

分类号 _____ 密级 _____
U D C _____

昆明理工大学 专业学位硕士学位论文

基于 Mahout 协同过滤算法在 KDD2010
比赛中的探索研究

研 究 生 姓 名 _____ 孟卓
指导教师姓名、职称 _____ 袁梅宇 副教授
学 科 专 业 _____ 软件工程
研 究 方 向 _____ 数据挖掘、机器学习
论 文 工 作
起 止 日 期 _____ 2014 年 3 月 ~ 2016 年 3 月
论 文 提 交 日 期 _____ 2016 年 3 月

学位论文出版授权书

我同意将本人学位论文著作权中的数字化复制权、发行权、汇编权和信息网络传播权的专有使用权在全世界范围内授予中国学术期刊（光盘版）电子杂志社（以下简称“杂志社”），同意其在《中国优秀博硕士学位论文全文数据库》和CNKI系列数据库中出版，未经杂志社书面许可，我不再授权他人以数字化形式出版本文。我同意《中国优秀博硕士学位论文全文数据库出版章程》规定享受相关权益。

如有任何第三方未经杂志社许可使用本人论文，杂志社应追究其法律责任，诉讼的全部费用由杂志社承担。胜诉后，由杂志社与本人按 5: 5 的比例分配所获赔偿金。

作者签名: 孟卓
2016年5月23日

学位论文作者信息

论文题目	基于Mahout 协同过滤算法在KDD2010 比赛中的探索研究				
姓名	孟卓	学号	2013704150	答辩日期	2016年5月21日
论文级别	博士 <input type="checkbox"/>	硕士 <input checked="" type="checkbox"/>			
院/系/所	电子信息工程与自动化学院	专业	软件工程		
联系电话			E-mail		
通信地址(邮编):					
备注:					

公开 保密 (____年____月至____年____月) (保密的学位论文在解密后应遵守此协议)

联系电话: 010-62791951 62793176 62790693 传真: 010-62791814

通信地址: 北京清华大学邮局 84-48 信箱 采编中心 邮编: 100084

学位论文使用授权书

本论文作者完全了解学校关于保存、使用学位论文的管理办法及规定，即学校有权保留并向国家有关部门或机构送交论文的复印件和电子版，允许论文被查阅和借阅。本人授权昆明理工大学可以将本学位论文的全部或部分内容编入学校有关数据库和收录到《中国博士/优秀硕士学位论文全文数据库》进行信息服务，也可以采用影印、缩印或扫描等复制手段保存或汇编本学位论文。

注：保密学位论文，在解密后适用于本授权书。

作者签名：孟卓

2016年5月23日

导师签名：袁梅宇

2016年5月23日

学院：信息工程与自动化学院

学号：2013704150

专业：软件工程

(一式三份，交研究生院学位工作处)

一 遵守学术行为规范承诺

本人已熟知并愿意自觉遵守《昆明理工大学研究生学术规范实施细则（试行）》的所有内容，承诺所提交的毕业和学位论文是终稿，不存在学术不端行为，且论文的纸质版与电子版内容完全一致。

二 独创性声明

本人声明所提交的论文是我个人在导师指导下进行的研究工作及取得的研究成果。尽我所知，除了文中特别加以标注和致谢的地方外，论文中不包含其他人已经发表或撰写过的研究成果，也不包含为获得昆明理工大学或其他教育机构的学位或证书而使用过的材料。与我一同工作的同志对本研究所做的任何贡献均已在论文中作了明确的说明并表示了谢意。本人完全意识到本声明的法律结果由本人承担。

三 关于论文使用授权的说明

本人完全了解昆明理工大学有关保留使用学位论文的规定，即：学校有权保留送交论文的复印件，允许论文被查阅和借阅；学校可以公布论文的全部或部分内容，可以采用影印、缩印或其他复制手段保存论文。（保密的论文在解密后应遵守此规定）

本学位论文属于（必须在以下相应方框内打“√”，否则一律按“非保密论文”处理）：

1、保密论文： 本学位论文属于保密。

2、非保密论文： 本学位论文属于内部论文，网上延后公开。

本学位论文不属于保密范围，适用本授权书。

是否同意授权以下单位（必须在以下相应方框内打“√”，否则一律按“同意授权”处理）：

同意授权 不同意授权

将本人学位论文著作权中的数字化复制权、发行权、汇编权和信息网络传播权的专有使用权在全世界范围内授予中国学术期刊（光盘版）电子杂志社，并在《中国优秀博硕士学位论文全文数据库》和 CNKI 系列数据库中出版。

研究生本人签名： 孟宇 签字日期：2016 年 5 月 23 日

研究生导师签名： 支柳宁 签字日期：2016 年 5 月 23 日

摘要

教育部于 2010 年出台了《教育信息化十年发展规划（2011-2020）》，提出了“到 2020 年，需形成与国家教育现代化发展目标相适应的教育信息化体系”。信息化教育已经成为影响国家现代化进程和时代发展的关键因素，其重要性可见一斑。本论文以 Mahout 协同过滤算法为手段，对 KDD Cup 2010 比赛数据集进行数据挖掘。KDD Cup 是国际顶级的数据挖掘赛事之一，2010 年其以教育数据挖掘作为竞赛命题。由此可见，本论文的研究方向具有很好的研究价值和实践意义。

本论文采用的数据集是从 ITS 智能导师系统中选取的 890 万条记录，该数据集具有以下几个难点：1. 数据量庞大：含 8918054 条记录，每条记录含 23 个特征，总共约 2 亿个值。2. 部分特征取值范围巨大（超过 10 万）。3. 数据的稀疏性：有些特征没有取值，或某些学生记录数过少。4. 具有强时间相关性，即学生的答题顺序是固定的。

协同过滤推荐算法是当前广泛应用到推荐系统领域的算法之一。它可以很好的解决个性化推荐的问题。但随着数据规模的递增，协同过滤算法也遇到了一些挑战：如数据稀疏性等问题。本文针对协同过滤算法进行了深入的研究，运用多类协同过滤算法对学生第一次尝试答题的正确与否进行预测，从而可以判断学生对问题的掌握程度，提高学习水平；并对比评价各类推荐方法的优劣，最后给出效果最优的推荐方法。

根据以上问题，本文的研究工作如下所列：

1. 对数据挖掘技术进行深入研究，对目前主流的数据挖掘方法进行学习，重点研究 Mahout 协同过滤推荐算法的相关知识；对教育数据挖掘发展现状和 KDD2010 比赛背景进行相关分析介绍。

2. 对协同过滤相关算法进行详细的比较分析。其主要包括三类算法：基于用户的协同过滤算法（User-Based CF）、基于项目的协同过滤算法（Item-Based CF）和基于模型的协同过滤算法（Model-Based CF）。

3. 应用 Apache Mahout 中的 taste 开源框架，针对 KDD2010 比赛数据集，运用三类协同过滤推荐算法分别进行仿真实验。使用 RMSE 评估标准，对相似度计算进行效果对比，从而选出最优的推荐算法。

关键词：协同过滤；数据挖掘；预测；KDD Cup 2010

ABSTRACT

The <Outline of China's National Plan for Medium and Long-Term Education Reform and Development (2010-2020)> pointed out that information technology has a revolutionary influence on education development, and must be attached great importance to. Educational informationization has been the key factor which affects the nation's modernization. The paper uses collaborative filtering as method, and doing data mining process to the dataset of KDD Cup 2010. KDD Cup is one of the world famous data mining competitions. It used educational data mining as topic in 2010. Therefore, the paper's research direction has very high value of practice.

The paper uses the data chosen from the Intelligent Tutor System, which contains 8.9 millions of data. The dataset has the following characteristics: 1. Large volume of data: There are 8918054 lines in the dataset, every line has 23 features, and there are about 200 billion values in total. 2. Huge scope of feature(over 450,000). 3. The data matrix is sparse: The contestants need to exploit relationships among problems to bring to bear enough data to hope to learn. 4. There is a strong temporal dimension to the data: the regular sampling method will make some mistakes.

The collaborative filtering recommendation algorithm has been widely used in recommendation system area. It can well solve personal recommendation problem. But with the increment of data, the collaborative filtering algorithm faces some challenges: some problems such as the sparse of data. This paper does deep research in collaborative filtering algorithm, using kinds of collaborative filtering algorithm to predict personal recommendation to students on problem items, and comparing these methods. Finally it gives the best recommendation.

Based on the questions above, the paper does some works below:

1. Making some deep learning about data mining technology. Learning the mainstream data mining methods, especially the knowledge about the Mahout collaborative filtering recommendation algorithm. Making analysis to the development of the educational data mining and KDD Cup 2010 competition.

2. Making analysis about the collaborative filtering algorithm. It mainly contains three

kinds of algorithms: User-Based collaborative filtering、Item-Based collaborative filtering and Model-Based collaborative filtering.

3.Using the taste frame in Apache Mahout to make simulation experiment with the three kinds of CF algorithms. Using RMSE value as the evaluative criteria to compare the recommendation effect. Finally choosing the best recommendation algorithm.

Keywords: collaborative filtering; data mining; prediction; KDD Cup 2010

目 录

摘要	I
ABSTRACT	II
目录	V
第一章 绪论	1
1. 1 选题背景	1
1. 1. 1 关于教育数据挖掘	1
1. 1. 2 关于协同过滤技术	2
1. 2 选题目的和意义	3
1. 3 论文结构	5
第二章 相关理论概述	7
2. 1 基本概念	7
2. 1. 1 数据挖掘与机器学习	7
2. 1. 2 数据和数据集	7
2. 2 预处理	8
2. 3 数据挖掘的常用算法	12
2. 3. 1 分类与回归	12
2. 3. 2 聚类分析	14
2. 3. 3 关联分析	14
2. 4 Apache Mahout 介绍	15
2. 5 协同过滤推荐算法	17
2. 5. 1 基于用户的协同过滤推荐算法	17
2. 5. 2 基于项目的协同过滤推荐算法	19
2. 5. 3 基于 SVD 的协同过滤推荐算法	20
2. 5. 4 相似度计算方法分析	22
2. 5. 5 协同过滤算法的特点	24
第三章 预处理过程	29
3. 1 问题定义与数据集分析	29

3.1.1 KDD Cup 2010 比赛命题	29
3.1.2 对相关特征数据的统计	30
3.2 数据集抽取	31
3.2.1 抽取方法	31
3.2.2 数据清理	33
第四章 实验的设计与效果评估	37
4.1 算法评估	37
4.2 基于 Apache Mahout 的算法实验	37
4.2.1 基于用户的协同过滤推荐 (User-Based CF)	39
4.2.2 基于项目的协同过滤推荐 (Item-Based CF)	40
4.2.3 基于 SVD 的协同过滤推荐 (SVD-Based CF)	41
4.2.4 三种相似度的计算方法	42
4.2.5 实验结论	47
第五章 总结与体会	49
5.1 论文主要工作总结	49
5.2 今后所要开展工作	50
致 谢	51
参考文献	52
附录 A 攻读硕士期间发表论文以及软件著作权	55

第一章 绪论

1.1 选题背景

1.1.1 关于教育数据挖掘

早在二十世纪八十年代，人们便已经将数据挖掘作为一个课题进行分析，发展至今，其对金融工商业与市场营销等行业的重要性也越来越明显；同时可汗学院等实时与远程教育模式的诞生，教育行业逐渐累积了大量的信息资源。怎样才可以在庞大且杂乱无章的信息海内获得存在一定意义的数据，交由教育专家与相关人员进行处理，建立正向循环教育体系，以此达到增强学生学习能力与有效管理的目标。在这样的形势下，有关教育数据挖掘的分析也随之应运而生。

2005 年，人工智能会议(AAAI)、人工智能教育应用会议(AIED)等国际会议针对“教育数据挖掘”这一课题进行了大量的研究与分析。在 2008 年，加拿大蒙特利尔地区正式举行了首届教育数据挖掘国际学术会议。2009 年，在第五届数据挖掘教育应用高级会议中增添了“数据挖掘在教育中的应用”这一内容，标志着正式把这一内容包含在数据挖掘课题中进行系统研究。2010 年，数据挖掘研究组织 (SIGKDD) 在国际顶级赛事 KDD CUP 中将教育数据挖掘作为竞赛赛题，2012 年 10 月美国教育部发布了《通过教育数据挖掘方法提高施教和学习》，帮助全部重点高校规划出“大数据”应用教育发展的目标，与此同时我国在教育数据挖掘领域也实现了较大的提升与进步^[1]。

教育数据挖掘 (Educational Data Mining, 即 EDM) 其本质是通过数据挖掘技术完成教育系统中相关信息的处理，以此获得部分具备一定意义与帮助的数据，最终得到的数据往往存在相应的格式，能够让教育学者、学生以及教育系统研发人员等与教育行业存在联系的人们受益^[2]。EDM 各个层面的研究存在各种各样的类型，站在 EDM 分析领域的层面而言，其本身包含“在教学研究中的应用”与“在教务管理中的应用”两种类型；若是站在数据获取途径的层面而言，则包含“在传统教育中的应用”与“在网络教育中的应用”两种类型，基于上述两类模式，实现 EDM 分析事项的细分处理^[1]。EDM 各研究方向如图 1.1 所示。

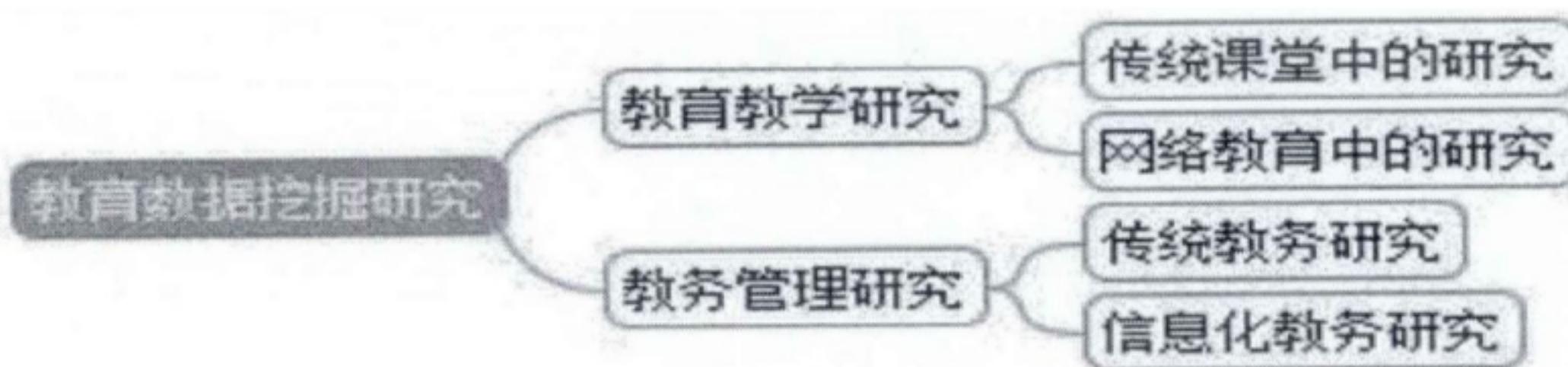


图 1.1 EDM 研究内容的划分

本论文的主题就是教育数据挖掘，其详细深入的理论研究阐述将在第二章进行介绍。

1.1.2 关于协同过滤技术

自 1992 年 Xerox 公司应用协同过滤系统设计的 Tapestry 解决在 Palo Alto 的研究中心资讯过载的问题开始^[3]，协同过滤算法在随后的 20 多年中已广泛的应用于商业推荐系统之中。“协同过滤”的发展主要经历了三个里程碑事件：1992 年设计的 Tapestry 邮件过滤系统、1994 年设计的 GroupLens 系统^[4]和现今的电子商务的推荐系统。其发展已经完成了从单一文件过滤、多领域跨系统到目前主流的电子商务领域三类系统的跨越。虽然在最终想要实现的目标上存在差异，但其便捷性与使用价值没有很大的不同。

推荐领域专家 F. Heylighen 给出的协同过滤的定义是这样的：“协同过滤（Collaborative Filtering，即 CF），其本质是通过相同兴趣爱好与经验的群体的意见，带来更多满足消费者需求与认可度的产品与信息，系统内所有使用者经过对商品的体验，利用相应的体系进行公平公正的评价（评论、评分等），通过对评价内容的整理可以使人们获得具备一定价值的信息。所评价的内容并非只是挑选有兴趣的商品或者给予单一的表扬，对于批评以及没有任何兴趣的反馈也需要十分注意，原因在于此类信息也可以帮助我们了解到十分关键的特征。协同过滤按照功能可以划分为评比过滤（rating）或者群体过滤（social filtering）”^[5]。

目前，随着个性化推荐系统持续有效的发展，相关研发工作者需要结合各种各样的状况形成符合个性化特点的操作模型。协同过滤推荐算法结合处理模式进行分类，主要包含建立在记忆前提（Memory-based）下以及建立在模型前提（Model-based）下两种协同过滤模式。以记忆为基础的协同过滤又包括基于使用者（User-based）的协同过滤和基于项目（Item-based）的协同过滤^[6]（“项目”代表着使用者的各项特征）。

●基于用户的协同过滤

又称为基于邻近者的协同过滤 (Neighbor-based Collaborative Filtering)，主要运用相似统计的方法得到具有相似爱好或者兴趣的相邻使用者。这里所说的相似用户一般情况下被叫做最近邻居^[7](Nearest Neighbor)。而对于用户的一致性或者相似性，仅仅需要考虑使用者现实表现中哪些地方可以达成一致，必须经过周密的运算了解它们在项目行为表现中存在的一致性。目前，存在大量方法可以有效求得使用者间存在的一致性，而普遍应用的模式则主要为皮尔森相关系数与完善后的余弦相似性等。

●基于项目的协同过滤

2001 年，Sarwar 成功验证了以项目为主体的协同过滤推荐算法^[8]。这种算法按照某种假设状况，也就是“用户比较认可的对象，和良性评价的对象之间在某种程度上存在着一致性”，如此一来便能够通过求出对象间存在的一致性了解用户间具备的一致性。

●基于模型的协同过滤

以记忆为主体的协同过滤算法不足之处在于：当保留的信息量过少时，最终表现无法满足标准需求，因此，当分析的数据达到一定规模时会使最终的结果存在偏差，无法达到最佳的效果，所以经过延伸得到以模型为主体的协同过滤算法。该算法首先用历史记录资料得到一个模型，再用此模型进行推荐计算。

以上这三类协同过滤算法几乎涵盖了当今所有主流的协同过滤算法，其包含的所有协同过滤算法总共有十几种。这里不可能详细地阐述每一种算法的原理与实现。由于协同过滤算法是本论文建立模型的主要理论来源，所以在论文中从每一类算法中都选取一项最有代表性和应用最广泛的算法，并使用其建立模型。对每一子类的协同过滤算法的详细阐述将在第四章进行介绍。

1.2 选题目的和意义

在教育部 2016 年印发的《教育信息化“十三五”规划》中提到，加快推动信息技术与教育教学融合创新发展。充分利用市场机制建设在线开放课程等优质数字教育资源，推进线上线下结合的课程共享与应用。进入二十一世纪，数据挖掘技术成为当今最炙手可热的技术之一。国际知名的 IT 企业几乎全部建立起了自己的数据库系统，并正在展开相关的数据挖掘各项课题，并取得了显著的经济效益。在线教育系统中的题目与题目之间，课程之间的影响等信息，通过数据挖掘技术能够有效的提取出来，

并且能够更生动具体的加以展现，进一步能够以全新的方式描述学生的学习效果。这些都具有着极其重要的价值。

数据挖掘技术涉及到的范围十分广泛，其中很有代表性的就包括协同过滤技术。协同过滤技术在商业推荐领域已经得到了十分广泛的应用，其相关的理论和应用研究也已经十分成熟。但到目前，协同过滤技术应用到教育领域的案例还十分稀少，取得的研究成果还处于初级的探索阶段。如果将协同过滤技术应用到教育数据挖掘中来，就能够对构建教育教学系统和探索全新的教育教学模式提供许多重要的理论依据和实现手段，这就是本论文的选题意义所在。同时本论文选取的数据集来源于 KDD Cup 2010 比赛数据集，由于 KDD Cup 在国际数据挖掘界属于比较有影响力的比赛，所以实验结果拥有较高的权威性以及实用价值。

2010 年 7 月 25 日至 28 日在美国华盛顿召开了第十六届知识发现和数据挖掘国际会议(the 16th ACM SIGKDD Conference on Knowledge Discovery and Data Mining)。本届比赛的主题是根据智能教学辅导系统和学生之间的交互日志，来预测学生数学题的考试成绩。该任务兼具实践性和科学趣味性。竞赛提供了 3 个开发 (develop) 数据集和 2 个挑战 (challenge) 数据集，每个数据集又分为训练 (train) 部分和测试 (test) 部分。Challenge 数据集的 test 部分被隐藏，参赛者需要开发一种学习模型，来准确预测这部分隐藏部分的成绩。2 个挑战集的相关数据如表 1.1 所示。

特征名\数据集名	Algebra 2008-2009	Bridge to Algebra 2008-2009
Lines(train)	8,918,054	20,012,498
Students(train)	3,310	6,043
Steps(train)	1,357,180	603,176
Problem(train)	211,529	63,200
Section(train)	165	186
Units(train)	42	50
KC(train)	2,097	1,699
Steps(new on test)	4,390	9,807

表 1.1 两个挑战集的特征统计表

由于时间的关系以及对问题的复杂度没有量化的判断，本论文所使用的数据集是 Algebra 2008-2009。论文主要建立的是 students 与 steps 之间的模型，并且通过分析各算法在预测学生首次尝试答题正确率的效果进而评判各算法的优劣。

1.3 论文结构

本论文的组织结构如下：

第一章：绪论。对教育数据挖掘的发展及其核心研究内容进行了探讨；同时对协同过滤推荐算法的衍化历程与子类算法有关内容进行了说明；依靠协同过滤推荐算法与教育数据挖掘选择论文的分析目标；说明论文选题的目的和意义。

第二章：理论与实验基础。对论文包含的理论基础进行说明，主要有：数据挖掘和机器学习对应的基础内容、数据预处理模式、数据挖掘普遍使用的算法种类。之后介绍与本文相关的推荐技术以及实验背景，其中主要介绍 Apache Mahout 开源框架、协同过滤推荐算法的原理公式介绍等。

第三章：数据预处理。介绍实验所涉及到的 KDD Cup 2010 比赛记录如何对实验数据及进行预处理，包括：数据清理、特征选取、抽样方法等。

第四章：对仿真实验及最终结果进行说明。结合 Apache Mahout 内部开源 Taste 组件完成了以使用者为主体、以项目为主体和以模型为主体的协同过滤推荐三种模式的仿真验证，同时将所得结果进行比较研究。

第五章：总结与展望。对论文中涉及的所有内容与最终的结果进行总结，深刻分析论文创新之处，同时对后期的工作内容进行规划。

第二章 相关理论概述

2.1 基本概念

2.1.1 数据挖掘与机器学习

数据挖掘与机器学习之间的联系十分密切。后者所采用的处理方式作为前者的核
心基础，而前者所使用的大量技术均属于对后者的学习与借鉴，同时推动着后者超更
高的标准发展。

数据挖掘实质上属于在大量信息中获得某种规律模式^[9]。这一阶段重点依靠自动
与半自动模式完成，同时需要确保信息量达到一定的规模，通过挖掘所得到的模式存
在相应的分析与商业意义。通常，数据挖掘需要分析数据库中的数据来解决问题，如
客户忠诚度分析、市场购物篮子分析，等等。

当今已经入海量数据时代。例如，全世界已经有约 1000,000,000,000 个网页；沃
尔玛仅一个小时就有一百万的交易量，其数据库里数据已有 2.5 拍（即 2.5×10^{15} ）字
节的信息量，等等。这些海量数据不可能采用手工方式进行处理，因此，迫切要求能
进行数据分析的自动化方法，这些都由机器学习提供。

机器学习定义为能够自动寻找数据中的模式的一套方法，使用所发现的模式来预
测将来的数据，或者在各种不确定的条件下进行决策。机器学习分为两种主要类型。
第一种称为有监督学习，或称为预测学习，其目标是在给定一系列输入输出实例所构
成的数据集的条件下，学习输入x到输出y的映射关系。这里的数据集称为训练集，实
例的个数称为训练样本数。第二种机器学习类型称为无监督学习，或称为描述学习，
即数据集中只包含相关实例，而且对于挖掘的目的没有任何概念，也没有特别的差
别度量帮助处理。而对于给定的x，有监督学习可以对所观察到的值进行比较。

2.1.2 数据和数据集

按照应用间存在的差异，若数据挖掘选择的目标中包含各种各样的信息，则最终
保管上述信息的设备同样存在较大的区别，如数据库、网页模式等^[10]。它们既可以被
集中存储在数据存储器或内存中，也可以分布在世界各地的网络服务器中。

一般情况下，我们把需要分析的综合型数据定义为数据集。在时间的推移下，所
选择的数据往往包含大量别名，如“记录”、“案例”等称呼。数据对象也是对象，

因此，可以用刻画对象基本特征的属性来进行描述。属性也有多个别名，如“变量”、“特征”、“字段”、“维”、“列”等等。

数据集可以类似于一个二维的电子表格或数据库表。在最简单的情形下，每个训练输入 x_i 是一个 N 维的数值向量，表示特定事物的一些特征，如人的身高、体重。将上述特征确定成属性概念，并且 x_i 能够作为复杂结构的目标，如图像、语句等。

属性可以分为四种类型：标称（nominal）、序数（ordinal）、区间（interval）和比率（ratio），首先，标称中确定的值代表保存于内部各种称谓信息，同时能够将标称值认为只是实现各种对象种类划分的有关数据^[11]，如性别（男、女）、衣服颜色（红、黄、蓝）、天气（阴、晴、雨、多云）等；序数属性的值可以提供确定对象的顺序的足够信息，如成绩等级（优、良、中、及格、不及格）、职称（初职、中职、高职）、学生（本科生、硕士生）等；而区间则重点了解值与值间出现的差，也就是测量单位，如温度区间、年月日区间等；比率属性值的差对于分析而言存在十分重要的意义，如数量比、长度比等。

标称与序数两种属性并未包含数的绝大多数特征，同时需要通过集合形式保存与获得取值，故而可以把这两种属性确定成分类型属性，通常条件下，人们往往把它们确定成符号实现操作，而没有通过仿真模式予以验证；区间与比率两种属性因为包含数据绝大多数特征，同时可以通过定量形式说明相应的属性，故而，可以把这两种属性确定成数值属性。

2.2 预处理

数据挖掘的实质是以一定规模的、包含潜在价值的信息量为基础，获得具备一定意义模式的流程。所以，数据源的合理性直接决定着数据挖掘是否可以实现预期的目标，合理挖掘的基础在于整理所得信息的质量^[12]。但是，由于数据挖掘所使用的数据往往不是专门为挖掘准备的，期望数据质量完美并不现实，人的错误、测量设备的限制以及数据收集过程的漏洞都可能导致一些问题，如缺失值和离群值。

由于无法在数据的源头控制质量，数据挖掘只能通过以下两个方面设法避免数据质量问题：1数据质量问题的检测与纠正；2使用能容忍低质量数据的算法。第一类模式即数据挖掘阶段需要检验与解决部分质量存在的不足，也就是数据预处理；第二类模式必须增强算法的适应性。

数据预处理是数据挖掘的重要步骤，数据挖掘者的大部分时间和精力都要花在预

处理阶段。数据预处理涉及的策略和技术非常广泛，主要包括如下技术：

1)聚集

这种方式可以把两个以及以上的类经过统一整理形成全新的类。通常条件下，处理定量数据源时能够利用获得加权平均以及方差的形式完成聚集，定性类数据则可以利用综合整理完成聚集。采取数据归约模式可以让数据量实现相应的缩减，如此一来，在处理规模一般的数据集时便能够节省更多的时间与精力，因此，我们认为这种技术可以在资源需求过高的挖掘算法中进行使用。而且，因为采取高层数据视图进行处理，可以让聚集呈现出最佳度量尺度。如此一来，便能够防止检视阶段只是对总体模糊数据进行了解，而忽视了细节问题。

2)抽样

若是因为信息量庞大占用过多的时间资源，可以采取抽样模式进行有效的处理，以此获得数据子集完成相应的分析。抽样实质上属于对数据量进行缩减处理，故而，运用这种模式也可以满足消耗较高的挖掘算法的需求^[13]。因抽样作为统计期间的子类，故而经统计得到的样本，其有效性直接可以体现抽样的价值，我们能够理解为样本和初始数据之间特征一致，样本能够作为初始数据进行相应的操作^[14]。

抽样模式包含大量种类，而相对容易实现的子类属性标记任意记录在抽取成样本时，其概率没有任何差别，即简单随机抽样。这种抽样形式根据回归初始数据行的形式可以再次区别成两种。一种是有放回抽样，样本子集主要包含在K个数据行内抽得k个数据行，同时出现的概率均是1/K；另一种是无放回抽样，和前一种操作步骤基本类似，只是通过这种模式进行处理必须与抽样完成时清理初始数据内部数据行。因此结合定义，前一种方法在操作阶段容易获得与初始数据处在同一行的相同数据，导致结果出现偏差。

因为数据集内部信息多种多样，采用随机的模式进行处理时容易造成部分冷僻的信息行始终得不到抽取，如此一来，造成最终得到的样本并不全面，此刻便需要采取分层抽样模式（Stratified Sampling）进行操作。运用该模式第一步便是运用事先整理的数据，同时若想提升样本的有效性需要采取相应的方法让样本与整体实现统一。采取该模式进行处理，首先需要根据特点完成数据集的分类，得到大量没有任何联系的“类”，接着通过简单随机抽样模式，形成全面合理的抽样数据子集。

3)维度归约

维度代表属性量。那么维度归约则代表形成全新属性，利用对信息的编码与转换，

让部分旧属性经过整理达到减少集合维度的目标^[15]。采用这种模式能够清理没有联系的属性对噪音进行控制，维度减小可以让大量数据挖掘算法作用更加明显，同时可以避免维灾难引起结果偏差。维灾难代表，在维度上升阶段，信息使用空间变得十分稀疏，如果是分类问题，则代表模型建立阶段失去了数据对象的支持；如果是聚类问题，点与点的密度、距离将没有任何价值。因此，对于高维数据，许多分类和聚类等学习算法的效果都不理想。维度归约使模型的属性更少，因而可以产生更容易理解的模型。

4) 属性选择

除了上述模式，我们也可以采取仅仅利用属性某个子集的方式达到减小维度的目的。虽然该模式容易让人们误解为丢失数据，然而绝大部分情况下，数据集表现出冗余以及没有关联的属性。前者代表某属性中存在不同属性的相关数据，而后者代表所选择的挖掘对象中并不具备有价值的数据。确定属性代表在数据集内选择出典型的属性子集，清理没有联系的属性，以此节省信息处理的时间，让模型定义更加方便^[16]。

确定属性时比较容易实现的模式是结合常识进行处理，从而清理部分没有联系的属性，然而，若想确定出合理的属性子集，则必须依靠系统全面的模式。确定合理的属性时我们可以按照以下流程进行处理：把所有潜在的属性子集确定成数据挖掘学习算法的子集，接着确定出可以得到理想结果的对象。该模式能够表现出后期采取的数据挖掘算法的认可度。然而，因为n个属性子集存在 2^n 个，故而使用率偏低^[17]。因此，需要考虑三种标准的属性选择方法：嵌入、过滤和包装。

嵌入方法（Embedded Approach）是把属性确定包含在数据挖掘算法中。采取挖掘算法处理阶段，算法可以表明属性是否存在价值。决策树算法通常使用这种方法。

过滤方法（Filter Approach）即采取数据挖掘算法操作时，通过其他模式完成属性的确定，也就是对数据集进行过滤处理形成全新属性子集。

包装方法（Wrapper Approach）是把学习算法所得结论包含在评价标准中，采取此前说明的最佳算法，然而，无法呈现所有具备价值的子集，最终得到的属性子集失去代表性。

按照属性确定期间有无采取类别信息，能够将其确定成有、无监督两种属性确定模式^[18]。前者利用度量类别信息与属性之间的相互关系来确定属性子集，后者不使用类别信息，使用聚类方法评估属性的贡献度，根据贡献度来确定属性子集。

5) 属性创建

即采取综合整理的模式找到其中的旧属性，形成全新数据集，如此一来可以更加

快速的了解其中比较重要的数据。一般情况下，新数据集其维度相对较低，所以，能够利用维度归约提升效果。形成全新属性的模式分为三类：属性捕捉、映射信息至新空间以及属性建造。

属性捕捉代表在初始数据的基础上形成全新属性集。如，分析照片数据时，获得具备一定价值的特征，如和人脸十分类似的边、区域等，便能够借此采取其他分类模式进行处理。

映射信息至新空间，代表站在某种差异明显的角度进行数据挖掘时将会产生具备一定价值的特征。如，采用傅里叶转换处理时间序列，改变成频率数据，将能够得到具备价值的周期格式。

如果初期数据集属性存在价值明显的内容，然而无法运用数据挖掘算法进行处理，此时便需要采取属性建造方式，在过去属性基础上形成全新属性。

6) 离散化和二元化

部分数据挖掘算法，特别是分类型算法，十分注重分类属性的意义。确定关联形式的算法强调数据必须符合二元属性要求。因此，需要进行属性变换，将连续的属性转换为分类属性称为离散化（Discretization），将连续和离散属性转换为一个或多个二元属性称为二元化（Binarization）。

连续属性离散化包含两类子任务：首先是表明分类值的具体数目，其次便是把连续属性值映射至上述分类值内^[19]。所以，操作阶段必须指出有效分割点的数量，同时指出分割点所在地。采取小部分分类值标签进行处理，以此尽可能让初始数据更加直观。

离散化技术按照有、无类别数据这一特征，可以确定成两种：前者即有监督离散化，后者则是无监督离散化。其中后者又包含等宽和等频离散化两类处理模式^[20]。等宽（Equal Width）离散化可以把属性值域划分成统一宽度的区域，其中数量按照使用者需求自由处理。该模式往往会导致分布差异化。等频（Equal Frequency）离散化即等深（Equal Depth）离散化，其本身是将具备统一数目的对象置于所有区域内，其中数量按照使用者需求自由处理。

7) 变量变换

变量变换（Variable Transformation）即属性变换，可以实现对变量全部值的转换。下面讨论两种重要的变量变换：简单函数转换和规范化。

简单函数转换属于采取某简单数学函数使其对所有值产生影响。对于统计学而

言，通过平方根、对数变换等方式进行处理，可以让数据获得高斯分布特征^[21]。

变量规范化（Standardization）可以让值的集合存在独立的特征。如，假定 \bar{x} 属于一属性平均值， s_x 作为它的标准差，那么转换公式 $x' = (x - \bar{x}) / s_x$ 所得变量可以将均值0与标准差1包含在内。而离群点会使均值与标准差出现明显的改变，所以，往往必须调整以上转换。如让中位数（Median）作为均值，以绝对标准差作为标准差等。

2.3 数据挖掘的常用算法

2.3.1 分类与回归

分类（Classification）与回归（Regression）是数据挖掘应用领域的重要技术。前者可以根据当前具备的数据完成某一分类函数的学习以及建立相关模型，即人们理解的分类器（Classifier）^[22]。这种模型可以将数据综合再映射至相应的类别之中，以此达到预测的目的。分类与回归作为预测主要方式，前者提供离散值，后者则提供连续值。

分类时，必须对数据集进行处理，使其分成训练集与测试集。分类主要包含两个流程，首先是对训练集进行研究同时创立分类模型，现在主要通过决策树、贝叶斯分类器、k-最近邻分类等分类模型进行处理；其次通过分类模型区分测试集，判定分类模型有效性等标准是否合理，最终确定合理的模型。

分类模型学习方法主要分为以下几类。

1) 决策树分类

这种模式需要首先完成训练集的训练工作，得到一棵二叉及以上的决策树。其本身主要存在三个结点，根节点处尚未入边，只是出现大量零条与出边；内部节点仅具备独立入边与两条及以上的出边；叶节点仅存在独立入边，无出边。其叶节点可以作为类别值，而非叶节点则作为某点至叶节点相应路径建立的独一分类规则，利用决策树可以快速获得大量规则，分析人员能够依照这种规则快速完成各种类别样本的合理估算^[23]。主要方法如下，由根节点处出发，依照测试要求完成样本的检验，结合最终的结果确定有效分支，顺着这一分支至另一个内部节点，再次使用新的测试规则：要么到达叶节点，结果是将叶节点的类别标号赋值给检验样本。

决策树所得学习算法需要实现对以下两个问题的处理。

第一，怎样分裂训练样本集？随着时间的推移，树也在不断发展，所有递归步需

要将某种属性确定成预测的标准，把样本集区别成小型子集。若想完成这一步，算法需要参照各种属性给予检测标准的模式，同时给予判定各检测标准的最佳度量。

第二，怎样结束分裂？必须具备相应的限制因素，使决策树停止发展。理想的方案时始终分裂，最终让各个样本达到相同类别，或让全部样本属性达成一致时停止。也可以使用其他策略提前终止树的生长过程。

不同决策树采用的技术不同，已经有很多成熟而有效的决策树学习算法，如 ID3、C4.5、CART、Random Forest 等。

2) 贝叶斯分类

这种模式存在指定的基础概率模型，从而确定样本作为相应类别标签的几率。这种模式可以依靠两种方式完成：朴素贝叶斯分类器与贝叶斯网络^[24]。前者建立在贝叶斯概念中统计分类模式的前提下，需要假设属性间不存在任何干扰，然而现实中无法真正达到这个标准。这种模式处理效率与有效性均符合相应的标准，满足增量学习需求。而后者主要通过贝叶斯网络对属性间存在的依赖性进行说明。

3) k-最近邻分类

前面所介绍的决策树分类器是一种积极学习器（Eager Learner），因为只要训练集数据可用，就开始学习从输入属性到类别标签的映射模型。另一类方案需要延长训练模型形成的时间，只有在分类测试样本阶段才实施后续的操作，我们将此方案命名为消极学习器（Lazy Learner）。这种算法主要通过该方案进行处理，属于以实例为主体的学习算法，无须提前采取训练样本形成分类器，依靠训练集便可以完成测试样本的分类处理，最终完成类别标签的设置。

这种模式通过有关训练实例实现预测，无须考虑在数据集内获得的模型。此类以实例为主体的学习算法必须通过邻近性度量了解实例中存在的类似度，同时让分类函数结合所检测的实例和相关实例邻近性推导出所检测实例间存在的类似度。即便消极学习方法无须考虑模型，但是，需要过多的投入才能完成测试实例分类处理，原因在于必须依次求得测试与训练两种样本间存在的类似度^[25]。反之，积极学习方法虽然需要运用庞大的资源得到模型，然而只要形成模型，便可以快速实现分类。最近邻分类器以部分数据为主体完成预测，但是决策树分类器需要确定满足输入空间的总体模型。因为建立在局部分类模式这一前提下，一旦 k 值减小，噪音将异常敏感。

4) 神经网络分类

神经网络（Neural Network）即各种简单神经元按照相应的规则组合得到的网络

系统，主要模拟人类大脑的组成与特征。通过相应的学习算法在训练样本内提取知识，接着把知识保存于网络模型权值内，与人类大脑一般，在利用相同脉冲循环刺激，使神经元间神经键接合强度发生变化，最终达到学习的目的。

根据神经元连接模式的差异，神经网络可以确定成前向与反馈两种网络形式。现在的神经网络模型多种多样，具有代表性的当属感知器模型、Hopfield 网络等几类。

2.3.2 聚类分析

聚类（Clustering）就是将数据集划分为由若干相似实例组成的簇（cluster）的过程，使得同一个簇中实例间的相似度最大化，不同簇的实例间的相似度最小化。可以理解为簇是将相互类似的一组对象经过处理得到的集合，而各类簇内部的实例往往没有明显的相似性。

聚类研究作为数据挖掘与机器学习至为关键的技术中，在很多领域均可以使用，如统计学、模式辨认等^[26]。

聚类本身属于无监督机器学习模式，在数据挖掘期间起到十分关键的作用，可以结合样本间相似性度量规范让数据集形成数个簇，而内部的簇并非提前设置的，而是结合相关数据的特点利用其中存在的相似性进行设置。这种算法的输入代表着样本与度量样本间相似性的规范，输出则代表簇集合。聚类研究同时能够完成所有簇的整体说明，此结果在研究数据集的表现形式时至为关键。聚类方法适合用于讨论样本间的关联，从而能不吃不评价其样本结构。

这种算法具备的特点是：能够对类属性进行处理、使大规模数据集得到延伸、能够获取所有形式的簇、能够对孤立点以及“噪声”信息进行处理、以约束为前提等几类。聚类研究包含：划分、层次、以密度为主体、以网络为主体等几类方式。

2.3.3 关联分析

随着时间的推移，运营中的公司往往累积着大量的信息。如超市需要对每日的消费者消费情况进行记录，也就是我们所说的购物篮状况。商家对分析这些数据很感兴趣，因为分析它可以了解顾客的购买行为，关联分析（Association Analysis）方法就是用于发现隐藏在大型数据集中有意义的联系，这种联系可以用关联规则（Association Rule）进行标识^[27]。比如，商家可以利用这种方法获取商场运营信息，了解消费者的重点需求，如消费者不仅选择了商品 X 而且选择了 Y，那么，可以在总体格局上进行

改变，让商品 X、Y 处在同一块区域，提升营业额。因此，关联分析为商场进行商品促销以及货架摆放提供了辅助决策信息。

例如，亚马逊从销售数据中发现这样一个令人匪夷所思的规则：

{电影}——>{红酒}

根据这种规则可知电影与红酒在营销过程中有着紧密的关联。理由是，美国群众十分喜爱观看电影，在受到爱情电影的感染的同时他们本能地想同家人亲密约会，这时候红酒就在其中起着调节氛围的作用。由于对这种情况的及时了解，为商家带来了商机，让“电影”与“红酒”两种原本没有任何关系的产品处于同一块区域内，实现了营业额的大幅度提升。

不仅仅只是购物篮信息这一点，这种分析方式也能够帮助其他行业获得发展，如生物信息学、医疗诊断等方面。比如，利用这种方式处理医疗信息，能够获知症状、疾病两者间存在的关联，让医生对病情实现更加快速、合理的了解。

关联分析中为人们所熟知的当属 Apriori 算法，这种模式属于以两阶段频繁项集思想为前提形成的递推算法^[28]。获得大规模项集（频繁项集）宗旨是：必须完成数据集的一系列操作。首先，初步统计全部独立元素项集对应的频数，删选得到部分项集必须超过最小支持度，也就是一维最大项集。其次便是重复进行筛选，确保不会出现从最大项集位置。重复具体步骤是：第 k 步，按照第 k-1 步得到的 (k-1) 维最大项集获取 k 维候选项集，接着查找信息库，了解候选项集支持度，和最小支持度对比，以此得到 k 维最大项集。

2. 4Apache Mahout介绍

Apache Mahout^[29]源自 ASF（Apache Software Foundation 的简写）推出的开源计划，能够让研发者在处理智能应用程序时节省大量的时间，也能够支持可延伸机器学习中典型算法进行处理。所含算法主要有聚类、分类等几类。

本文主要利用它的 Taste 组件进行处理。这种组件可以支持协同过滤算法的处理模式，具备研发者完成可延伸性编程时相应的接口。这种组件的功能表现是，能够建立典型的以使用者为主体的协同过滤算法和以内容为主体的协同过滤算法与 SVD 算法，而且拥有延伸性接口，让研发者能够完成推荐算法相应的操作。这种组件的出现，可以确保公司在实际操作阶段所使用的推荐引擎满足灵活性、延伸性等标准。

Taste 架构的主要构成如图 2.1 所示：

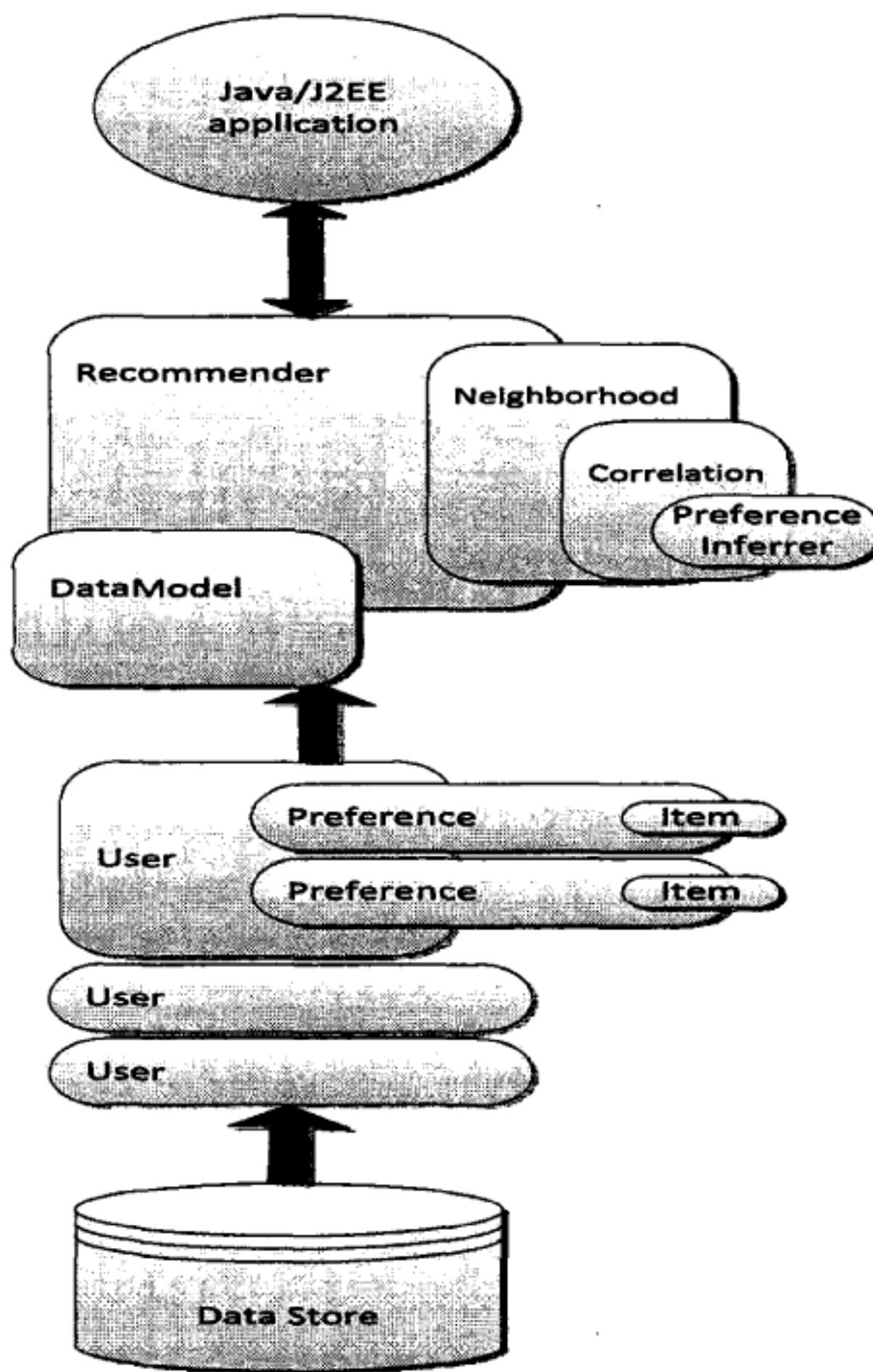


图 2.1 Taste 组件的主要组成

Taste 组件主要由 5 个部分组成：

DataModel: 作为使用者比较认可的数据抽象接口，可以对使用者比较认可的数据进行保存，属于该组件的信息模型，其默认数据存储分为数据库存储（默认为 Mysql 数据库）和文件存储两种方式，支持读取任意类型的数据源。

UserSimilarity（使用者相似度）：可以说明各个使用者间存在的相似度，从而求出使用者“最近邻居”，属于协同过滤引擎不可缺少的环节。

ItemSimilarity（项目相似度）：表示项目于项目之间的相似度。

UserNeighborhood: 能够让操作者获得“邻居用户”的基本模式，以建议的使用者相似度标准为前提，能够结合操作者给项目的评分找到“最近邻居”的最佳推荐结

果，主要建立在 UserSimilarity（用户相似度）计算这一前提下得到相应的处理方式。

Recommender: 同样属于该组件不可缺少的一部分，作为推荐算法抽象接口。对于推荐系统而言，首先需要为 Recommender 提供一个 DataModel(数据)，Recommender 可以针对不同的用户计算出不同的推荐结果。

2.5 协同过滤推荐算法

2.5.1 基于用户的协同过滤推荐算法

这种算法作为推荐算法中发展时间较长的一类，它的诞生和推荐系统的形成在时间上几乎一致。2002 年由专家指出，使邮件筛选系统性能得到了较大的提升，2004 年，通过研究组织 GroupLens 的努力，使其在新闻筛选中发挥出极佳的效果。而在 2010 年，这种算法依然属于推荐系统中广泛运用的一种模式，这种算法的实现并不复杂：第一步整理项目中使用者比较认可的信息，以此通过使用率较高的“K-邻居”算法求出和使用者认可度一致的“最近邻居”用户群，主要包含 K 个最近邻居所有认可数据，接着整理上述 K 个邻居用户的认可信息相似的特征，结合相似度，最后把非一致认可的数据提供给使用者，认可度一致性较高，那么非一致认可的数据在推荐中越有效，基本原理如图 2.2 所示：

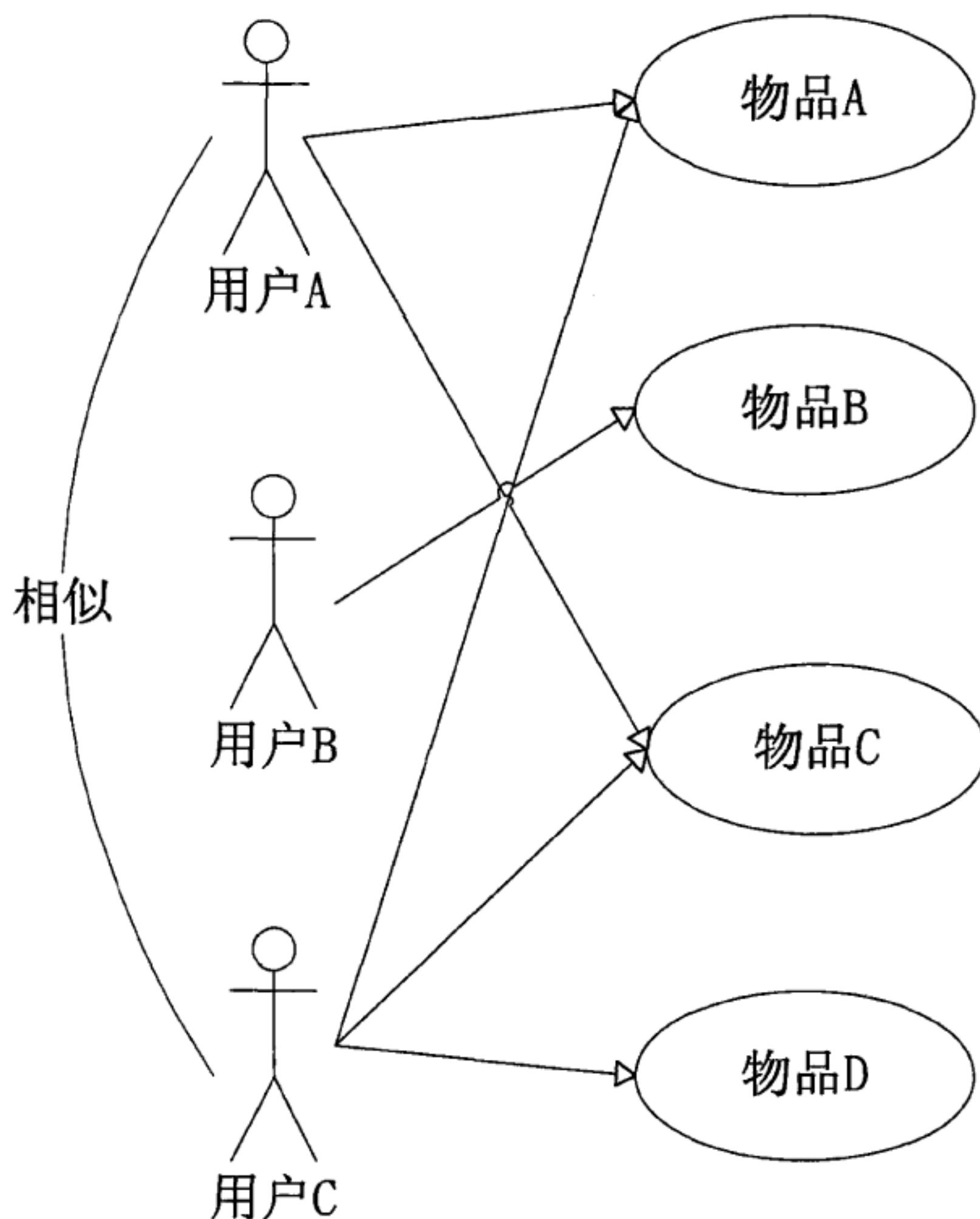


图 2.2 基于用户的协同过滤

结合原理图，假定消费者 A（用户 A）认可商品 A（物品 A），商品 C（物品 C），消费者 B（用户 B）喜欢商品 B（物品 B），消费者 C（用户 C）认可商品 A、商品 C 与商品 D（物品 D）。那么结合消费者 A、B、C 认可信息可知，消费者 A 与消费者 C 存在一致认可的商品 A 与商品 C，则表示消费者 A 与 C 喜好十分接近，但消费者 C 同样比较认可商品 D，那么我们可以假设消费者 A 也认可商品 D，商品 D 属于非一致偏好的商品，我们能够尝试把商品 D 提供给消费者 A。

以使用者为主体的协同过滤推荐算法主要是求出与当前使用者偏好信息类似的“最近邻居”用户群，借此了解群体中非一致偏好信息，最终进行合理的推荐。通过这种算法进行处理时必须提前完成假设：若是不同使用者在某方面出现相同的认可

度，那么一方认可的项目另一方也有很大可能会有所偏好，且偏好相同的项目数量越多，则不相同的项目给对方推荐获得偏好认可的可能性越大^[30]。

和其他算法相比，user-based CF 在推荐精度方面比较高，而且可以推荐图片流等项目。值得一提的是，在该系统之中，用户可以通过邻居的推荐获益，并将自己的推荐反馈给其他相似用户，从而形成一种良性循环系统。因此，它在现实中是比较实用的一类算法。

2.5.2 基于项目的协同过滤推荐算法

这种算法作为当今社会所有大型互相网企业使用率较高的一类算法。如 Youtube、Hulu 等企业均通过这种算法进行处理，最终给予的推荐也取得了不错的成果，这种算法会提供给消费者相应的商品，而所选择的商品和消费者此前认可的商品在某种程度上比较接近。比如，通过这一算法进行处理，当消费者在购买《机器学习》后，继续推荐《数据挖掘导论》，只是即便存在上述情况，但利用这种算法对项目进行处理时，最终结果的相似度和以内容为主体的算法存在着较大的差异，其本身并未包含内容属性，只是根据使用者的行为模式进行研究，这种算法将如下假设确定成前提：认可一类项目的用户群，部分使用者往往同时认可其他项目，则两种项目间存在一定相似度。该算法原理如图 2.3 所示。

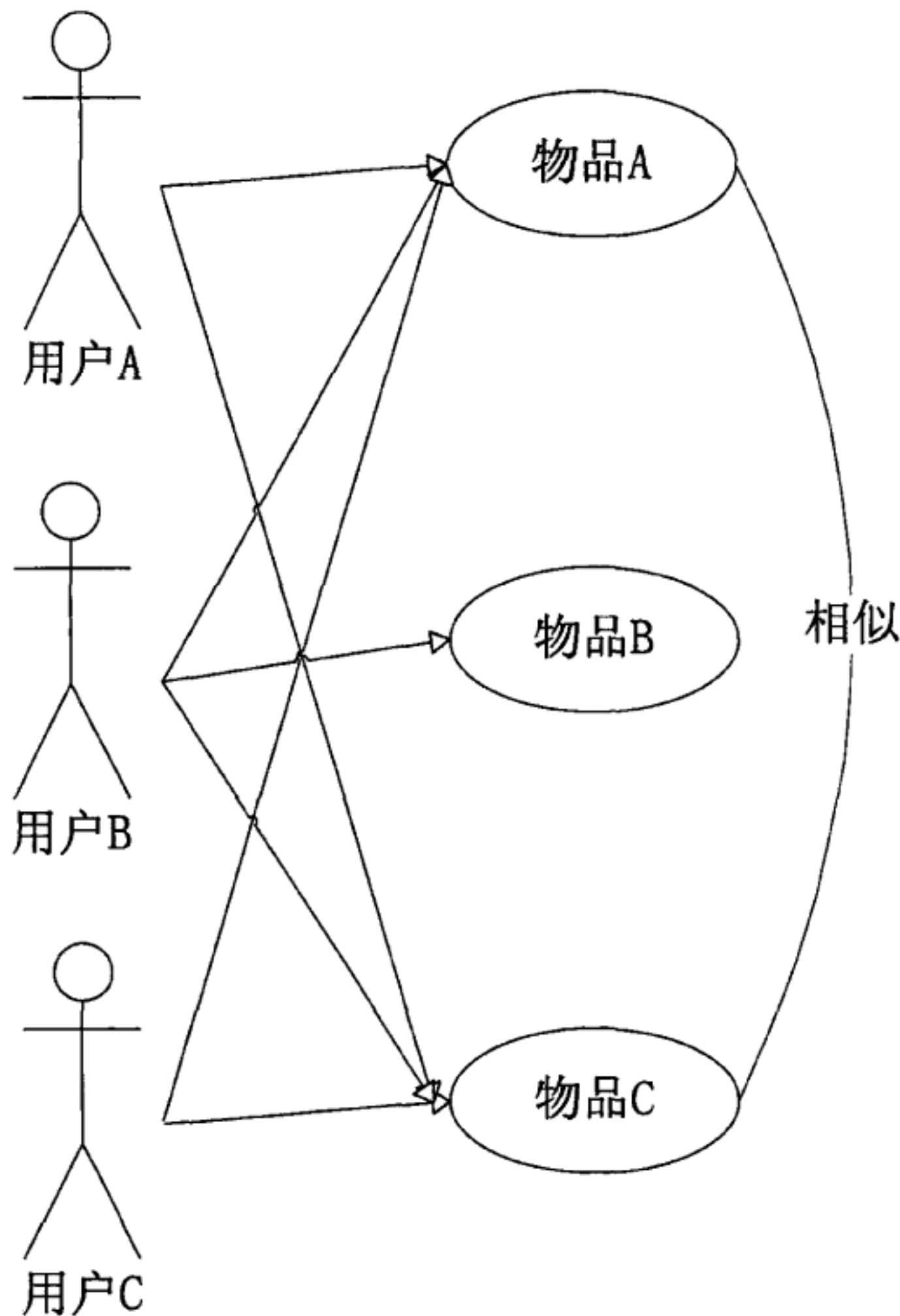


图 2.3 基于项目的协同过滤

结合基本原理进行说明：假定消费者 A 认可商品 A，商品 C；消费者 B 认可商品 A，商品 B，商品 C；消费者 C 认可商品 A，那么结合消费者 A、B、C 认可信息可知，消费者 A 与消费者 C 存在一致认可的商品 A，则表示消费者 A 与 C 喜好十分接近，但消费者 A 同样比较认可商品 C，那么我们可以假设消费者 C 也认可商品 C，商品 C 属于他非一致偏好的商品，我们能够尝试把商品 C 提供给消费者 C，而消费者 A、B 间同样存在这种相似性，根据上述说明能够尝试把商品 B 提供给消费者 A。

2.5.3 基于SVD的协同过滤推荐算法

即便协同过滤在各个领域中取得了不错的成绩，然而在系统规模快速发展的今天，使用人数与项目数量的上升，让用户评价信息逐渐变得过于稀疏。若想求得使用

者间存在的相似度，项目需要达到两个及以上的标准方能实施共同评分。如果用户评价信息过于稀疏，将会发生不同使用者分别给某些项目进行评分，然而项目却一致的情况，如此一来，便不能求得使用者间存在的相似度，无法确定最近邻居，最终完成不了推荐工作。其次，如果数据集包含的内容较多，那么在计算与查找时必定需要更多的时间与精力，从而让推荐过程无法满足实时性标准。最后，无论是 User-based 协同过滤还是 Item-based 协同过滤都是根据用户项目评分矩阵内部评分数据获得邻居，也就不可了解项目间具备的潜在联系，所以需要采取 SVD 降维模式进行处理，利用 SVD 计算离线操作性能^[31]，能够进一步加快系统反应速率。

SVD 降维模式发展时间较长，在很多领域中均取得了不错的成绩。而初期运用于推荐方式中时只是为了达到降维目的，然而经典的 GroupLens 推荐系统经过功能细化，使其达到了以下三个目的^[32]：

- 1.通过 SVD 模式处理用户-项目矩阵时，可以采取转换矩阵特征值的方式控制矩阵维度，以最终得到的矩阵为前提完成评分估计，不必进行求出使用者相似度以及获得邻居等操作。
- 2.利用 SVD 在矩阵中求出用户的相似度，接着利用以用户为主体的协同过滤进行处理，将用户邻居带入其中，以此产生推荐结果。
- 3.利用 SVD 细化初期的用户-项目矩阵，让系统得到合理的提升与发展。

将 SVD 引入推荐领域进行矩阵降维，可以避免矩阵过于稀疏，作用十分明显。基本原理是^[33]：

SVD 可以将一个 $m \times n$ 矩阵 R 分解为 3 个矩阵： $R = T_0 \times S_0 \times D_0'$, $S_0 = diag(\sigma_1, \dots, \sigma_r)$ 其中， $\sigma_1 \geq \dots \geq \sigma_r \geq 0$ ， T_0 和 D_0 分别代表了一个 $m \times r$ 和 $n \times r$ 的矩阵全部支持其正交性 ($T_0 T_0' = I, D_0 D_0' = I$)，矩阵 R 的秩用 r 表示 ($r \leq \min(m, n)$)。 S_0 代表一个 $r \times r$ 的对角矩阵，所有的 σ_i 大于 0 并按照降序排列且都成为矩阵 R 的奇异值。通常矩阵 R 表示为 $R = T_0 \times S_0 \times D_0'$, T_0 , S_0 , D_0 需要达到满秩状态，奇异值细化明显优势在于，可以让简化矩阵代替初始矩阵，而 S_0 矩阵由于对角线位置的元素依照大小顺序进行处理，我们只能获得部分极大值如前 K 个，通过细分矩阵均采取 0 作为余下 $r-k$ 个值，如此一来，便能够把 S_0 细分成只存在 k 个单值的矩阵 ($k < r$)^[34]。故将 S_0 矩阵中的全为 0 的列与行清理，形成全新对角矩阵 S，矩阵 T_0 , D_0 也可以按照相同模式细分成 T, D，得到矩阵 $R_k = TSD'$ ，且 $R_k \approx R$ ，重构的矩阵就是所有秩为 K 的矩阵中最逼近原矩阵 R 的矩阵。

因初期的评分矩阵内部只有少量评分选项，矩阵内全部表示为 0，以此为前提分解矩阵时将得到大量负值，若想避免负值过多增加求出相似度的难度，进行分解时需要使矩阵得到相应的处理，0 值项所在位置利用其所在列的均值取代，初期的矩阵内非 0 项若想方式各个用户评价差异造成干扰，将以 $r_{ij} - \bar{r}_j$ 取代 r_{ij} ，最终所得矩阵为 R' ，也可以作为算法对应输入矩阵。以 SVD 降维为主体的协同过滤推荐算法具体流程是：

输入：矩阵 R' 和用户 U 已评分过的项集合 I_u

输出：相关矩阵 T、S、D

步骤：

1)用 SVD 方法分解矩阵 R' 得到矩阵 T_0 ， S_0 ， D_0

2)将 S_0 简化为维数为 k 的矩阵，得到矩阵 S ($k < r$, r 为矩阵 R 的秩)

3)相应地将矩阵 T_0 、 D_0 简化为 T、D

4)矩阵 S 求平方根得到 $S^{1/2}$

5)计算两个相关矩阵 $TS^{1/2}$ 、 $S^{1/2}D'$

$TS^{1/2}$ 说明 k 维空间中使用者存在的联系，也就是使用者评价 k 个元素所得值，作为 $m \times k$ 型矩阵，也是用户矩阵，矩阵 $S^{1/2}D'$ 描述了项目在 k 维空间中存在的联系，规格是 $n \times k$ ，属于项矩阵，此处需要检验 k 值的确定是否有效性，仅需 k 值比矩阵 R 的秩 r 小就表示已经降维了，但是选择不同的 k 值对推荐效果还是有很大影响，当下不存在任何官方的选择模式，均以实验分析 k 值与推荐精度之间的关系，以此确定最佳 k 值进行处理。

然后便能够开始估计出使用者 u 给项目 t 的评分，可以用公式 (2-1) 表示：

$$pred_{u,t} = \bar{u} + TS^{1/2}(u) \cdot S^{1/2}D'(t) \quad \text{公式(2-1)}$$

这种模式不必求出用户、项目间存在的相似度，处理上十分简单，只是理解起来比较困难，同时降维有可能会丢失某些重要信息，推荐的准确性也会受到影响。

2.5.4 相似度计算方法分析

一般在计算数值属性的距离之前应该进行数据的规范化。这涉及到数据的变换，是数据落入较小的公共值域。如在图形图像处理中，将 [0,255] 的数值通过变换，映射到 [0,1] 的区域。

求数值属性类数据对象间存在的相异性时，需要依靠三种度量实现操作：欧几里得、曼哈顿以及闵科夫斯基三种距离^[35]。这三个距离又称作 L_2 范数、 L_1 范数、 L_p 范

数，范数的选取将在第 4 章进行分析。常见的数值属性相异度量计算公式如下：

(1) 欧几里得距离（直线距离）

$$d(i, j) = \sqrt{(X_{i1} - X_{j1})^2 + (X_{i2} - X_{j2})^2 + \dots + (X_{ip} - X_{jp})^2} \quad \text{公式 (2-2)}$$

(2) 曼哈顿距离（即城市块距离）

$$d(i, j) = |X_{i1} - X_{j1}| + |X_{i2} - X_{j2}| + \dots + |X_{ip} - X_{jp}| \quad \text{公式 (2-3)}$$

(3) 闵科夫斯基距离（又称 L_p 范数，或者一致范数）

$$d(i, j) = \sqrt[p]{|X_{i1} - X_{j1}|^p + |X_{i2} - X_{j2}|^p + \dots + |X_{ip} - X_{jp}|^p} \quad \text{公式 (2-4)}$$

●余弦相似性

它往往于数据挖掘期间文本挖掘时对文本相似度进行度量^[36]。实践运用时，需要将某一文档通过大量属性代替，任意属性均可以对文档内部特定词（如关键词）进行记录。利用该技术，可以将所有文档通过一个词频向量代替。如表 2.1 所示，通过两词频向量代表文档 1（摘自《我的奋斗史》第十章）和文档 2（摘自《童年》第三章）：

	我	祖国	牛	天空	计算机
文档 1	5	3	0	2	7
文档 2	3	2	5	1	0

表 2.1 文档的词频向量分布

经过词频向量的映射处理后，再用余弦相似度计算文档的相似性，公式为：

$$\text{sim}(x, y) = (x \cdot y) / (\|x\| \|y\|) \quad \text{公式 (2-5)}$$

$\|x\|$ 作为向量 x 欧几里得范数，即 $\sqrt{(x_1^2 + x_2^2 + \dots + x_p^2)}$ ，站在概念角度而言，属于向量长度。 $\|y\|$ 与其一致。所得余弦值近似于 1，那么两向量夹角将尽可能小，结果越合理。

当属性是二值属性时，余弦度量的一个简单变种如下：

$$\text{sim}(x, y) = (x \cdot y) / (x \cdot x + y \cdot y - x \cdot y) \quad \text{公式 (2-6)}$$

这个值也称作 Tanimoto 系数或 Tanimoto 距离。

●相关相似性

我们使用皮尔森（Pearson）相关系数计算其值。Pearson 相关系数是一个大于-1 小于 1 的值，它一般描述线性数据移动变化的趋势情况。两个变量线性联系不断紧密时，它们的相关系数也将无限近似于-1 或 1；若某个变量增大，剩下变量随之增大时，相关系数也将低于 0；若是相关系数为 0，那么两变量没有线性关联^[37]。

我们以 T_{ij} 作为用户 i、j 一致评价的项目集合，以 R_{ik} 作为用户 i 评估项目 k 时所得

值，以 \bar{R}_i 、 \bar{R}_j 作为用户 i 、 j 给相关项目的评价值。相关相似性计算公式表示为公式(2-7)：

$$sim(i, j) = \frac{\sum_{k \in T_g} (R_{ik} - \bar{R}_i)(R_{jk} - \bar{R}_j)}{\sqrt{\sum_{k \in T_g} (R_{ik} - \bar{R}_i)^2} \sqrt{\sum_{k \in T_g} (R_{jk} - \bar{R}_j)^2}} \quad \text{公式(2-7)}$$

●修正的余弦相似性

对于实际的情况下，不同用户的评价标准肯定是不同的，而由于他们的评价尺度不同，所以计算出的余弦相似度效果肯定不是理想的。调整后的余弦相似度处理模式将用户给项目的评分排除在外，让使用者评价标准达成一致。

我们假定用户 i 、 j 一致的评价集是 I_g ，用户 i 评价集为 I_i ，用户 j 则为 I_j ，那么调整后的余弦相似度是：

$$sim(i, j) = \frac{\sum_{k \in I_g} (R_{ik} - \bar{R}_i)(R_{jk} - \bar{R}_j)}{\sqrt{\sum_{k \in I_i} (R_{ik} - \bar{R}_i)^2} \sqrt{\sum_{k \in I_j} (R_{jk} - \bar{R}_j)^2}} \quad \text{公式(2-8)}$$

公式内，以 R_{ik} 作为 i 评估项目 c 得到的值，而 \bar{R}_i 与 \bar{R}_j 则表示用户 i 、 j 项目评价所得结果的平均值。一旦“最近邻居”集出现，便能够按照相似用户给目标用户带来推荐商品。往往选择加权平均法进行处理。若是以 u 作为目标用户，那么可以通过公式(2-9) 实现对项目 i 的预测^[38]：

$$P_{ui} = \bar{R}_u + \frac{\sum sim(u, n) \times (R_{ni} - \bar{R}_n)}{\sum |sim(u, n)|} \quad \text{公式 (2-9)}$$

公式内，以 $sim(u, n)$ 作为用户 u 、 n 间的相似度，将 R_{ni} 定义成 n 评估 i 后得到的值， \bar{R}_n 与 \bar{R}_u 作为 n 、 u 对应的项目评分平均值。

利用加权平均模式进行处理获得评分预测结果，选择前 N 个非用户 u 评价的项目，经过集合操作后可以确定成 Top-N 推荐结果。

2.5.5 协同过滤算法的特点

首先我们从亚马逊商城中的图书推荐谈起。亚马逊给出的图书推荐主要基于两点信息：首先是某用户浏览过的一本书，比如 Gene Reeves 写的《The Lotus Sutra》；其次是其他用户阅读过《The Lotus Sutra》后还查看过其他书籍。协同过滤中的“协同”

就是因为它基于其他人的影响来进行推荐。

上一小节所介绍的相似性的计算，在这里举一项简单的用例。亚马逊的图书评分有5个等级，分别用1、2、3、4、5表示，数字从小到大代表着用户对图书的喜爱程度由低到高。假设用户Amy、Bill、Jim分别对图书《Snow Crash》、《Girl with the Dragon Tattoo》两本书的打分如图2.4表示：

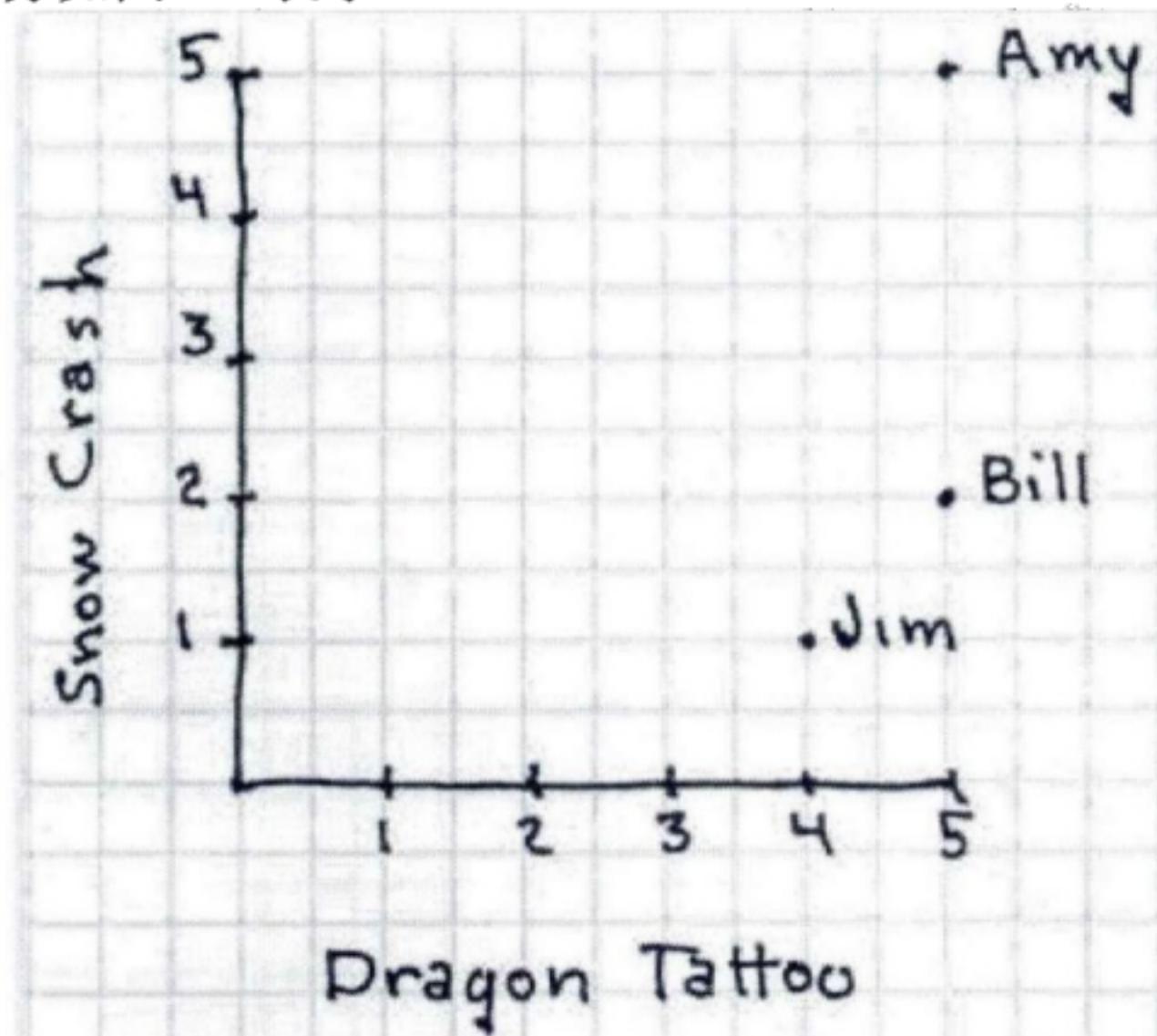


图2.4 用户对图书评分表

其中横坐标代表图书《Girl with the Dragon Tattoo》，纵坐标代表图书《Snow Crash》，坐标中的点代表用户对两本书的评分，图2.4也可以用表2.2进行表示。

	Snow Crash	Girl with the Dragon Tattoo
Amy	5	5
Bill	2	5
Jim	1	4

表2.2 用户对图书评分表

现在假定有一新用户X，其对图书《Snow Crash》的评分为4，对图书《Girl with the Dragon Tattoo》的评分为2，那么用户X和以上哪个用户更相似呢？这就需要进行相似度的计算。

首先计算曼哈顿距离，可以得到表2.3的结果。

	和用户 X 的距离
Amy	4
Bill	5
Jim	5

表 2.3 曼哈顿距离计算结果

可以观察发现，用户 X 和用户 Amy 的距离最短，也就可以认为用户 X 和用户 Amy 兴趣较为相似。如果 Amy 喜欢另一本图书《The Windup Girl》那么就可以将这本书推荐给 X。

接着计算欧几里得距离，可以得到表 2.4 的结果。

	和用户 X 的距离
Amy	3.16
Bill	3.61
Jim	3.61

表 2.4 欧几里得距离计算结果

结果和曼哈顿距离相似，这只是考虑两本图书时的情况。当我们考虑多维的情况，有很多本图书提供给多人来阅读时，由于不可能每本书都被所有人进行评分，所以计算的曼哈顿距离或欧几里得距离必定是不准确的。这也就说明了这两个距离的适用条件：数据集较为完整，不存在大量稀疏的数据。

下面考虑一下更为复杂的情况：

	Angelica	Bill	Chan	Dan	Hailey	Jordyn	Sam	Veronica
Blues Traveler	3.5	2	5	3	-	-	5	3
Broken Bells	2	3.5	1	4	4	4.5	2	-
Deadmau5	-	4	1	4.5	1	4	-	-
Norah Jones	4.5	-	3	-	4	5	3	5
Phoenix	5	2	5	3	-	5	5	4
Slightly Stoopid	1.5	3.5	1	4.5	-	4.5	4	2.5
The Strokes	2.5	-	-	4	4	4	5	3
Vampire Weekend	2	3	-	2	1	4	-	-

表 2.5 多维情况

如表 2.5 所示，横坐标代表着用户，纵坐标代表着图书。数字代表着用户对该图书的评分。从表中可以发现，用户 Bill 似乎总是避免给出极端值，他的评价范围在 2 到 4 之间；用户 Jordyn 似乎喜欢任何书籍，她的评价范围在 4 和 5 之间；用户 Hailey 只有两种评价，即 1 和 4。这就揭示出了一个问题：每个人的评价标准都是不一样的，

如何统一评价标准就需要用到皮尔森关联系数或者余弦相似性。皮尔森关联系数的适用范围就是可以统一不同的评价标准，并且可以忽略稀疏空缺值带来的影响。

介绍过相似度计算的几个方法之后，就要考虑将其应用到教育数据集的技术可行性问题。我们首先需要考虑教育数据集的特点。在教育数据集中，'CorrectFirstAttempt' 代表着学生首次尝试答题的正确与否，这是我们需要预测的属性，其值只有 0 或者 1，其他可能出现空缺值。它与亚马逊系统中的 5 项评分值截然不同，只有两种情况。但是考虑到可能出现大量的空缺值的情况，所以在计算相似性的时候，论文选择了皮尔森相关系数或余弦相似性或修正的余弦相似性。这些相似性计算公式在 Mahout 的 Taste 框架中都有具体的实现方法，例如皮尔森相关系数就使用了 PearsonCorrelationSimilarity() 方法。这些将在第四章进行详细的论证。

第三章 预处理过程

3.1 问题定义与数据集分析

3.1.1 KDD Cup 2010 比赛命题

KDD Cup 2010比赛最关键的环节是依照智能教学辅导系统与学生间的交互记录，推理出学生可能的回答。本论文则是讨论对学生第一次尝试答题的结果进行预测，从而判断学生的学习水平，并通过计算RMSE值来对比分析推荐效果。用于挖掘的数据集来源于2008年到2009年间一个叫做“Bridge to Algebra I”的数学辅导系统日志，该系统为学生提供习题练习，并记录学生相关答题信息。

竞赛提供来自两个系统“bridge_to_algebra_2008_2009”和“algebra_2008_2009”的挑战数据集，大小约合9G，含记录总计约700万条，本次数据挖掘选用“algebra_2008_2009”数据集，该数据集大小为3G，含记录890万条，每条记录含23个特征，总共约2亿个值，KDD Cup 2010各特征含义如下：

- Row: 行标号。
- AnonStudentId: 学生匿名唯一标识符。
- Problem Hierarchy: 该问题的所属的课程层级。
- Problem Name: 问题的名称，也是问题的唯一标识符。
- Problem View: 学生遇到该问题的总次数。
- StepName: 代表所有问题中的步骤称谓。系统内所有问题均会出现大量步骤的情况，这一步骤称谓独立于单个问题中，而各个问题间步骤称谓可以实现一致。例如，第一个问题里有a、b、c、d四个步骤，它们互不重复；第二个问题里有a、b、c三个步骤，它们也不重复；但是第一个问题和第二个问题里的a、b、c三个步骤名出现了相同的情况。
- StepStartTime: 步骤开始时间，可为空。
- FirstTransaction: 步骤首次处理时间。
- CorrectTransactionTime: 步骤正确作答时间，可为空。
- StepEndTime: 步骤结束时间。

- StepDuration(sec): 步骤持续时间。若开始时间为0，可为空。
- CorrectStepDuration(sec): 正确步骤持续时间。
- ErrorStepDuration(sec): 错误步骤持续时间（不正确的尝试或示意）。
- CorrectFirstAttempt: 首次提交是否正确。
- Incorrects: 学生在这个步骤上的错误提交次数。
- Hints: 学生请求提示的次数。
- Corrects: 学生同一问题正确次数（同一个问题遇到多次才会增加）。
- KC(KC Model Name): 解题中识别到的有效知识点。一个步骤可以有不同知识点，用“~”隔开。
- Opportunity(KC Model Name): 学生遇到KC项所列出的知识点次数，每次遇到该知识点增加一次。若有多个知识点，则数据以“~”隔开。

本次比赛命题为预测学生第一次回答问题的正确与否，其问题的本质还是机器学习领域的分类问题，即将学生的回答结果划分到1（代表正确）还是划分到0（代表错误）。所以本论文所要解决的问题是与分类问题密切相关的。

3.1.2 对相关特征数据的统计

确定了数据集中的各项特征之后，我们要对数据集中的各项数据进行统计。表3.1就是利用MySQL数据库对属性进行的统计列表^[39]:

特征名\数据集名	Algebra 2008-2009	Bridge to Algebra 2008-2009
Lines(train)	8,918,054	20,012,498
Students(train)	3,310	6,043
Steps(train)	1,357,180	603,176
Problem(train)	211,529	63,200
Section(train)	165	186
Units(train)	42	50
KC(train)	2,097	1,699
Steps(new on test)	4,390	9,807

表3.1 数据集中各属性数量统计

上表中的Lines代表总记录数，Students代表学生人数，Steps代表问题步骤数，Problem代表问题数，Section代表问题所属章节，Units代表问题所属单元，KC代表解

答问题所需的知识技能。Units、Section、Problem、Steps 所属关系为包含从属关系，相当于国家、省份、地市、区县的关系一样。

3.2 数据集抽取

3.2.1 抽取方法

既然论文本质解决的是分类问题，就要按照分类问题常用的解决办法，首先划分训练集和测试集。从训练集中抽取测试集的主要方法，根据 KDD Cup 2010 的测试集抽取方法，选取每个学生最后一个问题作为测试集。其次，由于数据集具有很强的时间相关性，所以训练集中最后部分的数据，成为了所要预测结果的关键所在。表 3.2 表示测试集抽取的数据。

学生	问题	步骤
1	A	1
		2
	B	1
		2
		3
	C	1
		2
		3
		4
		5
2	A	1
		2
	B	1
		2
		3

表 3.2 测试集抽取示意图

由表中所示，学生 1 作答了 A、B、C 三个问题，学生 2 作答了 A、B 两个问题，其中问题 A 包含步骤 1 和步骤 2，问题 B 包含步骤 1、步骤 2 和步骤 3，问题 C 包含步骤 1 到步骤 5。蓝色部分代表抽取出来作为测试集的数据，即抽取学生 1 问题 C 的所有步骤，抽取学生 2 问题 B 的所有步骤。

对数据集的选取方法有很多种，在尝试了多样的方法之后，最终选取分层抽取的方法，这种方法在时间复杂度和空间复杂度的度量上达到了最优的效果。

抽取过程尝试的方法主要有三种，如下所示：

1. 直接使用 Mysql 数据库抽取。Mysql 数据库在数据排序的问题上具有明显的优势，但是对于每个学生的问题顺序的逻辑判断的效果不太良好，并且单纯使用 SQL

语句进行逻辑判断，语句十分复杂；随着数据的累积，SQL 的时间效率将大大降低。

2.通过文件指针的方式进行抽取，把读文件的指针移到末尾，从最后往前读数据。这种方法可以明显降低内存的使用，节省资源。但存在的问题是，数据的读入是逆向的，也就是说读入的数据弯曲是颠倒反向的。最后还要用这种方式在还原数据，同样增加了过程的繁琐程度。

3.采用 Mysql 进行排序加 Java 逻辑判断的方式提取测试集。首先使用 Mysql 进行倒序排序，这样最终抽取的测试集就是每个学生的前几行数据，然后导出文件使用 Java 写入标记值，最后再次导入到数据库中通过标记值来进行抽样。最终经过 2 小时的抽样之后，抽取测试集条数 52610 条，训练集为 8863444 条，和 KDD Cup 2010 比例相近。

最终所使用的方法在算法上很容易理解，并且提高了训练集样本抽样的灵活性。它不仅可以按照时间相关性抽取测试集样本，还可以实现随机抽样等各种抽样方法，并且实现这些不同的抽样方法只需要改变一条 SQL 语句即可。测试集的抽取流程图如图 3.1 所示。

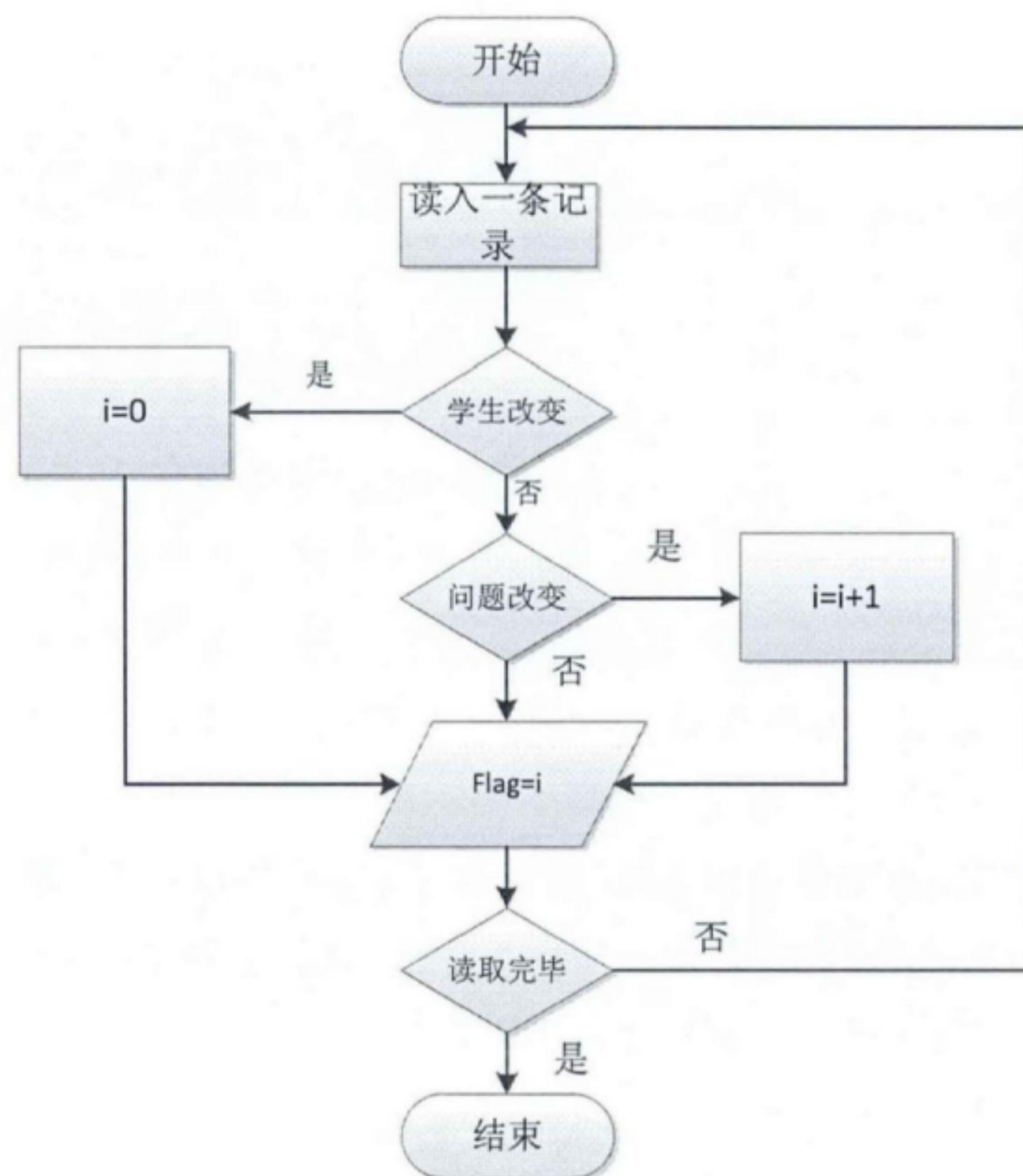


图 3.1 测试集提取流程图

3.2.2 数据清理

在3.1.1节中提到，数据集中总共有23个特征。但是协同过滤推荐算法显然是不需要如此多的特征的，所以要对无用的特征进行数据清理。

在这23个特征中，有10个特征在测试集中是隐藏的，原因是通过这部分特征可以直接推算出学生的作答结果。因此，将这10个特征从训练集中直接删除。删除的10个特征如下：

- StepStartTime
- FirstTransactionTime
- CorrectTransactionTime
- StepEndTime
- StepDuration (sec)
- CorrectStepDuration (sec)
- ErrorStepDuration (sec)
- Incorrects
- Hints
- Corrects

在剩下的特征中，“ProblemHierarchy”和“ProblemName”有着完全的相关性。每一个“ProblemHierarchy”下包含一个或多个“ProblemName”，它们的关系类似于问题与步骤的关系，但是“ProblemName”的值是唯一的，因此用“ProblemName”可以完全代替“ProblemHierarchy”的功能，且精度更高。所以将“Problem Hierarchy”删去。为了进一步提高模型精度，将“ProblemName”和“StepName”两个属性合并成一个叫做“ProblemStepName”的新属性，并将原来的“ProblemName”和“StepName”两个属性删去。

由于协同过滤算法主要涉及到三项属性，即用户ID项，商品ID项以及评分项，所以在本次数据挖掘中我们抽取“Anon Student Id”、“ProblemStepName”、“Correct First Attempt”(CFA)三项作为处理对象。在原数据集中数据的存储形式如表3.3所示。

Anon Student Id	ProblemStep	CFA
stu_6c94412bc1	1PTB02-1600=-400x	1
stu_6c94412bc1	1PTB02-4000=-400x	1
stu_c329962347	1PTB02-4000x+4000=2400	1
stu_c329962347	1PTB02-4000x=-1600	0
stu_b42eace9da	1PTB02-400k=-4000	1
stu_10127b1302	1PTB02-400x+4000=0	1
stu_c329962347	1PTB02-400x+4000=0	1
stu_526334d206	1PTB02-400x+4000=0	1
stu_e762f0fb85	1PTB02-400x+4000=0	1
stu_10127b1302	1PTB02-400x+4000=2400	1
stu_c329962347	1PTB02-400x+4000=2400	0
stu_526334d206	1PTB02-400x+4000=2400	1
stu_e762f0fb85	1PTB02-400x+4000=2400	1
stu_10127b1302	1PTB02-400x=-1600	1
stu_c329962347	1PTB02-400x=-1600	1
stu_526334d206	1PTB02-400x=-1600	1
stu_e762f0fb85	1PTB02-400x=-1600	1
stu_5eae5b29a1	1PTB02-400x=-1600	1
stu_10127b1302	1PTB02-400x=-4000	1
stu_c329962347	1PTB02-400x=-4000	1
stu_526334d206	1PTB02-400x=-4000	1

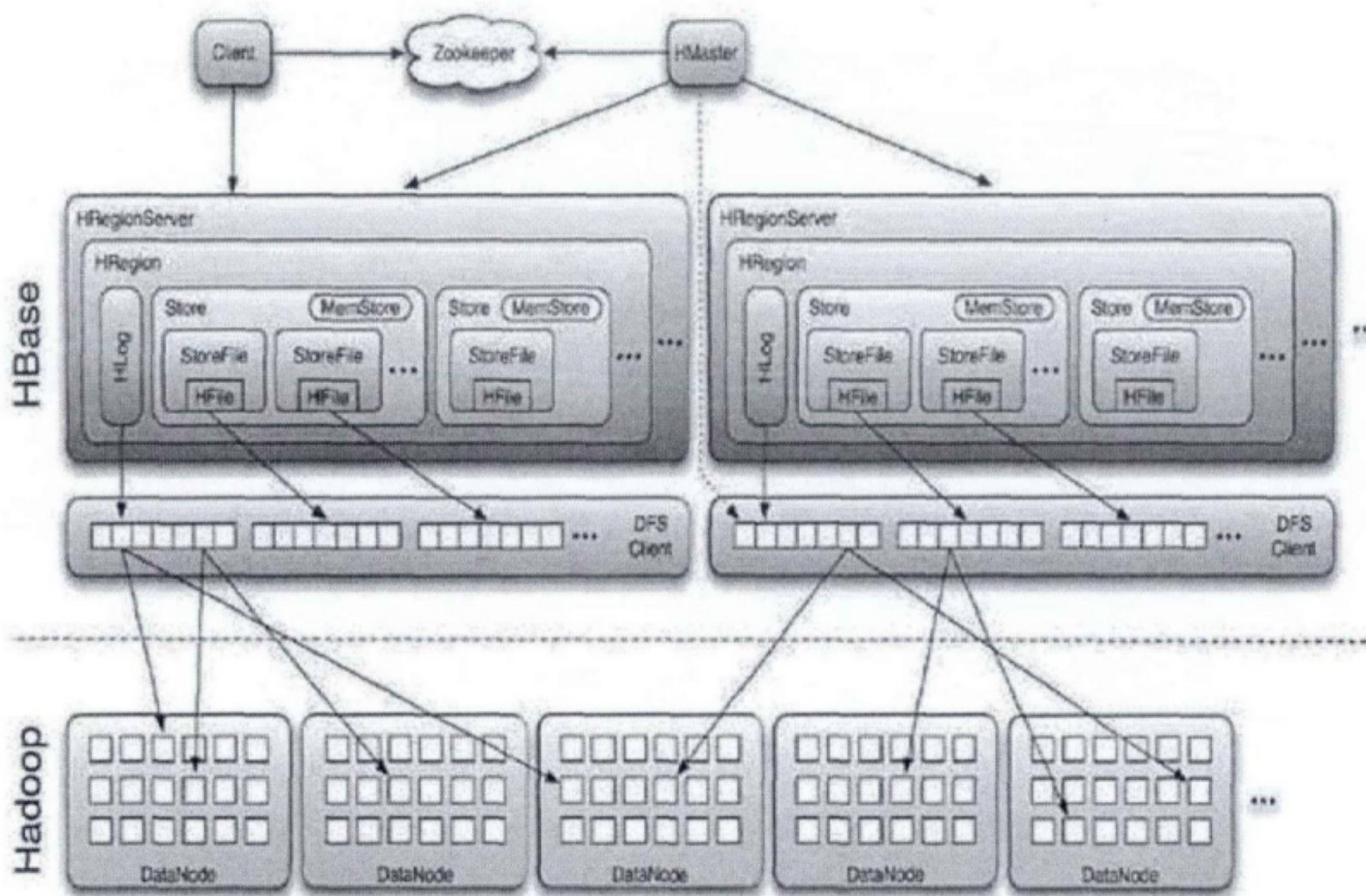
表 3.3 数据存储形式列表

从上图中可以发现，表中的三个属性除了 CFA 属性为数值型属性外，其余两个属性皆为标称属性。由于协同过滤推荐算法需要的属性都必须是数值型属性，所以要对“Anon Student Id”和“ProblemStepName”两个属性进行转换，来适应算法。处理后的数据源的形式如表 3.4 所示。

Sid	PSid	CFA
1	108184	1
1	108185	1
1	108186	1
1	108187	0
1	108188	1
1	108387	0
1	108388	1
1	108389	1
1	108390	1
1	108391	1
1	108392	1
1	108393	1
1	108394	0
1	108395	1
1	108396	1
1	108396	1
1	108397	1
1	108398	1
1	108399	1
1	108400	1
1	108796	1
1	108797	1
1	108798	1

表 3.4 处理后的数据存储形式

由表中所示，将“Anon Student Id”属性转化为“Sid”属性，将“ProblemStepName”属性转化为“PSid”属性。“Sid”中的数字 1 代表学生“stu_6af5d5e304”，“PSid”中的数字代表该学生所作答的所有问题步骤名，每一个数字都有唯一的问题步骤与之对应。由于原数据集是按照学生答题的时间顺序进行记录的，每个学生作答记录都是连续的，所以对学生 ID 的数值型转化比较简单。问题的难点在于，所有问题步骤数统计为 1,357,180 个，且都是分散的，即使是对其中 300 个学生统计的步骤数都已经达到了 88,791 个，所以对问题步骤的数值型转化是一项庞大而复杂的工作。这里我主要使用了 HBase 分布式数据库以及数据库集群等多项手段，终于将数据处理规模缩短到 10 分钟左右，提高了程序的运行时间预期。HBase 架构图如下所示。



从上图中可以看到，HBase 是建立在 Hadoop 的 HDFS（分布式文件系统）之上构建起来的。其具有很多优点，可以灵活拓展，容错性高，可靠性高。HBase 整个系统中包括 HMaster 一个，以及数个 HRegion，HRegionServer 和 HLog。HMaster 的作用是管理系统中全部 HRegionServer，并能够查看 HRegionServer 的状态信息。HRegionServer 负责管理 HRegion 对象。HRegion 和 HTable 中的 Region 单独对应。在其外部，HBase 需要 Zookeeper 和 HDFS 共同对其提供支持。Zookeeper 提供高可靠的调度，保证有效的 Master 的即时可用性。同时所有在集群中的数据都储存在 HDFS 中，需要 HDFS 提供持久性的存储服务。

第四章 实验的设计与效果评估

4.1 算法评估

若想确定推荐算法的有效性，必须运用相应的对比模式实现验证，而比较直观的一种便是让使用者进行评分。如果不具备使用者评分的条件，那么便需要考虑建立一套合理的评分指标进行处理。因此，用何种标准评价是协同过滤算法的重要环节之一。为了使推荐结果能够越来越准确，形成良性循环，就需要推荐结果与用户需求的契合度很高，用户对该系统的满意度很高，就能够向相似的用户介绍该推荐系统。

本文中通过计算均方根误差值RMSE（Root Mean Square Error）进行处理，也就是预测值与现实评分存在的差异，以此确定算法的有效程度。这种模式第一步便是求得某一给某一项目的评分值，接着求出RMSE值，一次确认最终推荐的有效性，若是RMSE值较小，则可以认为这种推荐算法相对可靠，相反，如果RMSE值较大，可以认为这种算法得到的结果并不理想。以下是其计算方法：

假定用I表示问题步骤集合，用S表示学生集合，用i表示某个问题步骤，用s表示某个学生，则 $i \in I, s \in S$ 。用矩阵 $C = [c_{is}]$ 表示每个学生对每项问题步骤的作答结果，用集合 $L = \{(i, s) | i \in I, s \in S, c_{is} \text{ is known}\}$ 表示学生s和问题步骤i之间的关系，将矩阵C划分成训练集 L_T 和测试集 L_p 。则RMSE公式如下：

$$RMSE = \sqrt{\frac{1}{|L_p|} \sum_{(i,s) \in L_p} (c_{is} - \hat{c}_{is})^2} \quad \text{公式 (4-1)}$$

其中 \hat{c}_{is} 表示学生s首次尝试正确回答步骤i的预测值。公式通过计算测试集中的未知项来评判预测效果的优劣性。由于RMSE就是计算数值分布在0、1之间的数据的误差分析，所以选择计算该项数值完全适合本实验的数据集。

4.2 基于Apache Mahout的算法实验

在完成Apache Mahout仿真算法实验之前，首先要建立数据模型，得到相关的实体信息以及它们之间的关系，这样之后能够对设计数据库和建立系统的数据模型带来帮助。如图4.1为学生和作答题目的建模图。

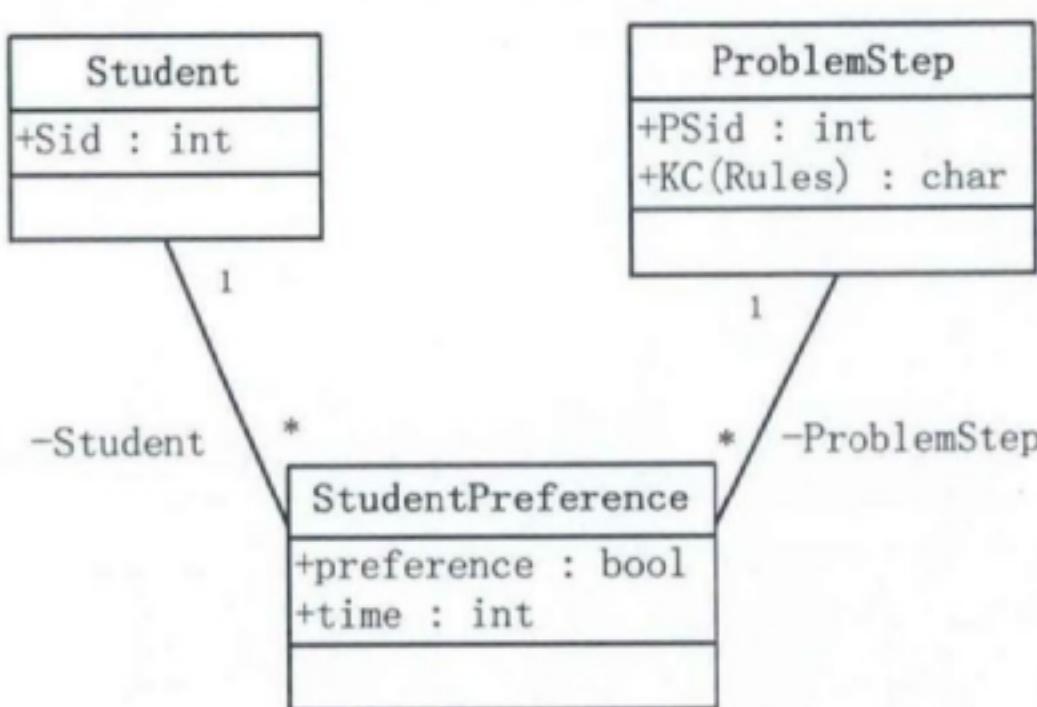


图 4.1 学生和作答题目类模型

本文所使用的数据模型主要包含 3 种实体类，如下所示：

- **Student** 类：表示学生类，对于该系统而言，类中主要包括学生 id 字段。
- **ProblemStep** 类：表示问题步骤，主要包含问题的 id 名称，问题所需要的知识子技能等属性。
- **StudentProference** 类：用来表示某个学生对某个问题步骤的作答结果。

基于上述的数据建模，以下为具体的实现方法。

1. 建立数据库存储。将以上三个类的数据分别存储在数据库中。由于实验数据还需要进行读写文件等相关数据操作，将转变后的特定格式的数据存储到 MySQL 数据库中。Mahout 引擎所能够接收的输入必须是自定义的 DataModel 类型，其他类型的数据将无法实现。

2. 实现推荐算法的数据存储。本文系统主要用到数据库相关读取的 JDBCDataModel 数据类型，它继承于 DataModel 类型，能够从任意形式的数据源中提取相关信息，同时它还对应有内存读取类型等其他类型的操作方法。本实验另需扩展 MySQLJDBCDataModel 来实现题目推荐算法中的 DataModel。

3. 推荐模型的实现

下面主要介绍三类协同过滤推荐算法的 Mahout 实现，这三类算法各选取其中一种代表性的算法，分别为：基于用户的协同过滤推荐（User-Based CF）、基于项目的协同过滤推荐（Item-Based CF）和基于 SVD 模型的协同过滤推荐（SVD CF）。

4.2.1 基于用户的协同过滤推荐 (User-Based CF)

这种算法能够达到预期的目的关键在于所求用户相似度的有效性。Apache Mahout 内部包含大量相似度求取模式，而且具备 UserSimilarity 接口。不仅如此，可以让 Recommender 接口实现正常运行，也就是将推荐引擎构造模式运用于实际，基本流程如下：

- 1.计算学生之间的相似程度，主要是建立 DataModel 数据类型；
- 2.学生相似度设置推理方法；
- 3.基于 User Similarity 计算学生的“最近邻居”。
- 4.通过相似度与相似用户两种计算模式求得推荐器实例。

实验通过 Mahout 延伸类型接口进行处理，达到了 User-Based 协同过滤推荐算法的目的，通过对试验数据（3310 名学生对 211529 项问题的 8918054 条答题记录）进行建模产生如图 4.2 所示的结果：

```
RecommendedItem[item:313, value:1.0]
RecommendedItem[item:311, value:1.0]
RecommendedItem[item:312, value:1.0]
RecommendedItem[item:225703, value:1.0]
RecommendedItem[item:305, value:1.0]
RecommendedItem[item:306, value:1.0]
RecommendedItem[item:225702, value:1.0]
RecommendedItem[item:304, value:1.0]
RecommendedItem[item:309, value:1.0]
RecommendedItem[item:310, value:1.0]
```

图 4.2 User-Based CF 推荐结果

图中是针对 Sid 为 1 的学生进行的推荐结果，item 代表 PSid，value 代表 CFA。论文主要选取了最有可能的 10 个推荐进行展示。其关键代码如下所示：

```
UserSimilarity similarity = new PearsonCorrelationSimilarity(model);
similarity.setPreferenceInferrer(new AveragingPreferenceInferrer(model));
//指定用户邻居数量，这里为 5
UserNeighborhood neighborhood = new
```

```
NearestNUserNeighborhood(5,similarity,model);
//构建基于用户的推荐系统
Recommender recommender = new CachingRecommender(new
GenericUserBasedRecommender(model,neighborhood,similarity));
```

4.2.2 基于项目的协同过滤推荐 (Item-Based CF)

这种算法和以用户为主体的推荐模式大体一致，而不同之处则是需要求解题目相似度。处理阶段基本流程如下：

1. 找到邻居 user。查找最近邻居的方法有查找固定数量的邻居 (K-neighborhoods) 和查找基于相似度门槛的邻居 (Threshold-based neighborhoods) 两种方法，而且这两种方法都有 Apache Mahout 的相应实现接口。

2. 以最近邻居为主体，得到以 item 为主体的协同过滤推荐，重点包含两步。第一步求得项目相似度，即 ProblemStep 之间的相似度，这样就可以需要时随时提取，而不需要每推荐一次就计算一次相似度。然后再创建基于项目的推荐器，即先从数据源中得到学生对某道题目的作答情况，第二步将最相似的题目推荐给该同学，以得到比较好的效果。

使用如上的方法，对所用的数据（3310 名学生对 211529 项问题的 8918054 条答题记录）进行试验，得到的运行结果如图 4.3 所示。

RecommendedItem[item:52, value:1.0]

图 4.3 Item-Based CF 推荐结果

如上图所示是针对学生 id 为 1 的学生的推荐结果，其中 item 代表 PSid，value 代表 CFA。由于基于 item 的协同过滤一次推荐只找最相近的 item，故只显示一行结果。其关键代码如下所示：

```
ItemSimilarity similarity = new PearsonCorrelationSimilarity(model);
//构建基于 Item 的推荐系统
Recommender recommender = new CachingRecommender(
new GenericItemBasedRecommender(model,similarity));
//得到指定用户的推荐结果，这里得到用户 1 的 1 个推荐
List<RecommendedItem> recommendation = recommender.recommend(1,1);
```

4.2.3 基于SVD的协同过滤推荐 (SVD-Based CF)

基于 SVD 的协同过滤推荐相对来说比较简单。其实现步骤在第 2 章中已经有了详细的描述，这里简单的再阐述一下。

假设数据矩阵 $Data$, 维度表示为 $m \times n$, 即有 m 个样本, 每个样本的特征数为 n 。则 SVD 可用如下公式分解表示:

$$Data_{m \times n} = U_{m \times m} \sum_{m \times n} V_{n \times n}^T \quad \text{公式 (4-2)}$$

$$Data_{m \times n} \approx U_{m \times k} \sum_{k \times k} V_{k \times n}^T \quad \text{公式 (4-3)}$$

上面的公式中 \sum 为只有对角元素, 其余元素都为 0, 一般 \sum 的对角元素都是按照从大到小排列的, 称为奇异值。那么得到

$$V = Data_{m \times n}^T \times U_{m \times k} \times \sum_{k \times k}^{-1} \quad \text{公式 (4-4)}$$

能够理解为用户与物品互相作表现成特征, 然而内部包含大量杂质, 通过 SVD 能够达到筛选目的, 增强推荐结果的有效性。基本流程是: 为用户 u 提供商品 v , 求出商品 v 和 u 之间所有经评分处理的商品存在的相似度, 求解期间, 它的特征主要利用上面矩阵细化得到 V 公式, 能够理解成把商品特征维度改变为 k , 商品特征矩阵是 $n \times k$, 并非此前的 $n \times m$ 。

针对所使用数据 (3310 名学生对 211529 项问题的 8918054 条答题记录) 得到的运行结果如图 4.4 所示:

```
RecommendedItem[item:6203, value:1.1249694]
RecommendedItem[item:6200, value:1.1249694]
```

图 4.4 SVD CF 推荐结果

如上图所示是针对学生 id 为 1 的学生的推荐结果, 其中 item 代表 PSid, value 代表 CFA。实验选取了最合适的前两个 step 进行推荐。其实现关键代码如下图所示:

```
//构建基于 SVD 的推荐系统
```

```
Recommender recommender = new CachingRecommender(new SVDRecommender(
    model, new ALSWRFactorizer(model, 10, 0.05, 10));
```

```
//得到指定用户的推荐结果, 这里得到用户 1 的两个推荐
```

```
List<RecommendedItem> recommendation = recommender.recommend(1, 2);
```

通过上述三类推荐算法处理相同数据实现推荐, 结合推荐结果不难发现, SVD 算法效率最高, 那么我们需要确定其中推荐结果最合理的一种方式。

4.2.4 三种相似度的计算方法

确定相似邻居、项目时需要求出相似度，目前使用率较高的两种相似度求解模式为余弦相似性与调整后的余弦相似性，以下则针对这两种求解模式进行说明。

使用两种相似度的计算步骤流程如图 4.5 所示：

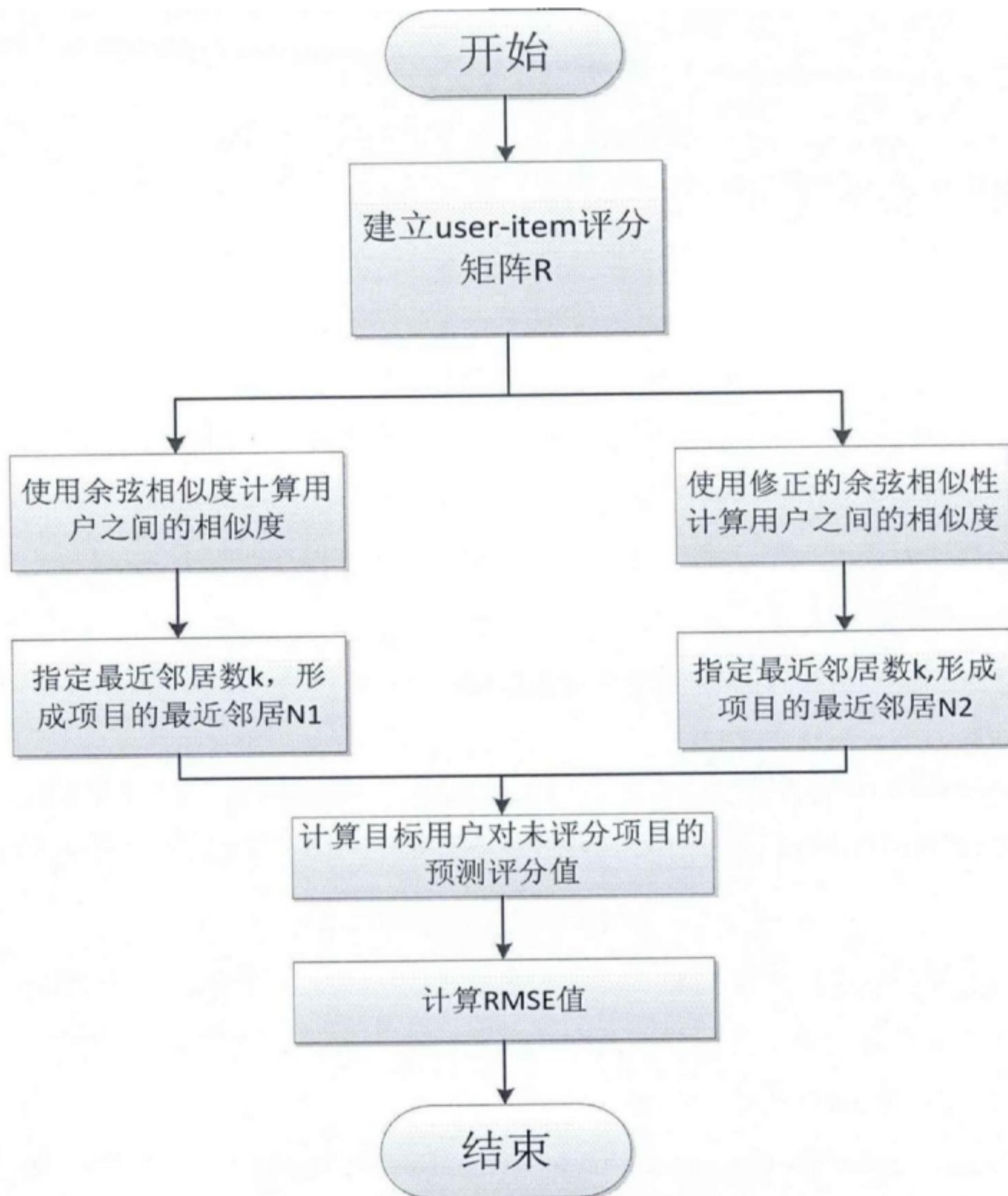


图 4.5 两种相似度进行计算的流程图

计算两种相似度所产生的 RMSE 值，并对 RMSE 值进行对比，得到结果如表 4.1 所示，可以看到随着邻居个数的变化所产生的 RMSE 值的相应变化。

相似度/邻居数 K	5	10	15	20	25
余弦相似度	0.3532	0.3465	0.3320	0.3300	0.3257
修正的余弦	0.3526	0.3433	0.3305	0.3279	0.3240

表 4.1 两种相似度计算的 RMSE 值

基于 item 的协同过滤算法所计算出的 RMSE 值如表 4.2 所示。

item 邻居数	5	10	15	20	25
RMSE 值	0.3520	0.3426	0.3300	0.3259	0.3220

表 4.2 基于 item 模型的 RMSE 值计算结果

从表中可以观察出，修正余弦相似性在预测效果上比余弦相似性要更准确一些，故基于用户的协同过滤一般采用修正的余弦相似性来进行推荐。基于 item 的协同过滤方法所计算的 RMSE 值要比基于用户的协同过滤要低一些，说明基于 item 的协同过滤在推荐效果上要更好一些。这是因为基于用户的协同过滤存在数据的稀疏性，而基于 item 的协同过滤则直接比较的是项目之间的相似性，跳过了用户之间的比较。接下来就要比较三种推荐算法的优劣性。在这之前，我们还要考虑数据稀疏性的问题。

所谓稀疏性问题，针对本文来说，是指对于 ITS 系统中的 21 万多个不同的问题，不可能每个学生都有作答记录，这样也就产生了很多的空缺值。而计算余弦相似性默认的是将未作答的题目的作答结果设定为 0 值或者其他学生对该题目的作答评价值。如果设定为 0 值，而另一用户作答为 1 值，在计算用户相似度时，可能二人的相似程度会明显降低；另一种情况是，对于某一题目，学生 i 和学生 j 都没有对其作答，这时如果将其作答结果设置为 0 值或者平均值，则两人的结果一致，这就有可能导致二人的相似程度大大增加，而实际上两学生的作答结果有可能是不相同的，所以说仅仅将作答结果设定为 0 是不明智的。

相关相似性计算方法则与余弦相似性不同，首先提取用户 i 和用户 j 都作答过的步骤形成一个集合，在这个前提下求出用户相似度，部分相似性采取皮尔森相关系数求解相似度，和余弦相似度通过 0 值进行处理的模式效果相比，得到的结果更加合理。不仅如此，调整后的余弦相似性和跟相关相似性比较一致，同时这种方法对评价标准进行了相应的规范^[40]。本文中虽不涉及评价尺度的问题，但使用修正的余弦相似性能够缩小误差的范围，它通过去除学生对题目的作答平均分来平衡评价尺度，所以在效果上会更好。本文就是采用的修正的余弦相似性进行相似度比较。

即便通过调整后的余弦相似性可以满足一定程度上的需求，但处理期间存在相应的限制，也就是稀疏度较低时，使用者回答同一问题的机会较多。然而实际处理阶段用户间极少会针对相同问题进行作答。对于小规模项目集合而言，就算评分相似度明显，也不可以确定用户间十分相似。足以表明这种方法同样存在相应的问题，若想增强结果的有效性，我们需要采取有关措施，增加用户-项目评分矩阵的内容，确保用户能够对同一问题进行作答，如此一来，便能提升推荐的意义。

填充评分矩阵的推荐流程如图 4.6 所示。

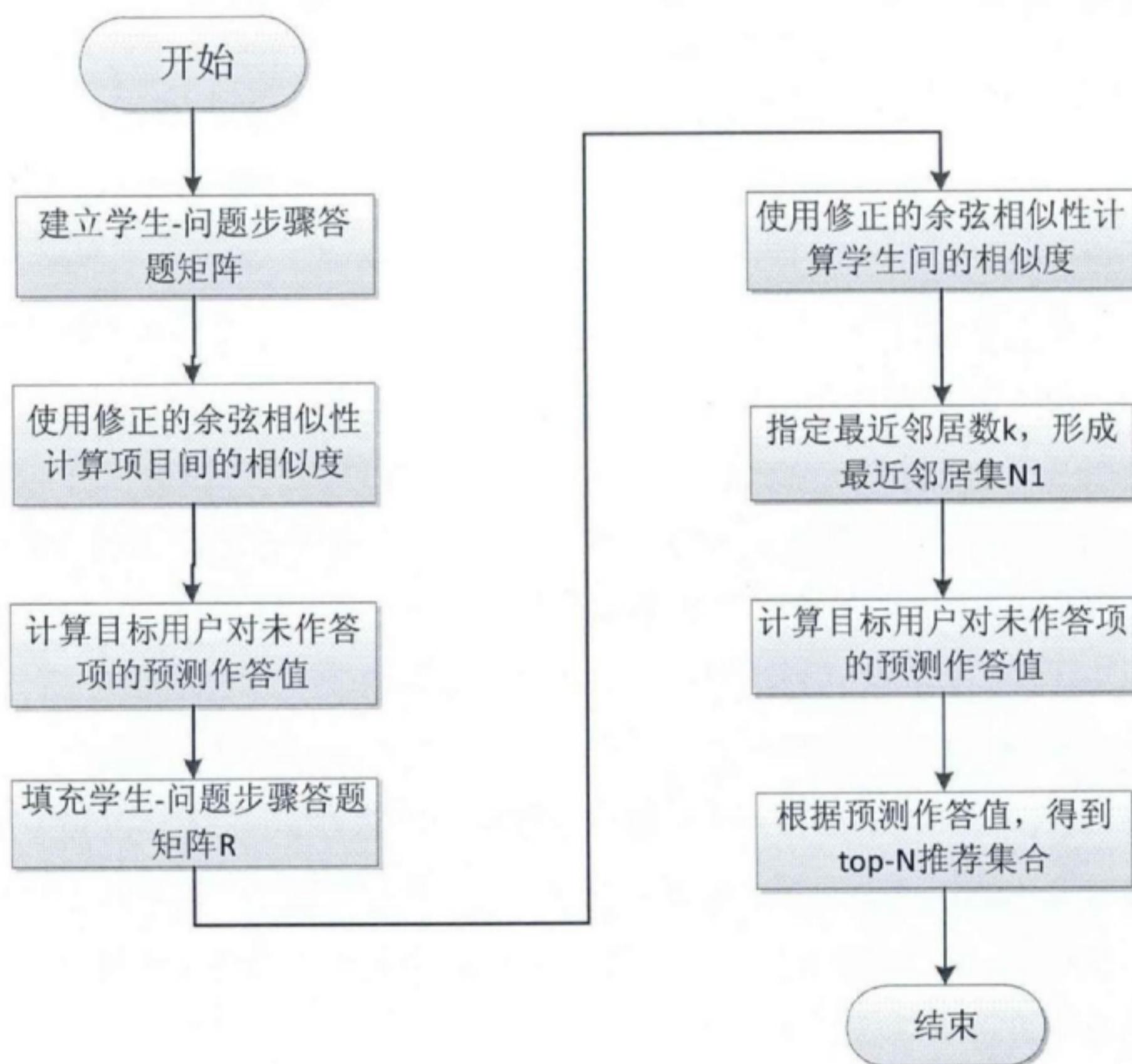


图 4.6 填充评分矩阵流程图

实验描述过程如下：

1. 设定学生-问题步骤答题矩阵 $R_{m \times n}$, m 代表学生的数量, n 代表问题步骤数, $R_{i \times j}$ 表示第 i 行, 第 j 列的元素, 其值代表学生 i 对步骤 j 的作答结果。没有作答的项则用空值表示。

2. 计算项目相似性

求解项目 i、j 存在的相似度时，采取调整后的余弦相似性进行处理，公式如下：

$$sim(i, j) = \frac{\sum_{u \in U_{ij}} (R_{ui} - \bar{R}_u)(R_{uj} - \bar{R}_u)}{\sqrt{\sum_{u \in U_i} (R_{ui} - \bar{R}_u)^2} \sqrt{\sum_{u \in U_j} (R_{uj} - \bar{R}_u)^2}} \quad \text{公式 (4-5)}$$

公式内 U_{ij} 作为题目 i、j 全部得到了答案的学生集合，学生 i、j 作答问题对应集合确定成 U_i 与 U_j 。学生 u 作答题目 i、j 的结果确定成 R_{ui} 与 R_{uj} ， \bar{R}_u 作为学生 u 作答所得结果的平均值。

3. 填充稀疏矩阵

运用如下公式对未作答的题目进行预测，公式如下：

$$P_{u,i} = \bar{R}_u + \frac{\sum_{u \in U_{ij}} sim(i, j) \times (R_{uj} - \bar{R}_j)}{\sum_{u \in U_{ij}} |sim(i, j)|} \quad \text{公式 (4-6)}$$

$sim(i, j)$ 表示学生 i 与最近邻居 j 的相似度， \bar{R}_j 表示题目 j 在所有作答结果中的平均值，循环计算 $P_{u,i}$ 得到的评分值回填到学生-题目作答矩阵中。

4. 学生用户间相似性计算

下面计算两个学生的相似度，计算公式如下所示：

$$sim(u, v) = \frac{\sum_{c \in I_{uv}} (R_{ui} - \bar{R}_u)(R_{vi} - \bar{R}_v)}{\sqrt{\sum_{c \in I_u} (R_{ui} - \bar{R}_u)^2} \sqrt{\sum_{c \in I_v} (R_{vi} - \bar{R}_v)^2}} \quad \text{公式 (4-7)}$$

公式中， I_{uv} 表示学生 u 和学生 v 共同作答的题目集合，学生 u 和学生 v 作答的题目步骤的集合分别用 I_u 和 I_v 来表示，学生 u 和学生 v 对题目步骤 i 的作答结果分别用 R_{ui} 和 R_{vi} 来表示，使用 \bar{R}_u 和 \bar{R}_v 来表示学生 u 和学生 v 对题目的平均作答结果值。

5. 最近邻居集的生成

结合所有用户集合，获取和目标学生 u 相似度明显的 K 个学生，确定成 u 的最近邻居集合 $N(u) = \{u_1, u_2, \dots, u_k\}, u \notin N(u)$ 同时 $N(u)$ 内学生 u_k 根据相似明显性 $sim(u_k, u) (1 \leq k \leq K)$ 进行排列。

6. 产生推荐结果

同样采用加权平均的策略，产生学生 u 对题目步骤 i 的预测作答结果为：

$$P_{u,i} = \bar{R}_u + \frac{\sum_{c \in I_{uv}} sim(u, v) \times (R_{vi} - \bar{R}_v)}{\sum_{c \in I_{uv}} |sim(u, v)|} \quad \text{公式 (4-8)}$$

公式内 $P_{u,j}$ 作为预测回答结果值，其余符号与此前的定义一致。求出最近邻居内相同问题差异化的回答结果，以此确定加权平均值，选择非集合 I_u 中的前 N 项值即推荐集 Top-N。

7.计算 RMSE 值对比观察

将填充后的计算出的预测结果用实际结果值进行比较，可以计算得到 RMSE 值，通过 RMSE 值来观察填充后的模型效果。实验结果如表 4.3 或图 4.7 所示：

算法/邻居数	5	10	15	20	25
K					
User-based	0.3526	0.3433	0.3305	0.3279	0.3240
Item-based	0.3520	0.3426	0.3300	0.3259	0.3220
填充后的结 果	0.3490	0.3380	0.3278	0.3170	0.3079

表 4.3 RMSE 值比较

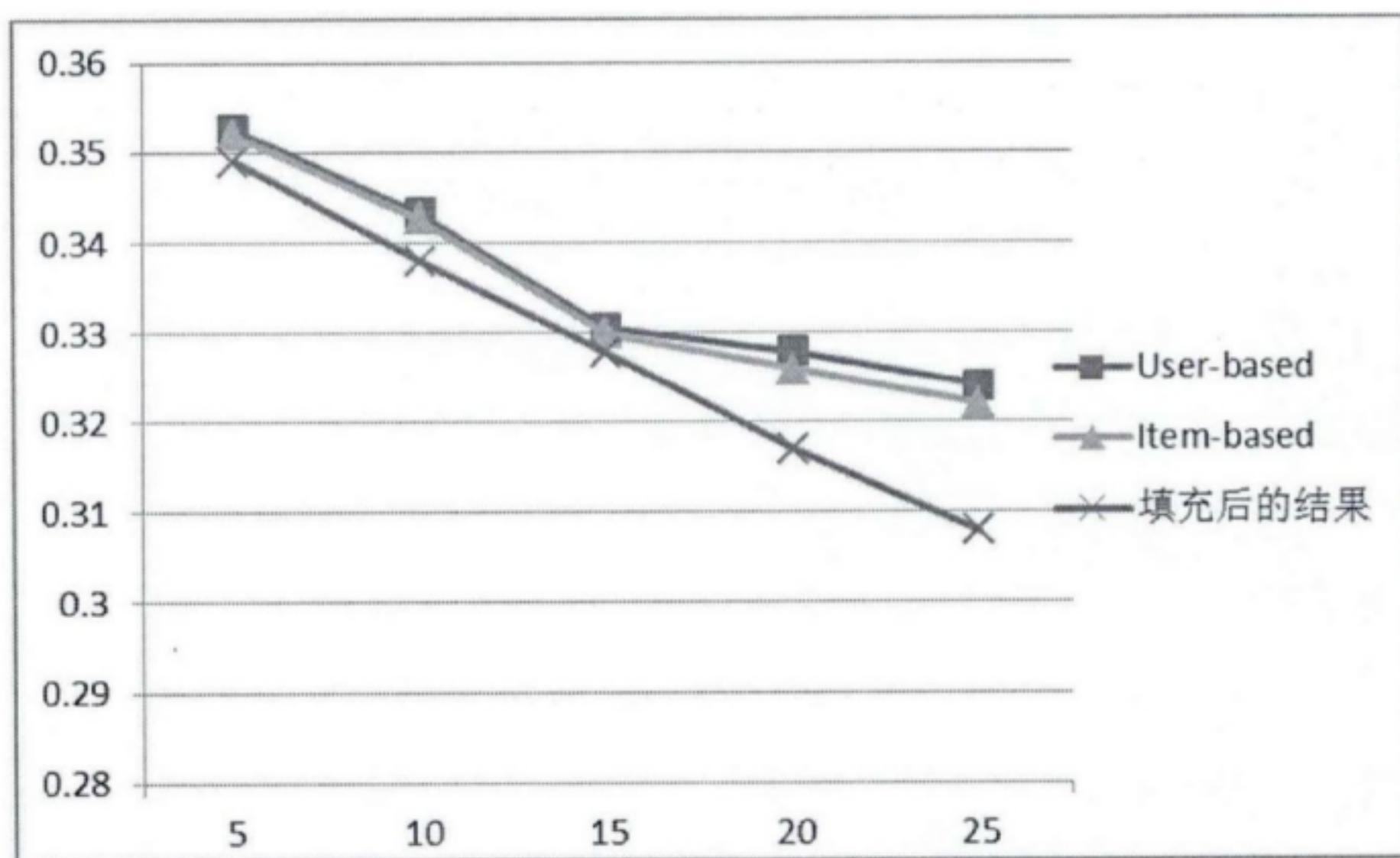


图 4.7 三种方法的 RMSE 值比较

由表和图中数据可以看出，在对学生-题目作答矩阵进行填充后，计算出的 RMSE 值较小，说明矩阵填充后得到的推荐结果相比填充前较好。因用户回答同一问题的矩阵规模快速增加，这种模式能够节省获取最近邻居用户的时间，确保推荐结果更加合

理。本次实验的结果要优于比赛中获得第三名的 0.3328 的结果^[41]，说明协同过滤算法能很好的契合到教育数据集的挖掘中来，并能获得比较好的效果。

4.2.5 实验结论

通过上述实验，得到以下结论：

通过三中协同过滤推荐算法仿真实验可知，改进后的推荐算法最终呈现出的预测结果更为准确；而分析相似度计算模式时，发现调整后的的余弦相似性计算方法更精确的结论；对于回填稀疏矩阵后再次运用推荐算法进行推荐，与原推荐算法比较，前者效果更准确的结论。

第五章 总结与体会

5.1 论文主要工作总结

随着网络远程教育、在线课程等在线教育的不断发展，个性化教育也逐渐发展起来。如何在网络教育中存储的海量用户数据中，提取出反映用户学习能力水平，对提高用户学习水平有帮助的有价值的信息，这就显得尤为重要了。教育数据挖掘就是在这种背景之下产生和发展起来的。协同过滤推荐算法在商业推荐领域应用广泛，它主要是针对用户使用者提供个性化推荐。协同过滤推荐算法作为该领域普遍使用的一种算法，因其内部包含大量分支算法，能够结合实际情况采取相应的推荐进行处理。本文将协同过滤技术结合教育数据挖掘，使用 Apache Mahout 中的 Taste 组件提供的各种方法，对教育数据集进行仿真建模。实验得到了很好的预测效果。

本文主要工作有如下内容：

1. 实现教育数据挖掘有关知识的了解，全面说明机器学习与数据挖掘两个领域采取的操作模式，概要说明 KDD2010 比赛命题；学习与分析了推荐系统与当下管饭使用的算法，同时系统全面的比较了不同推荐算法间存在的差距。

2. 重点研究协同过滤推荐算法各个环节。这种算法普遍使用的三种类型是，以用户为主体的协同过滤推荐算法（User-Based CF）、以项目为主体的协同过滤推荐算法（Item-Based CF）和以模型为主体的协同过滤推荐算法（Model-Based CF）。在基于模型的系统过滤推荐算法中重点介绍了基于 SVD 的协同过滤推荐算法，SVD 算法相对更加准确高效。本文对这三类算法的原理以及实现进行详尽的剖析，指出各自的优劣性以及相应的使用情景。

3. 应用 Apache Mahout 中的 Taste 开源框架，使用 KDDCup2010 比赛数据集和 RMSE 评估标准，对这三类协同过滤分别进行了仿真实验。第一步，和过去的系统过滤模式进行比较，系统说明实验步骤，求解对应结果。第二步，分析相似度求解模式的有效性，站在实现原理的角度对可以取得最理想结果的余弦相似性进行说明，同时进行相关验证。第三步，探索矩阵稀疏性情况，利用补充学生-题目评分矩阵内容的模式控制其稀疏度，增强推荐结果的合理性。

本论文的创新性主要体现在两点：一是预处理阶段的大数据并发处理的方式，将时间复杂度从原先十几个小时缩短为 10 分钟左右；二时将协同过滤算法应用到了教

育挖掘领域，将两个领域的研究融合到一起，为教育领域数据挖掘提供新思路。

5.2 今后所要开展工作

本文针对协同过滤推荐算法已经做了大量的实验处理，但是还有很多不足之处，主要包括如下几个方面：

1.预处理效率较低。本文针对数据集中的 8918054 条记录中去重提取的 1357180 个步骤作为 item 项，所消耗的时间复杂度巨大，仅仅是计算前三百个学生不同答题步骤就耗费了将近 1 个小时，而计算全部三千个学生不同做题步骤数所消耗的时间负责度则呈几何倍的数量级递增，初步计算所消耗的时间大概在 100 小时左右。这个数据集就目前来说还是比较小的，在现实情况下，数据系统的数据量非常庞大，按照去重计算项目的方法是不切实际的。最好的方法是单独另设置一个题目表，将所有的题目记录在题目表中，并将其与学生-题目作答结果表进行关联，这样就可以很方便的统计所有不同题目数以及学生共同作答题目数。这种方法为教育数据挖掘研究者提供了一种启示。

2.完成预测结果的深入探究。因协同过滤推荐算法仅仅面向推荐领域，最后给学生提供的只是部分明显能够回答正确的问题。由 4.2.1 节到 4.2.3 节的推荐结果就可以看出，推荐的题目都是学生作答结果为 1 的题目。但是现实情况下，作答结果为 1 的题目未必能够反映出学生的能力或反映一部分能力，我们还要将学生打错的题目综合起来共同评价学生的学习能力。这些在协同过滤领域并没有这一类方法的提供，这就为我今后研究的方向提供了很好的命题。我相信这一研究方向拥有很好的发展前景。

3.由于基于模型的协同过滤有很多种，例如基于随机森林模型的协同过滤和基于 Factor Model 模型的协同过滤，每种模型都有一定的适应场景。我没有一一分析每种模型的情况，只是单独挑选一个典型的 SVD 模型作为代表进行计算，这是不严谨的。所以在今后的工作中还要挑选更多的模型来参照对比，最终得到最好的效果。

由于时间的关系，我的论文完成的可能有些仓促，某些原理和实现可能没有清晰明了的表现出来，有些数据的统计可能出现了细微的误差。但是我的论文为教育数据挖掘提供了一种思路，并且取得了一定的成果。我认为协同过滤推荐算法同教育数据挖掘领域相互结合的研究方法很有研究价值，我将致力于这一研究领域，为大数据研究和智能教育的发展贡献一份力量。

致 谢

首先，本论文的理论部分的撰写和实验部分的规划，得到了我的导师袁梅宇老师的悉心指导，所以我十分感谢他的栽培。袁老师不仅在学习上对我进行指导和帮助，更在生活上是我的良师益友。他在专业研究上严谨的态度、对学生因材施教的治学理念，得到了校内外老师和同行们的一致好评，他也成为了我日后在工作中的榜样。在研一阶段袁老师就给我定下了眼下十分热门的机器学习和数据挖掘研究方向，从开题阶段就如何选题进行深入的探讨，到中期袁老师对我研究内容的严格把关，最后论文的批改审核，自始至终都注入了他的心血和精力，可以说没有袁老师就没有这篇论文的最终完成。在此向袁老师表达我最崇高的敬意和最深情的感谢。

其次，我要感谢我的同学刘翠翠和宿舍同寝室的王维、齐祥祥、司怀伟，感谢他们平时对我的各项帮助，在我困难的时候和因为科研心情烦闷的时候，是他们的开导和安慰，才是我重新树立了自信心，克服了各项困难完成了科研任务。我永远都不会忘记和你们在一起的共同进步的美好时光。

再次，自己在研究生阶段有幸加入了研究生协会，在这个大家庭中我结识了很多良师益友，并且参加了很多精彩纷呈的活动，研究生协会的日子真的让自己受益匪浅，它对我今后进入到社会积累了宝贵的经验阅历。

最后，感谢我的父母孟志强范宝婷，他们这么多年来不辞辛劳的抚养和教育了我，在我学习烦闷时是他们的鼓励让我看到了前行的希望。书本上的知识早晚会忘记，但父母的教诲一辈子都不会忘。他们教会了我做人的道理，让我学会做事之前先要学会做人，他们的人生观世界观已经深深影响融入我的血液中，在此，向我的家人表达我最深沉的祝福和感谢。

致谢人：孟卓

2016年3月1日

参考文献

- [1].李婷,傅钢善. 国内外教育数据挖掘研究现状及趋势分析[J]. 现代教育技术, 2010,20(010):21-25.
- [2].Educational Data Mining[DB/OL]. [2014-06-01].
<http://www.educationaldatamining.org>.
- [3].David Goldberg, David Nichols, Brian M. Oki and Douglas Terry. Using collaborative filtering to weave an information tapestry. Communications of the ACM, Volume 35, Issue 12, pp61-70, 1992.
- [4].Paul Resnick, Neophytos Iacovou, Mitesh Suchak, Peter Bergstrom, John Riedl. GroupLens:an open architecture for collaborative filtering of netnews. Computer Supported Cooperative Work, pp175-186, Chapel Hill, North Carolina, 1994.
- [5].F.Heylighen. Collaborative Filtering[EB/OL].
<http://pespmcl.vub.ac.be/COLLF-ILT.html>.
- [6].李春.协同过滤推荐算法的研究[D].湘潭大学,2010.
- [7].邓爱林,朱扬勇,施伯乐. 基于项目评分预测的协同过滤推荐算法[J]. 软件学报, 2003.09:1621-1628.
- [8].Badrul Sarwar, George Karypis, Joseph Konstan, John Riedl. Item-Based Collaborative Filtering Recommendation Algorithms. WWW 01 Proceedings of the 10th international conference on World Wide Web, Pages 285-295,ACM New York, NY, USA ©2001.
- [9].袁梅宇. 数据挖掘与机器学习—WEKA 应用技术与实践[M].北京: 清华大学出版社, 2015.
- [10].赵尔丹,张照枫. 基于数据仓库和数据挖掘的决策支持系统的研究与应用[J].河北软件职业技术学院学报.2015,7(1):47-50.
- [11].马金徽. 高维混合类型数据聚类算法研究[D].内蒙古科技大学,2011.
- [12].赵又霖,邓仲华,陆颖隽. 数据挖掘云服务分析研究 [J]. 情报理论与实践,2012,35(9):33-36.
- [13].朱勇. 基于关联规则的数据挖掘在邢嫌系统中的应用[D].华中科技大学,2006
- [14].王晓晖,风笑天,田维绪.论样本代表性的评估[J].山东社会科学,2015,(3)-2015:88-92.
- [15].郝媛,高学东,孟海东.高维数据对象聚类算法效果分析 [J].中国管理信息

- 化,2012,(8)-2012:51-53.
- [16].彭加红.一种基于粗糙集的混合特征选择算法[J].计算机工程与科学,2005.09:57-59.
- [17].方烈.分类方法在交通数据挖掘的应用研究[D].上海交通大学,2006.
- [18].王荣.分类技术及其在客户关系管理中的应用[D].浙江大学,2006.
- [19].林鑑.粗糙集在纹理图像分类中的应用研究[D].浙江师范大学,2011.
- [20].宫悦.基于粗集的不完备信息系统数据挖掘方法研究[D].大连海事大学,2008.
- [21].廖定安.一个基于聚类挖掘的信息协作分析模型[J].科技信息,2012,(10)-2012:117-118.
- [22].雷蕾,吴乃君,刘鹏,刘兰娟.灵敏度分析:分类器中的缺失数据[J].管理学报,2005.09:153-157.
- [23].曹宁,高莹,徐根祺.决策树方法的研究进展[J].科技视界,2014,(20):72.
- [24].范敏,石为人.层次朴素贝叶斯分类器构造算法及其应用研究[J].仪器仪表学报,2010,(4)-2010:776-781.
- [25].王维娜.基于相对位置视点的数据集精简算法研究[D].海南大学,2007.
- [26].欧阳浩,陈波,王萌,黄镇谨.基于网格的二次 K-means 聚类算法[J].广西工学院学报,2012,23(1)-2012:24-27,33.
- [27].方玮玮.基于关联规则的购物篮分析[J].四川理工大学学报:自然科学版,2010,23(4)-2010:430-434.
- [28].朱红蕾,李明.维护关联规则的算法研究[J].兰州理工大学学报,2004,30(5)-2004:104-107.
- [29].<http://mahout.apache.org/>.
- [30].Song-Jie Gong, Hong Wu Ye. Combining Memory-Based and Model-Based Collaborative Filtering Recommender System[C]. Circuits, Communications and Systems, 2009, Pacific-Asia Conference on Page(s):690-693.
- [31].J.D.M. Rennie and N.Srebro. Fast maximum margin matrix factorization for collaborative prediction[C]. In: Proc of ICML, 2005.
- [32].鲁为.协作过滤算法及其在个性化系统中的应用[D].北京邮电大学,2007.
- [33].G.H.Golub and C.Reinsch. Singular value decomposition and least squares solution[J]. Numerische Mathematik, 1997, 14(5):403-420.
- [34].赵亮,胡乃静等.个性化推荐算法设计[J].计算机研究与发展,2002,39(8):986-991..

- [35].赵正天.基于量子机制的分类属性数据聚类算法研究[D].兰州理工大学,2009.
- [36].程英英.web 挖掘技术及其在邮件系统中的应用[D].南开大学,2011.
- [37].梁成军,张红英.相关系数与关联度在体育科研中的应用对比[J].高师理科学刊,2002,22(1)-2002:52-54.
- [38].R.J.Mooney,P.N.Bennett, and L.Roy. Book Recommending Using Text Categorization withExtracted Information[C].In:Proc.Recommender Systems Papers from 1998 Workshop, Technical Report WS-98-08,1998.
- [39].Andreas Toscher, Michael Jahrer. Collaborative Filtering Applied to Educational Data Mining[J]. Journal of Machine Learning Research ,2010:576-587.
- [40].张亮.推荐系统中协同过滤算法若干问题的研究[D].北京邮电大学,2009.
- [41].Kun Liu, Yan Xing. A lightweight solution to the educational data mining challenge[C]/Proceedings of the KDD Cup 2010 workshop knowledge discovery in educational data.2010:76-82.

附录A 攻读硕士期间发表论文以及软件著作权

- [1].孟卓,袁梅宇.教育数据挖掘发展现状及研究规律的分析[J].教育导刊,2015,555(2):29-33.
- [2].软件著作权:博客管理系统.2014,登记号:2014SR217828.
- [3].软件著作权:铭卓购物商城系统.2015,登记号:2015SR008782.