**摘要**

随着社会的发展，推荐系统正在社会中变得越来越重要，这里设计了一个多引擎的推荐系统，并就不同引擎的效果和效率，以及多引擎融合推荐的效果和效率做了分析

（中文摘要是论文内容的总结概括，应简要说明论文的主要内容和观点，一般在300以内。摘要内容的字体为宋体小四号，1.5倍行间距。）

**关键词：**推荐系统，协同过滤，多引擎，×××

（另起一行，顶格写。3～5个关键词，不同关键词之间用逗号分隔。）

**Abstract**

××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××（英文摘要应与中文摘要内容相对应。英文摘要内容的字体、字号为Times New Roman小四号，1.5倍行间距，）

**Key words**: ×××××××××××××××（另起一行，顶格写。不同关键词之间用逗号空格分隔。）

**目录（示例）**

**第一部分毕业设计**

[**1.** **引言 1**](#_Toc479953242)

[1.1 研究背景与意义 1](#_Toc479953243)

[1.2 国内外研究现状 1](#_Toc479953244)

[1.3 主要设计任务及预期目标（指标） 1](#_Toc479953245)

[**2.** **系统设计方案 1**](#_Toc479953246)

[2.1总体设计方案 1](#_Toc479953247)

[2.2关键问题解决 1](#_Toc479953248)

[2.3设计结果呈现 1](#_Toc479953249)

[**3.** **系统测试、验证和结果分析 1**](#_Toc479953250)

[3.1测试、验证方法 1](#_Toc479953251)

[3.2测试、验证结果 1](#_Toc479953252)

[3.3测试、验证结果的分析 2](#_Toc479953253)

[**4.** **总结 2**](#_Toc479953254)

[4.1本设计的特色与总结 2](#_Toc479953255)

[4.2设计中出现的问题及思考 2](#_Toc479953256)

[4.3展望 2](#_Toc479953257)

[**参考文献 2**](#_Toc479953258)

[**附件 4**](#_Toc479953259)

[**作者简历 5**](#_Toc479953259)

[**《浙江大学本科生毕业设计任务书》 （可不编页码）**](#_Toc479953259)

[**《浙江大学本科生毕业设计考核表》 （可不编页码）**](#_Toc479953259)

**第二部分 文献综述和开题报告**

[**文献综述和开题报告封面 （可不编页码）**](#_Toc479953259)

[**指导老师对文献综述和开题报告具体内容要求 （可不编页码）**](#_Toc479953259)

[**目录 （可不编页码）**](#_Toc479953242)

[一、文献综述 1](#_Toc479953242)

[二、开题报告 3](#_Toc479953242)

[三、外文翻译 5](#_Toc479953242)

[四、外文原文 （可不编页码）](#_Toc479953242)

[**《浙江大学本科生文献综述和开题报告考核表》 （可不编页码）**](#_Toc479953259)

1. **引言**
   1. **研究背景与意义**
      1. **研究背景**

在现今知识过载这样的一个时代背景下，每个人都被淹没在信息的海洋里，如何为用户找到其感兴趣的内容，对用户进行个性化的推荐，已经成了一个重要的课题。

推荐系统作为一种有效的信息过滤手段，是当前解决信息过载问题及实现个性化信息服务的有效方法之一。尤其是在网络和电子商务环境下，得到了广泛的应用，带动了一大批中小型公司的快速成长：无论是淘宝、亚马逊的商品推荐，还是今日头条等新型内容网站的新闻推荐，都因为推荐系统得快速成熟而受益，从之前的千人一面做到了千人千面，极大地提高了点击率和转化率；

同时企业通过推荐系统，还可以有效保留用户，重建客户关系，提高用户的忠诚度，甚至可以根据企业需求，从企业的众多客户中识别最有价值和最有潜力的客户，向企业推荐客户。

推荐系统作为一门充满商业价值的学科，从诞生以来就得到了飞速的发展。下面简单讲解一下推荐系统研究的阶段[1]：

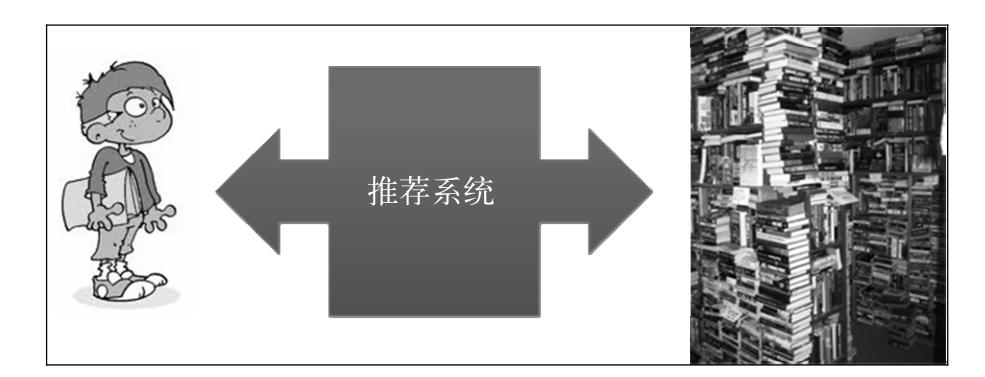
第一阶段：推荐系统形成的初期阶段。该阶段是面向系统的探索阶段，还没有出现大规模的商业化应用，但是基础理论已经提出。协同过滤、自动推荐、推荐系统等名词与想法在该阶段开始出现。

第二阶段：推荐系统商业应用的出现。推荐系统首先应用在了电子商务领域，其中最有名的是Amazon的电子商务推荐系统，其大大提高了Amazon的营业额——据统计，推荐系统的贡献率在20%~30%之间。

第三阶段：研究大爆发，新型算法不断涌现阶段。来自数据挖掘、人工智能、信息检索、安全与隐私以及商业与营销等各个领域的研究，都为推荐系统提供了新的分析和方法。同时在不同的应用场景下，基于上下文的推荐、基于社交网络的推荐、基于人口统计学数据的推荐、基于关联的推荐等新场景下的推荐系统开始不停地涌现；在应用领域，电子商务，在线教育[9,10]，商业旅游等都有所涉及；至于到今天面对着海量的数据，同样出现了大数据环境下的推荐系统，其靠着分布式的思想与技术来处理着海量的数据，提供了更加快速与准确的推荐。

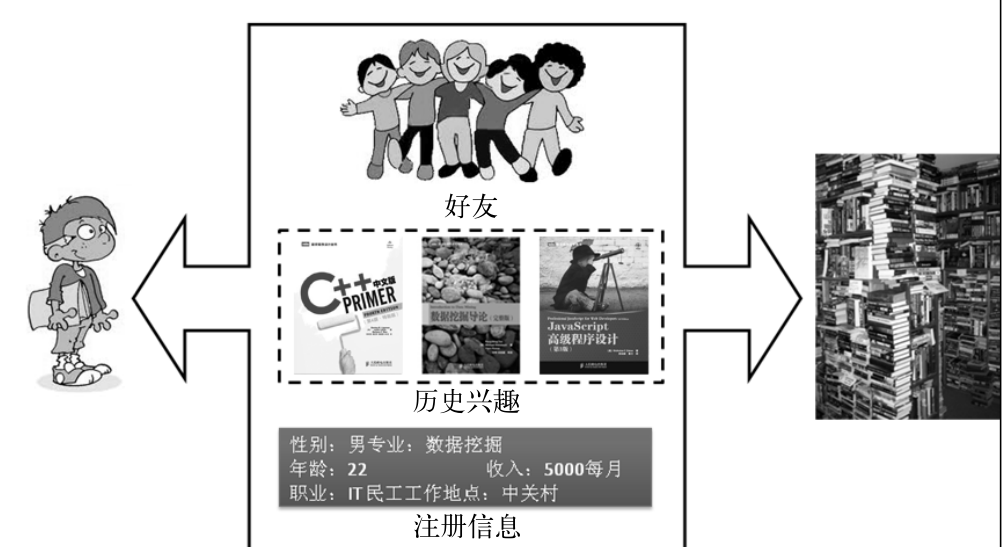
随着时代不断向前，推荐系统在越来越多领域大放光彩，如Amazon的个性化产品推荐、Netflix的视频推荐、Pandora的音乐推荐、Facebook的好友推荐和Google Reader的个性化阅读等[2]。但是面对着新的应用场景与海量的数据，以及一些仍然没有得到完全解决的历史问题，推荐系统仍然有许多困难等待着有志之人来挑战。

* + 1. **研究意义**

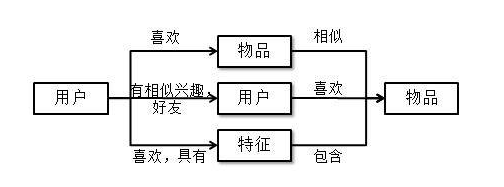


**图1.1 推荐系统的作用**

如图1.1所示，推荐系统的功能就是联系用户和物品（此处是广义的物品，也可以是其他用户），推荐系统的不同就是关联用户和物品的方法不同，如图1.2所示，关联方法可以是通过用户的好友，通过用户的历史兴趣、注册信息等。

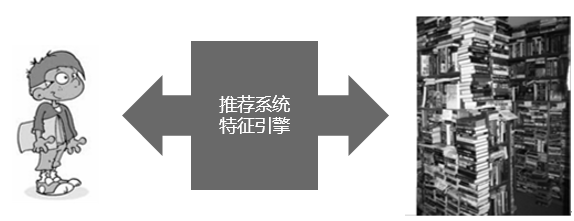


**图1.2 不同推荐方式的推荐系统**

****

**图1.3 推荐系统的实质是基于特征的关联**

图1.2还可以进一步简化成图1.3所示的样子，从中我们可以更明显的看出，不同的推荐方法之间的不同。我们可以把关联的方法抽象成一个个不同的特征引擎，而不同的推荐系统就是通过不同的特征引擎进行推荐。



**图1.4 推荐系统的实质是基于特征的关联**

目前对于推荐系统的研究，大多数都是基于对某一种算法的研究，而在实际生产使用中用到的也多是其中的一两种推荐方式，而在很多的推荐系统算法大赛上（如Netflix Prize），获得优胜的推荐系统往往融合了多个系统的推荐结果，才能最终拿的大奖。

同时，面对不同的应用场景，不同的用户来源，以及不同的物品分类，其适合的推荐系统都是不一样的：同一个应用的两个不同的用户可能一个比较适合基于历史兴趣的推荐，而另外一个比较适合基于好友的推荐。

但是，关于多种不同的推荐引擎之间的性能的对比，寻找到适合某个用户的推荐系统并进行个性化推荐，以及将多个引擎联合起来，生成的组合推荐的性能还没有系统的研究过。所以设计一种可以方便的增加、调整推荐引擎，并能利用多个引擎之间的互动来提高推荐性能的推荐系统是十分有必要性的。

虽然目前在实验室与实际生产中有着多种多样的推荐系统，但是多数推荐系统仅仅能够做到特定场景下的应用，并且在面对着需求发生变化的时候，并不能灵活的应对推荐引擎级别的调整。

本研究旨在开发一款通用的多引擎实时在线推荐系统的核心部分。其能够实现对多种推荐场景的有效应用；并能够实现推荐引擎级别的修改与调整，比如增强时间上下文信息在推荐系统中的权重或者减少用户间社会关系的权重等。

* 1. **国内外研究现状**

关于推荐系统的国内外研究现状，主要分为四个方向来阐述：算法研究、应用场景、推荐策略和存在的问题。

* + 1. **算法研究**

首先，在推荐系统所使用的算法上，主流的算法分为：随机推荐、最热门推荐、基于邻域的推荐算法、隐语义模型以及基于图的推荐算法。

随机推荐是最简单的一种推荐算法，即给定一个用户，将目标物品列表中的随机一个物品推荐给他。这种算法在早期使用较多，目前多用于冷启动物品的推荐以及对新加入用户推荐物品，同时在一些没有复杂的推荐系统的网站上进行新闻资讯的推荐。

最热门推荐是指，将用户对物品的行为，包括浏览、购买、打分等，按照一定的时间跨度进行统计，然后将其中最热门的物品推荐给用户。这种算法虽然原始简单，但仍是目前最普遍使用的算法之一，在各大网站上都可以看到今日排行、最多话题、十大热搜等推荐模块。

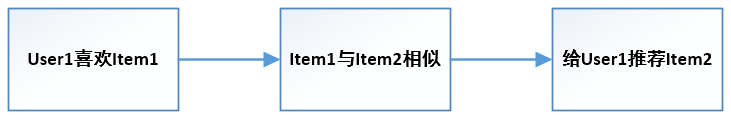
基于邻域的推荐算法比较复杂，主要分为两种：基于用户的协同过滤算法和基于物品的协同过滤算法。

其中，基于用户的协同过滤算法是指根据用户的行为计算用户之间的相似度，然后将与目标用户最相似的几个用户感兴趣的物品推荐给目标用户。



**图1.5 基于用户的协同过滤算法的核心思想**

而基于物品的协同过滤算法则需要先计算物品之间的相似度，然后将与目标用户产生行为的物品相似度高的物品推荐给目标用户。



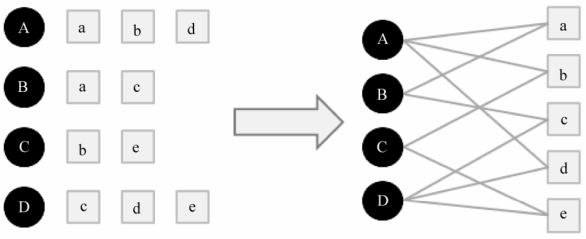
**图1.6 基于物品的协同过滤算法的核心思想**

隐语义模型（LFM，Latent Factor Model）又称为隐含类别模型（Latent Class Model）、话题模型（Topic Model）、 矩阵分解模型（Matrix Factorization Model）等，其都是在同一思想体系的不同扩展，其基本思想就是通过隐含的特征联系用户和物品。

首先，从上面关于推荐系统的本质的讨论可以看出，推荐系统的本质就是通过不同的方式来联系用户和物品，即可以抽象为通过特征集来联系用户和物品。但是，一般的推荐引擎其特征都是人为规定好的，比如用户的相似度、物品的相似度、用户的社交关系等。隐语义模型的不同是，其特征的生成是一个通过机器学习自动生成的黑箱过程，虽然有很好的推荐结果，但是不具有可解释性。

虽然隐语义模型中的特征不可解释，但我们仍然可以在此处做一个简单的类比：假设有100个人来自10个学校，不同学校的学生校服不同，校歌也不同，如果我们需要通过校服来推荐校歌，那么，此处机器自动学习到的隐含特征就可能是学校。需要注意的是，虽然对人来说校服对应校歌很简单，但对机器来说，他们都是平等的数据，校服是否对应校歌要根据训练结果来决定；其次，特征的数量也未必是10个，可能指定了是5个或者20个，到时候特征和结果也都会不一样。

基于图的推荐算法是指，将用户、物品以及其他特征（比如标签或者用户的社交关系）都表现在同一张图上，通过在该关系图上的随机游走，通过设定每一步决策的概率和随机试验的次数，来决定给用户推荐什么物品。该算法的优点是思想简单，不需要写复杂的算法，缺点是大部分图算法所共有的效率低下问题。



**图1.7 基于图的推荐算法的图生成过程**

以上讲解的都是最基础的算法思想，在推荐系统的不断发展之中，也出现了很多对算法的改进方式：比如在计算用户相似度或者物品相似度的时候，对过于热门的用户以及物品进行降权的惩罚，从而提高推荐的性能；或者在其中加入时间、用户的社交信息等额外信息，来提高推荐的性能。

同时，除了上诉几种算法之外，还有以下两种常用的算法：

基于内容的推荐源于信息检索技术，其不依赖于用户对项目的评价信息，而是侧重考察候选推荐项目与用户特征的匹配程度；

基于知识的推荐是一种基于特定领域规则或实例的推理方法，其优点在于不需要建立用户需求偏好模型，缺点是难以制定合理的推理规则。

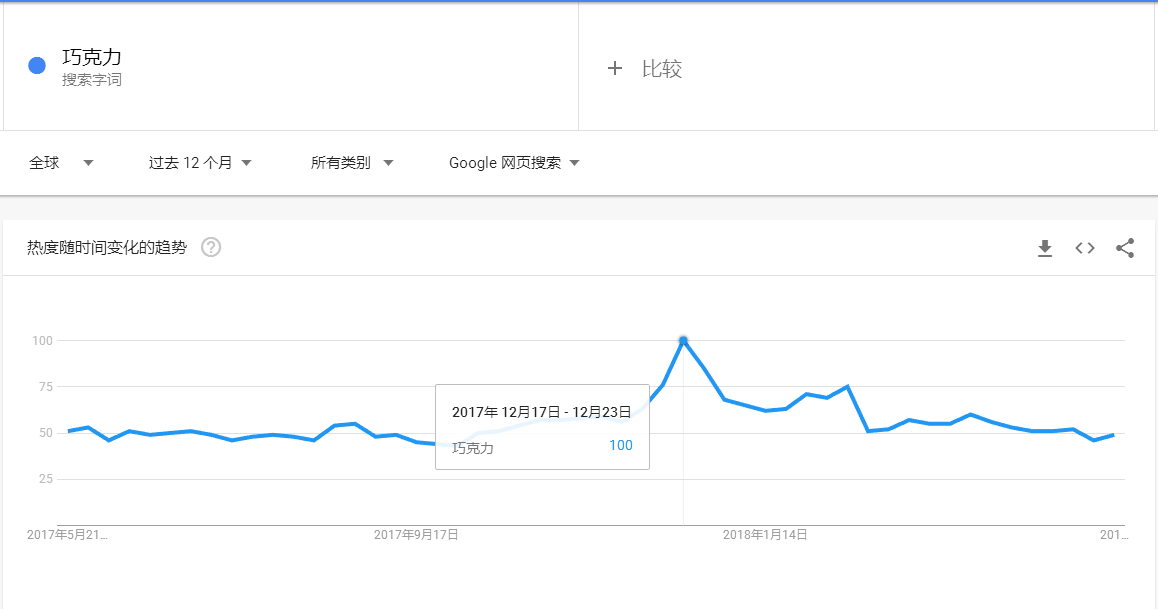
* + 1. **应用场景**

随着推荐系统的发展和社会的进步，推荐系统在不同的应用场景也大放光彩。虽然算法的思想是一致的，但在不同的应用领域，推荐系统的具体设计和具体算法却又有些不同。以下简单介绍一下利用用户标签、利用时间信息、利用社交网络的推荐系统。



**图1.8豆瓣读书的基于标签的推荐系统的页面**

利用用户标签的推荐系统，引入的一个新的变量就是标签，而通过标签这一个新的变量，能在以下几个方向上改进算法：通过标签计算用户相似度，然后通过用户相似度推荐物品；通过标签计算物品相似度，然后通过用户相似度推荐物品；通过用户常用的标签来给用户推荐该标签下的物品；通过用户所产生行为的物品上的标签，给用户推荐该标签下的物品。



**图1.9 巧克力的搜索热度随时间的变化折线图**

利用时间信息的推荐系统，引入了时间这个新变量，其考虑了用户或者物品在时间中的改变情况，比如用户的兴趣会变，对某个物品的评价会随着时代的改变而改变，如图1.9所示，在情人节前后和圣诞节前后巧克力的搜索热度会上升，因此在这个阶段可以提高巧克力的推荐比例。通过引入时间这一个新变量，有以下几种方式改良算法：加入时间的考虑计算用户的相似度，然后根据用户相似度推荐物品；加入时间的考虑计算物品的相似度，然后根据物品相似度推荐；再利用用户相似度推荐时，加大相似用户最近行为的权重；再利用物品相似度推荐时，加大用户最近喜欢的物品的权重；以及上面几种方式的组和改进方式。

利用社交网络的推荐系统，引入了用户之间的社交关系这一新变量。而用户的社交关系能在以下几个方向上改良算法：通过用户的社交关系重新计算用户之间的相似度，再利用用户相似度进行推荐；或者直接给用户推荐其社交网络中好友喜欢的物品。

同时，随着人们的不断研究，以及应用场景的不断开拓，推荐系统逐渐被应用在了更广阔的领域，新出现的领域有基于大数据的推荐系统，基于地理位置的推荐系统，组推荐系统等。

* + 1. **推荐策略**

过去对于推荐系统的研究，主要集中在如何提高单一引擎在单一场景下的推荐性能，随着时代的发展，逐渐出现了两种新的趋势。

第一种趋势，是考虑利用多种模型，多种算法的组合推荐来代替单一推荐引擎的推荐。这种转变往往在各种推荐系统的比赛上有良好的表现，但是因为性能的瓶颈，生产中并不多见。

第二种趋势，是使用更多的信息来进行推荐代替原来单纯利用用户的行为进行推荐。单纯的利用用户的行为数据，不论怎样优化算法，推荐性能都存在一个上限，所以近几年逐渐发展起了利用用户的社交网络、利用行为的时间信息等多种方式的推荐系统，而基于大数据的推荐则更是拥有海量的数据来源，会对这些信息综合进行推荐。

* + 1. **存在的问题**

推荐系统所面临的技术层面的问题，主要有以下几类：

冷启动问题，自推荐系统诞生之初就一直存在，目前尚且没有完美的解决方案，比较成熟的方法有：对新用户，采取其注册信息、社交关系等方式进行推荐或者推荐最热门物品，而对于新物品，一般专门使用一个引擎进行推荐或者预先处理出特征再进行推荐。

用户的多兴趣和用户兴趣转变问题，推荐系统直接对应的是社会中的行为，而行为总是会随着时间进行改变，目前比较好的解决方法有两种，一是可以在推荐的过程中加入时间变量，二是可以采取兴趣云的方式，对每个兴趣方面都单独推荐。

新技术的应用问题，如机器学习技术与数据挖掘技术如何在推荐系统中有效的应用、大数据环境下如何更有效的推荐等。

推荐系统所面临的社会问题，主要有以下几类：

隐私保护和安全性问题，推荐系统不可避免的会涉及到用户的行为，还有一些会涉及到用户的注册信息和社交网络数据，而这些信息的泄露或者滥用都会给用户造成潜在的威胁，因此在推荐系统的设计过程中也要考虑保护用户隐私的问题。

沉迷问题，推荐系统既可以帮助用户找到其感兴趣的事物，也会造成用户所接受的信息来源单一或者狭小的问题，“过度推荐”使得人们接触的世界越来越小，形成领域内的信息闭塞。尤其是在现象级应用今日头条、抖音等使用推荐算法获得巨大成功的同时，也使得越来越多的人沉迷其中，无法自拔，形成沉迷其中的“精神鸦片”现象。推荐系统的防沉迷问题也是一个重要的研究课题。

* 1. **主要设计任务及预期目标（指标）**
     1. **主要设计任务**

设计一个具有多个推荐引擎的推荐系统，并且能够方便的对其中的推荐引擎进行修改。同时设计对应的ETL模块、评估模块、测试模块和训练模块，从而形成一个完整的系统。

* + 1. **预期目标（指标）**

在工程系统方面，应该做到：

1. 拥有完整的从数据流入、数据处理、生成推荐结果、推荐结果评测整条工作流，形成一个可工作的完整系统；
2. 能够方便地对各个推荐引擎进行增删查改，并可以方便的设定其中的参数。

在推荐引擎方面，应该做到：

1. 实现常见算法的推荐引擎，如基于用户的协同过滤算法、基于物品的协同过滤算法、基于图的随机游走模型等；
2. 能够对每个引擎形成良好的接口，可以对其中的一部分结果进行调用，来生成融合推荐的结果。

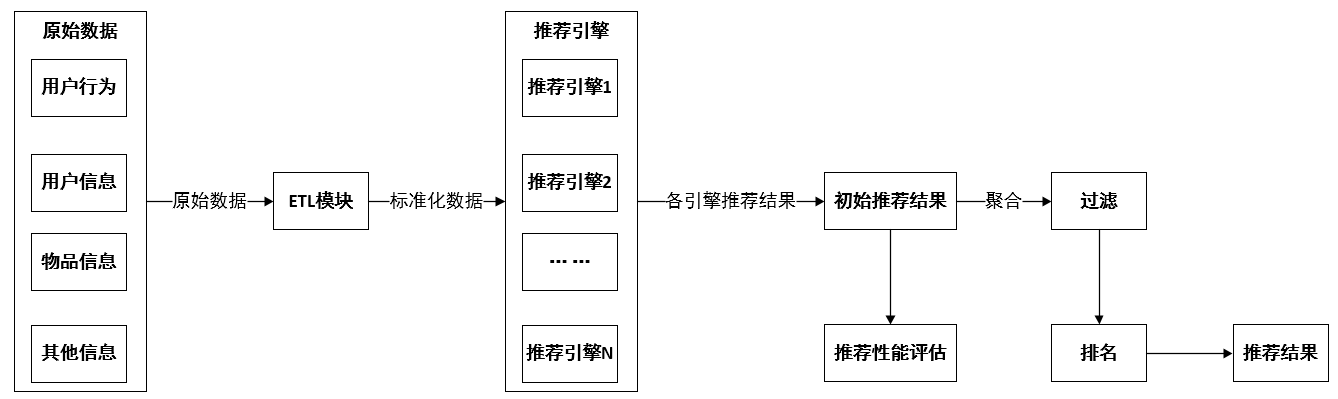
在推荐的效果方面，应该做到：

1. 简单的推荐引擎应该获得与目前研究性能；
2. 同时期待融合推荐引擎能够获得更加优秀的性能。
3. **系统设计方案**

**2.1总体设计方案**

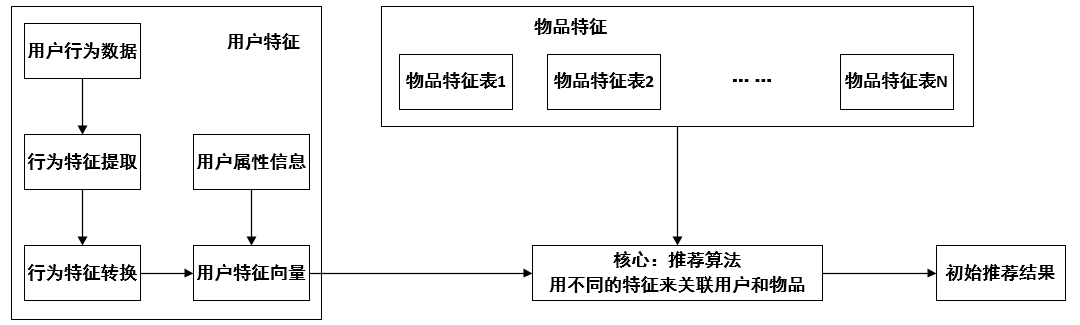
多引擎推荐系统的整体设计方案如图2.1所示，其核心为多种推荐引擎，包括不同算法上或不同应用场景下的推荐引擎。

多引擎推荐系统的辅助模块有ETL模块，负责将原始数据清洗成推荐引擎识别的标准数据；初始推荐结果的聚合、过滤和排名模块，负责将不同引擎生成的原始推荐结果处理成最终的推荐结果；推荐性能评估模块，负责对生成的推荐结果进行不同纬度的评测。



**图2.1 多引擎推荐系统整体设计方案**

对于推荐引擎部分，其通用的设计架构如图2.2所示，主要分为三个部分：用户特征生成模块，主要使用多种原始数据，如用户行为、用户属性、用户社交网络等来生成用户的特征；物品特征生成模块，通过使用多种原始数据，如用户在物品上的行为、物品的标签等来生成物品的特征；推荐结果生成模块，利用传入的用户特征和物品特征，通过不同的推荐算法，来生成最终的推荐结果。



**图2.2 推荐引擎设计通用架构**

**2.2关键问题解决**

1. 各个引擎部分如何设计——包括算法思想上、参数设定上

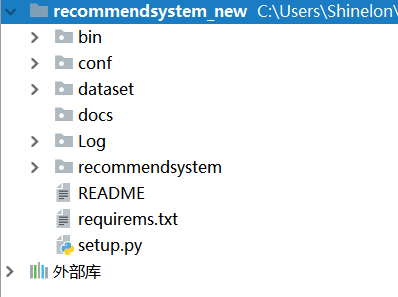
2. 不同的引擎如何组合到一起

3. 训练效率问题

（本部分要求详细记载和描述。）

**2.3设计结果呈现**

本次设计的多引擎推荐系统由python工程实现，最终实现的工程结构如图2.3和图2.4所示。



**图2.3 多引擎推荐系统整体设计方案**

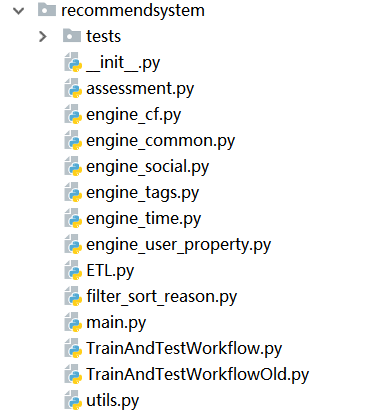
整个python的工程文件如图2.3所示，其中比较重要的有四个文件夹：

conf：主要保存工程中的一些设定，如一些推荐引擎的默认参数，或者进行大规模训练时的参数

dataset：存放训练所使用的数据集

Log：存放不同推荐引擎推荐性能的评估

recommendsystem：存放代码的主体部分，包括多种推荐引擎、各种辅助模块以及单元测试模块



**图2.4 多引擎推荐系统整体设计方案**

整个工程的代码部分如图2.4所示，主要分为以下几个部分：

main模块：整个程序的入口，可以调用其他各个模块的组件，形成推荐workflow；同时，不同组件的调用方式demo也在此模块。

ETL模块：负责将原始数据清洗成所需要的数据，能处理的原始数据格式主要有矩阵或者多元组列表，返回的标准数据为Dataframe或者清洗过的多元组数据。

engine模块：实现各种不同的推荐引擎，每种引擎同时采取尽可能多的实现方式。

filter\_sort\_reason模块：对推荐结果进行过滤和排序，同时对有些引擎可以加上推荐原因，增加结果可信度。

assessment模块：对推荐引擎的性能进行评测，由summary功能主导，也可分别查询推荐的准确率、召回率、新颖性、覆盖率等指标。

TrainAndTestWorkflow模块：用于本次测试不同推荐引擎的性能的模块，分为单引擎的测试和多种引擎组合推荐的测试，同时对其中所需的参数进行了多种测试。

其中，各种不同的推荐引擎的功能如下：

engine\_common：主要负责一些不需要复杂逻辑，但是又经常用到的推荐引擎的实现，包括最热门推荐（日热门、周热门、月热门和总最热门）和以随机推荐引擎为核心的对冷启动物品的推荐。

engine\_cf：该引擎模块实现了4种计算相似度的方法，分别是用户相似度、对热门用户进行惩罚的用户相似度、物品相似度以及对热门物品进行惩罚的物品相似度；在此之上实现了两种推荐引擎，分别是基于用户的协同过滤推荐引擎和基于物品的协同过滤推荐引擎；同时，基于用户行为的随机游走算法也由本模块实现。

engine\_user\_property：该模块实现了利用用户的信息（如性别、年龄、职业等）计算用户的相似度，并使用该相似度完成基于用户的协同过滤推荐。

engine\_tags：该模块实现了两种标签相似度的计算方法，分别是基于同一用户所打的标签计算标签、基于统一物品上所打的标签计算标签的相似度；同时实现了基于标签的用户相似度计算和基于标签的物品相似度计算；同时实现了利用基于标签得出的相似度的基于用户的协同过滤推荐和基于物品的协同过滤推荐。

engine\_social：该模块实现了利用用户的社交关系来计算用户的相似度，然后基于用户的社交关系进行推荐；同时，基于社交关系的相似度还可以拿来补充其他引擎。

engine\_time：该模块考虑了时间在推荐引擎中的作用，在计算用户相似度、物品相似度，以及在两种协同过滤算法中都考虑了时间的衰减项。

1. **系统测试、验证与结果分析**

**3.1****测试、验证方法**

首先针对各个不同的模块的可用性，由相应的单元检测负责，目前所有推荐引擎皆通过对应的单元检测，说明所有的单个推荐引擎功能正确，此处不再详述。

解决了可用性的问题之后，关键的问题就是对推荐系统的性能进行评测，该部分主要有assessment模块进行，其评测指标如下：  
 召回率recall：即推荐出来的结果中，命中的物品与测试集中用户实际产生行为的物品的比例，实际值在0:1之间，越接近1说明性能越好。

（公式）

准确率precision：即推荐出来的结果中，命中的物品与推荐的数量之间的比例，实际值在0:1之间，越接近1说明性能越好。

（公式）

流行度popularity：即推荐结果中的物品的平均热门程度，首先根据用户行为矩阵获得每个物品的流行度，然后在对推荐列表中的物品求加权平均。根据原来矩阵的稠密程度的不同而不同，一般而言，数值越大说明推荐的物品热门度越高，如果不是专门推荐热门物品，则应该对热门物品进行适当降权。

（公式）

新颖度novelty：流行度的倒数，一般而言，数值越大，推荐结果的新颖度越高，对长尾物品有更好的覆盖率。

（公式）

覆盖率coverage：即推荐结果对整个物品集的覆盖情况，即推荐的物品的个数与物品全集的个数的比例，其结果位于0:1之间，越接近1说明推荐系统的覆盖率越好。

（公式）

多样性diversity：推荐结果多样性的度量，计算方法为1减去推荐结果中所有物品相似度的加权平均，越接近1说明推荐系统的多样性越好。

（公式）

除此之外，还对推荐的每个环节和测试的每个环节所用的时间进行了统计，方便评估不同推荐系统的效率。

而在测试方法上，考虑到每次测试的随机性，所以按照数据集的大小安排了3到10次重复试验，同时为了使每个推荐引擎得到最佳的推荐结果，对其中的关键参数设定了多组对比。

既要对每个推荐引擎进行多组实验，同时考虑在大数据集下，每次实验都要耗费不少时间，整体所消耗的时间是巨大的。基于以上考虑，最终的测试在阿里云上，通过多台云服务器并行执行来获得不同引擎的性能评估。

**3.2****测试、验证结果**

以下测试结果都是在阿里云服务器（配置：4 vCPU 4 GB (I/O优化) ecs.s3.medium 5Mbps (峰值)）中测试得出，测试所使用的数据集为MovieLens的1M数据集的2个缩减版本，分别取其前100个用户和前1000个用户进行测试。

* + 1. **随机推荐引擎Random：**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **参数** | **recall** | **precision** | **coverage** | **popularity** | **novelty** | **diversity** |
| **结果** | **0.88%** | **5.71%** | **0.582** | **1.79** | **0.56** | **0.68** |

**小数据及**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **参数** | **recall** | **precision** | **coverage** | **popularity** | **novelty** | **diversity** |
| **结果** | **0.59%** | **4.55%** | **0.997** | **3.15** | **0.32** | **0.70** |

* + 1. **最热门推荐MostHot：**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **参数** | **recall** | **precision** | **coverage** | **popularity** | **novelty** | **diversity** |
| **结果** | **4.77%** | **7.09%** | **0.007** | **3.63** | **0.28** | **0.33** |

**小数据及**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **参数** | **recall** | **precision** | **coverage** | **popularity** | **novelty** | **diversity** |
| **结果** | **4.60%** | **8.15%** | **0.005** | **5.90** | **0.17** | **0.30** |

* + 1. **基于用户的协同过滤推荐UserCF：**

**Xiao**

N=5

Similarity=downhot

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **参数** | **recall** | **precision** | **coverage** | **popularity** | **novelty** | **diversity** |
| **k=5** | **4.48%** | **21.42%** | **0.09** | **3.12** | **0.32** | **0.74** |
| **k=10** | **5.00%** | **23.88%** | **0.06** | **3.29** | **0.30** | **0.79** |
| **k=20** | **5.33%** | **25.44%** | **0.03** | **3.44** | **0.29** | **0.79** |
| **k=40** | **5.05%** | **24.10%** | **0.052** | **3.56** | **0.28** | **0.84** |

norm

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **参数** | **recall** | **precision** | **coverage** | **popularity** | **novelty** | **diversity** |
| **K=5** | **4.56%** | **21.78%** | **0.079** | **3.19** | **0.31** | **0.77** |
| **k=10** | **5.02%** | **24.00%** | **0.058** | **3.33** | **0.30** | **0.76** |
| **k=20** | **5.14%** | **24.56%** | **0.039** | **3.46** | **0.29** | **0.80** |
| **k=40** | **4.90%** | **23.42%** | **0.023** | **3.58** | **0.28** | **0.89** |

N=10

Similarity=downhot

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **参数** | **recall** | **precision** | **coverage** | **popularity** | **novelty** | **diversity** |
| **k=5** | **7.61%** | **18.17%** | **0.143** | **3.03** | **0.33** | **0.73** |
| **k=10** | **8.30%** | **19.83%** | **0.099** | **3.20** | **0.31** | **0.73** |
| **k=20** | **8.70%** | **20.77%** | **0.064** | **3.36** | **0.30** | **0.77** |
| **k=40** | **8.38%** | **20.01%** | **0.039** | **3.48** | **0.29** | **0.79** |

norm

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **参数** | **recall** | **precision** | **coverage** | **popularity** | **novelty** | **diversity** |
| **k=5** | **7.51%** | **17.93%** | **0.132** | **3.08** | **0.32** | **0.72** |
| **k=10** | **8.32%** | **19.88%** | **0.093** | **3.24** | **0.31** | **0.74** |
| **k=20** | **8.56%** | **20.43%** | **0.060** | **3.38** | **0.30** | **0.78** |
| **k=40** | **8.04%** | **19.19%** | **0.036** | **3.49** | **0.29** | **0.79** |

N=16

Similarity=downhot

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **参数** | **recall** | **precision** | **coverage** | **popularity** | **novelty** | **diversity** |
| **k=5** | **10.59%** | **15.81%** | **0.194** | **2.96** | **0.34** | **0.72** |
| **k=10** | **11.64%** | **17.37%** | **0.132** | **3.14** | **0.32** | **0.72** |
| **k=20** | **12.10%** | **18.06%** | **0.085** | **3.29** | **0.30** | **0.74** |
| **k=40** | **11.49%** | **17.14%** | **0.052** | **3.41** | **0.29** | **0.78** |

norm

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **参数** | **recall** | **precision** | **coverage** | **popularity** | **novelty** | **diversity** |
| **k=5** | **10.35%** | **15.45%** | **0.181** | **3.00** | **0.33** | **0.71** |
| **k=10** | **11.40%** | **17.01%** | **0.125** | **3.17** | **0.32** | **0.72** |
| **k=20** | **11.91%** | **17.78%** | **0.078** | **3.32** | **0.30** | **0.76** |
| **k=40** | **11.32%** | **16.89%** | **0.049** | **3.42** | **0.29** | **0.79** |

Da

N=5

Similarity=downhot

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **参数** | **recall** | **precision** | **coverage** | **popularity** | **novelty** | **diversity** |
| **k=5** | **4.47%** | **25.12%** | **0.203** | **5.22** | **0.19** | **0.72** |
| **k=10** | **5.26%** | **29.56%** | **0.152** | **5.38** | **0.19** | **0.72** |
| **k=20** | **5.83%** | **32.76%** | **0.103** | **5.52** | **0.18** | **0.71** |
| **k=40** | **6.06%** | **34.08%** | **0.072** | **5.63** | **0.18** | **0.72** |

norm

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **参数** | **recall** | **precision** | **coverage** | **popularity** | **novelty** | **diversity** |
| **K=5** | **4.57%** | **25.68%** | **0.185** | **5.29** | **0.19** | **0.72** |
| **k=10** | **5.34%** | **30.00%** | **0.138** | **5.45** | **0.18** | **0.72** |
| **k=20** | **5.65%** | **31.74%** | **0.097** | **5.57** | **0.18** | **0.72** |
| **k=40** |  |  |  |  |  |  |

N=10

Similarity=downhot

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **参数** | **recall** | **precision** | **coverage** | **popularity** | **novelty** | **diversity** |
| **k=5** | **7.72%** | **21.69%** | **0.280** | **5.13** | **0.19** | **0.72** |
| **k=10** | **9.05%** | **25.43%** | **0.209** | **5.29** | **0.19** | **0.71** |
| **k=20** | **9.73%** | **27.35%** | **0.154** | **5.42** | **0.18** | **0.71** |
| **k=40** | **9.95%** | **27.97%** | **0.112** | **5.53** | **0.18** | **0.72** |

norm

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **参数** | **recall** | **precision** | **coverage** | **popularity** | **novelty** | **diversity** |
| **k=5** | **7.76%** | **21.80%** | **0.262** | **5.19** | **0.19** | **0.71** |
| **k=10** | **9.04%** | **25.39%** | **0.191** | **5.36** | **0.19** | **0.72** |
| **k=20** | **9.51%** | **26.73%** | **0.138** | **5.48** | **0.18** | **0.71** |
| **k=40** |  |  |  |  |  |  |

N=16

Similarity=downhot

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **参数** | **recall** | **precision** | **coverage** | **popularity** | **novelty** | **diversity** |
| **k=5** | **10.83%** | **19.02%** | **0.357** | **5.06** | **0.20** | **0.72** |
| **k=10** | **12.62%** | **22.16%** | **0.261** | **5.21** | **0.19** | **0.71** |
| **k=20** | **13.63%** | **23.93%** | **0.194** | **5.35** | **0.19** | **0.71** |
| **k=40** | **13.65%** | **23.98%** | **0.144** | **5.47** | **0.18** | **0.71** |

norm

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **参数** | **recall** | **precision** | **coverage** | **popularity** | **novelty** | **diversity** |
| **k=5** | **10.95%** | **19.23%** | **0.327** | **5.11** | **0.20** | **0.71** |
| **k=10** | **12.53%** | **22.00%** | **0.244** | **5.27** | **0.19** | **0.72** |
| **k=20** | **13.45%** | **23.62%** | **0.181** | **5.41** | **0.18** | **0.72** |
| **k=40** |  |  |  |  |  |  |

* + 1. **基于物品的协同过滤推荐ItemCF：**

Similarity=downhot

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **参数** | **recall** | **precision** | **coverage** | **popularity** | **novelty** | **diversity** |
| **k=5** | **7.21%** | **10.66%** | **0.253** | **2.51** | **0.40** | **0.85** |
| **k=10** | **9.32%** | **13.79%** | **0.216** | **2.72** | **0.37** | **0.85** |
| **k=20** | **10.45%** | **15.45%** | **0.191** | **2.88** | **0.35** | **0.84** |
| **k=40** | **10.91%** | **16.13%** | **0.176** | **2.94** | **0.34** | **0.83** |

norm

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **参数** | **recall** | **precision** | **coverage** | **popularity** | **novelty** | **diversity** |
| **k=5** | **6.29%** | **9.31%** | **0.272** | **2.46** | **0.41** | **0.62** |
| **k=10** | **7.88%** | **11.66%** | **0.231** | **2.62** | **0.38** | **0.60** |
| **k=20** | **8.53%** | **12.61%** | **0.197** | **2.74** | **0.37** | **0.58** |
| **k=40** | **8.81%** | **13.03%** | **0.173** | **2.80** | **0.36** | **0.56** |

大

Similarity=downhot

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **参数** | **recall** | **precision** | **coverage** | **popularity** | **novelty** | **diversity** |
| **k=5** | **12.77%** | **22.62%** | **0.228** | **5.25** | **0.19** | **0.84** |
| **k=10** | **13.44%** | **23.79%** | **0.180** | **5.41** | **0.18** | **0.82** |
| **k=20** | **13.40%** | **23.72%** | **0.146** | **5.50** | **0.18** | **0.81** |
| **k=40** | **13.14%** | **23.26%** | **0.130** | **5.56** | **0.18** | **0.80** |

norm

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **参数** | **recall** | **precision** | **coverage** | **popularity** | **novelty** | **diversity** |
| **k=5** | **12.03%** | **21.31%** | **0.252** | **5.14** | **0.19** | **0.62** |
| **k=10** | **12.79%** | **22.66%** | **0.196** | **5.30** | **0.19** | **0.58** |
| **k=20** | **12.92%** | **22.88%** | **0.159** | **5.40** | **0.19** | **0.55** |
| **k=40** | **12.56%** | **22.24%** | **0.141** | **5.46** | **0.18** | **0.53** |

* + 1. **基于图的随机游走推荐Personalrank：**

P=

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **参数** | **recall** | **precision** | **coverage** | **popularity** | **novelty** | **diversity** |
| **P=0.5** | **2.73%** | **4.34%** | **0.507** | **3.43** | **0.32** | **0.65** |
| **P=0.7** | **2.40%** | **3.84%** | **0.515** | **3.41** | **0.32** | **0.65** |
| **P=0.8** | **2.10%** | **3.06%** | **0.388** | **2.39** | **0.42** | **0.63** |

da

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **参数** | **recall** | **precision** | **coverage** | **popularity** | **novelty** | **diversity** |
| **P=0.5** | **5.69%** | **9.15%** | **0.212** | **4.22** | **0.25** | **0.54** |
| **P=0.7** | **6.61%** | **9.62%** | **0.165** | **3.11** | **0.32** | **0.52** |
| **P=0.8** | **6.52%** | **9.50%** | **0.169** | **3.11** | **0.32** | **0.52** |

* + 1. **基于用户属性的推荐UserProperty：**

100 333

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **参数** | **recall** | **precision** | **coverage** | **popularity** | **novelty** | **diversity** |
| **k=5** | **5.35%** | **7.84%** | **0.162** | **3.04** | **0.33** | **0.55** |
| **k=10** | **6.77%** | **9.92%** | **0.117** | **3.21** | **0.31** | **0.52** |
| **k=20** | **7.63%** | **11.18%** | **0.072** | **3.35** | **0.30** | **0.48** |
| **k=40** | **8.31%** | **12.19%** | **0.043** | **3.44** | **0.29** | **0.43** |

336

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **参数** | **recall** | **precision** | **coverage** | **popularity** | **novelty** | **diversity** |
| **k=5** | **5.00%** | **7.34%** | **0.160** | **3.04** | **0.33** | **0.55** |
| **k=10** | **6.43%** | **9.44%** | **0.113** | **3.21** | **0.31** | **0.52** |
| **k=20** | **7.47%** | **10.96%** | **0.074** | **3.35** | **0.30** | **0.48** |
| **k=40** | **8.22%** | **12.06%** | **0.046** | **3.44** | **0.29** | **0.43** |

363

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **参数** | **recall** | **precision** | **coverage** | **popularity** | **novelty** | **diversity** |
| **k=5** | **5.38%** | **7.89%** | **0.160** | **3.04** | **0.33** | **0.54** |
| **k=10** | **6.88%** | **10.10%** | **0.116** | **3.20** | **0.31** | **0.52** |
| **k=20** | **7.83%** | **11.49%** | **0.071** | **3.35** | **0.30** | **0.48** |
| **k=40** | **8.40%** | **12.32%** | **0.042** | **3.45** | **0.29** | **0.42** |

633

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **参数** | **recall** | **precision** | **coverage** | **popularity** | **novelty** | **diversity** |
| **k=5** | **5.28%** | **7.75%** | **0.163** | **3.03** | **0.33** | **0.55** |
| **k=10** | **6.59%** | **9.66%** | **0.144** | **3.19** | **0.31** | **0.52** |
| **k=20** | **7.67%** | **11.25%** | **0.068** | **3.35** | **0.30** | **0.48** |
| **k=40** | **8.08%** | **11.85%** | **0.042** | **3.44** | **0.29** | **0.43** |

1000

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **参数** | **recall** | **precision** | **coverage** | **popularity** | **novelty** | **diversity** |
| **k=5** | **5.37%** | **9.53%** | **0.291** | **5.21** | **0.19** | **0.59** |
| **k=10** | **6.44%** | **11.41%** | **0.195** | **5.40** | **0.19** | **0.55** |
| **k=20** | **7.50%** | **13.30%** | **0.130** | **5.56** | **0.18** | **0.52** |

* + 1. **基于物品标签的推荐Tags：**

100

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **参数** | **recall** | **precision** | **coverage** | **popularity** | **novelty** | **diversity** |
| **k=5** | **2.16%** | **3.17%** | **0.155** | **1.72** | **0.58** | **0.75** |
| **k=10** | **2.33%** | **3.42%** | **0.155** | **1.70** | **0.59** | **0.75** |
| **k=20** | **2.52%** | **3.70%** | **0.167** | **1.65** | **0.61** | **0.75** |
| **k=40** | **2.60%** | **3.82%** | **0.178** | **1.67** | **0.60** | **0.74** |

**3.3测试、验证结果的分析**

1. **总结**

**4.1本设计的特色与总结**

（包括：本设计的成果展示；对完成工作的总结，并以条目形式罗列出本文所实现研究成果的一些特点和优点，使人一目了然。）

* 1. **设计中出现的问题及思考**

（总结完成过程中遇到的一些问题的解决办法，最好能把一些细节再回顾一下，尽量把遇到的问题的解决办法都写出来；可能有的问题在前文中已经提到了，但仍需在这里再罗列归纳一下。）

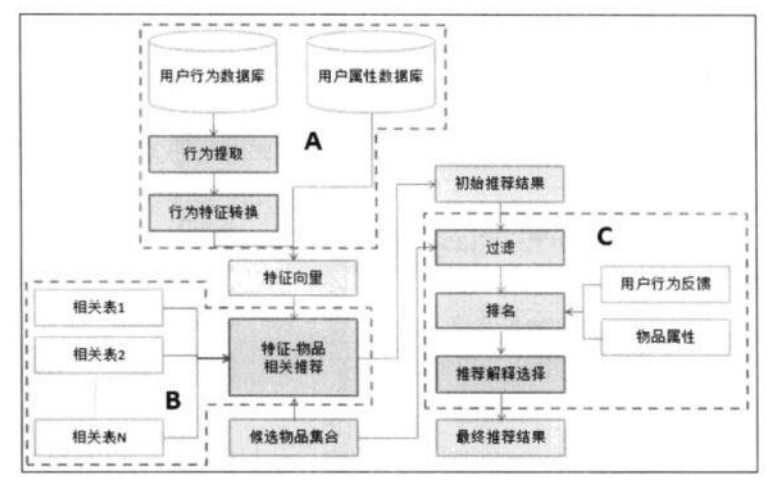
* 1. **展望**

（总结本文尚待完善之处，展望对今后改进和完善方向的思考；接着站在更全局性的角度和高度展望一下相关研究领域今后的发展，最后说一段展望发展的概括性的话。）

开题报告

### 2. 本研究的意义和目的

系统整体的框架如下图所示：



**图1 多引擎推荐系统架构图**

作为主流推荐系统中效果最好的组合推荐系统，常常因为其推荐效率低下，以及各个组合推荐引擎之间的组成与协调需要大量的实验才能得到最优的结果所诟病。

所以，本研究还将对主流推荐系统中效果最好的组合推荐系统进行深入的研究，对包括组合推荐的算法组成、运算效率、推荐效果以及适用场景进行详细的分析，给出不同情境下最佳的组合推荐配置。

二、主要研究开发内容

### 1. 主要研究内容

设计部分：

设计一个通用的多引擎实时在线推荐系统的核心部分，只需要输入相应特征，以及所需要的推荐引擎的类型以及比例，即可给出推荐结果，外部只需要调用相应的接口即可。

研究部分：

基于设计部分所编写的通用推荐系统，来进行组合推荐系统的研究。主要研究方向为在不同的应用场景下，不同推荐算法的组成对推荐效果的影响，包括推荐效果，推荐效率等。

优化部分：

在基于研究部分结论的基础上，测试系统是否可以使用机器学习算法自行调整内部推荐引擎的组成到最佳状态，实现仅输入特征的条件下，系统就可以自动进行推荐引擎程度的调整，从而生成最佳推荐策略的效果；

设计简单的交互系统与接口，实现在真实情况下的测试与仿真；

将系统部署到云服务器中，测试在大规模数据推荐请求下，系统的响应情况，并进行相应的优化，争取做到大规模数据下的实时性推荐。

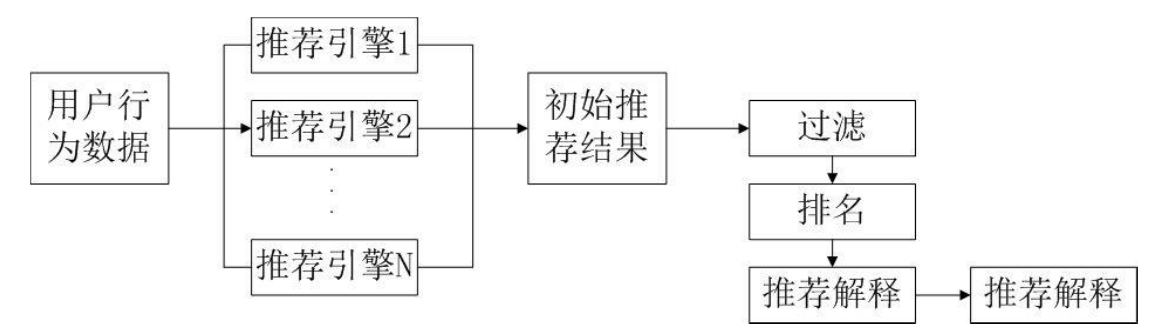
### 2. 技术路线

推荐系统技术原理部分：

主要参考了项亮的《推荐系统实践》以及国内外论文，对常见推荐系统的推荐思想，以及主要算法理论有了基础的了解，并对其中的利用用户行为的推荐系统、利用上下文信息的推荐系统和利用社交网络的推荐系统部分进行了比较深入的研究。

推荐系统的代码实现部分：

单独的每个推荐引擎或者每个推荐算法的实现代码主要参考项亮的《推荐系统实践》【1】，并进行单个模块的调试。而联合了各个推荐算法的推荐系统则主要有以下几个部分组成：



**图2 多引擎通用推荐系统架构图**

组合推荐算法的调优部分：

在上一个阶段所构建的系统的基础上，完成组合推荐系统性能的研究，初步测试主要对应图2中的从初始推荐结果到过滤排名的阶段；而在得出最佳的配置后，将在推荐引擎级别进行调整。

系统上云部署部分：

考虑到阿里云目前占有云市场的主流地位，且提供了完善的上云支持【6】，所以在完成了系统的测试与优化后，考虑利用阿里云技术完成部署与真实环境的模拟。

### 3. 可行性分析

推荐系统作为一个机器学习领域的一个重要的应用，已经从机器学习中分离了出来，成为了一个独立的学科。经过几十年的不断探索，推荐系统已经得到了完备的发展，有着完善的资源：不论是书籍，论文期刊，还是数据集，源代码都有很多可以参考的资源，因此给项目的完成提供了很好的后台支持。

虽然整体上作为一个复杂的系统的代码因为各种原因而难以找到，但目前对于每个单一的推荐算法，基本上都有了比较完善的思想与可以参考的代码，因此在构建系统模块的时候拥有较多的参考资源。

在组合推荐算法的测试中，有很多开源的数据集，比如MovieLens，Jester Joke，Netflix数据集，可以进行各种环境下的测试。

最后，借助阿里云平台强大的计算能力和并行执行能力，可以在尽可能短的时间内得出测试的结果，而不需要等待大量的时间来得到结果，使得在几个月内测试各种不同的组合成为可能。

### 2.

### 预期目标

基本目标：完成一个通用的多推荐引擎的推荐系统设计，并使用该系统完成对组合推荐算法的测试，寻找出在不同的推荐环境下最佳的推荐引擎组合配置。

优化目标：

* 提高系统的实时性，使得利用用户的反馈，能实时更新推荐列表；
* 使用相关技术将此推荐系统部署到云端，从而实现对真实使用环境的模拟；
* 使用机器学习方法，使得推荐系统能够针对不同应用场景，自行寻找并配置最优的推荐引擎组合，省去人为调优的过程。

**参考文献***（开题报告的参考文献）*

[1] 项亮编. 推荐系统实践. 北京: 人民邮电出版社, 2017

[2] 刘鲁，任晓丽.推荐系统研究进展及展望. 信息系统学报, 2008, 2(1): 82～90

[3] 朱扬勇，孙婧. 推荐系统研究进展. 计算机科学与探索, 2015, 9(5): 513～525

[4] 孟祥武，刘树栋，张玉洁等. 社会化推荐系统研究. 软件学报, 2015, 26(6): 1356～1372

[5] 许海玲，吴潇，李晓东等. 互联网推荐系统比较研究. 软件学报, 2009, 20(2): 350～362

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

[7]阿里云计算. 阿里云计算学习平台. <https://edu.aliyun.com/promotion/28>.

[8] Li Xie, Wenbo Zhou, Yaosen Li. Application of Improved Recommendation System Based

on Spark Platform in Big Data Analysis. BULGARIAN ACADEMY OF SCIENCES, 2016, 16(6): 245～255

[9] Chih-Fong Tsai, Chihli Hung. Cluster ensembles in collaborative filtering recommendation. Applied Soft Computing, 2012, 12: 1417～1425

[10] MIHAI GABROVEANU. RECOMMENDATION SYSTEM BASED ON ASSOCIATION RULES FOR DISTRIBUTED E-LEARNING MANAGEMENT SYSTEMS. ACTA UNIVERSITATIS CIBINIENSIS – TECHNICAL SERIES, 2015, 67: 99～104

[11] Mehmet Ali SALAHLI Tokay Gasimzade, Flora Alasgarova, Akber Guliyev. The use of predictive models in intelligent recommendation systems. Procedia Computer Science, 2016, 102: 515～519

[12] Jian Wei, Jianhua He, Kai Chen etc. Collaborative filtering and deep learning based recommendation system for cold start items. Expert Systems With Applications, 2017, 69: 29～39

**参考文献**

文献排列顺序按照在论文正文中出现的先后循序进行，并在正文中将参考文献引用的位置标注出来，两者编号应该一致。

1. Morgan R M, Hunt S D. The commitment-trust theory of relationship marketing [J]. Journal of Marketing, 1994, 58(3): 20-38.
2. 李里特, 候文义. 农业产业化和结构调整的几个关键问题[J]. 科技导报, 2002(1): 36-39.
3. Beck C A J, Sales B D. Family mediation: Facts, myths, and future prospects[M]. Washington, DC: American Psychological Association. 2001: 10-12.
4. 郭健. 哈佛大学发展史研究[M]. 石家庄:河北教育出版社, 2000: 2-3.
5. 辛希孟. 信息技术和信息服务国际研讨会论文集: A集[C]. 北京:中国社会科学出版社, 1994.
6. 张筑生. 微分半动力系统的不变集[D]. 北京: 北京大学数学系数学研究所, 1983.
7. 冯西桥. 核反应堆压力管道和压力容器的LBB分析[R]. 北京: 清华大学核能技术设计研究院, 1997.
8. 钟文发, 赵玮. 运筹学的理论和应用——中国运筹学会第五届大会论文集[C]. 西安: 西安电子科技大学出版社, 1996: 468-471.
9. Spivak G. "Can the Subaltern Speak?" [A]. In Nelson C &GrossbergL(eds.). Victory in Limbo: Imigism [C]. Urbana: University of Illinois Press, 1988: 271-313.
10. GB/T 16159-1996, 汉语拼音正词法基本规则[S].
11. 姜锡洲. 一种温热外敷药制备方案[P]. 中国专利: 881056073, 1989-07-26.
12. 王明亮.关于中国学术期刊标准化数据库系统工程的进展[M/CD]. <http://www.cajed.Cn/pub/wml.txt/980810-2.html,1998.08.16/1998-10-04>.

[13] 万锦坤. 中国大学学报论文文摘(1983-1993)英文版[DB/CD]. 北京:中国大百科全书出版社, 1996.

**本科生毕业设计任务书**

**一、题目：**

**二、指导教师对毕业设计的进度安排及任务要求：**

**起讫日期 2017年11月20日至 2018年6月8日**

**指导教师**（**签名） 职称**

**三、系或研究所审核意见:**

**负责人**（**签名）**

**毕 业 设计 考 核**

**一、指导教师对毕业设计的评语：**

**指导教师(签名）**

**年 月 日**

**二、答辩小组对毕业设计的答辩评语及总评成绩：**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **成绩**  **比例** | **文献综述**  **占（10%）** | **开题报告**  **占（15%）** | **外文翻译**  **占（5%）** | **毕业设计质量及答辩**  **占（70%）** | **总评成绩** |
| **分值** |  |  |  |  |  |

**答辩小组负责人（签名）**

**年 月 日**