

基于协同过滤与蚁群聚类的文章推荐模型

摘要

在这个信息技术和互联网迅速发展的时代，人们面临的各种数据极其繁杂，对于用户而言，要找到自己想要的信息极为困难，因此推荐系统的研究应用而生。

为了确定信息推荐的变量，考虑到推荐系统的评估主要的体现是用户的满意度，我们通过参考其它领域的推荐系统的指标，引用到教育行业，并且进行了分析调整，给出教育推荐系统的关于准确性、覆盖率、推荐过程实时性，平均误差等的指标体系。

在该指标体系的基础上，为了构建推荐算法和建立高等学校信息推荐数学模型一方面，针对多次进入系统的用户（老用户），我们初步根据用户对文章的点击量和下载量，利用协同过滤方法，在当前文献数据库基础上，利用 k 近邻聚类算法给出与用户浏览文章相似度较高的一些文章，并对其进行排序，从中挑选出相似度较高的 10 篇文章，在用户下次进入系统时，推荐给他们。但是考虑到 k 近邻聚类算法本身的局部最优解问题会降低系统推荐的准确性，同时考虑到传统协同聚类方法对稀疏性问题、冷启动问题、扩展性等问题的束手无策性，我们在传统协同过滤推荐技术基础上，采用蚁群聚类算法代替协同过滤算法中的 k 近邻聚类对用户进行分类，在整个用户空间高效的查找邻居用户群，从而提高了系统对目标用户推荐的准确率和推荐质量。另一方面，考虑到第一次进入系统的用户没有行为表现，我们也通过对用户身份的辨别，在模型的基础上给出了一个文件下载次数检索的算法，基于该算法给出了适用于新用户的推荐方法。最后，我们对模型进行了准确性和实时性指标检验，同时对模型进行了实际性推送结果的检验，检验结果说明了我们改进模型的优越性和良好的准确性。最后，我们根据模型，给出了基于协同过滤与蚁群聚类的文章推荐系统，并基于该方案，我们形成了一份不超过 1000 字的信，在信中向有关部门推介了我们的推荐方案。

关键词： 身份辨别 推荐模型 协同过滤 自动搜索算法 蚁群聚类算法

目录

一、引言	3
二、问题分析	6
2.1 评价指标体系的分析	6
2.2 推荐系统模型的分析	6
2.3 推荐系统模型算法的分析与解决方案	6
2.4 推荐系统的设计与实现	7
三、模型假设	7
四、评价指标体系的建立和推荐算法的设计	7
4.1 评价指标体系的建立	7
4.1.1 准确性指标的建立 ^[1]	7
4.1.2 实时性指标的建立	9
4.1.3 覆盖率指标的建立	10
4.1.4 指标综述	10
4.2 协同过滤算法	10
4.2.1 协同过滤算法基本思想	10
4.3 蚁群聚类算法	11
4.3.1 蚁群算法思想方法 ^[8]	11
4.4 文件下载次数检索算法	12
4.5 身份辨别算法	13
五、基于协同过滤算法的文章推荐系统模型	13
5.1 基于协同过滤算法的文章推荐系统模型的建立	13
5.1.1 数据的评分源的获取	13
5.1.2 推荐结果的输出	15
5.1.3 协同过滤模型的评估	15
六、文章推荐系统模型的改进以及推荐方案的形成	17
6.1 基于协同过滤与蚁群聚类的文章推荐模型的建立	17
6.2 基于文章推荐模型的推荐方案的形成	18
七、蚁群聚类算法的文章推荐系统模型的求解及其算法的实现	19
7.1 算法实现	19
7.1.1 蚁群算法的实现	19
7.1.3 蚁群聚类算法 (MATLAB)	23
7.1.4 新用户信息检索算法 (SQL)	24
7.1.5 文件下载次数检索算法 (JSP)	24
7.2 基于蚁群算法的文章推荐系统模型求解	25
7.3 基于蚁群算法的文章推荐方案的形成	25
八、模型的检验	27
8.1 平均准确度 (MAE) 检验	27
8.2 推荐过程耗时性检验	29
8.3 实际推荐结果检验	31
九、模型的评价	31
9.1 模型的优点	31

9.2 模型的缺点.....	32
十、推荐信.....	32
十一、参考文献.....	33

一、引言

1. 1 选题的背景和意义

现在的时代是互联网时代，也是信息爆炸的时代，云计算、大数据、数据挖掘、物联网、移动计算等技术已经广泛应用到社会发展的方方面面，各行各业信息化步伐不断加快，社会信息化程度不断提高，信息技术对教育的革命性影响日趋凸显。内容太多，而用户的时间太少，如何选择成了难题。所以对海量信息进行筛选、过滤，将用户最关注最感兴趣的信息展现在用户面前，能大大增加这些内容的转化率，对各类应用系统都有非常巨大的价值。

而在教育体系中亦是如此，虽然教育信息化在教务信息系统、教学与学习系统、自动化办公系统、校园一卡通系统、目标服务系统已经成为了各高校教育信息化建设最基本和核心的内容。但是，无论在学术研究还是教育信息化的推进过程中，有一项重要的教育信息化内容被严重忽略，那就是信息推荐系统。所谓信息推荐系统就是指利用教师、科研人员、学生及管理人员的一些学术、学习、管理行为，通过一些数学算法，推测出个人可能感兴趣的最新文献、发表新成果的作者以及准备开设的课程。

1. 2 国内外现状和发展趋势

随着图书馆资源的不断丰富和现代信息技术的飞速发展，各大高校纷纷建立了藏书丰富的图书馆以及拥有海量资源的数字图书馆。而且大量跨学科及新兴学科的出现使得图书馆资源不仅数量庞大而且学科类型繁杂各异。学生到图书馆借书愈发茫然，内容太多，而用户的时间太少，如何选择成了难题。同时在现有网络环境下，人们想要准确、快速地查找到自己所需的图书信息反而越来越困难，一是图书检索结果可能有成千上万条，从中寻找到目标结果必然消耗大量时间和精力；二是用户需求千奇百怪，不同学科专业、习惯偏好等导致用户需求个性化和多样化，而传统的图书检索系统满足不了用户个性化需求。所以，对海量信息进行筛选、过滤，将用户最关注最感兴趣的信息展现在用户面前，进一步大大增加这些内容的转化率，对各类应用系统都有非常巨大的价值。个性化推荐研究成果主要应用于电子商务领域，目前，数字图书馆领域推荐系统有：卡内基·梅隆大学开发的主动协作过滤推荐系统，主要用于电子文档的推荐；MIT 开发的 GroupLens 协同过滤推荐系统，主要用于新闻信息的推荐；明尼苏达大学开发的协作过滤推荐系统可以用于 Web 电影资源的推荐；孙博阳等提出了图书馆数字资源的管理系统，对文献资源进行有效管理；冯太琴针对北京工业大学图书馆实际问题提出了个性化选书导购系统；肖红等引入 RSS 技术到图书馆数字资源个性化推荐。国内外图书馆领域个性化推荐服务大多是基于数字资源的协同过滤推荐，寻找最近邻产生推荐结果；结合学生背景进行专业类书籍、公共类书籍推荐的研究很少。我们尝试建立一个个性化图书推荐系统，结合西北师范大学实际情况，从历史借阅数据中实现对图书馆馆藏情况、

读者的需求情况等方面进行分析，并以专业、图书类别、新书推荐等五方面进行图书推荐，依据读者背景信息提供图书的智能检索和推荐服务，帮助用户进行信息筛选提高检索效率，积极引导大学新生的图书借阅，拓宽读者的阅读视野，满足不同用户的个性化需求。

1.3 研究思路

在本课题中，我们需要解决的问题就是：

1. 建立信息推荐的指标体系，确定信息推荐的变量；
2. 建立高等学校信息推荐数学模型；
3. 解决传统推荐算法存在的问题：稀疏性问题、冷启动问题、扩展性等问题
3. 就信息推荐数学模型设计推荐算法；
4. 推荐系统设计与实现；
5. 给有关部门写一份不超过 1000 字的信，推介我们的推荐方案。

1.4 研究方法

(1) 文献研究法：通过文献资料收集，了解现有推荐系统研究的进展情况和已取得的成果以及存在的问题。

(2) 协同过滤算法：协同过滤算法的基本思想是：如果用户对一些项目的评分比较相似，则他们对其他项目的评分也会较为接近。通过与目标用户相似的邻居用户群的评分预测推荐结果，从而达到推荐给目标用户喜欢的项目的目的。其核心就是通过最近邻的评分产生最后的推荐结果，当前用户对未评分项目的评分通过最近邻对该项目评分的加权平均值逼近。例如：用户 A 喜欢物品 a，用户 B 喜欢物品 a、b、c，用户 C 喜欢物品 a 和 c，由此可知，物品 a 和 c 非常相似，又由于喜欢物品 a 的用户也同时喜欢物品 c，而用户 A 喜欢物品 a，因此把物品 c 推荐给用户 A。

(3) 蚁群算法：蚁群算法 (ant colony optimization, ACO)，又称蚂蚁算法或蚁群优化，是一种用来在图中寻找优化路径的机率型技术。它由 Marco Dorigo 博士于 1992 年在他的博士论文中引入，蚁群聚类算法的灵感来源于自然界中蚂蚁寻找食物的过程。蚂蚁之间通过一种叫做信息素 (pheromone) 的物质进行互相通信，协作完成共同的目标；蚂蚁会在走过的路径上留下信息素，走过的蚂蚁越多，路径上的信息素积累的量就越大，信息素会随时间消失，但也会对其他蚂蚁产生吸引的作用；通过信息素交流的蚂蚁群体便产生一种正反馈的现象：某一条路径上走过的蚂蚁越来越多，留下的信息素量越来越大，会吸引更多的蚂蚁选择这条路径，蚂蚁个体之间就是通过这种方式找到距离食物最短的路径。蚁群算法是一种启发式的模拟进化算法。初步的研究表明该算法具有许多优良的性质。并且蚁群算法具有一种新的启发式模拟进化优化方法的有效性和应用

价值，是一种求解组合最优化问题的新型通用启发式方法，该方法具有正反馈、分布式计算和富于建设性的贪婪启发式搜索的特点。^[5]

二、问题分析

教育推荐系统主要是想通过教师、科研人员、学生及管理人员的一些学术、学习行为，通过一些数学算法，推测出个人可能感兴趣的最新文献、发表新成果的作者以及准备开设的课程。然后在他们下次进入系统时，推荐他们可能喜欢的文章。考虑到第一次进入系统的用户没有行为表现，文中也给出适用于新用户的推荐方法。

2.1 评价指标体系的分析

对于该问题，考虑到推荐系统的评估主要的体现是用户的满意度，我们通过参考其它领域的推荐系统的指标，引用到教育行业，并且进行了分析调整，给出教育推荐系统的关于准确性、覆盖率、平均误差等的指标体系。

2.2 推荐系统模型的分析

对于该问题，我们首先利用协同过滤的方法，初步给出了基于协同过滤方法的信息推荐模型，再在原有模型基础上，利用蚁群聚类方法改进了协同过滤方法；最后给出我们的设计思路和设计过程的总体框架；再在该框架下给出具体的数学模型。

2.3 推荐系统模型算法的分析与解决方案

由于新老客户浏览量的不同和用户行为表现的不同，我们利用用户身份，设计针对新用户信息检索算法和文件下载次数检索算法，在该基础上利用辨识算法进行身份识别，从而区分了新老用户，再在当前文献数据库基础上，利用 k 近邻聚类算法给出与用户浏览文章相似度较高的一些文章，并对其进行排序，从中挑选出相似度较高的 10 篇文章，在用户下次进入系统时，推荐给他们，但一方面考虑到 k 近邻聚类算法本身的局部最优解问题会降低系统推荐的准确性，另一方面考虑协同聚类方法对稀疏性问题、冷启动问题、扩展性等问题的束手无策性，我们在传统协同过滤推荐技术基础上，因此我们采用蚁群聚类算法来代替协同过滤算法中的 k 近邻聚类对用户进行分类，在整个用户空间高效的查找邻居用户群，来达到提高系统对目标用户推荐的准确率和推荐质量的目的。

2.4 推荐系统的设计与实现

根据已确定和建立的算法将前面建立的数学模型进行程序化处理，给出基于协同过滤与蚁群聚类的文章推荐系统模型，并对该模型进行系统设计和算法实现。

三、模型假设

1. 我们的模型只针对高校的文章推荐系统。
2. 假设我们已经建立了完整的学生信息数据库。该学生信息库中包含学生的学号、专业、年级等一些基本信息。
3. 假设我们已经建立了文献数据库。

四、评价指标体系的建立和推荐算法的设计

4.1 评价指标体系的建立

对于一个推荐系统好坏的评价，最主要的体现是用户的满意度。而满意度的调查大多数是基于用户评分、问卷调查或窗口弹出提问等的方法，如果使用调这些方法，由于用户的随意性，可能会有用户随意评分，或者不会评分，更有甚者出于厌恶情绪，会给出不符合实际情况的恶意评分等，这一系列情况的出现都会使得最终的结论失去准确性。所以我们只能通过一些能够量化的指标来建立评价指标体系，比如说点击率、是否下载、停留时间增幅等。

4.1.1 准确性指标的建立^[1]

推荐系统的准确性是最根本的指标，也是最受我们关注的指标。准确性越高，表示该推荐系统推荐的内容用户正是用户喜欢的类型，而且这也表明了该推荐系统的实用性很高。

而准确性又可以分为分类准确性、平均绝对误差、排序准确性。下面分别给出这三种准确性的具体定义。

平均绝对误差，在本文中，我们采用 MAE 的计算方法来计算推荐系统的平均绝对误差，平均绝对误差是最常用的评估方法之一，它是通过统计预测值和真实评分值之间的绝对误差值得到的，计算公式如下：

$$\text{MAE} = \frac{\sum_{u,i \in T} |r_{ui} - \hat{r}_{Mi}|}{n} \quad (1)$$

其中， n 为评分的总数， r_{ui} 为用户 u 对项目 i 的实际评分， \hat{r}_{Mi} 为用户对项目 i 的预测评分， MAE 数值与推荐的预测准确性呈反比。^[6]

4.1.1.1 分类准确性指标

指对于推荐系统推荐的文章与用户喜欢的文章相符的程度。

1. 点击率计算公式如下：

$$\zeta_i = \frac{W_i}{W_{\text{总}}} \quad \zeta = \frac{1}{M} \sum_{i=1}^M \zeta_i \quad (2)$$

式中， ζ 表示平均点击率， ζ_i 表示用户 i 的点击率， W_i 表示用户在推荐文章中点击的文章数， $W_{\text{总}}$ 表示推荐系统推荐文章数， M 表示用户总数。

2. ACU 指标：指 ROC 曲线下的面积，用来衡量推荐系统多大程度将用户喜欢与不喜欢的文章分开。^[2]假如我们已经得到了所有样本的概率输出（属于正样本的概率），我们根据每个测试样本属于正样本的概率值从大到小排序。每次选取一个不同的 threshold，我们就可以得到一组 FPR 和 TPR，即 ROC 曲线上的点，将它们画在 ROC 曲线中。当我们将 threshold 设置为 1 和 0 时，分别可以得到 ROC 曲线上的 (0, 0) 和 (1, 1) 两个点。将这些 (FPR, TPR) 对连接起来，就得到了 ROC 曲线。当 threshold 取值越多，ROC 曲线越平滑。

下面给出一个模拟的 ROC 曲线样例：

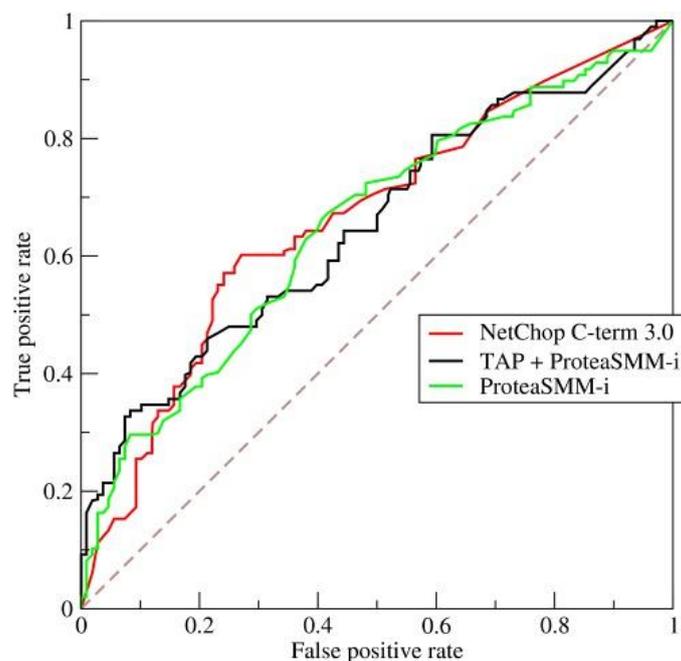


图 4.1.1.1^[2] ROC 曲线

除此之外，常用的指标还有 F1 指标和召回率。由于召回率一般和点击率成反比，点击率越高，召回率越低，所以在这里不给出计算公式；而 F1 指标是基于点击率和召回率建立的一种指标，我们只需知道，F1 越大，体系越好。

4.1.1.2 排序准确性指标

指对于推荐的文章内容排序与用户喜欢程度的排序拟合度。计算公式如下：

$$R_{im} = \frac{l_{im}}{L_i} \quad (3)$$

式中， R_{im} 表示推荐系统的排序准确性， l_{im} 表示用户 i 对于文章 m 在推荐列表中的排序， L_i 表示用户 i 没有选择的文章总数。

4.1.2 实时性指标的建立

对于一个推荐系统，评判其好坏程度的另一重要指标就是实时性。好的系统，一方面，要能够自动实时跟新文献数据库信息，并且进行推荐；另一方面，要在最短的时间内给用户推荐其想要的文章。在文中，我们主要利用推荐过程耗费时间来作为判断模型优良的重要指标。

4.1.3 覆盖率指标的建立

覆盖率：推荐系统向用户推荐的文章占全部文章的比例。计算公式如下：

$$C_R(L) = \frac{N_d(L)}{N} \quad (4)$$

式中， $C_R(L)$ 表示推荐系统的覆盖率， N 表示文章总数， $N_d(L)$ 表示推荐系统可以为用户推荐的文章总数， L 表示推荐系统为用户推荐的文章总数。

当然，除了推荐系统覆盖率，相似的，可以分析其预测覆盖率和种类覆盖率。

4.1.4 指标综述

综合前面的分析，我们按照指标的等级，将指标体系总结如下：

文章推荐系统指标体系		
一级指标	二级指标	三级指标
准确性	分类准确性	点击率
		ROC 曲线
		F1 指标
	平均绝对误差	召回率
	排序准确性	平均排序分类
实时性	推荐过程耗费时间	
覆盖率	预测覆盖率	
	用户覆盖率	
	推荐覆盖率	
	种类覆盖率	

接下来，我们利用所建立的这些指标进行算法的设计和模型的建立。

4.2 协同过滤算法

4.2.1 协同过滤算法基本思想

协同过滤算法的基本思想是：如果用户对一些项目的评分比较相似，则他们对其

他项目的评分也会较为接近。通过与目标用户相似的邻居用户群的评分预测推荐结果，从而达到推荐给目标用户喜欢的项目的目的。其核心就是通过最近邻的评分产生最后的推荐结果，当前用户对未评分项目的评分通过最近邻对该项目评分的加权平均值逼近。例如：用户 A 喜欢物品 a，用户 B 喜欢物品 a、b、c，用户 C 喜欢物品 a 和 c，由此可知，物品 a 和 c 非常相似，又由于喜欢物品 a 的用户也同时喜欢物品 c，而用户 A 喜欢物品 a，因此把物品 c 推荐给用户 A，因此协同过滤推荐系统的前提首先是获取用户对项目的数据评分源。

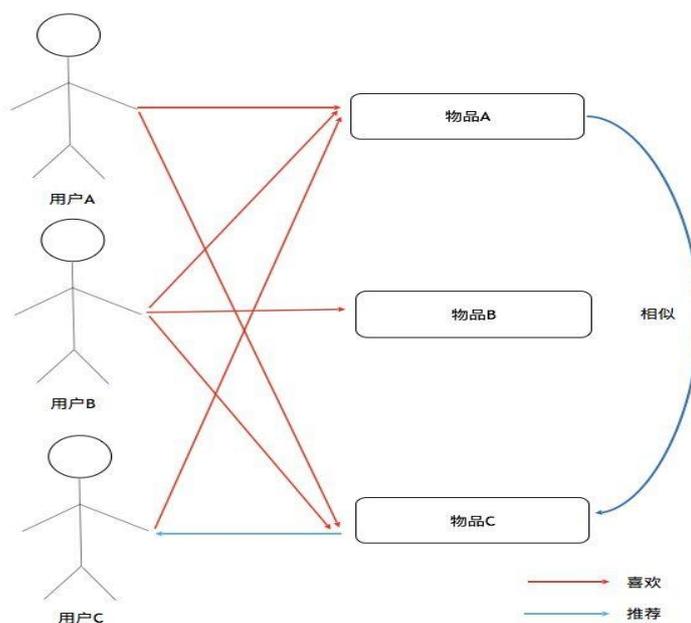


图 4.2.1^[3] 协同过滤算法推荐系统示意图

4.3 蚁群聚类算法

4.3.1 蚁群算法思想方法^[8]

蚁群算法(ant colony optimization, ACO)，又称蚂蚁算法或蚁群优化，是一种用来在图中寻找优化路径的机率型技术。它由 Marco Dorigo 博士于 1992 年在他的博士论文中引入，蚁群聚类算法的灵感来源于自然界中蚂蚁寻找食物的过程。蚂蚁之间通过一种叫做信息素 (pheromone) 的物质进行互相通信，协作完成共同的目标；蚂蚁会在走过的路径上留下信息素，走过的蚂蚁越多，路径上的信息素积累的量就越大，信息素会随时间消失，但也会对其他蚂蚁产生吸引的作用；通过信息素交流的蚂蚁群体便产生一种正反馈的现象：某一条路径上走过的蚂蚁越来越多，留下的信息素量越来越大，会吸引更多的蚂蚁选择这条路径，蚂蚁个体之间就是通过这种方式找到距离食物最短的路径。

蚁群算法是一种启发式的模拟进化算法。初步的研究表明该算法具有许多优良的性质。并且蚁群算法具有一种新的启发式模拟进化优化方法的有效性和应用价值，是一种求解组合最优化问题的新型通用启发式方法，该方法具有正反馈、分布式计算和富于建设性的贪婪启发式搜索的特点。^[6]

1. 利用蚁群聚类能够有效地降低推荐算法的计算量，有效的自发聚类提高了系统的伸缩性和鲁棒性；

2. 基于蚁群算法实现用户进行聚类，提高了协同过滤推荐系统的最近邻查询速度，降低了搜索开销，同时避免了使用 k 近邻聚类方法受初始聚类中心和聚类个数的影响。

3. 解决了（冷启动问题）新用户得不到推荐的问题，并提高了协同过滤推荐算法的精确。

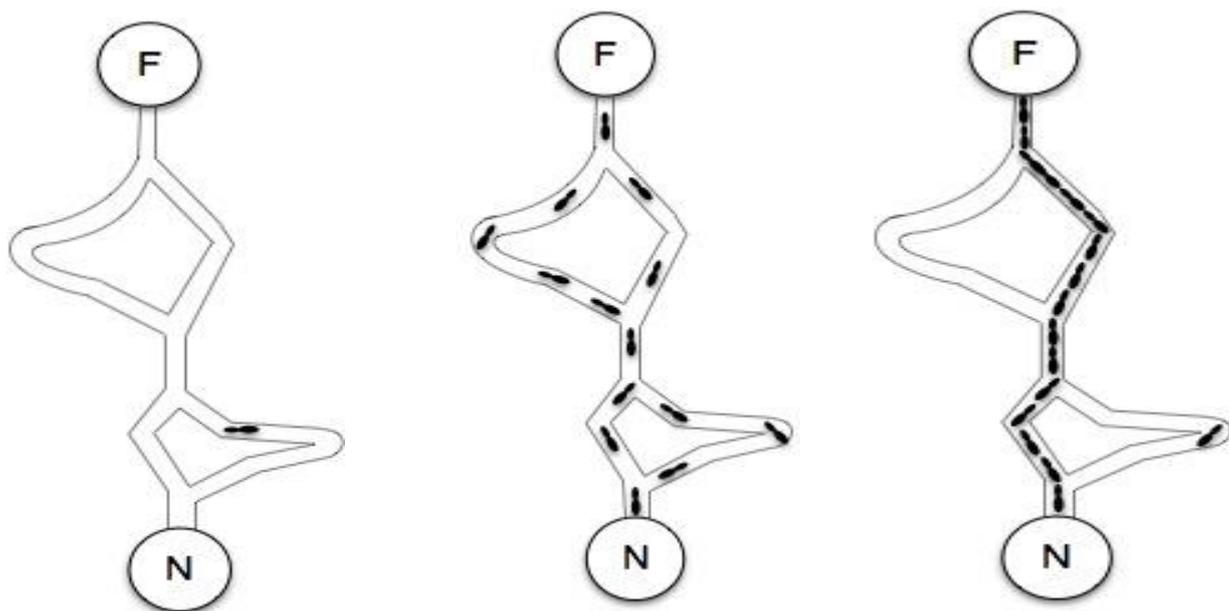


图 4.3.1 蚁群算法示意图

4.4 文件下载次数检索算法

对于文库中的文章，我们首先使用聚类算法，将相似的文章归类，再采用 JSP 对文件下载次数统计算法，对文库中的相关文章进行检测，然后按照检测的结果给出下载次数最多的 10 篇文章。

4.5 身份辨别算法

考虑到新老用户的不同，推荐文章也应有所不同，我们将利用程序设计一个身份识别算法，具体操作是，用户在使用身份代码（如学号等）登录系统时，首先调用该算法和在学生信息数据库和用户行为数据库中进行身份检索，然后对其进行辨别，如果是老用户，调用蚁群聚类算法，进行聚类，然后形成文章 TOP-N 推荐集；如果是新用户，我们首先将身份信息在信息数据库中搜寻，找到相关的信息，将其身份信息，传送到文件下载次数检索算法中。然后根据身份信息，在数据库中找到相关信息。给其推荐该领域热门文章。

五、基于协同过滤算法的文章推荐系统模型

5.1 基于协同过滤算法的文章推荐系统模型的建立

5.1.1 数据的评分源的获取

以 $D(U, I, R)$ 表示数据源，其中表 $U = \{U_1, U_2, \dots, U_m\}$ 示含有 m 个用户的集合， $I = \{I_1, I_2, \dots, I_n\}$ 表示含有 n 个基本用户的集合， $R^{(m,n)}$ 表示为用户对项目的评价矩阵， R_{ic} 表示用户 i 对项目 c 的评分，若用户 i 对项目 c 没有评分，则 R_{ic} 为 0。

用户\项目	I_1	I_2	...	I_c	...	I_n
U_1	R_{11}	R_{12}	...	R_{1c}	...	R_{1n}
U_2	R_{21}	R_{22}	...	R_{2c}	...	R_{2n}
...
U_i	R_{i1}	R_{i2}	...	R_{ic}	...	R_{in}
...
U_m	R_{m1}	R_{m2}	...	R_{mc}	...	R_{mn}

表 5.1.1.1 用户—项目评分矩阵

我们利用层次分析法确定出用户对项目喜好程度的权重，继而确定出评分标准。最近邻查询（k 近邻聚类）具体方法如下：最近邻查询是整个协同过滤推荐算法的核心，其效果和效率很大程度上决定了协同过滤推荐算法的效果和效率。而所谓最近邻居，就

是评分行为与当前用户比较相似的若干用户。例如：若用户给出的评分数据如表 3.2.1 所示，其中行代表用户，列代表项目，行列中的元素代表用户对项目的评分。

用户\项目	物品 a	物品 b	物品 c	物品 d
用户 A	5	3	4	5
用户 B	4	5	2	4
用户 C	3	2	5	2
用户 D	5	2	4	?

表 5.1.1.2 用户评分数据表

通过数据分析，我们可以发现 A 与 D 对物品的评分（喜好程度）非常相似，从而用户 A 是用户 D 的最近邻，因此用户 A 对物品 d 的评分与用户 D 对物品 d 的评分可能极为相似，由此我们可以通过用户 A 对物品 d 的评分来预测出用户用户 D 对物品 d 的评分。

而协同过滤推荐系统是以用户之间的相似性查找最近邻作为预测基础的。因此为了度量用户 i 和用户 j 之间相似性,我们基于以上做出的用户-项目评分矩阵来计算其相似度，考虑到不同用户的评分尺度问题，我们采用修正的余弦相似度量方法，具体计算方法如下：

设用户 i 和用户 j 共同评分过的项目集合为 I_{ij} ， I_i 和 I_j 分别表示用户 i 和用户 j 评分过的项目集合，则用户 i 和用户 j 之间的相似性为

$$sim(i, j) = \frac{\sum_{c \in I_{ij}} (R_{i,c} - \bar{R}_i)(R_{j,c} - \bar{R}_j)}{\sqrt{\sum_{c \in I_i} (R_{i,c} - \bar{R}_i)^2} * \sqrt{\sum_{c \in I_j} (R_{j,c} - \bar{R}_j)^2}} \quad (5)$$

式中， $R_{i,c}$ 表示用户 i 对项目 c 的评分， $R_{j,c}$ 加表示用户 j 对项目 c 的评分， \bar{R}_i 和 \bar{R}_j 分别表示用户 i 和用户 j 对所有项目的平均评分。

我们发现最近邻查询的目标就是对每一个用户 u，对于目标用户 u，协同过滤推荐系统需要在整个用户空间中搜索用户 u 的最近邻居集合 $C = \{C_1, C_2, \dots, C_k\}$ 。使得 $u \notin C$ ，且按照 C_i 与目标用户 u 之间的相似性 $sim(u, C_i)$ ($1 \leq i \leq k$) 的大小排列， $sim(u, C_i)$ 的取值范围为 $[-1, 1]$ 。 $sim(u, C_i)$ 的值约接近于 1 说明用户之间相似性越高， $sim(u, C_i)$ 的值越接近于 -1 说明用户之间拥有越明显的相反兴趣爱好。 $sim(u, C_i)$ 为 0, 说明用户之间不具有相似性。^[5]

通过 MATLAB 进行聚点分类，下面是 K 聚类算法聚点前后对比图

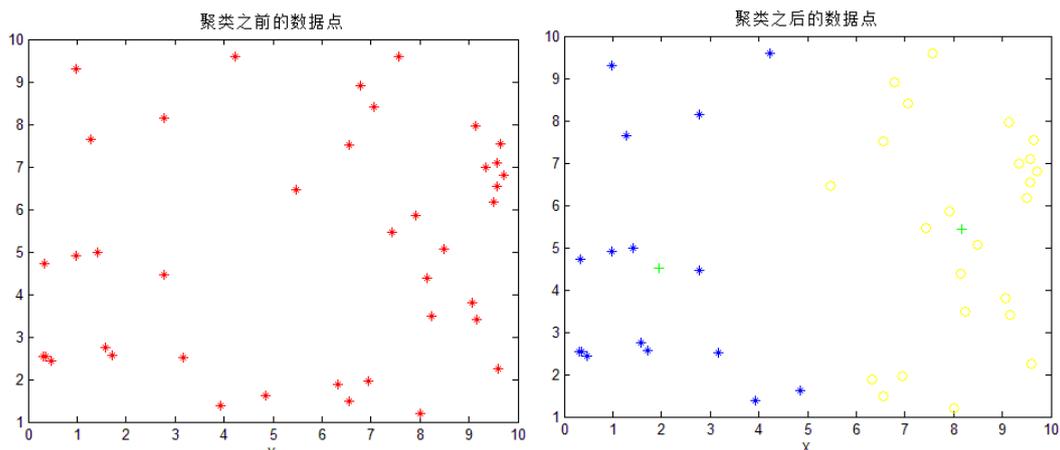


图 5.1.1 协同过滤的 K 聚类算法聚类前后示意图

5.1.2 推荐结果的输出

当目标用户的最近邻居集合产生以后，我们利用其计算两类结果：用户对任意项的兴趣度的预测值和基于该预测值的 Top-N 形式的推荐集。

①目标用户对任意项目评分的预测值。

我们采用的是加权平均策略来产生预测评分：目标用户 i 对项目 C 的预测评分公式如下：

设我们前面生成的用户 i 的最近邻居集合用 NN_i 表示，则用户 i 对项目 c 的预测评分 $P_{i,c}$ 可以通过用户对最近邻居集合 NN_i 中项目的评分得到，计算方法如下：

②Top-N 形式的推荐集。

$$P_{i,c} = \bar{R}_i + \frac{\sum_{j \in NN_i} sim(i, j)(R_{j,c} - \bar{R}_j)}{\sum_{j \in NN_i} |sim(i, j)|} \quad (6)$$

计算在最近邻居集中目标用户 i 对各个未评分项目的预测评分值，然后按照评分的大小进行排序，选择最高的前 N 个项目作为 Top-N 推荐。成为按用户的兴趣偏好大小进行排序的推荐列表。使之最终为用户产生一个推荐列表。

5.1.3 协同过滤模型的评估

在现实中，最近邻搜索的时间代价随着用户数量和项目数量的增加而急剧增加。因此当用户数量和项数量都数以万计时，协同过滤推荐算法需要同时为数以万计的用户同时提供实时的推荐服务，这时协同过滤推荐算法的伸缩能力和实时性要求难以保证。而且随着系统规模的扩大，用户数目和项目数目的指数级也随之增长，在这种情形下会导

致用户评分数据极端稀疏。在用户评分数据极端稀疏的情况下，协同过滤推荐算法无法对某些用户产生任何推荐。同时，推荐算法的推荐精度的显著下降。另外，目前大部分的推荐算法都只利用了一部分可用信息来产生推荐。然而随着研究的深入，推荐算法应该利用尽可能多的信息，收集多种类型的数据并有效集成，从而提供更加精确有效的推荐。推荐算法的推荐精度和伸缩能力是一对矛盾。此外还有冷启动问题等。^[6]

为了解决这一问题，我们利用蚁群算法具有灵活性高、鲁棒性强及分布式计算的特点，在传统协同过滤推荐技术基础上，采用蚁群聚类算法对用户进行分类，在整个用户空间高效的查找邻居用户群，结合用户行为模型和基于项目内容的推荐，建立了基于机器学习的蚁群聚类的协同过滤算法，实践证明，该算法很好的解决了上述问题。改进算法的设计流程图如图所示：

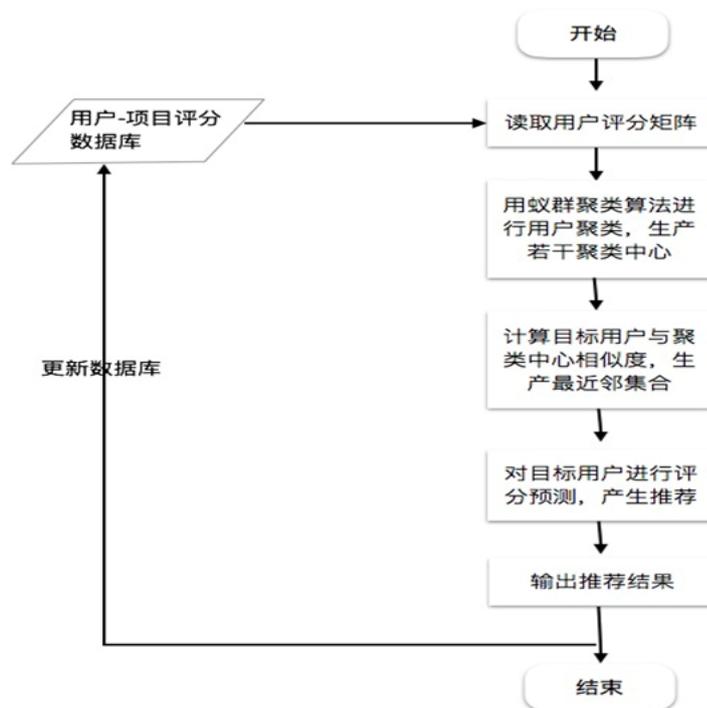


图 5.1.3 改进算法的设计流程图

六、文章推荐系统模型的改进以及推荐方案的形成

6.1 基于协同过滤与蚁群聚类的文章推荐模型的建立

基于蚁群聚类算法和协同聚类算法，我们将推荐过程分为离线和在线两个部分，首先在离线部分，系统将利用蚁群聚类算法对用户进行聚类分析，通过蚁群聚类算法产生若干具有不同特征的用户类别，从而大大缩减了寻找最近邻的范围，通过由具有喜爱程度较高的类别（美感类别）用户的加权评分产生一个代表该类别的虚拟用户，最后生成虚拟用户一项目评分矩阵；而在在线时，通过算法只需计算目标用户与虚拟用户一项目评分矩阵的相似性（相似度），进而获得与各个聚类中心的相似度，将这些相似度数据组成一个度量矩阵，然后通过该矩阵搜索目标用户的最近邻居集合，此时由于通过蚁群聚类算法使得虚拟用户的数目已经远远小于系统中实际用户的数目，所以在线时只需计算目标用户与少数聚类中心的相似性，从而将在线查找目标用户最近邻居集合的时间大大缩短，很好的解决了系统的可扩展性和实时性问题，最终提高系统对目标用户推荐的准确率和推荐质量。在最近邻居集合的基础上利用协同推荐系统预测出目标用户的评分，将这些评分按照由高到低的顺序进行排列，从而依据系统的要求为用户推荐最喜欢的项目或者输出推荐 Top-N 推荐列集。^[6]

首先，我们利用蚁群聚类算法对 100 位用户进行偏好数据的输出，输出情况如下：

用户偏好数据表									
0.0280	0.8312	0.5894	0.8814	0.9960	0.6683	0.3772	0.0295	0.0881	0.841
0.9601	0.3595	0.1531	0.8676	0.2002	0.2811	0.9245	0.5417	0.2981	0.652
0.4323	0.6930	0.9680	0.5731	0.8019	0.0048	0.6008	0.7995	0.2181	0.721
0.4904	0.5379	0.6446	0.1119	0.6753	0.5622	0.4400	0.3131	0.2607	0.324
0.0961	0.7649	0.8121	0.9689	0.6793	0.4583	0.0876	0.5108	0.7085	0.625
0.9437	0.3096	0.5793	0.9620	0.7695	0.2939	0.5697	0.1780	0.8875	0.346
0.0671	0.9389	0.0138	0.5034	0.8003	0.8636	0.5335	0.7382	0.8050	0.521
0.2579	0.4843	0.0685	0.3974	0.6130	0.1266	0.3547	0.9496	0.4114	0.625
0.4775	0.9136	0.5885	0.7969	0.2821	0.7381	0.2684	0.7756	0.9633	0.417
0.3467	0.4055	0.0850	0.6702	0.2481	0.1259	0.0408	0.5424	0.9017	0.569
0.1265	0.5562	0.1703	0.4222	0.1531	0.8180	0.4957	0.8825	0.2428	0.712

表 6.1 用户偏好数据表

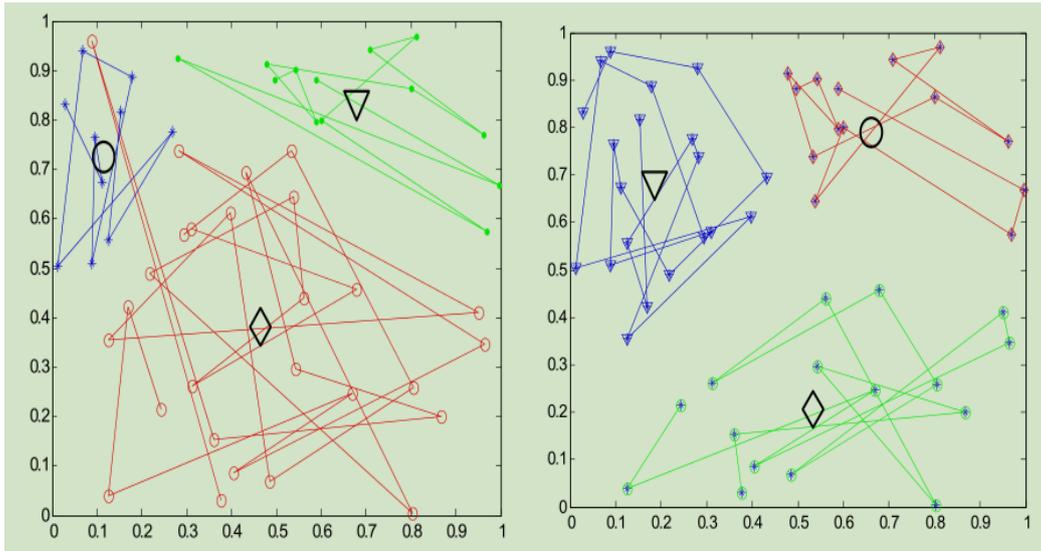


图 6.1.1 K 聚类算法对数据的聚类示意图

图 6.1.2 K 聚类算法对数据的聚类示意图

由图可以看出，蚁群聚类算法可以很好的解决 K 聚类算法存在的问题，大大提高了聚类效果。使得在整个用户空间上可以高效的查找邻居用户群，从而提高了系统对目标用户推荐的准确率和推荐质量。

下面我们给出基于蚁群算法的协同过滤文章推荐算法的流程图：

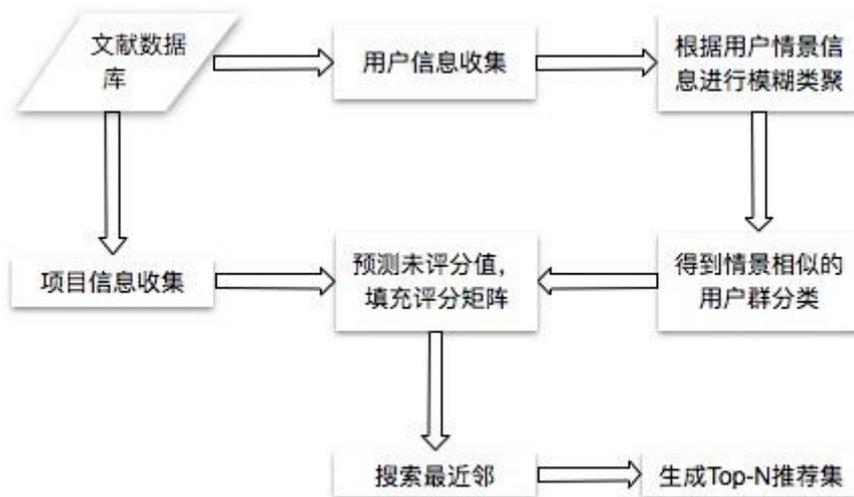


图 6.1.3^[6] 蚁群算法生成 Top-N 推荐集过程

6.2 基于文章推荐模型的推荐方案的形成

文章推荐系统，是根据用户平时的浏览行为，形成记忆，并且运用算法，提取出用户喜欢的文章的要素，然后对大量的文章进行筛选，最终选取一定数量的文章进行推荐。

对于文章推荐系统的建立与形成，思路结构图如下：

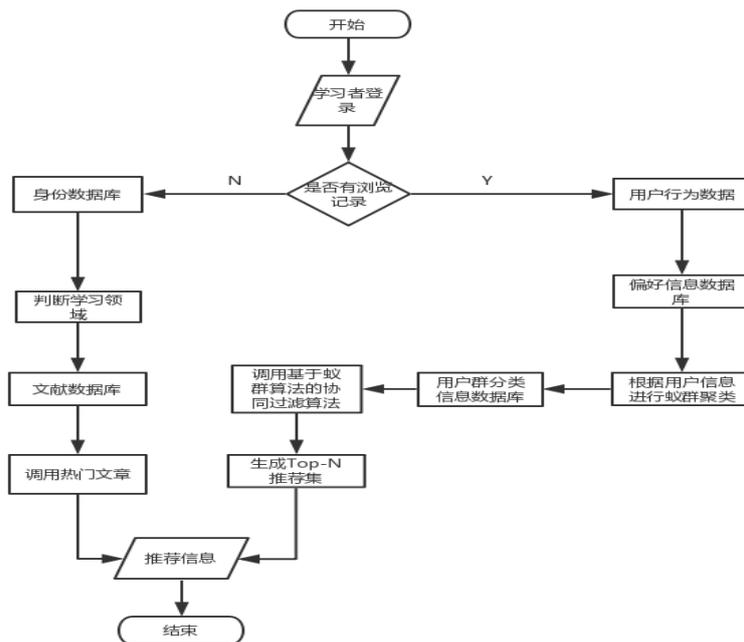


图 6.2 思路结构图

我们建立的文章推荐模型，基本的思路为：在用户进入之后，先对用户身份进行验证，若为老用户，系统自动检索系统中对该用户最常浏览的文件类型的记忆，然后，再借助蚁群聚类方法，在文库中找出 10 个相似的文章进行推荐；若是新用户，就应用文件下载次数检索算法，将文库中下载次数最多的 10 篇文章推送给新用户，当新用户在此进入时，系统中已经有了浏览记录，就可以将用户转变成老用户，对其推荐相关文章。其中，最核心的是算法的理论建立和编程实现。

七、蚁群聚类算法的文章推荐系统模型的求解及其算法的实现

7.1. 算法实现

7.1.1 蚁群算法的实现

我们借鉴蚁群算法的原理，将用户数据视为具有不同属性的蚂蚁，用户聚类中心看作是蚂蚁所要找的“食物源”，所以用户聚类过程就看作是蚂蚁要找“食物源”的过程。

我们将准备聚类的 m 个项目评分数据作为聚类对象随机分布在二维网格中，将 P 只蚂蚁随机的放置在二维网格中，由于二维网格代表了我们所确定的模式样本，因此也就是相当于在每个模式样本处分别放置 1 个蚂蚁，在我们的模式样本中（二维网格中）随机选取一只蚂蚁，并随机选取一个方向移动这只蚂蚁；蚂蚁倾向于选择信息素

最多的一条路径，也就是距离最近的一个模式样本。

定义网格中 P 单元格的邻域是以 p_i 为中心，边长为 B 个单位的正方形区域，记为 $S(p_i)$ 。若 p_i 处有对象 O_i ，则 O_i 的邻域为 $S(O_i)$ 。每个蚂蚁都有一个长度为 L 的短期记忆，根据蚂蚁本身拥有的短期记忆 L，让蚂蚁朝着匹配位置进行随机移动。否则随机移动到其他的数据点上。单个蚂蚁如果找到一个数据对象 O_i ，计算该数据对象的领域半径 r 与其他数据对象的相似度 $f(O_i)$ 。^[2]

$$f(o_i) = \begin{cases} \frac{1}{S^2 * o_j \in S(o_j)} \left| 1 - \frac{d(o_i - o_j)}{\partial} \right| f(o_i) > 0 & S \neq 0 \\ 0 & \end{cases} \quad (7)$$

其中， $d(o_i, o_j)$ 为欧氏距离，

$$d(o_i, o_j) = \sqrt{\sum_{r=1}^m (o_i - o_j)^2} \quad r = \frac{S-1}{2} \quad (8)$$

通过以 $f(O_i)$ 获得蚂蚁捡起 O_i 的概率 P_p ，

$$P_p = \left(\frac{k_1}{k_1 + f(o_i)} \right)^2 \quad (9)$$

其中， k_1 是处于 (0, 1) 之间的一个计算该对象与周围其他对象的相似性 $f(O_i)$ 和蚂蚁对该对象的拾取概率 P_p ，并生成一个随机概率 P_r ，且 $P_r \in [0, 1]$ ；，如果 $P_p(o_i)$ 大于等于某个随机产生的概率 P_r ，该蚂蚁将捡起聚类对象 O_i ，并标识该蚂蚁为负载状态，标识聚类对象 O_i 为被拾起状态，同时也被标识为曾被拾起状态。否则考察该蚂蚁对邻域内其他未曾被拾起的聚类对象的拾起概率 $P_p(o_i)$ ，重复上述操作，直到蚂蚁为负载状态或移动到下一位置。同时对于每一只有负载的蚂蚁，计算该负载的聚类对象在蚂蚁所在邻

$$P_d(o_i) = \begin{cases} 2^{f(o_i)k} & f(o_i) < k_1 \\ 1 & f(o_i) < k_1 \end{cases} \quad (10)$$

域内的平均相似度 $f(O_i)$ 以及放下概率 $P_d(o_i)$ ，

若放下概率 $P_d(o_i)$ 大于随机概率 P_r ， $P_r \in [0, 1]$ ，则将聚类对象放于蚂蚁当前所在的单元格内，并标识蚂蚁为未负载状态，标识聚类对象为未拾起状态。否则蚂蚁负载该聚类对象继续移动，蚁将数据点放下的同时，在自己的短期记忆中记录放下数据点的位置。^[4]

随机分布在二维网格的聚类对象通过蚂蚁的不断拾起与放下逐渐聚集，通过计算拾起对象与其领域内对象的相似度，并与随机概率比较决定下一步操作。如果拾起或放下概率高于随机概率，蚂蚁则会将聚类对象执行拾起或放下操作，若低于随机概率则继

续移动。经过蚁群的移动、拾起和放下不断迭代操作，聚类对象最终被聚为若干类簇，聚类结束输出聚类结果。因此，蚁群在经过若干次的移动、拾起与放下动作后将会实现对所有对象的聚类，进而最终输出聚类结果。

下面是蚁群算法实现流程图

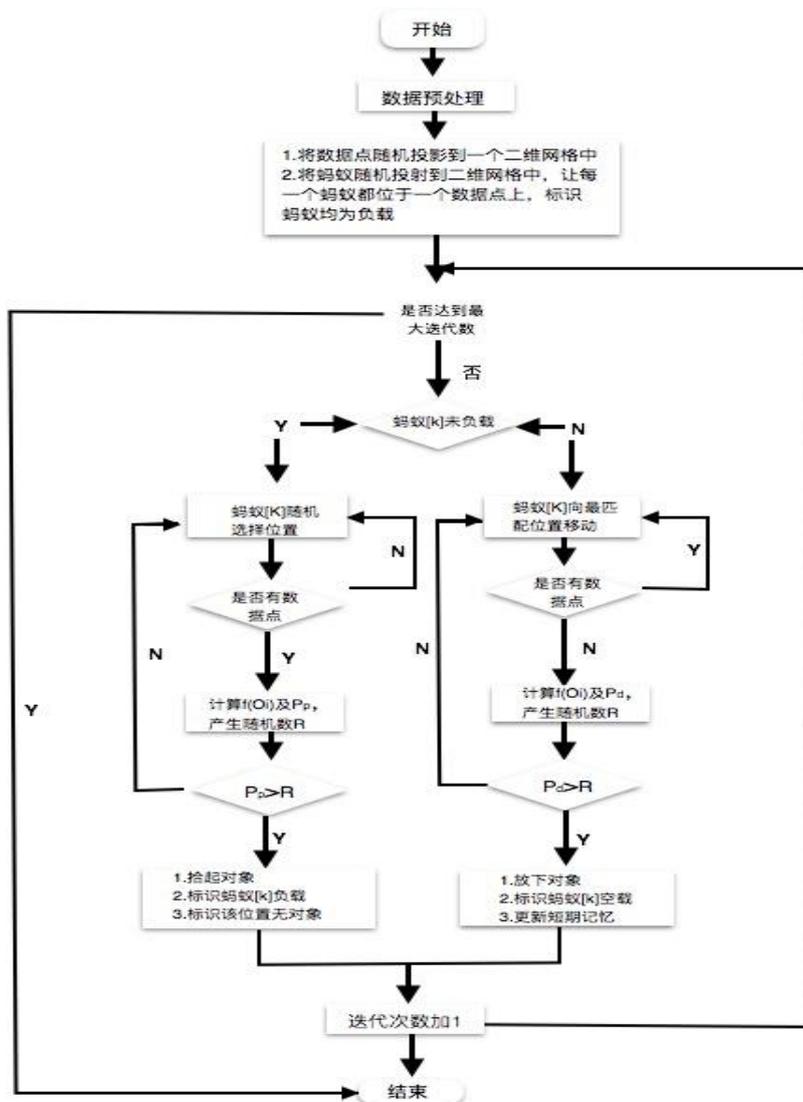


图 7.1.1 蚁群算法实现流程图

7.1.2 身份辨识算法（JSP）

设计一个窗口，输入身份代码（如学号等），然后对其进行辨别，如果是老用户，调用蚁群聚类算法；如果是新用户，调用下载次数检索算法。

附录给出了算法的实现代码

最后利用 java 进行算法的实现，最后输出的窗口为：

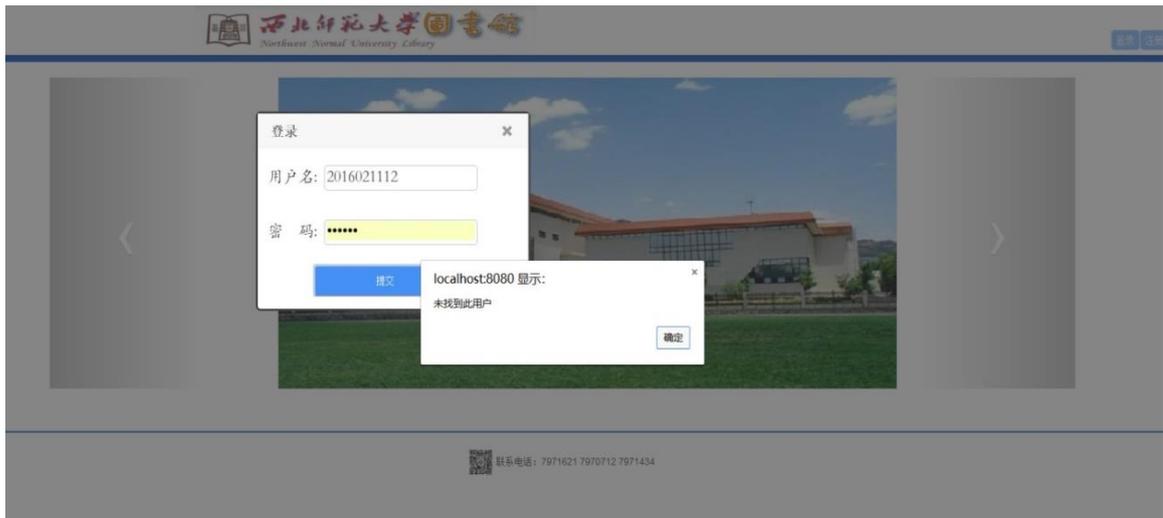


图 7.1.2 身份辨别算法实现后窗口示意图



添加学生信息

学生学号:

上传证件照: 未选择任何文件

学生学号:

学生姓名:

性别: 男 女

专业名称:

学院名称:

班级名称:

高中就读学校:

入学日期:

毕业日期:

证件类型:

身份证号:

出生日期:

民族:

家庭地址:

联系电话:

图 7.1.3 用户注册界面



图 7.1.4 推荐系统登录窗口示意图



图 7.1.5 推荐系统登出窗口示意图

7.1.3 蚁群聚类算法 (MATLAB)

对于蚁群聚类后的数据，我们利用 MATLAB 进行分类，分类后的输出示意图如下：

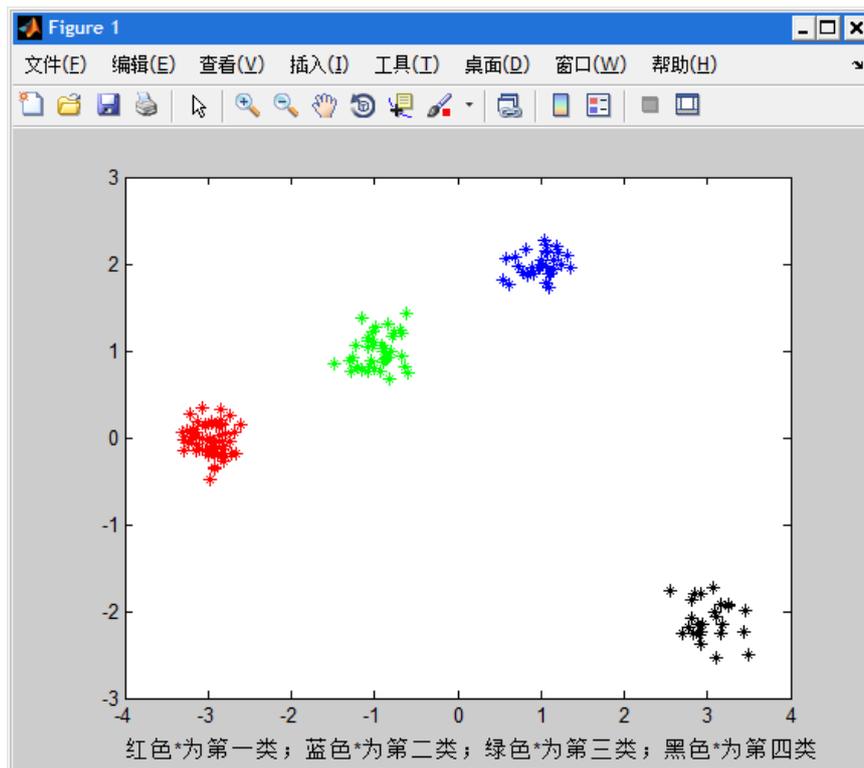


图 7.1.3 蚁群算法聚类前后示意图

7.1.4 新用户信息检索算法 (SQL)

对于新用户，我们首先将身份信息在信息数据库中搜寻，找到相关的信息，将其身份信息，传送到文件下载次数检索算法中。这里用到的是 SQL 语句，大致的过程为：

建立信息数据库——根据输入身份信息，在数据库中找到相关信息。

下面以学生信息数据库 Message(学号、姓名、学院、专业、性别、年龄)为例，给出具体程序：

```
SELECT 专业
FROM Message
WHERE 学号='*****'
```

7.1.5 文件下载次数检索算法 (JSP)

对于文库中的文章，我们首先使用聚类算法，将相似的文章归类，再采用 JSP 对文件下载次数统计算法，对文库中的相关文章进行检测，然后按照检测的结果给出下载次数最多的 10 篇文章。

附录给出了主要的程序——JSP 对下载文件的次数统计算法^[7]

7.2 基于蚁群算法的文章推荐系统模型求解

基于蚁群算法的实现，我们对我们的模型进行求解，一方面，针对多次进入系统的用户（老用户），我们初步根据用户对文章的点击量和下载量，利用协同过滤方法，在当前文献数据库基础上，利用 k 近邻聚类算法给出与用户浏览文章相似度较高的一些文章，并对其进行排序，从中挑选出相似度较高的 10 篇文章，在用户下次进入系统时，推荐给他们。但是一方面考虑到 k 近邻聚类算法本身的局部最优解问题会降低系统推荐的准确性，另一方面考虑协同聚类方法对稀疏性问题、冷启动问题、扩展性等问题的束手无策性，我们在传统协同过滤推荐技术基础上，采用蚁群聚类算法代替协同过滤算法中的 k 近邻聚类对用户进行分类，在整个用户空间高效的查找邻居用户群，从而提高了系统对目标用户推荐的准确率和推荐质量，另一方面，考虑到第一次进入系统的用户没有行为表现，我们也通过对用户身份的辨别，在模型的基础上给出了一个文件下载次数检索的算法，基于该算法给出了适用于新用户的推荐方法。并根据模型，给出了基于协同过滤与蚁群聚类的文章推荐系统，从而给出了问题四的推荐方案。

我们随机选取了西北师范大学图书馆文献数据库的 100 篇文章，通过用户日志统计了用户对这 100 篇文章的每篇文献的关注数量和共同数量，并通过蚁群聚类算法对其进行聚类，对聚类后的 100 篇文章进行了相似度计算。

计算结果见附录图 7.2 蚁群算法聚类前后示意图

7.3 基于蚁群算法的文章推荐方案的形成

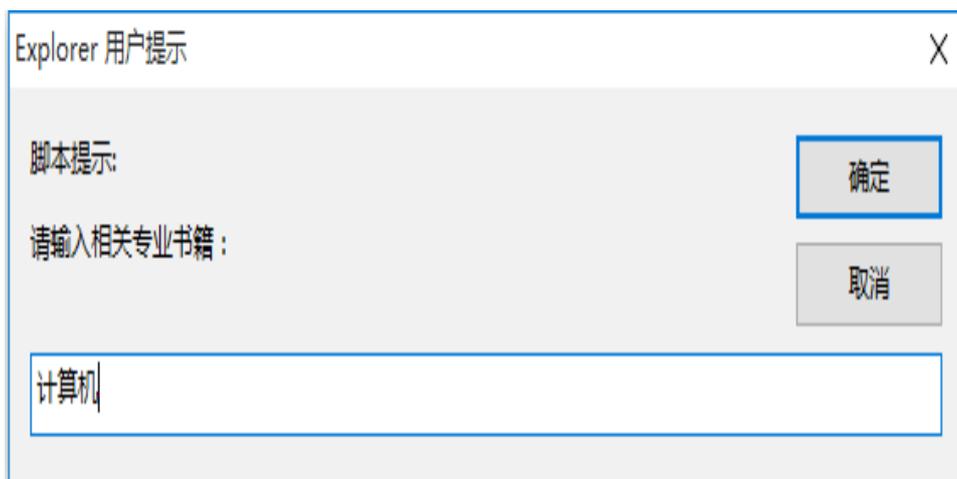
一方面，针对多次进入系统的用户（老用户），我们初步根据用户对文章的点击量和下载量，利用协同过滤方法，在当前文献数据库基础上，利用 k 近邻聚类算法给出与用户浏览文章相似度较高的一些文章，并对其进行排序，从中挑选出相似度较高的 10 篇文章，在用户下次进入系统时，推荐给他们。但是考虑到 k 近邻聚类算法本身的局部最优解问题会降低系统推荐的准确性，同时考虑到传统协同聚类方法对稀疏性问题、冷启动问题、扩展性等问题的束手无策性，我们在传统协同过滤推荐技术基础上，采用蚁群聚类算法代替协同过滤算法中的 k 近邻聚类对用户进行分类，在整个用户空间高效的查找邻居用户群，从而提高了系统对目标用户推荐的准确率和推荐质量。另一方面，考虑到第一次进入系统的用户没有行为表现，我们也通过对用户身份的辨别，在模型的基础上给出了一个文件下载次数检索的算法，基于该算法给出了适用于新用户的推荐方法。最后，我们对模型进行了实际性检验，检验结果如下：

对于新用户，我们以汉语言文学专业的学生为例，通过身份验证后用搜索算法搜索出本专业学生最近阅读量较高的书籍，对其进行自动推荐，实际推荐界面如下：



图 7.1.3 新用户推荐系统推荐结果实现后窗口示意图

对于老用户，我们以计算机专业的学生为例，通过其字词或者标题的输入，系统利用我们所建立的算法，根据用户的爱好兴趣进行自动推荐。推荐界面如下：



以下可能是你喜欢的文献

《计算机网络》
《C语言程序设计》
《Java核心编程》
《大数据》
《path》
《PS教程》
《计算机组成原理》
《网页设计》
《数据结构》
《数据结构与算法》

图 7.1.5 老用户推荐系统推荐结果实现后窗口示意图

八、模型的检验

8.1 平均准确度 (MAE) 检验

我们采用基于 K 近邻聚类协同过滤算法模型 (KNN-CFR) 与基于蚁群聚类的协同过滤算法模型 (ACO-CFR) 按照聚类中心 K 从 5~100 之间 5 的倍数进行了 300 组试验, 选取其中具有代表性的 5 组数据进行分析, 计算出其平均准确度, 实验参数及结果如表 8.1 与图 8.2 所示

实验次数	K=50, i=50		K=50, i=100		K=100, i=100	
	K 聚类	蚁群聚类	K 聚类	蚁群聚类	K 聚类	蚁群聚类
1	0.957	0.954	0.955	0.991	0.966	0.969
2	0.974	0.962	0.974	0.951	0.942	0.979
3	0.965	0.946	0.968	0.991	0.971	0.969
4	0.982	0.962	0.984	0.976	0.955	0.959
5	0.966	0.941	0.966	0.964	0.968	0.971

表 8.1 不同聚类中心 K 值和文献数 i 下, 平均准确度 (MAE) 的值

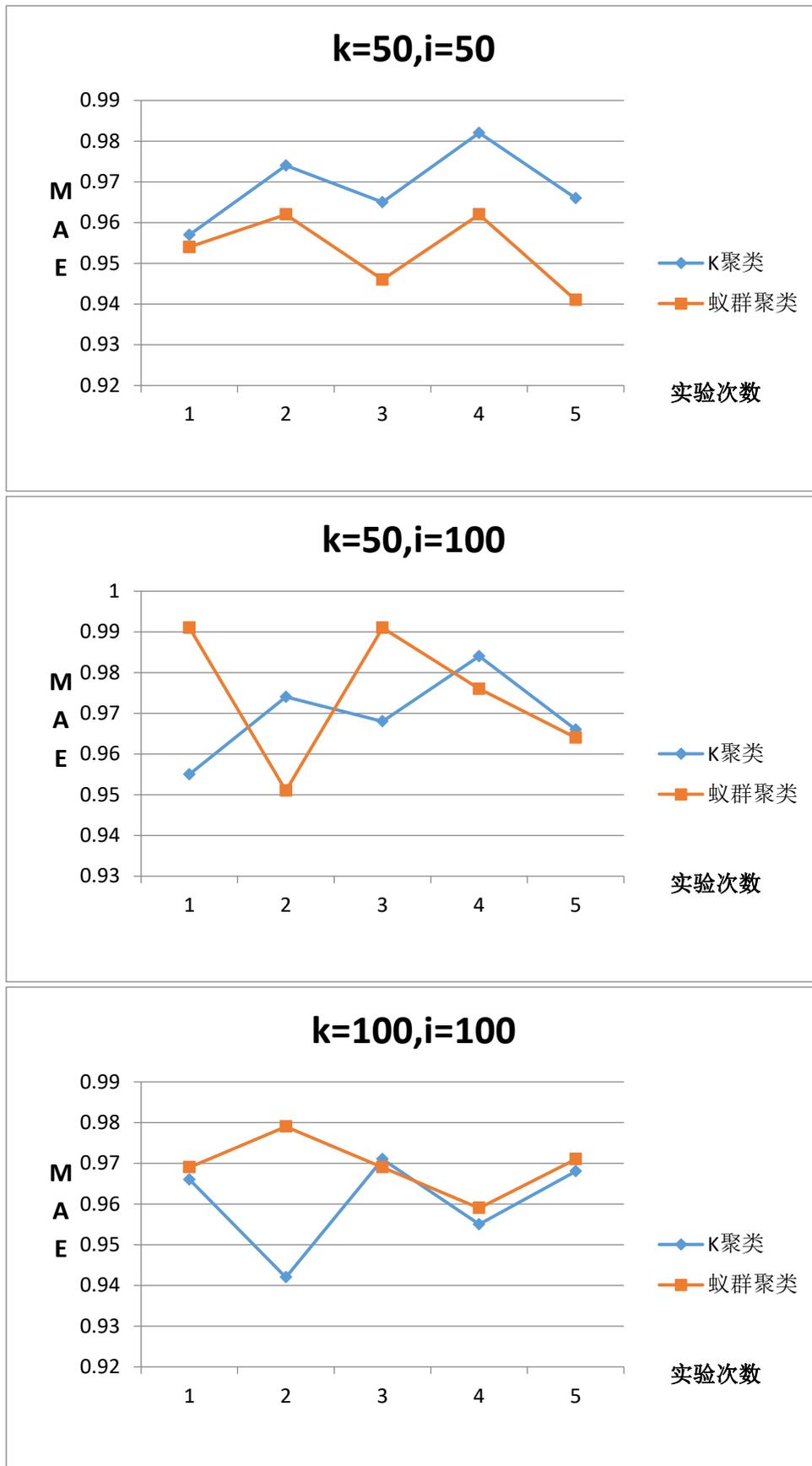


图 8.2 不同聚类中心 K 值和文献数 i 下，平均准确度 (MAE) 值的折线图

由表格数据和折线图分析可知，在样本量（文献数）较少和聚类中心较低的情况下，基于K近邻聚类协同过滤算法的文章推荐模型的推荐平均准确度要高于基于蚁群聚类的协同过滤算法的文章推荐模型，然而随着样本量（文献数）和聚类中心的增大，我们发现，基于蚁群聚类的协同过滤算法的文章推荐模型的推荐平均准确度明显高于基于K近邻聚类协同过滤算法的文章推荐模型，而且，我们从中可以看出K近邻聚类协同过滤算法对于聚类的稀疏性，而蚁群聚类的协同过滤算法很好的解决了该问题。

8.2 推荐过程耗时性检验

我们采用基于K近邻聚类协同过滤算法模型（KNN-CFR）与基于蚁群聚类的协同过滤算法模型（ACO-CFR）按照聚类中心K从5~100之间5的倍数进行了300组试验，选取其中具有代表性的5组数据进行分析，计算出其推荐耗时，实验参数及结果如表8.1与图8.2所示

实验次数	K=50, i=50		K=50, i=100		K=100, i=100	
	K 聚类	蚁群聚类	K 聚类	蚁群聚类	K 聚类	蚁群聚类
1	66.54	62.12	76.14	42.36	72.58	41.58
2	69.25	54.31	62.42	55.51	68.64	45.25
3	68.37	71.06	69.21	56.45	77.48	55.21
4	49.55	42.36	64.25	61.21	79.24	46.29
5	72.02	52.18	71.54	58.54	74.53	50.06

表 8.1 不同聚类中心 K 值和文献数 i 下，推荐耗时的值（单位：秒）

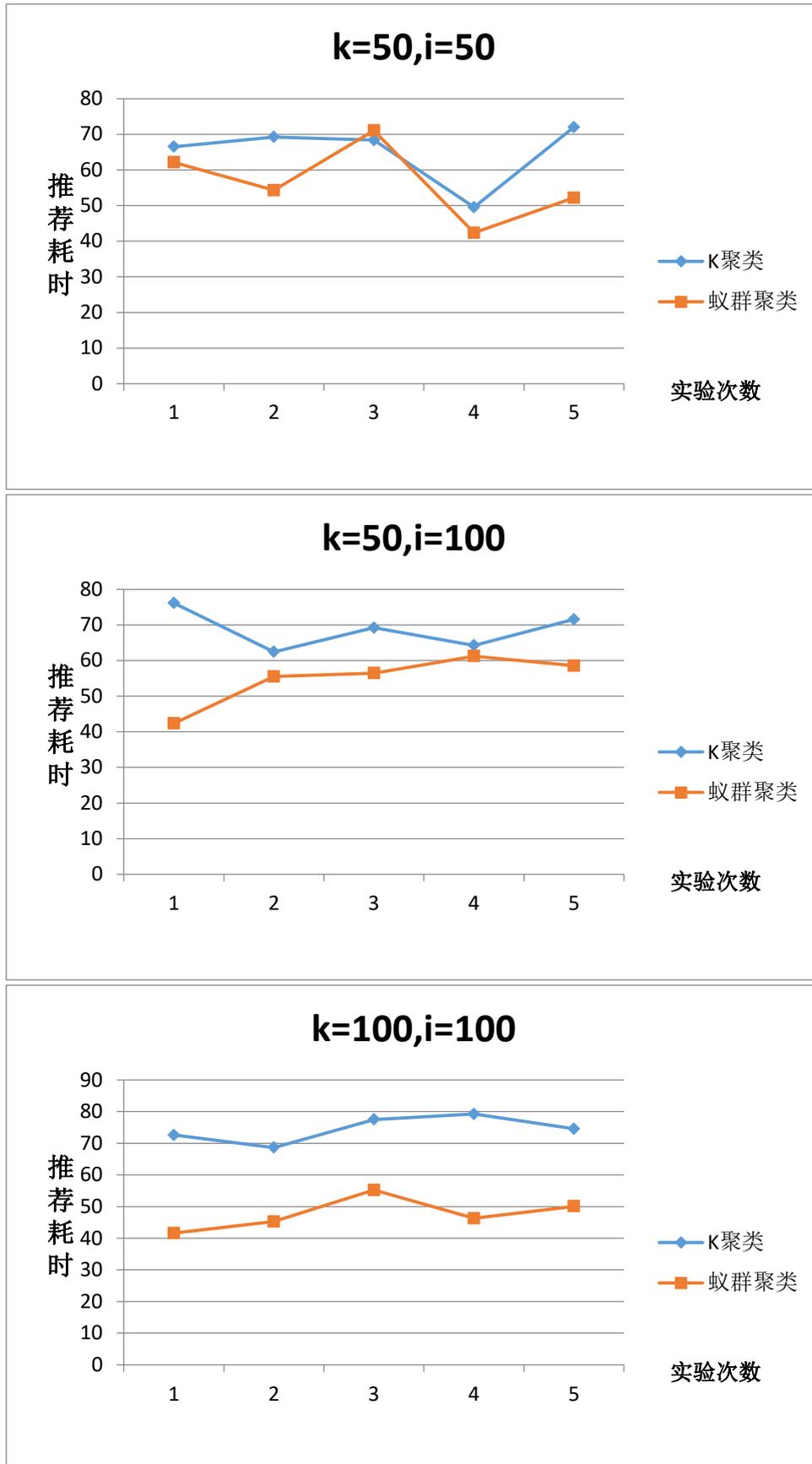
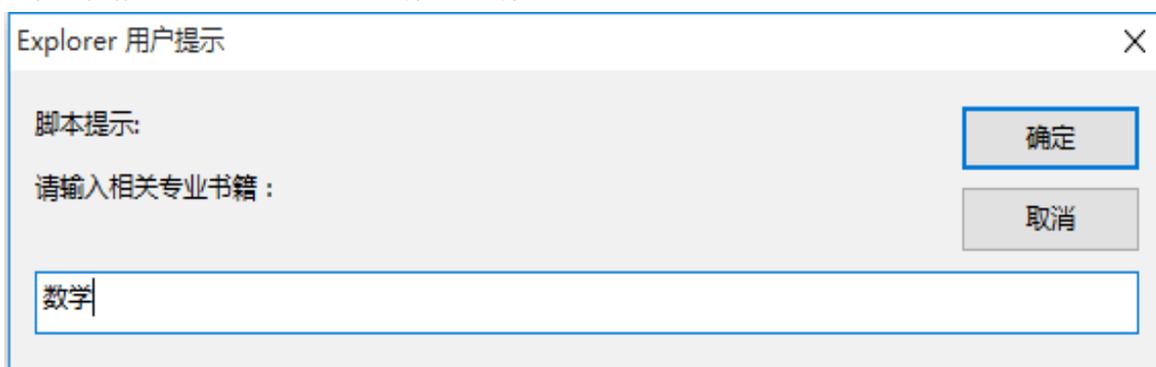


图 8.2 不同聚类中心 K 值和文献数 i 下，推荐耗时的值的折线图

由表格数据和折线图分析可知，无论是在样本量（文献数）较少和聚类中心较低的情况下，还是在样本量（文献数）和聚类中心不断增大的情况下，基于 K 近邻聚类协同过滤算法的文章推荐模型的推荐耗时都要高于基于蚁群聚类的协同过滤算法的文章推荐模型，而且，我们从中可以看出 K 近邻聚类协同过滤算法对于聚类的鲁棒性和冷启动问题，而蚁群聚类的协同过滤算法很好的解决了该问题。

8.3 实际推荐结果检验

我们以数学专业的学生为例，通过对其身份进行辨别，区分出新老用户，对于老用户，通过字词或者标题的输入，系统利用我们所建立的算法，根据用户的爱好兴趣自动进行推荐。对于新用户，我们通过身份验证后用搜索算法搜索出本专业学生最近阅读量较高的书籍，对其进行自动推荐，推荐界面如下：



以下可能是你喜欢的文献

《解析几何》
《数学分析》
《常微分方程》
《高等代数》
《概率论与数理统计》
《复变函数》
《高等数学》
《线性代数》
《概率统计》
《数据结构与算法》

图 8.3 实际检验结果图

九、模型的评价

9.1 模型的优点

1. 可以针对不同的用户，推荐不同的文章。
2. 对于经常进入系统的用户，能够记录其行为，通过蚁群聚类方法，自动分析其行为爱好，推荐符合用户爱好的文章。

3. 利用蚁群聚类能够有效地降低推荐算法的计算量，有效的自发聚类提高了系统的伸缩性和鲁棒性；

4. 基于蚁群算法实现用户进行聚类，提高了协同过滤推荐系统的最近邻查询速度，降低了搜索开销，同时避免了使用 k 近邻聚类方法受初始聚类中心和聚类个数的影响。

5. 解决了（冷启动问题）新用户得不到推荐的问题，并提高了协同过滤推荐算法的精确。

6. 模型将蚁群聚类算法与协同过滤技术相结合有效的避免了 K 聚类算法带来的局部最优解问题，在一定程度上提高了系统推荐的准确性。

9.2 模型的缺点

1. 算法我们用了不同的编程软件，相互调用时可能不相容；

2. 对于检索文档的算法过于庞大，如果调用一次，如果需要 2ms，若是文库里有十万篇文章，则大约需要 17 分钟，若是更多，则需要更多，耗时较大。

3. 目前世界对蚁群聚类算法的理论研究还不足，这导致我们的模型还有待改进。

所以，在后续模型改进中，我们主要的方向就是算法的改进。用同样的设计软件得出一个整体的推荐系统，并且在短时间内进行推荐。

十、推介信

尊敬的领导：

您好！

我们是 xxxx 大学的学生，针对教育信息系统推荐问题，我们运用数学建模的方法，针对文章推荐，我们建立了一种推荐方案，希望对你们的工作有所帮助。

我们的方案具体实现过程是：

1、设计进入文库的窗口，对用户进行身份验证；

2、如果是系统中有浏览记录，认定为老用户，根据记录，运用协同算法筛选文库，对其推荐文章；

3、系统中没有浏览记录，认定其是新用户，搜索身份数据库，找出相应的个人信息，对其进行推荐相关的热门文章。

其中，我们算法最大的亮点在于，运用协同过滤的算法之前，用蚁群聚类的方法代替了 k 近邻聚类，这样的方法优势在于：

①利用蚁群聚类能够有效地降低推荐算法的计算量，有效的自发聚类提高了系统的

伸缩性和鲁棒性;

②基于蚁群算法实现用户进行聚类,提高了协同过滤推荐系统的最近邻查询速度,降低了搜索开销,同时避免了使用k近邻聚类方法受初始聚类中心和聚类个数的影响。

③解决了(冷启动问题)新用户得不到推荐的问题,并提高了协同过滤推荐算法的精确。

并且在我们的论文中,还给出了推荐系统实现的每一步所需要的程序实现,尽管我们的程序存在一定的缺陷,但我相信,在后续的努力中,一定可以改进这些缺点,将我们的系统进一步完善。

虽然我们不知道,对于信息推荐模型,有许多学生都提出了自己的想法,但是针对他们的方案,我们的方案优势就在于,根据我们的模型,不仅能根据用户的浏览记录进行推荐,还能够根据用户的身份,判断其是否是老用户,针对新老用户的不同然后推荐不同的文章。

感谢您在百忙之中阅读我们的信件。

XXXXXX

XXXX年xx月xx日

十一、参考文献

- [1]朱郁筱,吕琳媛. 推荐系统评价指标综述[J]. 电子科技大学学报. 2012. 3(2): 163-171
- [2]孔明. ROC 和 ACU 介绍及如何计算 AUC[OL]. www.douban.com/note/284051363/. 2017. 6. 3
- [3]杨昭. 基于用户行为模型和蚁群聚类的协同过滤推荐算法[D]. 复旦大学, 2013
- [4]李华,张宇,孙俊华. 基于用户模糊聚类的协同过滤推荐研究[J]. 计算机科学, 2012, 39(12):83-86
- [5]刘伟. 基于蚁群用户聚类的协同过滤推荐系统研究[D]. 南开大学, 2013
- [6]曹毅. 基于内容和协同过滤的混合模式推荐技术研究[D]. 中南大学, 2007
- [7]JSP 统计下载次数[OL]. <https://zhidao.baidu.com/question/262477414.html>. 2017. 6. 3
- [8]任帅,王浙明,王明敏. 基于用户行为模型和蚁群聚类的协同过滤推荐算法[J]. 微型电脑应用, 2014, 30(3):5-8

[9] 基本蚁群聚类算法及其改进算法[OL].

<http://www.ilovematlab.cn/thread-90764-1-1.html>. 2017. 6. 4

[10] 协同过滤聚类算法

<http://www.cnblogs.com/technology/p/4467895.html>. 2017. 6. 4

[11] 西北师范大学图片

http://image.baidu.com/search/index?tn=baiduimage&ct=201326592&lm=-1&cl=2&ie=gbk&word=%CE%F7%B1%B1%CA%A6%B4%F3%D5%D5%C6%AC&hs=2&xthttps=000000&fr=ala&ori_query=%E8%A5%BF%E5%8C%97%E5%B8%88%E5%A4%A7%E7%85%A7%E7%89%87&ala=0&alatpl=sp&pos=0. 2017. 6. 4
