****

**本科毕业设计**

**（2018届）**

|  |  |
| --- | --- |
| **题 目** | **个性化新闻推荐系统** |
| **学 院** | 计算机学院 |
| **专 业** | 物联网工程 |
| **班 级** | 14058911 |
| **学 号** | 14051336 |
| **学生姓名** | 张慧 |
| **指导教师** | 李珣 |
| **完成日期** | 2018 年 6 月 |

**摘要**

**ABSTRACT**

1. 引言

1.1项目背景

1.2发展历程

1.3研究现状

1.4本人主要工作

1.5文本组织结构

2. 相关技术

本结主要是对开发个性化新闻推荐系统相关技术的介绍。

2.1 SSM框架

SSM框架主要是指的spring、springMVC、mybatis三大框架整合的框架。

2.1.1 spring框架

Spring是一个开源框架，Spring是于2003 年兴起的一个轻量级的Java 开发框架，由Rod Johnson创建。

Rod Johnson在2002年编著的《Expert one on one J2EE design and development》一书中，对Java EE 系统框架臃肿、低效、脱离现实的种种现状提出了质疑，并积极寻求探索革新之道。以此书为指导思想，他编写了interface21框架，这是一个力图冲破J2EE传统开发的困境，从实际需求出发，着眼于轻便、灵巧，易于开发、测试和部署的轻量级开发框架。Spring框架即以interface21框架为基础，经过重新设计，并不断丰富其内涵，于2004年3月24日，发布了1.0正式版。

Spring框架可以说是Java开发人员使用的最流行的应用程序开发框架之一。它目前由大量提供一系列服务的模块组成。包括模块容器，为构建横切关注点提供支持的面向切面编程（AOP），安全框架，数据存取框架，Web应用框架和用于模块测试提供支持的类。Spring框架的所有组件都通过依赖注入粘在一起。依赖注入（也称为控制反转）使设计和测试松散耦合的软件模块变得更容易。 

图1 spring框架总览

2.1.2 SpringMVC框架

springMVC是spring系列中的一个开源项目。Spring MVC 是一个模型-视图-控制器（MVC）的Web框架建立在中央前端控制器servlet（DispatcherServlet），它负责发送每个请求到合适的处理程序，使用视图来最终返回响应结果的概念。Spring MVC是Spring产品组合的一部分，它享有 Spring IoC容器紧密结合Spring松耦合等特点，因此它有Spring的所有优点。

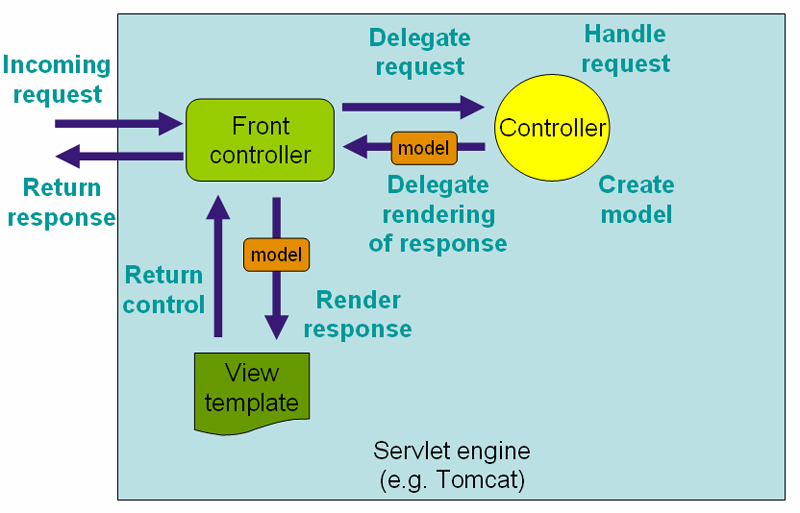


图2 springMVC框架

2.1.3 Mybatis框架

MyBatis的前身是apache的一个开源项目iBatis，2010年这个项目由apache software foundation 迁移到了google code，并且改名为MyBatis。MyBatis是一个数据持久层(ORM)框架。

MyBatis 是支持普通 SQL查询，存储过程和高级映射的优秀持久层框架。MyBatis 消除了几乎所有的JDBC代码和参数的手工设置以及结果集的检索。MyBatis 使用简单的XML或注解用于配置和原始映射，将接口和 Java 的POJOs（Plain Old Java Objects，普通的 Java对象）映射成数据库中的记录。每个MyBatis应用程序主要都是使用SqlSessionFactory实例的，一个SqlSessionFactory实例可以通过SqlSessionFactoryBuilder获得。SqlSessionFactoryBuilder可以从一个xml配置文件或者一个预定义的配置类的实例获得。

2.2 存储相关

2.2.1 MySQL数据库

MySQL 是最流行的关系型数据库管理系统，在WEB应用方面 MySQL 是最好的RDBMS(Relational Database Management System：关系数据库管理系统)应用软件之一。

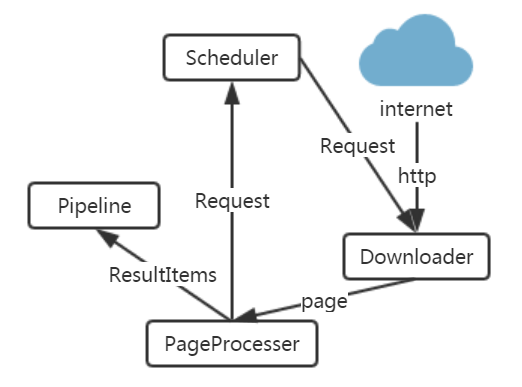
2.2.2 Mongodb数据库

MongoDB是当前最流行的Nosql数据库之一。 MongoDB 是由C++语言编写的，是一个基于分布式文件存储的开源数据库系统。在高负载的情况下，添加更多的节点，可以保证服务器性能。MongoDB 旨在为WEB应用提供可扩展的高性能数据存储解决方案。MongoDB 将数据存储为一个文档，数据结构由键值(key=>value)对组成。MongoDB 文档类似于 JSON 对象。字段值可以包含其他文档，数组及文档数组。

2.3爬虫技术

爬虫使用webmagic框架。

WebMagic是一个简单灵活的Java爬虫框架。WebMagic的结构分为Downloader、PageProcessor、Scheduler、Pipeline四大组件，并由Spider将它们彼此组织起来。这四大组件对应爬虫生命周期中的下载、处理、管理和持久化等功能



2.4 相关框架

2.4.1任务调度框架Quartz

Quartz是OpenSymphony开源组织在Job scheduling领域又一个开源项目，是完全由java开发的一个开源的任务日程管理系统，“任务进度管理器”就是一个在预先确定（被纳入日程）的时间到达时，负责执行（或者通知）其他软件组件的系统。

Quartz由Scheduler（任务调度器）、Trigger（触发器）、Calendar（一些日历特定时间点的集合）、JobDetail（描述Job实现类及其它相关的静态信息）、Job（开发者实现该接口定义运行任务）五个核心元素构成。Quartz 支持丰富多样的调度方法，可以满足各种常规及特殊需求；支持任务和调度的多种组合方式，支持调度数据的多种存储方式等优点。Quartz 很容易与 Spring 集成实现灵活可配置的调度功能。



图 核心类关系图

2.4.2 消息队列框架ActiveMQ

ActiveMQ是一种开源的，实现了JMS1.1规范的，面向消息(MOM)的中间件，为应用程序提供高效的、可扩展的、稳定的和安全的企业级消息通信。ActiveMQ使用Apache提供的授权，任何人都可以对其实现代码进行修改。

ActiveMQ 支持两种截然不同的消息传送模型：PTP（即点对点模型）和Pub/Sub（即发布 /订阅模型），分别称作：PTP Domain 和Pub/Sub Domain。

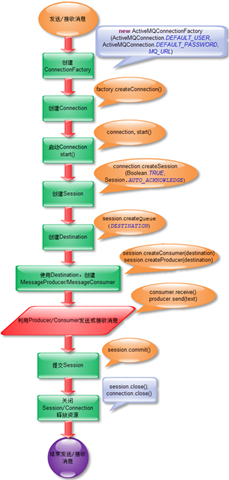


图 ActiviteMQ接收和发送消息基本流程

2.5 前端技术

1.AngularJS介绍

AngularJS 是一款来自 Google 的前端 JavaScript 框架，也是 SPA（single-page-application，单页应用）框架。AngularJS 框架的体积非常小，但是设计理念和功能却非常强大，极大地简化前端开发的负担，它快速成为了 JavaScript 的主流框架，帮助开发者从事 web 开发。

AngularJS有五个主要核心特性如下：

①双向数据绑定：实现了把model与view完全绑定在一起，model变化，view也变化，反之亦然。

②模板：在AngularJS中，模板相当于HTML文件被浏览器解析到DOM中，AngularJS遍历这些DOM，也就是说AuguarJS把模板当做DOM来操作，去生成一些指令来完成对view的数据绑定。

③MVVM：吸收了传统的MVC设计模式针但又并不执行传统意义上的MVC，更接近于MVVM(Moodel-View-ViewModel)。

④依赖注入：AngularJS拥有内建的依赖注入子系统，可以帮助开发人员更容易的开发，理解和测试应用。

⑤指令：可以用来创建自定义的标签，也可以用来装饰元素或者操作DOM属性。

2.6 本章小结

本章旨在介绍开发过程中使用的相关技术与框架。本章介绍了后端使用的SSM框架、Quartz、WebMagic、ActiveMQ以及前端使用的AngularJs、Angular Material 技术等。

3. 推荐算法研究

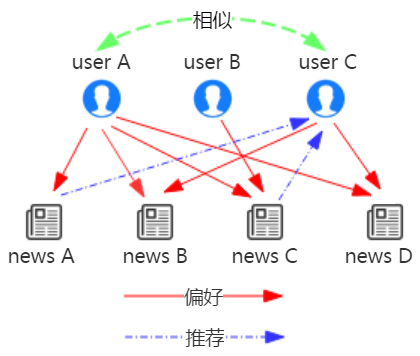
个性化新闻推荐系统的核心是推荐引擎。现阶段推荐算法已经相当成熟，本结主要是对比较流行的协同过滤与SVD推荐算法研究介绍。

3.1 基于协同过滤(Collaborative Filtering)的推荐

协同过滤推荐算法是诞生最早，并且较为著名的推荐算法。主要的功能是预测和推荐。协同过滤可简单的分为两类，分别是基于用户的协同过滤算法(user-based collaboratIve filtering)，和基于物品的协同过滤算法(item-based collaborative filtering)。

3.1.1 基于用户的协同过滤算法（userCF）

基于用户的协同过滤算法根据某一用户偏好找到与该用户偏好相似的用户集，然后预测该用户相对相似用户集的未评分项的偏好值，最后根据定义的过滤规则推荐给该用户相关未评分项。



以个性化新闻推荐举例说明：

设用户新闻偏好矩阵（user-item）为。众所周知在计算机的世界只有数字，因此用户对某一新闻的偏好只能用数字表示，数字的大小表示了该用户对该新闻的喜好程度。如下表所示

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| News  Score  User | News1 | News2 | News3 | News4 | News5 | …… |  |  |
| User1 | 0 | 1 | 2 | 2 | 5 | …… | 0 | 5 |
| User2 | 1 | 1 | 1 | 0 | 5 | …… | 4 | 3 |
| User3 | 2 | 3 | 3 | 3 | 3 | …… | 4 | 4 |
| ……… | …… | …… | …… | …… | …… | …… | …… | …… |
|  | 2 | 2 | 2 | 0 | 0 | …… | 0 | 2 |
|  | 3 | 2 | 1 | 0 | 0 | …… | 5 | 5 |

上表中用户对于某一条新闻的评分可根据用户相应的历史行为如点击、阅读、收藏、评论等给出，开发者应该合理的制定打分规则，以便真实表达出用户对该新闻的偏好程度。表中0表示该用户对该新闻无任何正反馈行为，即不知道用户对该新闻的偏好程度，默认对某一条新闻无任何反馈行为的为0，且不能为负值；

基于用户的协同过滤算法的核心是找出相似用户。计算两个用户的相似度的方法有多种比如欧几里得距离（Eucledian Distance）、余弦相似度（Cosine Similarity）、调整余弦相似度(Adjusted Cosine Similarity)、皮尔森相关系数(Pearson Correlation Coefficient)、K-近邻算法(K-neighborhood)、曼哈顿距离（Manhattan Distance）、明可夫斯基距离（Minkowski distance）、杰卡德相似性（Jaccard Similarity）等方式。这里我们不对相似度计算算法做过多赘述，我们选取调整余弦相似度(Adjusted Cosine Similarity)作为举例说明。调整余弦相似度(Adjusted Cosine Similarity)公式如下：

调整余弦相似度(Adjusted Cosine Similarity)是由余弦相似度(Adjusted Cosine Similarity)变化而来。计算时U表示用户的评分集即, ,;

表示用户u对的评分，表示用户u评分集的平均值，同理。

计算出用户之间的相似度后，我们得到了相似度由高到低的N个用户集，我们选取其中前K个用户作为预推荐相似用户即邻居。

最后推荐时我们是从已选出的K个相似用户集中找到推荐用户为阅读的N条新闻推荐给用户。由于K个相似用户中可能或多于或少于N条推荐新闻所以我们需要预测评分。以上面使用的调整余弦相似度(Adjusted Cosine Similarity)为前提，预测评分有加权和回归两种，这里采用加权平均法举例，公式如下：

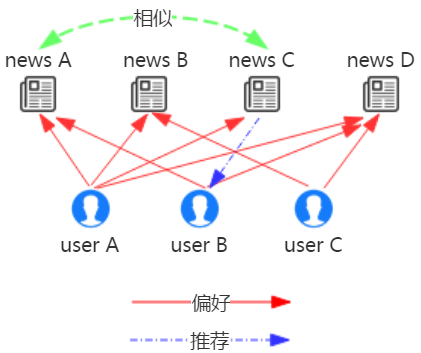
表示用户u对未评分的的预测评分，表示用户的相似度，表示用户对的实际评分值。

由上面的举例知协同过滤算法的一般步骤如下：



3.1.1 基于项的协同过滤算法（itemCF）

基于项的协同过滤算法是根据用户喜欢的物品与用户未知的物品的相似度，推荐给与用户喜欢的物品最相似的物品。



基于项的协同过滤算法的计算与基于用户的协同过滤算法大致相同。举例说明如下：

输入数据与上面基于用户的协同过滤算法相同，计算相似度时以上面的调整余弦相似度(Adjusted Cosine Similarity)为例

计算时表示新闻项，U表示新闻的评分集即，,；表示用户对新闻的评分，表示新闻的平均评分，同理。

预测评分时同理可采用回归或加权，但是计算向量都必须是新闻向量。

3.2基于SVD的推荐算法

奇异值分解(Singular Value Decomposition，简称SVD)是在机器学习领域广泛应用的算法，它不仅可以用于降维算法中的特征分解，还可以用于推荐系统，以及自然语言处理等领域。

3.2.1特征值

定义设**A**是n阶方阵，如存在数λ和n维非零向量**X，**使关系式

成立，则称数λ为方阵**A**的**特征值；**非零向量**X**称为**A**的对应于特征值**λ**的**特征向量**

解特征值与特征向量可以使用公式构建特征方程组。

定理 n阶矩阵**A**与对角矩阵相似的充分必要条件是**A**有n个线性无关的特征向量

当n阶矩阵**A**有n个不同特征值时**A**一定可相似对角化即 **=>**

由上知如果我们求出n阶矩阵**A**的n个特征值以及n个特征值所对应的特征向量，那么矩阵**A**可表示如下

其中**W**是n个特征向量所组成的阶矩阵，**Σ**是n个特征值为对角线的阶矩阵。当**W**为正交矩阵时即**，**因此。

3.2.2 SVD

从上面的特征值知我们可以将一个n阶矩阵**A**分解为3个矩阵相乘的形式。但对于一个阶矩阵特征值的方式去不行，因此就有了SVD。SVD是对特征值分解的扩展，它适用于任意阶矩阵。

SVD公式如下

其中都是正交矩阵。

求解三个矩阵步骤如下

1. 求出，；
2. 计算的m个特征值与m个单位正交特征向量，得到由所有特征向量组成的的左奇异向量。
3. 计算的n个特征值与n个单位正交特征向量，得到由所有特征向量组成的矩阵的转置矩阵，的右奇异向量。
4. 计算奇异值σ，因为所以即所以

（注特征值由大到小一一对应）

计算举例如下：

设矩阵则

；

的特征值与特征向量为

；

的特征值与特征向量为；

求奇异值利用

得奇异值分解如下

3.2.3奇异值分解SVD推荐算法

SVD中前10%甚至1%的奇异值的和就占了全部的奇异值之和的99%以上的比例，因此采用近似的描述原矩阵即如下公式：

SVD推荐算法的基本原理是我们计算出奇异值然后取k个奇异值在与左右奇异向量相乘最后得到与原矩阵相似的矩阵，得到的相似矩阵会出现原矩阵未评分的项出现了打分，它可以看成是预测出的，为了使这个预测的分数接近真实值我们需要将相似矩阵与原矩阵的已评分项进行比较，使用某种方法使相似矩阵与原矩阵的已评分项的误差减小，当误差减小到一定程度我们默认相似矩阵与原矩阵已无差别因此相似矩阵中出现的预测评分被认为是用户的真实评分，最后根据这些预测评分推荐给用户合理的项。

SVD推荐算法的核心是预测。SVD推荐的衍生算法有Basic SVD、RSVD、ASVD、SVDPP（SVD++），这些算法都旨在解决预测中的各种问题，预测中问题解决这里不做赘述。

3.3奇异值分解SVD与协调过滤的结合

3.4本章小结

3. 需求分析

3.1需求概述

3.2功能性需求

3.2.1新闻处理功能性需求分析

3.2.2日志分析功能性需求分析

3.2.3推荐处理功能性需求分析

3.2.4用户行为功能性需求分析

3.2.5消息处理功能性需求分析

3.3数据字典

3.3.1新闻处理数据流条目

3.3.2相似度计算数据流条目

3.3.3标签计算数据流条目

3.3.4推荐处理数据流条目

3.3.5消息处理数据流条目

3.4非功能性需求

3.5本章小结

4. 系统概要设计

4.1系统技术架构

4.2系统功能模块概述

4.3新闻处理模块概要设计

4.3.1实现目标

4.3.2新闻处理模块功能结构

4.3.3数据存储

4.4日志分析模块概要设计

4.4.1实现目标

4.4.2日志分析模块功能结构

4.4.3数据存储

4.5推荐处理模块概要设计

4.5.1实现目标

4.5.2推荐处理模块功能结构

4.5.3数据存储

4.6用户行为模块概要设计

4.6.1实现目标

4.6.2用户行为模块功能结构

4.6.3数据存储

4.7消息处理模块概要设计

4.7.1实现目标

4.7.2消息处理模块功能结构

4.7.3数据存储

4.8本章小结

5. 系统详细设计与实现

5.1新闻处理模块详细设计与实现

5.2日志分析模块详细设计与实现

5.2.1用户相似度分析功能

5.2.2用户标签计算功能

5.3推荐处理模块详细设计与实现

5.3.1用户相似度推荐功能

5.3.2用户标签推荐功能

5.4用户行为模块详细设计与实现

5.5消息处理模块详细设计与实现

5.6本章小结

6. 系统测试

6.1单元测试

6.2压力测试

6.3本章小结

7. 总结与展望

7.1总结

7.2展望

参考文献

图表目录

源码附录

源代码已上传github地址：https://github.com/MaloneRe/graduate.git