

基于 Spark 的智能菜品推荐系统设计与实现

周杨玥, 李世锋, 李 林

(黄河交通学院, 河南 焦作 454000)

✉ yangzhoumxn@163.com; 1006061024@qq.com; 18839508953@163.com



摘 要: 为解决人们“每天不知道吃什么”的烦恼,采用 Spark 分布式处理框架,结合分布式存储数据库(MongoDB)、日志收集系统(Flume)、分布式系统文件(HDFS)等工具,实现对菜品实时评分及特色化推荐。系统包含菜品评分及储存模块、评分数据处理模块、菜品推荐模块、推荐结果展示模块等,其中推荐模块采用协同过滤推荐算法。使用测评方法和指标验证 3 种推荐模型的有效性,根据测评结果设计并实现以基于物品的推荐模型为主、以基于 Spark ALS 的推荐模型为辅的智能菜品推荐系统。该系统能够快速准确地推荐顾客喜欢的菜品,提高了商家的服务效率和顾客的满意度,可用性较高。

关键词: 菜品推荐; Spark; 协同过滤算法; HDFS

中图分类号: TP311 **文献标志码:** A

Design and Implementation of Intelligent Dish Recommendation System Based on Spark

ZHOU Yangyue, LI Shifeng, LI Lin

(Huanghe Jiaotong University, Jiaozuo 454000, China)

✉ yangzhoumxn@163.com; 1006061024@qq.com; 18839508953@163.com

Abstract: To solve the problem of people not knowing what to eat every day, this paper proposes an intelligent dish recommendation system by using the Spark distributed processing framework and tools such as MongoDB (a distributed storage database), Flume (a log collection system), HDFS (Hadoop Distributed File System), etc., to achieve real-time rating and personalized recommendation of dishes. The system includes a dish rating and storage module, rating data processing module, dish recommendation module, and recommendation result display module, etc. The recommendation module adopts a collaborative filtering recommendation algorithm. Evaluation methods and indicators are used to validate the effectiveness of three recommendation models. Based on the evaluation results, the intelligent dish recommendation system is designed and implemented, which is primarily based on item-based recommendation models and supplemented by Spark ALS-based recommendation models. The system can quickly and accurately recommend dishes that users like so as to improve the business service efficiency and customer satisfaction, with high availability.

Key words: dish recommendation; Spark; collaborative filtering algorithm; HDFS

0 引言(Introduction)

随着社会信息化水平的不断提高,大数据技术越来越多地被运用到人们的生产和生活中。基于互联网的智能推荐系统

在人们的生活中扮演越来越重要的角色^[1]。以往在点菜过程中,顾客大多是通过服务人员介绍的方式选择菜品,耗费的时间较长,灵活程度不高,受服务人员主观因素的限制较大,占据

较多的服务资源,而且不能精准地把握客人的喜好,因此有必要借助计算机和大数据技术完成对菜品的合理推荐。

餐饮企业通过信息技术和设备实现对菜品信息的管理,可以提高工作效率,也能更加精准地把握顾客对菜品的喜好。此外,还可以设计一款智能菜品推荐系统,既能让顾客品尝到自己心仪的美味,又节省了选菜时间,提高了商家的服务效率和顾客的满意度。

1 系统概要设计(System outline design)

1.1 总体结构设计

以通过系统实现菜品的智能推荐为研究对象进行分析与设计,旨在抓住顾客的喜好和提高顾客满意度以及减少选菜时间。餐饮推荐系统采集到的评分数据存储在 HDFS 上,使用推荐模块中的推荐模型输出符合要求的推荐数据。餐饮推荐系统结构图如图 1 所示。

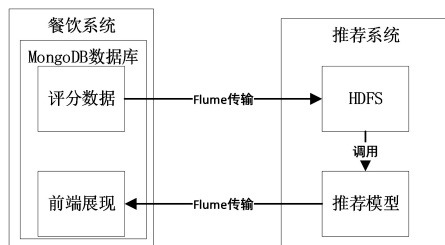


图 1 餐饮推荐系统结构图

Fig. 1 Structure diagram of catering recommendation system

1.2 用户功能

(1)用户注册:用户进行账号注册时,系统会要求用户填写邮箱和密码,之后系统会验证这些信息的有效性,并将是否已注册成功的结果返回给用户,注册成功则回到登录页面。

(2)用户登录:用户在登录页面填写真实的邮箱和密码,由系统与自身数据库进行比对与校验。校验通过后,将自动跳转到推荐网页,否则将被提示校验失败。

(3)菜品搜索:用户可以在网页顶部的搜索框中搜索想要品尝菜品的名称或者关键词,系统会根据这些关键词查找相应的菜品,并将查找到的结果返回给用户。

(4)菜品评分:用户查看菜品的平均评分以及菜品推荐信息,评价分值的范围是 0~5 分。

(5)查看菜品:用户可以查看推荐的菜品、爆款菜品以及所有菜品。

(6)退出登录:用户点击“退出”,就会从已存在的账号中退出,并自动进入登录网页。若想再次进入系统,就得重复登录操作。

1.3 推荐功能及方案设计

(1)推荐功能是根据菜品评分,通过推荐算法向用户推荐菜品信息。为提高用户的交互体验,推荐模块中包含了爆款数据。

(2)详情页推荐是在用户感兴趣菜品的基础上再次进行推送。进行符合用户喜好的推送,往往会达到更好的推荐效果。

推荐系统的核心是推荐算法,它是整个系统的根基,推荐算法的性能好坏直接关系到系统的运行状态。目前,主要有基

于内容、关联规则、知识推荐以及协同过滤等已经广泛应用的推荐算法。

1.3.1 基于内容的推荐

基于内容的推荐可以在不依赖其他用户信息的情况下,比较准确地识别用户兴趣,信息直观易懂,具有较强的解释性。只要用户在网上发生了浏览记录或购买行为,系统就可以根据已知项目的特征标签推测用户喜欢的物品。这种方法依赖于项目标记特征,因此系统推荐的商品不能保证产品的质量,并且推荐范围狭窄、缺乏新颖性,获取内容来源和特征提取相对困难,因此相对小众^[2]。

基于内容的推荐方法通常会建立一个用户兴趣模型和一个项目特征模型,并使用相似度度量衡量两种模式之间的相似性,然后通过相似度排序,由高到低地向目标用户推荐物品。基于内容的推荐通常会忽略用户行为,只考虑物品之间的相似性。因此,与协同过滤推荐算法相比,它可以解决协同过滤算法中一些不可避免的问题,如冷启动问题、评分数据稀疏问题等。此外,基于对物品内容的分析,它可以提供推荐物品的内容特征,并向用户提供推荐的理由^[3]。

1.3.2 协同过滤推荐

协同过滤算法是目前最成熟、使用最早、应用最广泛的一种推荐方法,它通常采用最近邻技术,获取用户与物品的共现矩阵,以用于计算用户间或物品间的距离,并使用已被确定为最偏好用户的项目,然后根据最接近目标用户的项目得分权重对该产品进行预测。尽管协同过滤技术是目前最具代表性的推荐方法之一,但其也存在不少缺点,典型的问题有稀疏问题和可扩展问题等^[4]。

1.3.3 基于知识推荐

在某些特定的场景,使用者的行为会在很短的时间内发生,例如购买电脑后,会持续使用此种电子产品,通常几年内不会再次购买,因此在此类场景下,基于用户过去的行为是无法进行推荐的。基于知识的推荐则可以根据用户及产品的信息,得出哪种产品应该推荐给哪个用户,从而完成推荐^[5]。

1.3.4 主要推荐方法对比

各种推荐方法的优点和缺点如表 1 所示。

表 1 各种推荐方法的优点和缺点

Tab.1 Advantages and disadvantages of various recommendation methods

推荐方法	优点	缺点
基于内容推荐	结果直观,而且不需要具备太多的专业知识	稀疏问题是新用户问题中难以解决的属性,要有充足的数据进行分类
协同过滤推荐	不需要事先对商品或用户进行分类或标注,适用于各种类型的数据;算法简单易懂,容易实现和部署;推荐结果准确性较高,能够为户提供个性化的推荐服务	存在数据量和数据质量问题、稀疏问题、可扩展性问题、新主体问题等,推荐效果取决于历史数据
基于规则推荐	能发现新兴趣点,不需要具备领域知识	规则提取困难的商品名同义性问题在个体上表现得不够个人化

续表

推荐方法	优点	缺点
基于效用推荐	非冷启动和稀疏问题对于使用者需要输入的是一个客户喜好的改变很灵敏,静态的效用功能,并且存在以将非产品特性考虑在内	较少的属性重叠问题
基于知识推荐	能够将使用者的需要与产品相对应,能够将非产品的特性考虑在内	知识难获得,推荐是静态的

综合以上各种推荐方法的优点和缺点,考虑到新兴趣点发现与推荐个性化方面的表现,本设计选择协同过滤推荐方法。

1.4 数据库设计

系统中主要包含的对象为用户和菜品,除用户和菜品的基本属性外,还有两者间产生的交互属性,需要使用数据库存储。系统对数据的存储主要使用 MongoDB,存储用户信息表(User)和菜品信息表(Menu),以及用户的日志数据,其中日志数据以 json 的形式进行存储,在用户信息表中可以查到相应的信息,如表 2 所示;在菜品信息表中可以查到相应的信息,如表 3 所示。

表2 用户信息表

Tab.2 User information table

Key	Types	Comment
_id	ObjectID	
userid	Int32	
username	String	用户名
Password	String	用户密码
Email	String	邮箱
First	Bool	
Timestamp	Int64	
prefGenres	Array	

表3 菜品信息表

Tab.3 Dish information table

Key	Types	Comment
_id	ObjectID	
Name	String	菜品名
Sco	Number	平均评分
Dish	String	菜品用料
lab	String	标签

2 推荐系统实现 (Implementation of recommendation system)

2.1 数据采集

本系统需要与菜品网站数据库结合使用,因此数据来源分为两个部分,一部分从黄河交通学院东、西两个校区(4个年级的学生、餐饮公司负责人、学校后勤人员、教师、校外商家等)收集真实有效的数据,另一部分通过爬虫技术爬取“美团”数据采集原数据,以 json 格式存储,原数据如图 2 所示。

2.2 数据预处理

原始数据中存在很多异常数据,需要对其进行异常数据处理。为了便于存储以及加快数据处理效率,需要对原始的数据格式进行编码处理。将数据集划分为训练集、校验集和测试

```
1 {"UserID": "A2M0H395HGSOT", "Rating": 5.0, "ReviewTime": 1496177056, "MealID": "B0040HNTZW"}
2 {"UserID": "A32KH50VW0N0HB", "Rating": 3.0, "ReviewTime": 1496177108, "MealID": "B006Z48TZS"}
3 {"UserID": "A1VQ4Z5U9NIGP", "Rating": 5.0, "ReviewTime": 1496177276, "MealID": "B00C0BTQCH"}
4 {"UserID": "A3E5V5T5AY3R9", "Rating": 4.0, "ReviewTime": 1496179256, "MealID": "B00751IYQ4"}
5 {"UserID": "A1V50CTTD373ZM", "Rating": 5.0, "ReviewTime": 1496180009, "MealID": "B00C0OLT6S"}
6 {"UserID": "A3G0BQR550A608", "Rating": 5.0, "ReviewTime": 1496180056, "MealID": "B006HC63X8"}
7 {"UserID": "A8662ZYJVQ42D", "Rating": 5.0, "ReviewTime": 1496180098, "MealID": "B0099JKR6U"}
8 {"UserID": "A2FPYIU94BNC73", "Rating": 5.0, "ReviewTime": 1496182040, "MealID": "B0007VWHNT"}
9 {"UserID": "A2M0H395HGSOT", "Rating": 5.0, "ReviewTime": 1496183065, "MealID": "B002A493NY"}
10 {"UserID": "ANOLNXVIXZKKH", "Rating": 3.0, "ReviewTime": 1496184000, "MealID": "B002KGEORM"}
11 {"UserID": "A1VQ3C5KL50MGV", "Rating": 5.0, "ReviewTime": 1496184000, "MealID": "B0045TGR5G"}
12 {"UserID": "A2FPYIU94BNC73", "Rating": 3.0, "ReviewTime": 1496184000, "MealID": "B007335X58"}
13 {"UserID": "AMXJDI60YD39", "Rating": 5.0, "ReviewTime": 1496184000, "MealID": "B006Z48TZS"}
```

图2 原数据

Fig. 2 Raw data

集,从而对推荐模型进行评估。因为系统开发处于探索与分析阶段,所以选择在 Spark Shell 环境下进行编程实现。

2.2.1 异常值数据处理

对重复评分的数据进行统计,并且使用时间戳对重复数据进行判断,最新的评分被认定为该用户对菜品的最终评分,删除旧数据,实现异常值处理(图 3)。

```
scala> repeaterRatingList.select("userid","mealid","rating","reviewtime").show(10)
+-----+-----+-----+-----+
|userid|mealid|rating|reviewtime|
+-----+-----+-----+-----+
|A1029QPGAKN80T|B000QISQX6|5.0|1497465600|
|A1029QPGAKN80T|B000QISQX6|3.0|1496961600|
|A10410535ID37WNBGTT8|B004AUGJ58|5.0|1493664000|
|A10410535ID37WNBGTT8|B004AUGJ58|5.0|148752000|
|A108QNVQPC1W2|B003NRWVMC|5.0|1494099600|
|A108QNVQPC1W2|B003NRWVMC|5.0|1486889600|
|A108QNVQPC1W2|B009AP2626|3.0|1487120000|
|A108QNVQPC1W2|B009AP2626|5.0|14966616000|
|A10CJIMP7KBR4W|B000MPC68|5.0|1496326400|
|A10CJIMP7KBR4W|B000MPC68|3.0|1495822400|
only showing top 10 rows
```

图3 异常值处理

Fig. 3 Exception handling

2.2.2 数据变换处理

数据变换是将文件内容转换成系统需要的格式以便于下一步调用。由于用户和菜品的 ID 都是以字符串的形式表示,存储空间较大,并且在调用计算时效果差、效率低,因此将其转换为数值类型。将用户和菜品的 ID 编码都转换为 Integer 类型。编码步骤如下:(1)对用户数据与菜品数据进行去重后,再进行排序;(2)使用排序后的原始用户与菜品的下标值代替该用户或菜品;(3)使用编码后的值替换原始数据中的值。

编码后的用户评分数据格式为(用户编码,菜品编码,评分,评分时间戳),编码数据图如图 4 所示。

```
1 (236,89,5.0,1493576000)
2 (4882,42,2.0,1493576000)
3 (1753,74,4.0,1493576000)
4 (2764,152,3.0,1493576000)
5 (4866,162,5.0,1493576000)
6 (867,195,5.0,1493576000)
7 (2983,588,4.0,1493592000)
8 (2498,701,4.0,1493592000)
9 (815,940,4.0,1493592000)
```

图4 编码数据图

Fig. 4 Encoding data graph

2.2.3 数据集分割

在对数据集进行编码后,再将其进行分割、存储,将训练集、校验集、测试集按时间戳的次序进行分割。原数据与编码之后的数据全部存储在 HDFS 中。

2.3 推荐模型建立

系统选取了分别基于用户、项目和 Spark ALS 的 3 种不同的协作过滤算法,建立了相应的推荐模型并进行了比较,对推荐模型进行总体评价后,再决定选择哪一个模型。

2.3.1 以基于用户的协同过滤算法建模

基于用户的协同过滤算法是根据消费者对相同产品的不同评价判定其是否具有相似度,从而找到最接近目标商户的用

户,然后将评价最高的商品按距离最近的商户推送给目标客户。

通过每个用户的得分向量统计用户间的接近程度。通过统计各个用户间的相似程度,能够找出与目标用户相似程度较大的用户,进而按照相似程度得分对这些用户进行排列,从而得出前 k 个近邻用户。给目标用户介绍前 k 个近邻用户的高分项目,算法流程图如图 5 所示。

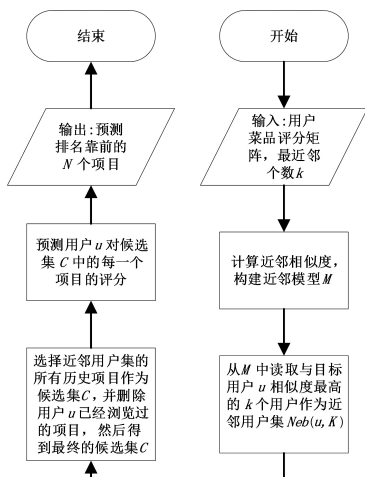


图 5 算法流程图

Fig. 5 Algorithm flow chart

2.3.2 以基于菜品的协同过滤算法建模

在计算过程中,基于物品的推荐方法和基于用户的推荐方法类似,但不同之处在于基于物品的推荐方法是通过项目之间的相似性推荐与目标用户喜欢的物品相似的物品^[6],其基本思想是消费者对某一产品的预期评估可以基于其对某一产品的加权平均数,计算得出相似性最大的 k 个近邻商品的评价,基于此建立本系统的菜品推荐模型。

2.3.3 以基于 Spark ALS 的协同过滤算法建立模型

交替最小二乘法 (Alternating Least Squares, ALS) 属于 Spark 机器学习组件 Spark MLlib 的内置算法,是 Spark MLlib 中最重要的算法,也是协同过滤算法的重要思想^[7]。ALS 的基本思想是利用用户对各类美食进行评分获得的矩阵,在用户群中寻找高度相似的用户群,然后综合这些用户对同一信息的评价,预测指定用户对此信息的喜好程度^[8],用 $P(m \times k)$ 和 $Q(n \times k)$ 的积近似表达 $m \times n$ 的得分矩阵 R ,然后不断最优优化两个小矩阵,最大化地还原原始评分矩阵 R ,从而填充评分缺失值,降低数据的稀疏性,提高算法的推荐质量^[9]。这样,就得到了一个基本模型,利用这个模型可以对未出现的用户或者物品进行评分预测^[10]。ALS 矩阵分解图如图 6 所示。

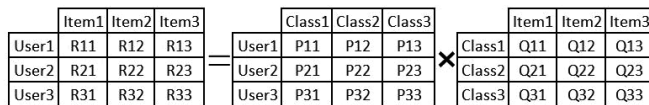


图 6 ALS 矩阵分解图

Fig. 6 ALS matrix decomposition diagram

采用基于用户、基于菜品、基于 Spark ALS 的 3 种不同的协同过滤算法建立推荐模型,其中基于用户和基于菜品的算法

是采用直接编码的方法来完成的,这两种算法虽然简单直观,但是在处理大数据集时,资源占用较大^[11]。与其他算法比较,Spark ALS 算法内置在 Spark MLlib 中,不用通过编码实现,因此调用速度快、效率高。而且,由于 Spark ALS 算法在矩阵计算时应用了降维处理,因此在相同的计算条件下可以处理更大的数据量。

2.4 推荐模型的评测

对建立的模型进行评测,有利于更好地选择模型,并能够提高推荐准确度。因为模型是通过离线数据集构建的,所以本研究选择离线测试。离线测试是根据真实的菜谱系统对各个算法进行评分,再利用所建立的不同的推荐算法对其进行评估。该方法不需要真正的使用者参与,方便了测试过程。

测评指标:由于不同推荐模式的样本集呈现的评价指数不同,根据不同的呈现模式,其测评指标公式如表 4 所示。

表 4 测评指标表

Tab.4 Evaluation index table

数据表现方式	指标 1	指标 2	指标 3
预测准确度	$RMSE = \sqrt{\frac{1}{N} \sum (r_{ui} - \hat{r}_{ui})^2}$	$MAE = \frac{1}{N} \sum r_{ui} - \hat{r}_{ui} $	—
分类准确度	$P = \frac{TP}{TP + FP}$	$R = \frac{TP}{TP + FN}$	$F1 = \frac{2PR}{P + R}$

主要评价指标为均方根误差 (RMSE) 和平均绝对误差 (MAE)。其中, r_{ui} 为用户 u 在项目 i 上的真正分数, $\hat{r}_{ui}$ 为用户 u_i 的预测分数, N 为实际参与评分的项目的总分。RMSE 的值越小,说明推荐结果越接近实际评分值。MAE 的值越小,说明推荐算法的准确度越高^[12]。RMSE 和 MAE 为评价推荐质量提供了直观的度量方法,是两种比较常用的评估模型^[13]。

精确率 (P) 表示推荐物品被用户认可的程度和准确度;召回率 (R) 表示消费者想要某一种产品的程度; $F1$ 指数综合考虑精确率与召回率,能够更客观地评估推荐方法的优点和缺点 ($F1$ 值越大,则该方法越优)。其公式如公式 (1) 至公式 (3) 所示,分类准确度指标表如表 5 所示。

$$\text{精确率: } P = TP / (TP + FP) \quad (1)$$

$$= \text{正样本预测结果数} / \text{推荐物品数}$$

$$\text{召回率: } R = TP / (TP + FN) \quad (2)$$

$$= \text{正样本预测结果数} / \text{正样本实际数}$$

$$\text{综合指数: } F1 = 2PR / (P + R) \quad (3)$$

$$= \text{准确率与召回率的综合评价指标}$$

表 5 分类准确度指标表

Tab.5 Classification accuracy index table

预测表现	推荐物品数 (正)	未被推荐物品数 (负)	合计
实际表现			
用户实际喜欢物品数 (正)	TP	FN	$TP + FN$
用户不喜欢物品数 (负)	FP	TN	$FP + TN$
合计	$TP + FP$	$TN + FN$	—

由于基于用户和物品的协同过滤算法的预测结果并没有直接预测出评分,因此根据实际情况选用以上两种测评方式中

的分类器评估法,通过此测评方法和指标对三种推荐模型进行测评,结果如下。

(1)对基于用户的推荐模型进行评测,加载推荐模型与测试数据集,为每个用户选取 k 个推荐结果,与测试数据集的实际记录进行匹配。

(2)对基于菜品的推荐模型进行评测,首先加载推荐模型与测试数据集,为每个用户选取 k 个推荐结果;其次与测试数据集的实际记录进行匹配。

(3)对基于 Spark ALS 的推荐模型进行评测,首先加载推荐模型与测试数据集,为每个用户选取 k 个推荐结果;其次与测试数据集的实际记录进行匹配。

对评测结果进行对比,选出适合的算法模型,评测结果对比表如表 6 所示。

表6 评测结果对比表

Tab.6 Comparison table of evaluation results

算法模型	建模参数	模型评价参数	k 值	R	P	$F1$
基于用户的协同过滤算法模型	单用户评价过的最小物品数=2	单用户评价过的最小物品数=2	10	6.5	1.7	2.7
			20	11.1	1.4	2.5
			30	14.8	1.3	2.4
			40	18.2	1.2	2.2
			50	21.0	1.1	2.1
基于物品的协同过滤算法模型	单用户评价过的最小物品数=2	单用户评价过的最小物品数=2	10	14.9	3.8	6.0
			20	21.4	2.8	4.9
			30	26.1	2.3	4.1
			40	30.4	2.0	3.7
			50	33.4	1.7	3.3
基于 Spark ALS 算法的推荐模型	单用户评价过的最小物品数=1 $Rank=50$ $Iteration=20$ $Lambda=30$	单用户评价过的最小物品数=2	10	1.06	0.29	0.5
			20	2.13	0.29	0.5
			30	2.83	0.26	0.5
			40	3.76	0.26	0.5
			50	4.61	0.25	0.5

由表 6 可知,单纯以综合指标 $F1$ 值评判,基于物品的推荐模型表现是最适合本次设计的,但当训练数据中的用户数很大而物品数相对较小时,适合使用基于物品的协同过滤。如果训练数据中的用户数相对较小,而物品数较大时,则适合使用基于用户的协同过滤。此外,如果数据量非常大或为了提高计算效率,则考虑使用 Spark ALS 进行建模。基于以上结果,选取基于物品的推荐模型为主、基于 Spark ALS 的推荐模型为辅,分别向用户进行菜品推荐。

2.5 推荐模型实现

展示推荐结果首先要根据日志评分数据,使用选定好的推荐模型生成推荐数据,推荐数据运行图如图 7 所示;其次需要将推荐结果通过 Flume 传输到菜品平台上,根据推荐结果和 MongoDB 数据库中的菜品信息来进行结合展示。

为了更好地服务用户,注册菜品推荐系统账号进入系统后,采用基于项目的推荐模型为用户提供定制化的推荐信息;使用 Spark ALS 推荐模型为用户推荐热门菜肴并让用户能够查看完整菜单,支持用户点击进入菜品详情页,同时用户可查询该菜品的平均评分、菜品评价、菜品用料等。根据页面停留

```
scala> val realRecommend=recommendationrecords.map(case (user,meal)=>reverseuserzipcode.get(user)).
realRecommend: org.apache.spark.rdd.RDD[(string,string,string)]=MapPartitionsRDD[44]at map at
scala> realRecommend.collect.foreach(println)
(A1PTBXZG04GT29,B0054TK99E,自制锅巴)
(A1PTBXZG04GT29,B00CZ18114,火腿鲫鱼煲)
(A1PTBXZG04GT29,B008LUC32E,爆炒孜然鸡心)
(A1PTBXZG04GT29,B0081N4PPWk,香菇扣三丝)
(A1PTBXZG04GT29,B003PDL346,干锅包菜)
```

图 7 推荐数据运行图

Fig. 7 The operation chart of the recommended data
时间、菜品浏览记录、历史订单等数据,可以对系统进行优化,以达到更精确的菜品推送目标。

3 结论(Conclusion)

基于 Spark 的餐饮平台菜品智能推荐系统专注于解决顾客点餐时菜品选择难的问题,可以为顾客智能化地推荐符合其偏好的菜品,提高顾客点餐效率,从而提高顾客满意度及回头率,同时也能提高餐厅的备餐效率。当然,该系统仍然存在欠缺之处,例如对商家的恶意评分,无法通过系统自动判断并删除,因此需要进一步改进。

参考文献(References)

[1] 张奥多,张昕,李怡婷. 基于关联规则的餐饮服务智能推荐系统[J]. 广西科技大学学报,2017,28(3):117-123,131.
[2] 殷明康. 基于 Spark 平台的推荐算法研究与应用[D]. 南京:南京邮电大学,2021.
[3] 杨志伟. 基于 Spark 平台推荐系统研究[D]. 合肥:中国科学技术大学,2015.
[4] 肖芳,张良均. Spark 大数据技术与应用[M]. 北京:人民邮电出版社,2018:240-251.
[5] 胡可响. 基于 Spark 的推荐系统的设计与实现[D]. 杭州:浙江大学,2015.
[6] 丁一凡. 基于 Spark 的电影混合推荐系统的设计与实现[D]. 上海:华东师范大学,2021.
[7] MENG X R, BRADLEY J, YAVUZ B, et al. MLlib: machine learning in apache spark[J]. Journal of machine learning research,2016,17(1):1235-1241.
[8] 徐林. 基于 Spark MLlib 协同过滤算法的美食推荐系统研究[J]. 吉林大学学报(信息科学版),2019,37(2):181-185.
[9] 刘佳耀,王佳斌. 基于 Spark 的 K 近邻 ALS 的推荐算法[J]. 电脑知识与技术,2018,14(11):6-7,10.
[10] 麻括朗. 基于 Spark 平台的协同过滤推荐算法研究[D]. 天津:天津理工大学,2021.
[11] 黄庚长. 基于改进 SVD 的医生推荐算法研究与应用[D]. 长沙:长沙理工大学,2018.
[12] 潘妍虹. 基于用户评论的菜品套餐推荐算法设计与实现[D]. 上海:华东师范大学,2018.
[13] 陈丽芳,陈宏松,孙海民. 大数据平台下 LDA-ALS 智能推荐算法研究[J]. 华北理工大学学报(自然科学版),2022,44(1):89-97.

作者简介:

周杨玥(1995-),女,硕士,助教。研究领域:计算机视觉。
李世锋(1993-),男,硕士,助教。研究领域:人工智能与信息处理。
李林(2000-),男,本科生。研究领域:大数据技术与应用。