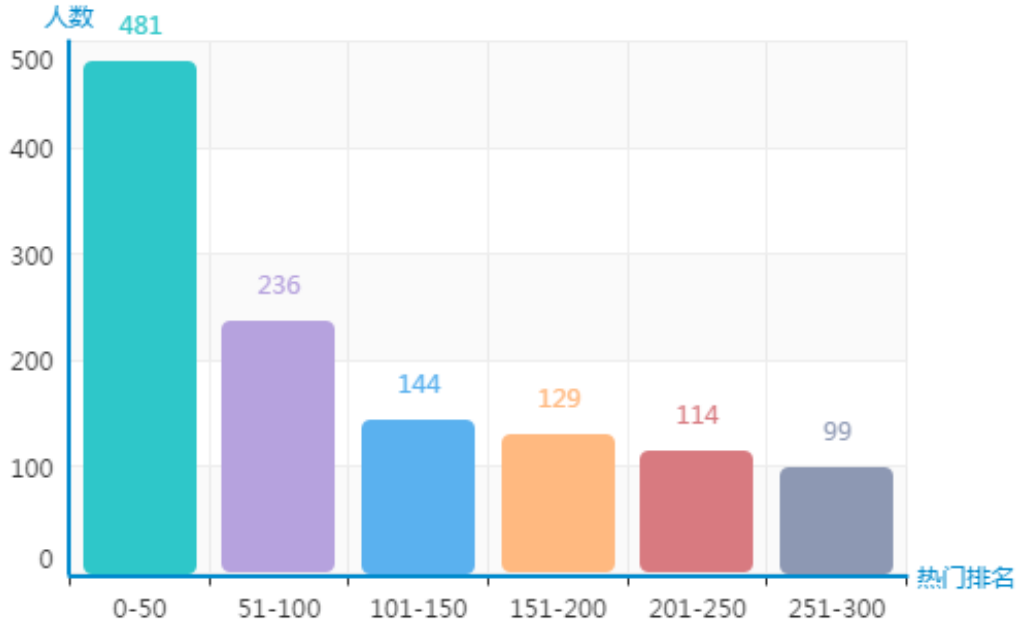

图 3.5 学习数较少的用户与热门课程的关系

另外，我们对用户学习课程的时间进行了一些研究，选取了 1 万名用户的学习记录，统计了一天中各个时段的学习人数，统计结果如图 3.6 所示。

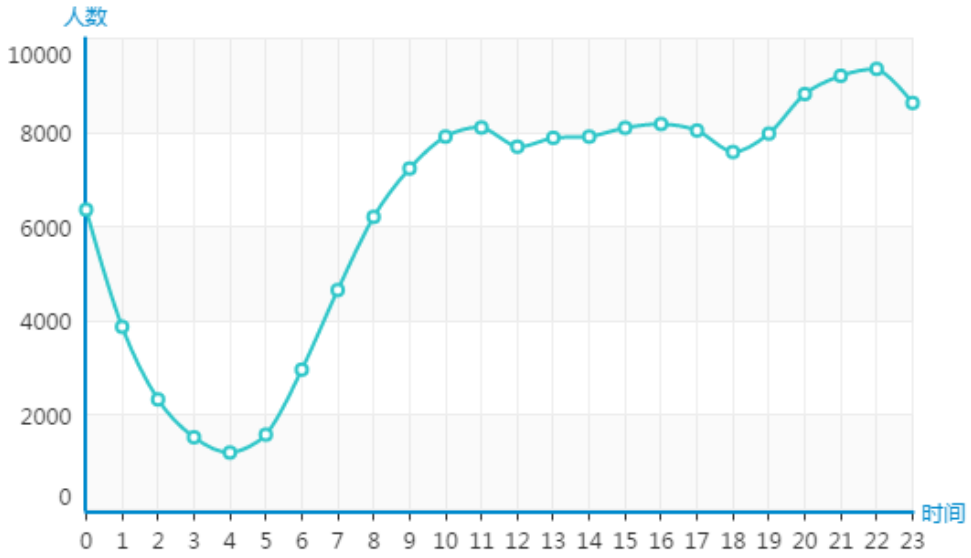

图 3.6 用户学习时间分布统计

可以看到时间分布基本与作息情况一致，而晚上 8 点到 11 点是用户活跃度最高的时刻，到凌晨 12 点以后人数才逐渐下降。活跃时间的偏晚说明云课堂的用户大多数是年轻人，白天学习的主要以学生为主，晚上学习的大多是有工作的社会人士等。

3.5 本章小结

本章从多个角度对云课堂的课程及用户数据进行了分析，通过一系列统计数据和图表的展示，说明了网络课程推荐应用场景独有的特点。从这些分析中我们可以看到，课程的热门程度、讲师的权威性都会影响用户对课程的选择，在后文设计推荐算法时应当对这些信息进行考量。

第4章 基于多特征排序模型的网络课程推荐算法

本章将详细介绍本文所设计的推荐系统用到的算法，算法将充分利用网络课程场景中包括的诸多信息，通过特征提取和学习排序两个步骤，实现为用户进行个性化课程推荐的目的。

4.1 概述

在用户进行在线网络课程学习的过程中，将涉及到诸多内容信息。包括课程相关的信息，如文本描述、课程难度、学生活跃度、课程的讲师情况等。以及用户相关的信息，如课程学习时间、学习进度、课程收藏情况、用户社交关系等。本文将利用这些信息，为不同的信息设计各自的特征提取方法，再使用排序学习框架将提取的特征整合起来，最终得到网络课程的推荐结果。另外，本文算法还将利用基于主题的用户偏好信息，生成用户的兴趣标签，详细算法处理流程如图 4.1 所示。

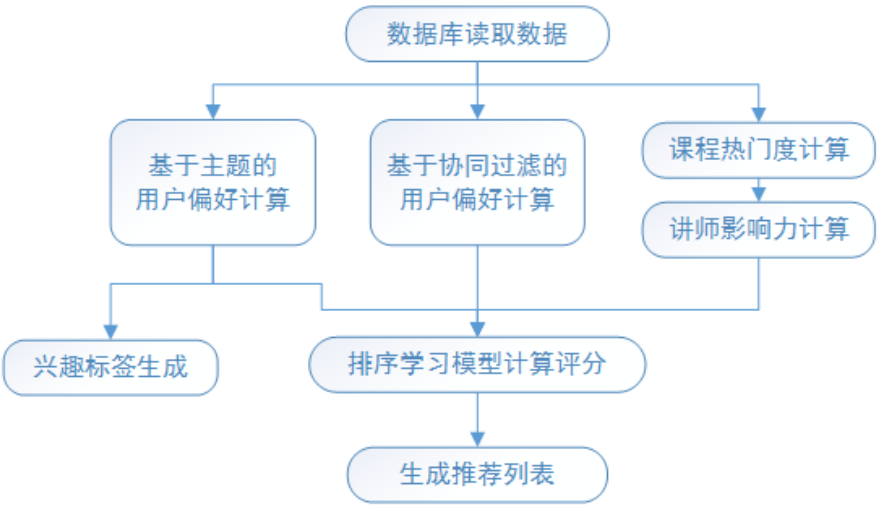

图 4.1 推荐算法流程图

算法的课程推荐部分包括特征提取和排序学习两个步骤。在特征提取阶段，我们首先将根据用户的课程学习记录、课程的描述及章节信息来构造主题向量空

间，根据抽象出的课程特征向量以及用户特征向量，将用户对课程的偏好关系对应为向量的相似度值。接下来我们使用原始的用户-物品学习关系矩阵计算出用户间的相似度，采用基于用户的协同过滤方法预测出用户对物品的评分，将其作为特征之一。为了有效缓解推荐算法中经常会遇到的冷启动问题，我们又对课程的流行程度进行了特征抽取，考量了课程的热门度以及评价情况。并基于课程热门度，对讲师的影响力进行了计算。在排序学习阶段，为了合理的利用前一步骤中获得的各个特征，我们使用排序函数将其整合到一起，将推荐转化为排序学习问题，采用对级排序方法来生成最后的推荐列表。

在用户兴趣标签生成步骤中，我们使用了课程的二级分类作为标签内容，通过标签与用户间特征向量相似度的计算，得到最能够描述用户课程偏好的若干个主题，构造每个用户的个性化兴趣标签。

4.2 特征提取

在网络课程推荐的过程中，有诸多因素可以作为影响推荐结果的特征。由于在线课程系统中存在大量的用户行为数据，且随着时间的推移行为数据会越来越丰富，因此非常适合使用基于协同过滤的的偏好特征。而作为推荐算法研究中应用最广、发展最成熟的方法，基于协同过滤的算法也通常能够得到非常好的推荐结果，并且运算效率也比较高。因此，我们在进行网络课程推荐时将主要参考基于协同过滤的用户偏好。

在用户学习网络课程的过程中，由于课程的学习时间较长，大多用户学习过的课程数都是非常少的，另外还有很多新加入的用户完全没有学习记录。由于用户历史数据不足，单纯的基于协同过滤的推荐可能无法得到较好的效果，而基于内容的推荐算法则能够较好的解决冷启动问题。网络课程中包含了多种文本信息，包括课程标题、简介、章节信息。这些信息内容较短，但却能够非常准确的反映课程的特点，尤其是章节信息，简短明确的对课程内容进行了概括。对课程主题以及用户主题的提取能够反映用户对某一类课程的兴趣偏好，也将作为我们进行推荐时的参考依据。

对于网络课程的学习用户来说，无论是新加入的用户还是学习过很多课程的老用户，当他在选择一门新的课程进行学习时，往往会发现有很多课程内容非常相似，这种情况下用户通常会查看课程的评价情况，选择质量较好、相对更热门的课程进行学习。云课堂网站中每门课程都显示了总学习人数、用户评分、打分人数、评论人数，以及所有的用户评论，这些信息都会对用户的课程选择造成影响，我们在推荐中也需要对课程质量进行综合评价。在这些因素中，课程的评论人数并不能代表课程的质量，因为评论的内容有好评和差评，需要通过查看评论内容来对课程质量进行判断。另外，由于云课堂中有许多第三方机构，很多课程存在严重的刷好评情况，对于用户来说他能够根据自己的识别判断出有用评价和无用评价，但对于我们的算法来说，要做到这一点是非常困难的，因此我们在算法中并没有将评论人数和评论内容作为评价课程质量的因素。除此之外，课程的学习人数能够反映出其热门程度，而用户评分则能够反映出课程的质量，但考虑到评分中也存在刷好评的现象，我们又加入了打分人数的衡量，将这三个因素作为课程热门程度特征进行综合考量。

由于基于协同过滤的算法无法对新物品进行推荐，并且课程热门程度的考量对于新物品也不成立，但是我们又希望给新物品提供一定的推荐机会，因此考虑引入课程讲师影响力这一特征。在用户选择课程时，讲师的权威性也会对他造成一定的影响，比如对于用户主动关注的讲师，或曾经上过他课程的讲师，可以认为他们的新课程更容易被用户所接受。除此之外，讲师自身的情况也会产生影响，比如云课堂中有合作的高校教师，其权威性是被大多用户认可的。然而我们在分析云课堂数据时发现，关于讲师个人的资料非常少，无法对讲师的权威性进行判断，但是系统中有讲师与课程的关系信息，通过对讲师开设的课程情况进行考量，能够对讲师的影响力进行合理的判断。因此，我们将通过每一位讲师开设的课程情况对他们的影响力进行量化，同时再针对每个用户，加入该用户与讲师的关注、上过课等关系，将这些因素综合概括为讲师影响力特征。

综上所述，我们选择了基于协同过滤的用户偏好、基于主题的用户偏好、课程热门程度、课程讲师影响力四个特征，作为进行排序推荐的依据。

4.2.1 基于协同过滤的用户偏好

我们将根据用户的学习记录，对用户间的相似度进行分析，根据相似用户的课程学习情况，评估用户对未学习课程的兴趣偏好。用户间的相似度将根据用户-物品矩阵进行计算，用户 u 和用户 v 的相似度计算公式如下，其中 r_u 和 r_v 分别表示两个用户对所有物品的评分向量，r_u' 和 r_v' 分别表示两个评分向量的转置矩阵，用于进行相似度的计算。

$$simi(u,v)=\frac{r_u r_v'}{\sqrt{(r_u r_u')(r_v r_v')}} \tag{4.1}$$

在这里我们主要使用了基于用户的最近邻推荐算法，即根据与目标用户最相似的 K 个用户的评分，来对用户与物品的评分进行预测，相似度计算使用了余弦相似度，而 K 值的大小将通过实验确定。在得到了用户两两之间的相似度后，我们将用户 u 对课程 i 的协同过滤偏好值定义如下，将 K 个用户的评分与相似度加权得到最终结果，公式中 $r(v,i)$ 表示用户 v 对课程 i 的评分。

$$pref_{CF}(u,i)=\frac{\sum_{v=1}^{K} simi(u,v)\cdot r(v,i)}{\sum_{v=1}^{K} simi(u,v)} \tag{4.2}$$

该偏好值即相当于协同过滤推荐算法预测出的评分，范围在 0 到 1 之间，评分越高的说明用户对该物品可能的喜好程度就越高，我们将使用该数值作为一个偏好特征。在该特征提取的过程中，重点是对用户相似度的计算，计算过程的时间复杂度为 $O(N_{user}*N_{item})$，与用户及项目的数量相关。

4.2.2 基于主题的用户偏好

在网络课程学习中，内容高度相似的课程并不会被用户重复学习，但却能很好的反馈用户的兴趣倾向，因此我们将使用主题模型来进行内容偏好特征的抽取。由于文本之间的相似度比较比较困难，且课程的文本介绍往往比较短，使得资料的稀疏度很高，导致高维数据计算效果比较差，因此我们将使用 LDA 算法将文本信息映射到低维向量空间。

首先我们将课程文本信息抽象为主题向量，课程信息包括名称及介绍。然后根据用户资料构造他的主题特征向量，用户资料包括课程学习记录和个人标签，标签为用户自己填写的兴趣爱好信息。但在对数据集中的用户资料进行研究后发现，在整个表的 1 万多用户中，只有 154 个用户填写了个人兴趣标签，由于填写该项信息的用户太少，我们最终没有将该特性作为获取用户偏好的数据来源。我们使用用户学习记录来获得用户的偏好信息，将该用户学习过的课程在主题向量空间上的分布进行加权平均，这样做即可以突出用户频繁学习的课程类别在主题上的权重，又能够有效的避免用户突发兴趣造成的权重噪声。

通过对用户特征向量与课程特征向量相似度的计算，可以得到用户对课程的偏好情况。我们将用户的特征向量，以及基于主题的用户偏好定义如下。其中，θ_i 表示课程 i 的特征向量，在计算用户的特征向量 θ_u 时，n 为用户学习过的课程数量。

$$\theta_u = \frac{1}{n}\sum_{i=1}^{n}\theta_i \tag{4.3}$$

$$pref_{LDA}(u,i) = \frac{\theta_u\theta_i'}{\sqrt{(\theta_u\theta_u')(\theta_i\theta_i')}} \tag{4.4}$$

在基于主题的特征提取中，训练 LDA 模型的时间复杂度为 $O(N_{iter} * V * K)$，其中 N_{iter} 为迭代的次数，V 是集合中单词的数量，K 是主题的个数。而计算用户偏好的时间复杂度为 $O(N_{collect} * N_{word})$，其中 $N_{collect}$ 为用户学习的课程数量，N_{word} 为课程描述中单词的数量。

4.2.3 课程热门程度

课程热门程度特征的计算将综合考量课程的学习人数、用户评分、打分人数。在云课堂的数据中，通用课程的评分范围为 1 到 5 分。我们在查看了数据库表中的评分字段后发现，单纯的按照平均评分高低进行排序得到的结果中，排名靠前的大多课程都是只有极少用户评分且都打了 5 分的情况，不排除存在一些第三方

教育机构为了得到靠前的排名而进行刷分行为。因此我们在评价课程的热门程度时，加入了对评分人数和学习人数的考量，得出了综合的课程热门度值的公式，如下所示。其中 c_i 表示课程 i 的学习人数，d_i 表示课程 i 的打分人数，s_i 表示课程 i 的平均得分。通过对数据集的分析我们发现，平均只有约 5%的学习者会对课程进行评分，因此在平均情况下 c_i 的值大约是 d_i 的 20 倍，另外课程评分s_i的值在 0 到 5 之间。由于绝对数值的这些差异，在计算公式中若单纯的将两项相加，将使得打分人数的影响比较微小。因此为了平衡学习人数和评分情况对课程热门值的影响，我们将学习人数 c_i 乘上了系数 0.25。

$$val_{HOT}(u,i) = c_i \cdot 0.25 + \ s_i \cdot d_i \tag{4.5}$$

通过该公式的计算，在增大了评分的影响力的同时，也考虑到了学习人数在热门课程评定中起的作用。由于课程热门程度的绝对值并没有实际意义，我们更加看重的是课程之间的相对值，所以在对每个课程计算上述的热门度后，我们对其分数进行了从高到低的排序。由于不同分类下的课程学习人数会有较大区别，对数据集中所有课程的排序意义不大，所以我们在排序时是针对每个二级分类下的课程单独进行的。在得到了排序列表后，我们使用如下公式对对最终所要使用的课程热门度特征值进行了计算。

$$pref_{HOT}(u,i) = 1 - \frac{R_i}{N} \tag{4.6}$$

上式中 N 表示相应分类下的课程总数，R_i 表示课程 i 的排名，R_i 为 1 的课程为 $val_{HOT}(u,i)$ 值最高的课程，即该分类下相对最热门的课程。经过这一步计算后，我们即可得到范围在 0 到 1 之间的课程热门度特征值。由于在课程热门度特征提取的过程中，排序部分是最主要的计算步骤，因此该特征提取的时间复杂度为 $O(N_{item} * \log N_{item})$。

4.2.4 课程讲师影响力

由于课程讲师个人信息的缺乏，我们无法从中对其影响力进行较好的判断，

而讲师的开课记录数据比较丰富，因此我们将主要根据讲师所开课程的热门程度来对讲师的影响力进行量化。讲师开设的热门课程数越多，也就意味着这个讲师的影响力越大，因此我们将利用上一节中定义的课程热门程度来计算讲师的影响力。考虑到平台中有些课程的讲师实际是一些教育机构，他们会发布大量的课程，绝对数量的叠加会出现发布大量低质内容的讲师却获得了高影响力的情况，所以我们将通过取讲师所有课程的影响力均值的方法来定义讲师的影响力。我们将讲师影响力因子定义如下，其中 n 为课程 i 的讲师名下其他课程的总数。

$$pref_{TEACH}(u,i) = C \cdot \frac{1}{n}\sum_{k=1}^{n} pref_{HOT}(u,k) \tag{4.7}$$

上述公式中的系数 C 是对用户与讲师关系的考量，用户与讲师的关系通常分为关注、上过课程、无关三种，三种关系分别反映了用户对该讲师的潜在信任程度，因此我们将系数 C 的值也分为 2.0、1.5、1.0 三档，在讲师开课热门程度的基础上乘以该系数。在云课堂中有用户关注的功能，当用户关注了某个老师，则从很大程度上能够说明，他对这个老师的权威性及课程质量是非常信任的，该老师开设的课程也更容易被用户接受。对于这种情况下的讲师，我们将系数 C 的值设定为 2，以增加这一特征的重要性。在用户没有关注该讲师的情况下，我们将考察用户学习过的课程中是否有该讲师的课程，如果学习过则将系数 C 的值设定为 1.5，而除此之外的一般情况则系数 C 取默认值为 1。由于讲师影响力的计算使用了课程热门度特征的计算结果，因此此处计算复杂度仅与讲师开设的课程数量有关，时间复杂度为 O(n)。

4.3 排序学习

经过上一节的特征提取阶段后，我们从原始数据中抽象出了若干特征，这些特征都会对最后的推荐结果产生作用。我们将这些特征整合到一个函数中，用户 u 对课程 i 的预测评分即为这些特征的加权线性组合，我们将组合函数的权重计算问题建模为排序学习问题。在课程推荐问题中，用户与课程的关系只有学习和

未学习两种，对于这种二元评分问题，使用对级排序算法比较适用，我们将使用 Ranking SVM 方法来进行排序学习。

对于给定的用户与物品来说，每一对关系将被表征为一个向量x，向量中的每一个维度对应了提取出的不同特征，向量的维度即为特征的个数。排序函数$f(x) = wx$，w是权重向量，也是我们需要进行训练的参数。在模型训练阶段，我们根据训练数据中用户对课程的学习情况，将课程两两组合为物品对。训练集中的第 i 组数据的表示形式如下 ：

$$\left(x_i^{(1)} - x_i^{(2)}, y_i\right), i = 1,2, \dots, n \tag{4.8}$$

训练集中的每一组数据都是针对同一个用户的，其中 n 为训练集中的样本数量。训练样本的标记值 y_i 的取值包括+1 和-1，对于两个物品 $x^{(1)}$和$x^{(2)}$ 来说，我们将使用其特征向量的差来进行样本标记。若用户学习过课程 1 而未学习过课程 2，我们认为课程 1 在排序列表中的顺序应该在课程 2 前面，因此将 $x^{(1)} - x^{(2)}$ 标记为正样本，而 $x^{(2)} - x^{(1)}$ 则为负样本。以此类推，将训练样本中所有用户学习过的课程与未学习过的课程组队标记。若两门课程用户都学过或都未学过，因其排序关系无法判断，将忽略该训练样本。

标记好训练样本后，我们使用 SVM 模型对其进行排序训练，我们将其损失函数（Loss Function）设置如下：

$$lossfunc = min \sum_{i=1}^{n} \max(1 - \omega^T\left(x_i^{(1)} - x_i^{(2)}\right), 0) + \frac{1}{2}\|\omega\|^2 \tag{4.9}$$

通过学习训练出排序函数后，我们便可使用模型得到待推荐课程列表的排序情况，进而选择排序靠前的课程推荐给用户。

4.4 用户标签生成

我们在 4.2.2 节中抽取出了课程及用户的主题特征向量，该向量除了如前面描述的作为课程推荐的一个特征之外，还将用于生成用户的兴趣标签。云课堂中

课程的一级分类有 7 个，二级分类有 45 个，三级分类则有超过 150 个。我们认为一级分类范围太广泛，无法明确的反映用户的具体兴趣爱好，而三级分类则太精细，很可能出现用户学习的每一门课程都属于一个单独的三级分类的情况，不利于从这些课程中提取出公共的特征。因此我们将选用课程的二级分类作为用户兴趣标签的内容，其在细粒度和准确度上都更有效。

在生成用户偏好标签的过程中，我们首先将使用前述算法中抽取的主题向量空间，分别计算得到每个二级分类标签在主题空间上的向量。然后将该向量与用户在主题空间的向量计算相似度，将相似度排序后即可得到和用户相似度高的二级分类作为用户的兴趣标签。计算二级分类标签在主题空间上的向量的方法与计算用户的兴趣标签方法一致，即将对应二级分类标签下的所有课程在主题空间上的向量进行加权平均。计算用户和标签的相似度时，我们采用余弦相似度来计算这两个向量之间的相似度。

标签 t 的特征向量 θ_t，以及用户 u 和标签 t 的相似度计算公式如下，其中 n_t 为标签 t 分类下的课程总数。

$$\theta_t = \frac{1}{n_t}\sum_{i=1}^{n_t}\theta_i \tag{4.10}$$

$$simi(u,t) = \frac{\theta_u \theta_t'}{\sqrt{(\theta_u \theta_u')(\theta_t \theta_t')}} \tag{4.11}$$

4.5 本章小结

本章首先对本文提出的课程推荐算法进行了介绍，该算法将课程推荐问题转换为了排序问题，使用了排序学习的方法，综合考虑了影响用户课程选择的多种因素，以得到更合理的推荐结果列表。本章对算法的具体步骤、各特征的内容、排序方式进行了详细介绍。另外，本章还对用户标签生成的算法进行了简要介绍，该算法将用于后续的系统实现中。

第5章 系统设计与实现

本章是系统的设计实现部分，本文实现的功能主要包括课程推荐和用户兴趣标签功能，本章对这两个功能的设计分别进行了详细的介绍，并展示了最后的实现效果。

5.1 课程推荐功能

5.1.1 功能描述

在云课堂的网站中，每个用户会有自己的个人学习主页，其中展示了用户的课程学习情况，包括用户已经学习过的课程、正在学习的课程和收藏的课程。个人主页的主要作用是方便用户掌握自己的学习情况，并对自己拥有的课程进行管理。我们将在该页面的基础上，添加课程推荐的功能，为用户未来的课程学习提供建议。推荐课程将以列表的形式展示，在学习课程和收藏课程的页面都会显示推荐列表，但推荐所使用的数据主要是用户的学习课程记录。

5.1.2 详细设计

我们实现的网络课程推荐系统总体架构如图 5.1 所示。

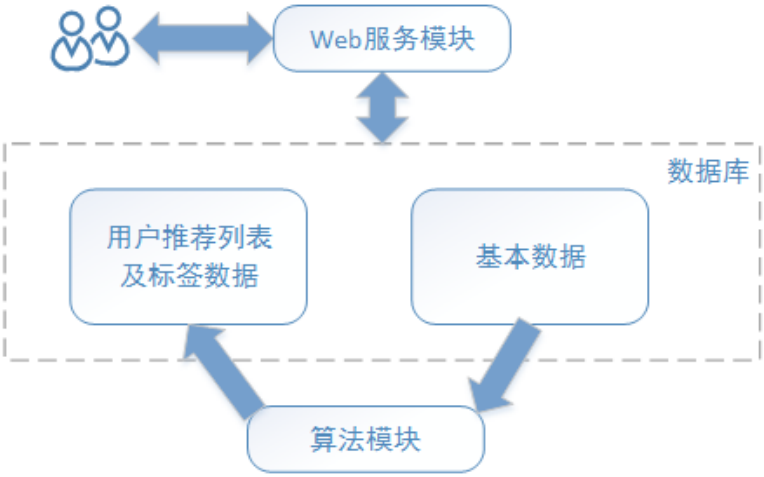

图 5.1 课程推荐系统架构图

我们在推荐算法中主要使用的是用户的学习记录，而对于每个用户来说，这部分数据并不会频繁更新，对推荐的实时性要求不高。因此我们将采用定时计算的方式，每隔一段时间计算一次课程推荐列表及用户兴趣标签，并将计算结果保存至数据库，当用户访问个人主页时，服务器从数据库中获取基本信息和推荐相关信息，并将结果返回给用户。

其中，Web 服务模块是页面展示以及用户进行操作的接口，模块将从后台读取数据展示给用户，同时也将用户活动产生的新的行为数据更新到数据库中。

数据库模块存储了两部分数据，一部分是课程、讲师、用户等的基本数据，以及用户学习课程产生的学习记录数据，另一部分是算法模块根据基本数据计算出的推荐列表数据及用户标签。

算法模块是进行推荐课程计算的部分，模块将从数据库中读取课程信息、学习记录等基本信息，使用排序算法计算出推荐列表，然后更新至数据库中的推荐数据表，该模块的执行将定时进行。

5.2 兴趣标签功能

5.2.1 功能描述

根据用户已经学过的课程可以提取出他的偏好信息，生成个性化标签云，这些内容将显示在用户的个人学习主页上，帮助用户掌握自己的学习情况，也为访问用户主页的其他用户提供了该用户的兴趣内容，帮助他们发现兴趣相投好友进行社交。标签云反映了用户对每个分类的偏好程度，这个程度值是从用户的学习记录中计算出来的，所以随着用户学习课程数量的增多，标签云也会越来越丰富，越来越准确的概括出用户的兴趣分类图。

兴趣标签功能与推荐功能是相关联的，我们在推荐过程中使用了基于内容的用户偏好作为特征之一，在计算用户偏好的过程中可以生成用户的兴趣标签，因此该功能也是推荐的一部分。

5.2.2 详细设计

5.2.2.1 文本的选择

用户的兴趣偏好信息将从他学习过的课程中提取，但课程信息中包含了很多文本内容，主要有标题、简介、大纲，我们首先需要对主题提取所使用的文本内容进行选择。一般来说，标题和简介是由课程创建者自己编写的，通常能够准确的概括出课程的主要内容以及特色之处。但这些内容一般比较简单，而且大多是比较笼统的介绍性语言，并不能有效的反映出课程的具体教学内容，从而也就无法有效的将课程的特点区分出来。而课程大纲信息则非常详细，尤其是会出现一些课程专有的名词，这些名词往往能够反映出一门课程或一类课程的特色，在主题提取的过程中非常有用。我们从云课堂中选取了一门课程进行说明，其课程信息如表 5.1 所示。

表 5.1 课程文本信息示例

课程 ID	167002	课程名称	计算机科学及编程导论
课程简介	课程致力于使学生理解计算机在解决问题中的作用,并且帮助学生，不论其专业，使他们对于能够完成有用的小程序的目标充满信心。		
教学目录（部分）	课时 1：课程目标，数据类型，运算，变量 课时 2：分支，条件和循环 课时 3：一般代码样式，循环式程序 课时 4：函数抽象与递归简介 课时 5：浮点数和二分法（逐次近似） 课时 6：二分法，牛顿，拉复生方法，对于数组的简介 课时 7：数组以及可变性，字典，伪码，对于代码运行效率的简介 课时 8：算法的复杂度：对数级，线性级，平方级，指数级 课时 9：二分法搜索，冒泡排序与选择排序 课时 10：分治法，合并排序，异常 课时 11：测试与调试 课时 12：调试的更多内容：背包问题，动态规划简介 课时 13：动态规划，重叠的子问题，最优子结构 课时 14：背包问题的分析，面向对象编程简介		

从表中可以看到，课程简介非常简短，只出现了“计算机”、“程序”这两个能够代表课程属性的词语。而在课程目录中，完整的概括了课程的主要内容，包含了大量与计算机相关的词语，非常具有代表性。综上所述，为了增强用户兴趣标签的说服力，我们将把课程的名称、简介、大纲合并，作为课程的内容信息进行主题提取。

5.2.2.2 分词和停用词

在得到了每门课程的文本信息后，我们首先需要对文本进行分词处理。实际应用时中文分词一般比较困难，在开发中我们使用了中文分词组件 jieba[1]来进行分词，该分词库有以下几个特点：

- 基于前缀词典实现高效的词图扫描，生成句子中汉字所有可能成词情况所构成的有向无环图（DAG）。
- 采用了动态规划查找最大概率路径，找出基于词频的最大切分组合。
- 对于未登录词，采用了基于汉字成词能力的隐马尔科夫模型（Hidden Markov Model，简称 HMM），使用了 Viterbi 算法。

该分词库效率高、使用比较方便，而且分词效果比较好，对于已收录词和未收录词都有相应的算法处理，能很好的完成我们的分词任务。

在进行分词之后我们还需要将停用词剔除，我们首先使用了常用的中文停用词表，将表中的词全部删除，如我们、虽然、但是、比如、不仅、结果等普通文档中经常出现的无意义词语。另外，我们还根据 LDA 的训练结果，筛选出了一些网络课程中会出现的停用词，比如课程、学习、绪论、第一章、知识、part1 等，这些词语在一般文本中具有标志性，但在我们的网络课程相关分词任务中，是普遍出现且无明确指向含义的词，在 LDA 主题抽取中也无法起到区分的作用，因此也将作为停用词剔除掉。

5.2.2.3 计算用户的学习标签

我们将使用 LDA 算法来计算用户的学习标签，为了更准确的描述用户的兴

[1] https://github.com/fxsjy/jieba

趣，我们使用了云课堂课程的二级分类标签，包括前端开发、办公软件、摄影影视等。我们首先根据用户的历史学习记录，将他学习过的所有课程在主题空间的分布进行加权平均，得到用户的兴趣偏好在主题空间上的分布。然后按照同样的方法，计算出所有课程二级分类标签在主题空间上的分布。在得到了用户和分类标签的主题空间向量之后，我们使用余弦相似度来计算用户和分类标签之间的喜好程度，然后将相似度值排序，取相似度最高的几个作为用户的兴趣标签。

我们选取了系统中的一个用户，对他的学习课程进行了分析，如表 5.2 所示为该用户学习过的部分课程，如表 5.3 所示为我们训练得到的用户和标签的近似度排名情况。可以看到兴趣标签比较准确的反映了用户学习过的课程情况。

表 5.2 用户 1 学习过的部分课程

编号	课程名称
1	淘宝美工教程 ps 实战教程平面设计海报设计教程
2	Servlet+JSP（JavaEE 开发进阶Ⅰ）
3	Hibernate 框架（JavaEE 开发进阶Ⅲ）
4	贝太新煮艺：十分钟搞定早餐
5	HTML+CSS+JS【有答疑】（JavaEE 开发基础Ⅱ ）
6	Android 基础视频教程
7	大学就业综合实训—JavaWeb 开发
8	Spring MVC 从入门到精通视频教程

表 5.3 用户 1 的兴趣标签前 10 项

编号	标签	编号	标签
1	后端开发	6	用户体验
2	前端开发	7	生活
3	基础语言	8	职场技能
4	移动开发	9	个人管理
5	平面设计	10	健康

5.3 效果展示

我们将兴趣标签云以及课程推荐功能集成在了云课堂的用户个人学习主页中，如图 5.2 所示，为所选用户 1 的个人学习主页。

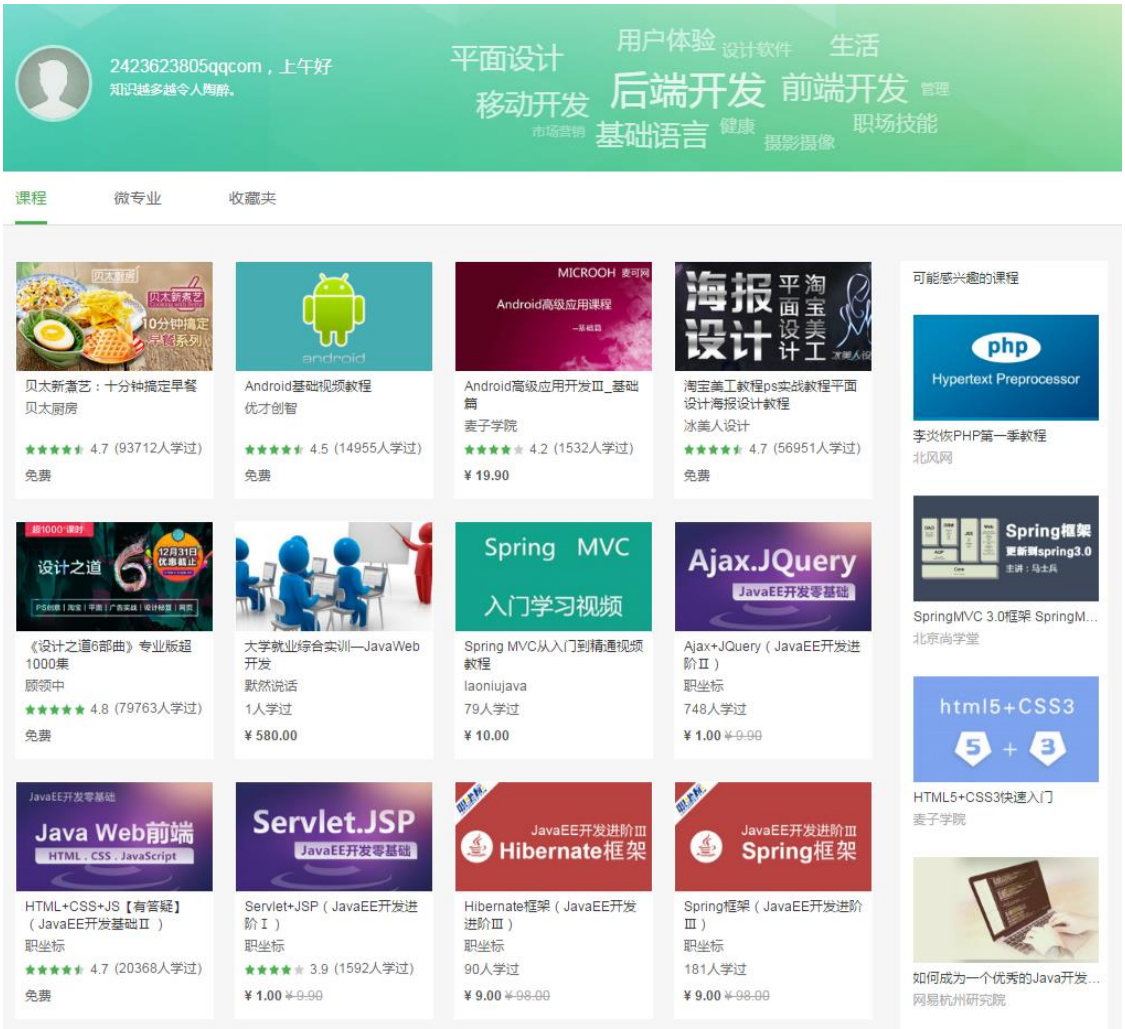

图 5.2 用户 1 的个人学习主页

从上图中可以看到，页面的最上面显示了用户的头像和昵称，旁边是用户的兴趣标签云，根据用户在每个标签上偏好程度的不同，各个标签会有不同的字体大小及透明度。页面主体部分是用户学习过的课程列表，列表的右侧则是我们推荐给用户的课程。图中仅截出了用户学习过的部分课程，从中已经可以看出该用户主要学习的都是计算机方向的课程，并且以网站开发为主。用户的标签云也准确的反映了用户的兴趣偏好，匹配度较高的标签包括后端开发、前端开发、基础

语言、移动开发等。系统推荐给用户的课程也是相关领域的课程，与用户兴趣比较一致。

我们又选取了另一名用户进行说明，图 5.3 为其学习主页。

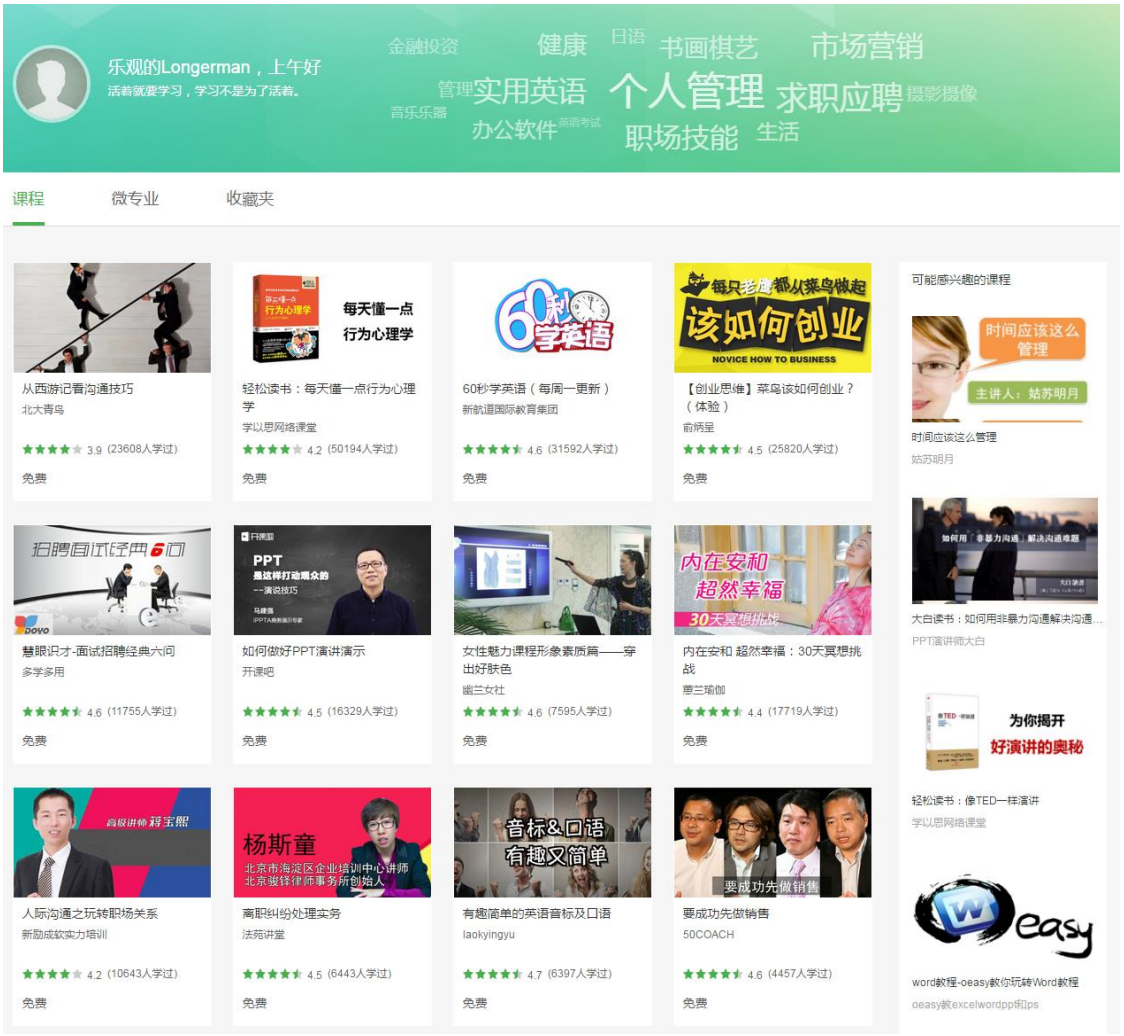

图 5.3 用户 2 的个人学习主页

从图中可以看到，用户 2 学习的课程涉及的类别比较广泛，有人际沟通自我管理类，也有英语、办公软件类，从他的兴趣标签中也可以发现这一点，而推荐给他的课程也比较符合用户的兴趣倾向。

5.4 本章小结

本章是推荐系统的设计实现部分，我们将课程推荐以及用户兴趣标签功能进行了实现，详细描述了功能的架构设计以及各模块具体的内容，最后展示了各个功能的效果，验证了本文工作的实用性。

第6章 实验结果与分析

本章是本论文的实验部分，将在云课堂真实数据上将本文算法与其他算法进行对比实验，本章将详细介绍实验环境、实验数据、实验步骤，以及实验结果的展示和分析等。

6.1 实验配置

6.1.1 运行环境

本文实现算法的主要语言为 Matlab，使用的环境版本为 Matlab 2016a，主题空间及标签提取部分使用 Python 实现，所用语言版本为 Python 3.5，其中用到了中文分词库 jieba[1]。系统功能开发是网页的形式，主要使用到的语言为 HTML、CSS、JavaScript、Java。算法实验和功能实现均在作者个人电脑上进行，实验的硬件环境如表 6.1 所示。

表 6.1 实验硬件环境

项目	内容
处理器	Intel(R) Core(TM) i5-4590
内存	16.0GB
操作系统	Windows 7 64bits

6.1.2 对比算法

为了验证本文算法的效果，我们选取了若干算法进行对比实验。由于本文算法综合了多个特征进行排序推荐，而每个特征都从不同的方面反映了用户对课程的偏好程度，因此每个特征都可以单独作为一种算法来进行推荐，我们将使用这些单特征算法作为对比实验。实验中使用的对比算法如表 6.2 所示。

[1] https://github.com/fxsjy/jieba

表 6.2 实验对比算法介绍

算法名称	算法描述
RS	随机选取算法
LDA	基于主题模型的推荐算法
CF	基于用户协同过滤的算法
HOT	基于课程热门程度的推荐
TEA	基于讲师影响力的推荐
LTR	本文提出的学习排序推荐算法

其中 LTR 即为本文提出的基于多特征排序模型的推荐算法。LDA、CF、HOT、TEA 为本文排序算法中使用的四个特征，对于每个单特征算法，我们将根据每个项目的得分情况进行排序，选择分数最高的若干个作为推荐结果。RS 为随机选取（Random Select）算法，即从候选课程中随机选出若干个作为推荐结果。RS 算法是最基本的对比算法，在使用单特征进行推荐时，需保证其效果至少比随机选取算法要好。

6.1.3 衡量指标

本文将推荐问题转换为了排序问题，算法的效果主要将使用准确率（Precision）和召回率（Recall）来进行衡量。在实验中，我们为测试集中的每个用户推荐 N 门课程，并与其真实值进行比较，我们将使用Precision@N、Recall@N来表示推荐数量不同的情况下的结果，其计算公式如下所示，其中N_{real}表示用户实际学习的课程数量，$N_{predict}$表示系统推荐的课程数量。

$$\text{Precision@N} = \frac{\left|N_{real} \cap N_{predict}\right|}{N_{select}} \tag{6.1}$$

$$\text{Recall@N} = \frac{\left|N_{real} \cap N_{predict}\right|}{N_{predict}} \tag{6.2}$$

准确率表示返回结果中用户学习过的课程，占用户所有学习过课程的比率。

召回率表示用户学习过的课程占返回结果总数的比例。这两个指标的值越大，说明推荐的结果越理想。但是由于我们实验中使用的用户评分矩阵只有 0 和 1 两个值，其中 0 并不代表用户不喜欢该课程，可能是还没有进行评分，而为 1 的项则能够准确反映出用户的喜好。准确率指标较难计算，无法正确的反映推荐结果的好坏，后续的实验中也证明了这一点，因此我们将主要使用召回率指标来衡量推荐的效果。

6.2 实验过程与步骤

6.2.1 特征提取

在特征提取阶段，我们将分别提取基于协同过滤的用户偏好、基于主题的用户偏好、课程热门度评分以及讲师影响力评分。在实验过程中，我们从全部的用户-课程矩阵中按照 80%训练集、20% 测试集的比例进行了随机抽取，然后在此基础上计算基于协同过滤的用户偏好和基于主题的用户偏好。另外，由于课程热门程度和讲师影响力是基于全部学习数据来评定，并且我们最终考虑的是考虑课程及讲师在各自分类下的相对评分，所以我们将选择用全部的用户学习记录来计算这两个评分。

6.2.1.1 基于协同过滤的用户偏好

为了计算基于协同过滤的用户偏好，我们使用云课堂的原始数据集生成了用户-物品矩阵，矩阵中 1 表示用户学习过该课程，0 表示用户未学习过。得到矩阵后，我们使用协同过滤算法 userKNN 来进行用户偏好的计算。该算法中包含参数最近邻数量 K，在实验中我们也发现该参数会影响算法结果。为了确定参数 K 的选值，我们将 userKNN 作为单独的推荐算法，在数据集上进行实验，对推荐结果的 recall 指标进行衡量。实验结果如图 6.1 所示，可以看到召回率随 K 值变化的的曲线为上凸状，K 值大约在 25 处时达到最大值，此处我们兼顾计算速度和推荐效果将 K 值选定为 20。

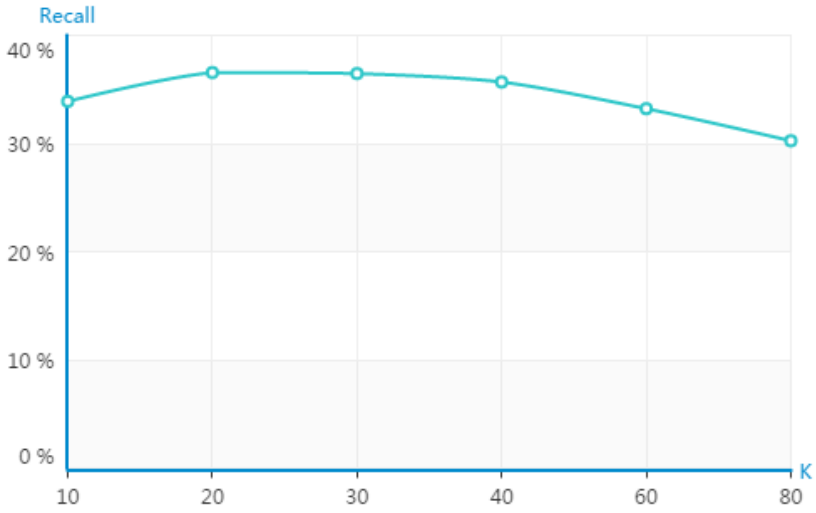

图 6.1 userKNN 算法参数 K 对结果的影响

6.2.1.2 基于主题的用户偏好

在使用 LDA 计算基于主题的用户偏好时，算法中主题数量是重要的参数，我们在实验中发现，LDA 算法中主题数量越多实验的结果就会越好，但相应的模型训练时间和执行时间也会相应的增大，如何在效率和结果间进行平衡是非常重要的问题。我们将 LDA 作为单独的推荐算法进行了实验，为每个用户选取主题向量最相似的课程作为给他的推荐结果。我们将使用推荐结果的召回率来衡量 LDA 算法在不同主题数量下的表现，实验结果如图 6.2 所示，实验中推荐课程数量设为 100，同等条件下随机选取算法的召回率在 0.024 左右。

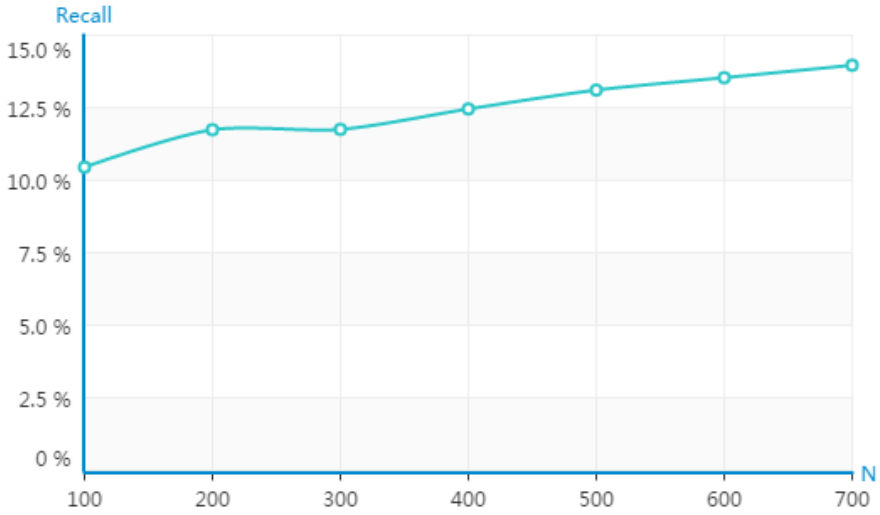

图 6.2 主题数量对 LDA 结果的影响

经过上述实验后，我们综合考虑了算法的运行速度和实验效果，决定选取主题数 200 作为后续开发实现的基准。我们使用该参数对数据集中的课程信息进行了主题提取，部分主题和单词的关系如表 6.3 所示，可以看到每一个主题的代表词都能够明确的反映出该主题的特点，如第一个主题是与企业及招聘等相关，第二个主题是与产品经理这一职业有关，第三个主题则是高等数学的内容，说明主题提取的结果比较准确。

表 6.3 LDA 算法主题与单词关系示例

编号	主题代表词（按主题与单词关系强弱排序）
1	企业、发展、职业、招聘、人才、竞争、小企业、就业、行业、需要
2	产品、经理、用户、需求、问题、研究、分析、案例、笔试、设计
3	函数、应用、概念、方程、积分、数学、导数、性质、定理、微分
4	游戏、案例、iOS、安全、Cocos2d、项目、碰撞、开发、飞机、大战
5	制作、动画、课件、效果、场景、模型、PPT、渲染、材质、3D

6.2.1.3 课程热门度生成

课程热门程度评分兼顾了课程的学习人数、平均评分和评分人数，根据我们在第 4 章中定义的计算方法，我们将数据集中的所有课程按照二级分类进行了分组，并对每组中的课程计算热门度值从高到低排序，最终得到所有课程在 0 到 1 范围内的热门度指标。该评分将在下一节的排序学习过程中，作为反映课程热门程度对用户选课的影响因子来使用。在热门度单特征算法 HOT 中，我们针对测试集中的每一个用户，将其课程的热门度值进行排序，选取值最高的 N 门课程作为推荐结果。

6.2.1.4 讲师影响力生成

我们在第 4 章中定义了讲师影响力评分，在实验中我们对数据集中的所有讲师根据其拥有的课程的热门程度计算了讲师影响力。针对特定用户和特定课程，我们将对用户与课程讲师的关系进行判断，查看用户是否关注了该讲师，以及他

上过的课中是否有该讲师的课，然后对相应的讲师影响力因子进行调节，最终能够得到用户-课程的讲师影响力评分矩阵。在下一节的排序过程中，该指标将作为反映讲师影响力对用户选课与否的影响因子来使用。而在单特征算法 TEA 中，我们将计算出每门课程所属讲师的影响力，选择影响力最高的 N 们课程作为给用户的推荐结果。

6.2.2 模型训练

在上一小节中，我们依次得到了用户在主题上的偏好、基于协同过滤的用户偏好、课程热门度评分以及讲师影响力评分这四个特征，接下来将在此基础上进行基于 Ranking SVM 的排序学习过程。

针对每一个用户，我们根据训练集中该用户的所有用户-课程关系生成对应的新记录 $(\mathrm{x}, \mathrm{label})$，其中 x 向量为上述四个特性对应的值，label 为正负样本标注。如果训练集中该用户确实学习过了此课程，则 label 值为 1，否则值为 0。接下来我们将根据成对排序学习方法构造实验中使用的新训练集，把所有 label 为 1 的记录依次和所有 label 为 0 的记录两两结合，假设 $x_i^{(1)}$ 的 label 为 1，$x_i^{(2)}$ 的 label 为 0，则将 $(x_i^{(1)} - x_i^{(2)}, +1)$ 和 $(x_i^{(2)} - x_i^{(1)}, -1)$ 作为新的记录加入训练集。得到每个用户新的训练集以后，就可以利用 Ranking SVM 方法为每个用户训练一个用户偏好模型，即该用户在各个特征上面的偏好系数。利用训练得到的用户偏好模型，我们在测试集上预测用户对课程的评分，并将课程按评分的大小降序排列，得出该用户的课程推荐列表。

6.3 实验结果与分析

6.3.1 推荐数量对结果的影响

我们在实验中发现，推荐课程的数量会对结果造成影响，我们将推荐数量 N 取不同值，对各个算法的表现进行了实验，其在 precision 和 recall 两个指标上的实验结果如图 6.3 和图 6.4 所示。

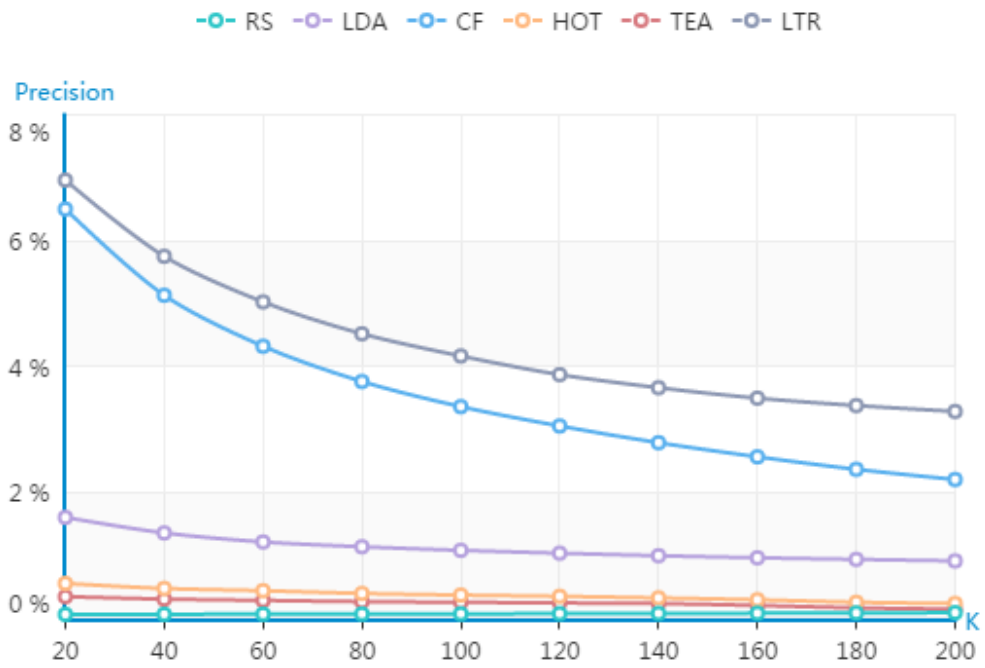

图 6.3 推荐数量 N 对各算法 precision 结果的影响

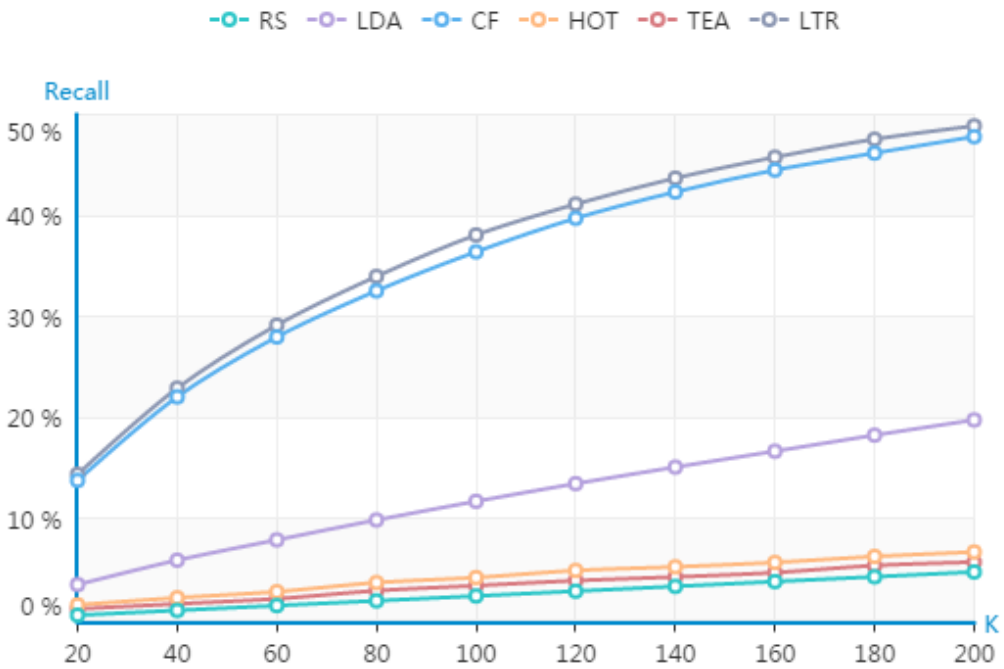

图 6.4 推荐数量 N 对各算法 recall 结果的影响

从图 6.3 中可以看到，各个算法的 precision 值均随着 N 的增大而减小，并且绝对数值都非常低，均在 10%以下。造成这种现象的原因主要在于数据集的稀疏

性比较大，测试集中有大量为 0 的项，而同时每个用户平均学习的论文数又比较小，因此随着 K 的增大会有越来越多的无关项出现在结果集合中，导致 precision 值越来越低。从图 6.4 中可以看到，各个算法的 recall 值随着 N 的增大而增大，这是因为随着推荐数量的增加，结果集中包含的真值数量也在增加，而测试集中正值的数量保持不变，所以 recall 的值会越来越大。

在各个算法的总体表现上，随机选取 RS 算法作为基本对比算法，结果比较差。而单特征算法中的热门程度 HOT 以及讲师影响力 TEA 两种方法，由于所使用的算法非常简单，所考虑的影响因素也非常单一，因此推荐效果仅比随机选取算法略好。另外可以看到 HOT 算法比 TEA 算法整体效果要好，也说明了用户在选择课程时，相比于讲师的影响力来说，更加看重的还是课程本身的热门程度以及质量。除了这两种单特征算法之外，比较复杂的单特征算法包括 LDA 和 CF 两种。其中基于内容的 LDA 算法由于只考虑了课程的内容，所以推荐效果一般。而基于协同过滤的 CF 算法是个性化推荐问题中最常用的算法，我们的实验也证明了其在网络课程的推荐场景中也能够有较好的表现。本文所提出的基于多特征排序模型的推荐算法，综合考虑了多种影响因素，其中最主要参考的是基于协同过滤的算法，从实验结果中也可以看到 LTR 算法的推荐结果相对于 CF 算法有所提升，并且比其他几种单特征算法结果有显著提升。

6.3.2 用户学习数量对结果的影响

理论上基于内容的算法无论用户学习记录多少都能发挥其效果，而基于协同过滤的算法则会随着用户学习数据的增加而有越来越好的表现。本文提出的综合推荐算法将这两种算法进行了综合考虑，其在用户学习数不同的情况下都应该有比较好的表现。为了验证这一点，我们将数据集根据用户学习数不同进行了分割，对 LDA、CF、LTR 三种算法进行了实验，结果如图 6.5 所示，其中 N 表示所选数据集中用户学习课程数的上限，实验中我们将推荐课程数设为 180。

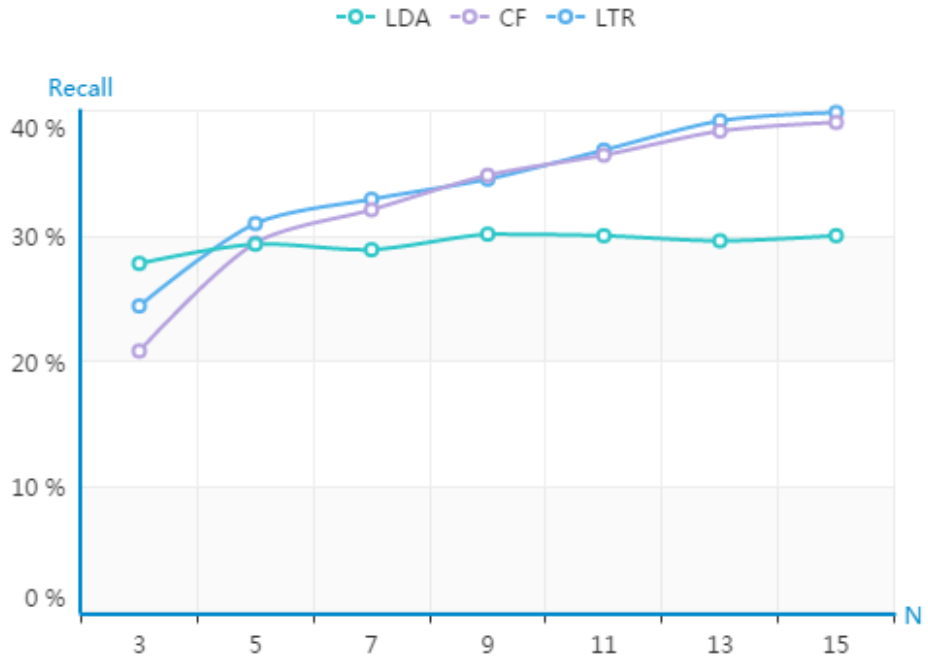

图 6.5 用户学习课程数量对结果的影响

从图中可以看到，当N为3时用户学习数量比较少，协同过滤算法的效果是非常差的，我们选用的协同过滤算法是userKNN，它通过寻找相似用户来进行推荐，在这种情况下仅通过有限的课程数，很难对用户的相似度进行准确的判断。另一方面LDA算法的效果却比较好，这也是基于内容的推荐算法在冷启动问题上优良表现的证明。而本文所提出的LTR算法综合了这两种算法的优点，在这种情况下的表现也是介于两者之间。

随着用户学习数量的增加，越来越丰富的用户行为数据使得协同过滤算法的表现变得非常好。本文的LTR算法也有较好的表现，且总体上比协同过滤算法略好，仅在N为9时比CF算法差一些，可能是因为限定了学习课程数后，实验数据中用户的数量比较少，实验的结果并不理想。另一方面，LDA算法虽然在课程数较少的情况下有不错的表现，但随着课程数的增加，其效果一直比较平均，没有太大的提升。在实际应用中，我们更希望得到的效果是随着时间的推移，随着用户学习的课程越来越多，个性化推荐的效果应该越来越好。LDA算法虽然对于冷启动问题有较好的解决，但却无法满足实际系统以及用户的需要。

综上所述，本文提出的排序学习算法，综合了协同过滤算法的 LDA 算法的

优点，在用户课程数变化的情况下，整体效果保持在较好的水平。一方面对协同过滤算法的冷启动问题进行了改善，另一方面相比基于内容的推荐算法效果有显著提升，达到了预期的效果。

6.4 本章小结

本章将本文的算法与其他推荐算法进行了对比实验，将特征提取中涉及到的每个特征单独作为一种推荐方式，分析了各个算法的优劣性，验证了本文提出的综合推荐算法的有效性。本章详细介绍了实验的环境、数据集、衡量指标、过程步骤，以及最后的结果。

第7章 总结与展望

7.1 本文工作总结

自从大规模网络开放课程出现以来，在线教育的发展如火如荼，越来越多的用户开始习惯于这种新颖且丰富的学习形式。随着在线课程资源数量的增加以及种类的越来越多样化，用户在考虑想要学习的课程时经常会遇到选择难题。课程推荐能够为用户的学习提供可靠的建议，但由于这一领域比较新颖，相关的研究工作还不够丰富。另外，由于网络课程自身的特点，如文本信息较少、用户行为信息不够丰富、评价信息缺乏等，传统的推荐算法无法直接应用到网络课程的推荐中，使得相关的工作还有很大的研究空间。

通常的推荐算法功能以评分预测为主，但在实际应用中用户往往只会关注推荐结果中排在前面的几个，因此推荐系统更多的是解决排序问题。另外，在用户进行网络课程选择的过程中，课程的内容、课程的热门程度、讲师的权威性等诸多条件都会对用户造成影响。单一的推荐算法条件太过局限，将无法为用户提供最佳的推荐结果。因此在进行网络课程推荐时，应当针对网络课程及其用户独有的特点，设计出最佳的推荐方法。

在综合考虑了现有研究工作，以及对云课堂用户学习数据进行了充分分析的前提下，本文研究并实现了一个基于排序模型的网络课程推荐算法，该算法综合考虑了网络课程的多方面特征，并将推荐问题概括为了排序问题。我们的算法使用了基于主题的用户偏好、基于协同过滤的用户偏好、课程热门程度、讲师影响力这四个特征，使用排序学习方法计算出各特征的权重，以线性组合的方式将各特征结合起来，为每一位用户进行个性化课程推荐。

为了验证本文工作的有效性，我们在云课堂真实数据集上对算法效果进行了验证，实验证明本文算法相对于传统单一的推荐算法有较大的改进。另外，我们基于云课堂用户个人学习主页实现了课程推荐以及用户兴趣标签的功能，对算法的有效性也进行了验证。

概括来说，本文的主要工作包括：

1. 对云课堂的课程、讲师、用户学习记录数据进行了充分的统计分析，从多个角度对网络课程及用户的特点进行了概括总结。
2. 提出了一种基于排序模型的网络课程推荐算法，综合考虑了多种可能会影响用户课程选择的因素，将这些因素综合考虑并整合为基于排序问题的推荐算法。
3. 在云课堂真实数据集上进行了对比实验，将本文算法与几种常用算法的效果进行了对比，模拟了多种不同情况下各算法的表现情况，并对实验结果进行了详细分析。
4. 在云课堂用户个人学习主页的基础上，添加了课程推荐以及用户兴趣标签云的功能。

7.2 未来工作展望

随着互联网的不断发展，尤其是移动互联网的普及，网络课程这种教育方式将会被越来越多的用户所接受，在线教育网站的课程数量以及用户数量也会出现大幅增长，这给网络课程推荐的研究工作也带来了新的挑战。如何在保证系统性能的条件下，对用户行为进行深入的挖掘分析，为每一位用户提供个性化推荐服务，是相关研究面临的重要问题。

本文对网络课程推荐问题进行了研究，但由于数据集的规模不足、考虑条件不够充分等问题，我们研究工作仍然存在许多不足之处，未来主要考虑从以下几个方面进行进一步的完善和扩充：

1. 课程相关的文本信息除了简介及教学目录外，还有大量用户的评论，其内容能够很好的反映课程的质量、难度等，未来可以考虑在这方面进行挖掘，从更丰富的角度对课程进行考量。
2. 用户的学习数据中包含很多反映学习情况的内容，比如学习的进度、学习的时间等，这些信息从一定程度上反映了用户对这门课的兴趣情况。比如有的课程只学习了一个课时就没有再继续过，说明用户学习后发现

对课程内容并不感兴趣。未来可以通过这些信息，加强对用户的课程偏好的评价情况。

3. 在系统实现方面，现在的实现主要以离线计算为主，但未来的系统中可能会对实时性提出更高的要求，如何在及时响应用户行为的前提下，保证系统的运行性能，将是未来系统研究的重点。

7.3 本章小结

本章是本论文的最后一章，首先对整篇论文所做的研究工作进行了概括，总结了突出的贡献，也反思了现有工作的不足。然后对未来的研究工作及改进方向进行了展望，从算法研究和应用实现两个方面进行了概括。

参考文献

[1] Ricci F, Rokach L, Shapira B. Introduction to recommender systems handbook[M]. Springer US, 2011.

[2] Leskovec J, Rajaraman A, Ullman J D. Mining of massive datasets[M]. Cambridge University Press, 2014.

[3] Lops P, De Gemmis M, Semeraro G. Content-based recommender systems: State of the art and trends[M]//Recommender systems handbook. Springer US, 2011: 73-105.

[4] Liu T Y. Learning to rank for information retrieval[J]. Foundations and Trends in Information Retrieval, 2009, 3(3): 225-331.

[5] Apaza R G, Cervantes E V, Quispe L C, et al. Online Courses Recommendation based on LDA[C]//SIMBig. 2014: 42-48.

[6] Kravvaris D, Ntanis G, Kermanidis K L. Studying massive open online courses: recommendation in social media[C]//Proceedings of the 17th Panhellenic Conference on Informatics. ACM, 2013: 272-278.

[7] Lee Y, Cho J. An Intelligent Course Recommendation System[J]. SmartCR, 2011, 1(1): 69-84.

[8] Hang L I. A short introduction to learning to rank[J]. IEICE TRANSACTIONS on Information and Systems, 2011, 94(10): 1854-1862.

[9] Karatzoglou A, Baltrunas L, Shi Y. Learning to rank for recommender systems[C]//Proceedings of the 7th ACM conference on Recommender systems. ACM, 2013: 493-494.

[10] Shi Y, Larson M, Hanjalic A. List-wise learning to rank with matrix factorization for collaborative filtering[C]//Proceedings of the fourth ACM conference on Recommender systems. ACM, 2010: 269-272.

[11] Blei D M. Probabilistic topic models[J]. Communications of the ACM, 2012, 55(4): 77-84.

[12] Blei D M, Ng A Y, Jordan M I. Latent dirichlet allocation[J]. Journal of machine

Learning research, 2003, 3(Jan): 993-1022.

[13] Cremonesi P, Koren Y, Turrin R. Performance of recommender algorithms on top-n recommendation tasks[C]//Proceedings of the fourth ACM conference on Recommender systems. ACM, 2010: 39-46.

[14] Su X, Khoshgoftaar T M. A survey of collaborative filtering techniques[J]. Advances in artificial intelligence, 2009, 2009: 4.

[15] Dumais S T. Latent semantic analysis[J]. Annual review of information science and technology, 2004, 38(1): 188-230.

[16] Hofmann T. Probabilistic latent semantic indexing[C]//Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval. ACM, 1999: 50-57.

[17] Goldberg D, Nichols D, Oki B M, et al. Using collaborative filtering to weave an information tapestry[J]. Communications of the ACM, 1992, 35(12): 61-70.

[18] Mozina M, Sadikov A, Bratko I. Recommending videolectures with linear regression[C]//Proc. of ECML-PKDD 2011 Discovery Challenge Workshop. 2011: 41-49.

[19] Kuang W, Luo N, Sun Z. Resource recommendation based on topic model for educational system[C]//Information Technology and Artificial Intelligence Conference (ITAIC), 2011 6th IEEE Joint International. IEEE, 2011, 2: 370-374.

[20] Farzan R, Brusilovsky P. Social navigation support in a course recommendation system[C]//International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems. Springer Berlin Heidelberg, 2006: 91-100.

[21] Bishop C M. Pattern recognition[J]. Machine Learning, 2006, 128.

[22] Herbrich R, Graepel T, Obermayer K. Large margin rank boundaries for ordinal regression[J]. Advances in neural information processing systems, 1999: 115-132.

[23] Joachims T. Optimizing search engines using clickthrough data[C]//Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining. ACM, 2002: 133-142.

[24] Romero C, Ventura S. Educational data mining: a review of the state of the art[J]. IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and

Reviews), 2010, 40(6): 601-618.

[25] Fuhr N. Optimum polynomial retrieval functions based on the probability ranking principle[J]. ACM Transactions on Information Systems (TOIS), 1989, 7(3): 183-204.

[26] Cao Y, Xu J, Liu T Y, et al. Adapting ranking SVM to document retrieval[C]//Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval. ACM, 2006: 186-193.

[27] Song Y, Wang H, He X. Adapting deep ranknet for personalized search[C]//Proceedings of the 7th ACM international conference on Web search and data mining. ACM, 2014: 83-92.

[28] Freund Y, Iyer R, Schapire R E, et al. An efficient boosting algorithm for combining preferences[J]. Journal of machine learning research, 2003, 4(Nov): 933-969.

[29] Miao Z, Wang J, Zhou A, et al. Regularized Boost for Semi-supervised Ranking[C]//Proceedings of the 18th Asia Pacific Symposium on Intelligent and Evolutionary Systems, Volume 1. Springer International Publishing, 2015: 643-651.

[30] Fischer G. Beyond hype and underestimation: identifying research challenges for the future of MOOCs[J]. Distance education, 2014, 35(2): 149-158.

[31] Martin F G. Will massive open online courses change how we teach?[J]. Communications of the ACM, 2012, 55(8): 26-28.

[32] Joachims T, Galor T, Elber R. Learning to align sequences: A maximum-margin approach[M]//New algorithms for macromolecular simulation. Springer Berlin Heidelberg, 2006: 57-69.

[33] Tsochantaridis I, Joachims T, Hofmann T, et al. Large margin methods for structured and interdependent output variables[J]. Journal of Machine Learning Research, 2005, 6(Sep): 1453-1484.

[34] Joachims T. Making large scale SVM learning practical[R]. Universität Dortmund, 1999.

[35] Joachims T. Learning to classify text using support vector machines: Methods,

theory and algorithms[M]. Kluwer Academic Publishers, 2002.

[36] Joachims T, Finley T, Yu C N J. Cutting-plane training of structural SVMs[J]. Machine Learning, 2009, 77(1): 27-59.

[37] Sun JK. Research and implementation of ranking based personalized recommender algorithms [D]. Shandong University, 2014.

[38] Liu N N, Yang Q. Eigenrank: a ranking-oriented approach to collaborative filtering[C]//Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval. ACM, 2008: 83-90.

[39] Croux C, Dehon C. Influence functions of the Spearman and Kendall correlation measures[J]. Statistical methods & applications, 2010, 19(4): 497-515.

[40] Puth M T, Neuhäuser M, Ruxton G D. Effective use of Spearman's and Kendall's correlation coefficients for association between two measured traits[J]. Animal Behaviour, 2015, 102: 77-84.

[41] Rendle S, Freudenthaler C, Gantner Z, et al. BPR: Bayesian personalized ranking from implicit feedback[C]//Proceedings of the twenty-fifth conference on uncertainty in artificial intelligence. AUAI Press, 2009: 452-461.

攻读硕士学位期间主要的研究成果

致谢

首先感谢实验室的各位导师：陈刚教授、寿黎但教授、陈珂副教授、胡天磊副教授、伍赛副教授。在研究生的两年多时间里，各位老师广博的专业见识、严谨的治学态度给我留下了深刻的印象，对我自身专业素养的提升有很大的启发和帮助。尤其要感谢陈珂老师，除了学术上的指导，还在平时的生活学习上给了我很多关心和照顾，同时也对我未来的职业规划方面给了建议和指导。另外要特别感谢胡天磊老师，在我个人的工程实践能力的提升方面给予了极大的帮助，同时也对我毕业论文的完成提供了耐心的指导。

其次要感谢实验室的各位师兄师姐，在科研实践以及求职方面都分享了丰富的经验。尤其是王铖微师兄，他花费了很多时间对我的毕业论文进行了指导，是他的无私帮助使得我能够顺利完成毕业论文，衷心地祝愿师兄未来能够工作顺利、身体健康。另外也要感谢各位同窗同学，和大家一起相处的时光里我收获了很多快乐，尤其在即将毕业的这一年里，同学们互相帮助、互相鼓励，一起找工作、写论文，希望大家的这份友谊能够永远保持下去，也希望同学们在未来的人生道路上都能一帆风顺。

最后要感谢浙江大学对我的栽培，转眼间我在母校已经度过了将近七年的时光。从当初孤身一人来到陌生的城市，到现在熟悉了这里的一切，浙大已经成了我的第二个家。这七年里的所有快乐、痛苦、进步、迷茫，都将深刻地印在我的记忆里，成为我未来人生道路上宝贵的财富。我会把母校的求是精神带到未来的工作生活中，发扬母校的优良传统，定不辱浙大之名。

署名

当前日期