

大数据电商用户画像及推荐系统

张冲*, 刘雨菡

湖南交通工程学院, 湖南 衡阳
Email: 307649360@qq.com

收稿日期: 2021年6月10日; 录用日期: 2021年7月5日; 发布日期: 2021年7月14日

摘要

互联网的飞速发展使得社会数据量变得越来越庞大。在大数据的背景下, 用户对获取自己想要的信息的方式也变得更高了。本文通过搭建电商用户画像推荐系统给用户推荐他们想要的信息。为保证推荐结果的不单一性和推荐结果的新颖性, 本文通过基于协同过滤算法, 计算用户或物品的相似度, 将其作为推荐的依据给用户推荐其想要和新颖的物品。利用Embedding向量的相似性来计算用户和物品之间的Embedding相似度, 实现推荐系统的召回, 大大地加快了召回效率。同时基于Python使用flask实现推荐系统在线API给某个确定的用户, 在他某个使用场景下, 推荐他最喜欢的列表。对于冷启动、A/B测试、深度学习技术、难以设计的知识图谱、可交互数据规模有限且奖励信号稀疏、用户画像研究和推荐系统的解释性差这些问题和挑战, 本文给出了相应的解决和优化方法来完善推荐系统, 使用户更加便捷地获取自己想要的信息并在使用推荐系统时更加满意。

关键词

大数据, 用户画像, 推荐系统, 协同过滤, Embedding, 在线API

Big Data E-Commerce User Portrait and Recommendation System

Chong Zhang*, Yuhan Liu

Hunan Institute of Traffic Engineering, Hengyang Hunan
Email: 307649360@qq.com

Received: Jun. 10th, 2021; accepted: Jul. 5th, 2021; published: Jul. 14th, 2021

*第一作者。

Abstract

The rapid development of the Internet has made the amount of social data become more and more huge. In the context of big data, users become higher about the way to get the information. This article recommends the information they want to users by building an e-commerce user portrait recommendation system. To ensure the nonsimplicity of the recommendation results and the novelty of the recommendation results, this paper calculates the similarity of the user or items based on the collaborative filtering algorithm. Using the similarity of Embedding vectors to compute the Embedding similarity between users and items, realize the recall of the recommendation system, greatly accelerate the recall efficiency, use flask to implement the recommendation system online API to a certain user based on Python, recommend the favorite list in certain use scene. For the problems and challenges such as cold start, A/B testing, deep learning technology, design knowledge map, limited interactive data scale and sparse reward signal, poor interpretation of user portrait research and recommendation system, this paper gives corresponding solution and optimization methods to improve the recommendation system, then users can easily obtain the information they want and get more satisfactory when using the recommendation system.

Keywords

Big Data, User Portrait, Recommendation System, Collaborative Filtering, Embedding, Online API

Copyright © 2021 by author(s) and Hans Publishers Inc.

This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

1. 引言

在众多领域中, 电子商务是其中最重要也是最核心的领域, 像京东、阿里和网易这些以 TB 计的高质量、多维度数据记录着用户大量的网络行为的电商平台, 他们对用户的兴趣偏好的研究, 进行用户个性化计算并给用户推荐用户感兴趣的物品是有着极大的需求, 无论是信息消费者还是信息生产者都遇到了极大的挑战。在用户没有明确的目标需求时, 想要找到符合自己的东西并不是一件轻松的事情。而大数据电商推荐系统可以帮助消费者发现自己偏好的信息并且也可帮助信息生产者给用户进行个性化推荐。

大数据电商用户画像及推荐系统通过研究用户的兴趣偏好, 进行个性化计算, 由系统发现用户的兴趣点, 从而引导用户发现自己的信息需求。一个优秀的推荐系统不仅可以给用户提供个性化的服务, 而且还可以抓住用户, 让用户感受到推荐系统带来的不同体验。

用户画像就是对这些数据分析得到用户基本属性、购买能力、行为特征、社交网络、心理特征和兴趣爱好等方面的标签模型, 进而指导并驱动业务场景和运营, 发现和把握在海量用户中的巨大商机。其核心价值在于了解用户、猜测用户的潜在需求、精细化地定位人群特征、挖掘潜在的用户群体。

那么怎么搭建优秀的大数据电商推荐系统呢? 本文基于协同过滤的推荐算法搭建大数据电商推荐系统, 下文给出了搭建推荐系统的框架和搭建系统所需算法和技术并给出了一定的建议。

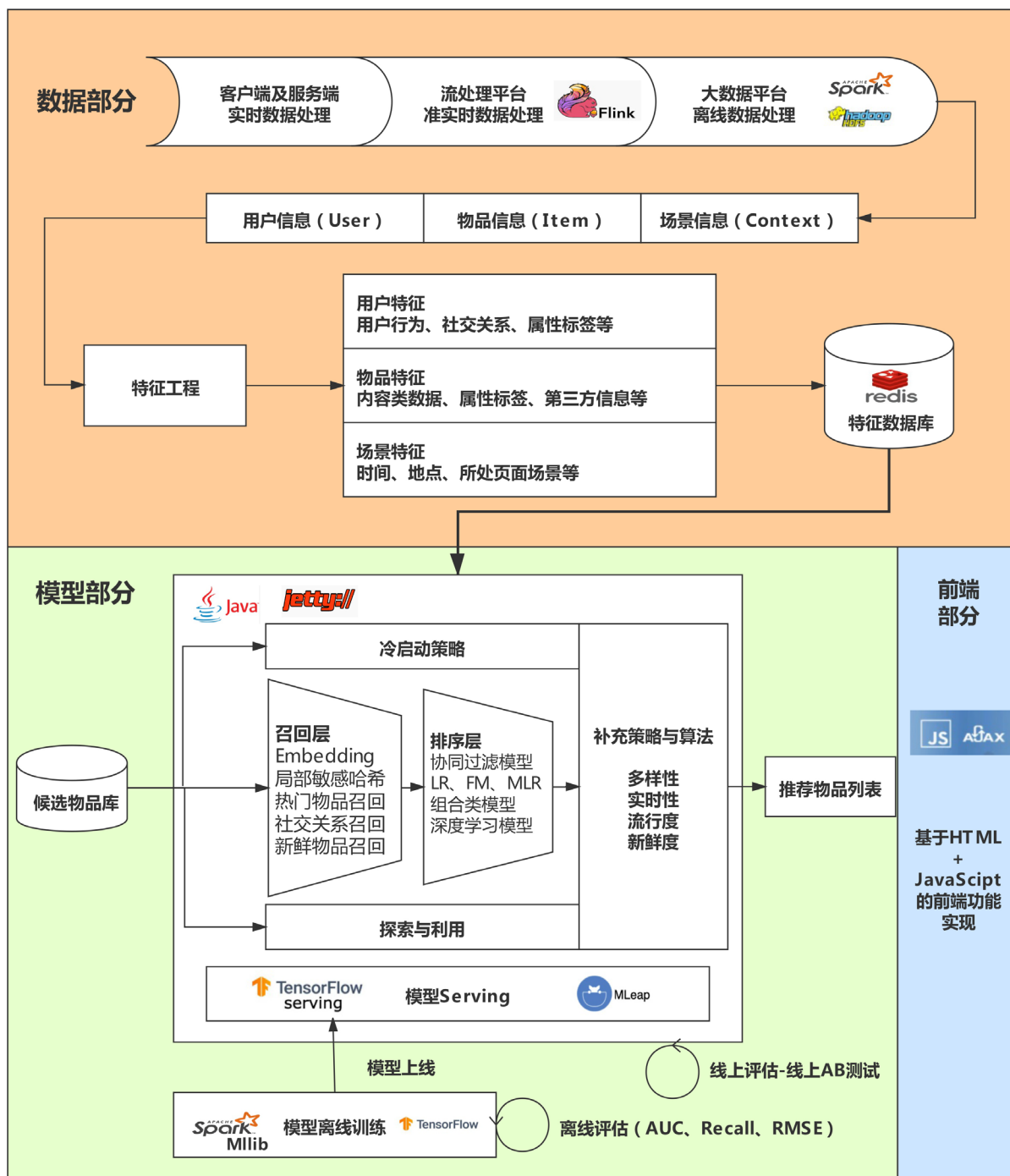


Figure 1. Recommended system technical architecture diagram (Stem from: deep learning recommendation system)
图 1. 推荐系统技术架构示意图(来源: 深度学习推荐系统)

2. 基于深度学习的电商推荐系统的技术架构

2.1. 电商推荐系统架构

深度学习推荐系统的架构与经典的推荐系统架构是一脉相承的, 它对经典推荐系统架构中某些特定模块进行了改进, 使之能够支持深度学习的应用, 如图 1。本推荐系统的技术架构围绕两类问题展开:

一类问题与数据和信息相关, 即“用户信息”、“物品信息”、“场景信息”分别是什么? 如何存储、更新和处理数据? 另一类问题与推荐系统算法和模型相关, 即推荐系统模型如何训练、预测, 以及如何达成更好的推荐效果? 其中“数据和信息”部分逐渐发展为推荐系统中融合了数据离线批处理、实时流处理的数据流框架; “算法和模型”部分则进一步细化为推荐系统中, 集训练(Training)、评估(Evaluation)、部署(Deployment)、线上推断(Online Inference)为一体的模型框架[1]。

第一部分: 推荐系统的数据部分推荐系统的“数据部分”主要负责的是“用户”、“物品”、“场景”信息的收集与处理。根据处理数据量和处理实时性的不同, 我们会用到三种不同的数据处理方式, 按照实时性的强弱排序的话, 它们依次是客户端与服务器端实时数据处理、流处理平台准实时数据处理、大数据平台离线数据处理。

在实时性由强到弱递减的同时, 三种平台的海量数据处理能力则由弱到强。因此, 一个成熟推荐系统的数据流系统会将三者取长补短, 配合使用。我们也会在今后的课程中讲到具体的例子, 比如使用 Spark 进行离线数据处理, 使用 Flink 进行准实时数据处理等等。

大数据计算平台通过对推荐系统日志, 物品和用户的元数据等信息的处理, 获得了推荐模型的训练数据、特征数据、统计数据等。那这些数据都有什么用呢? 具体说来, 大数据平台加工后的数据出口主要有 3 个:

1) 生成推荐系统模型所需的样本数据, 用于算法模型的训练和评估。

2) 生成推荐系统模型服务(Model Serving)所需的“用户特征”, “物品特征”和一部分“场景特征”, 用于推荐系统的线上推断。

3) 生成系统监控、商业智能(Business Intelligence, BI)系统所需的统计型数据。

第二部分: 推荐系统的模型部分推荐系统的“模型部分”是推荐系统的主体。模型的结构一般由“召回层”、“排序层”以及“补充策略与算法层”组成。

其中, “召回层”一般由高效的召回规则、算法或简单的模型组成, 这让推荐系统能快速从海量的候选集中召回用户可能感兴趣的物品。“排序层”则是利用排序模型对初筛的候选集进行精排序。而“补充策略与算法层”, 也被称为“再排序层”, 是在返回给用户推荐列表之前, 为兼顾结果的“多样性”“流行度”“新鲜度”等指标, 结合一些补充的策略和算法对推荐列表进行一定的调整, 最终形成用户可见的推荐列表。

从推荐系统模型接收到所有候选物品集, 到最后产生推荐列表, 这一过程一般叫做“模型服务过程”。为了生成模型服务过程所需的模型参数, 我们需要通过模型训练(Model Training)确定模型结构、结构中不同参数权重的具体数值, 以及模型相关算法和策略中的参数取值。

模型的训练方法根据环境的不同, 可以分为“离线训练”和“在线更新”两部分。其中, 离线训练的特点是可以利用全量样本和特征, 使模型逼近全局最优, 而在线更新则可以准实时地“消化”新的数据样本, 更快地反应新的数据变化趋势, 满足模型实时性的需求。

除此之外, 为了评估推荐系统模型的效果, 以及模型的迭代优化, 推荐系统的模型部分还包括“离线评估”和“线上 A/B 测试”等多种评估模块, 用来得出线下和线上评估指标, 指导下一步的模型迭代优化。

2.2. Sparrow Recsys 的功能

Sparrow RecSys 是一个电影推荐系统, 像所有经典的推荐系统一样, 它具备“相似推荐”“猜你喜欢”等经典的推荐功能, 在页面设置上, 主要由“首页”“电影详情页”和“为你推荐页”组成[2], Sparrow Recsys 中的技术点如下表 1。

Table 1. The technical point of Sparrow Recsys
表 1. Sparrow Recsys 中的技术点

| 模板 | 技术点 | 设计工具 |
|------|---|---------------------------|
| 数据部分 | Spark 进行批量特征处理 | Spark |
| | Flink 进行流式数据处理 | Flink |
| | Redis 保存线上推荐服务器所需特征 | Redis |
| | HDFS 保存离线训练所需训练样本和特征 | HDFS |
| 模型部分 | 使用 Spark Mllib 训练 Embedding 和传统推荐系统 | Spark Mllib |
| | 使用 Tensor Flow 训练深度学习推荐模型 | Tensor Flow |
| | 使用 MLeap、Tensor Flowserving 进行模型上线和在线推断 | MLeap, Tensor Flowserving |
| 前端部分 | 使用简单的 HTML 和 JavaScript 实现前端用户体验部分 | HTML, JavaScript |

首先, 是 Sparrow RecSys 的首页, 如下图 2。

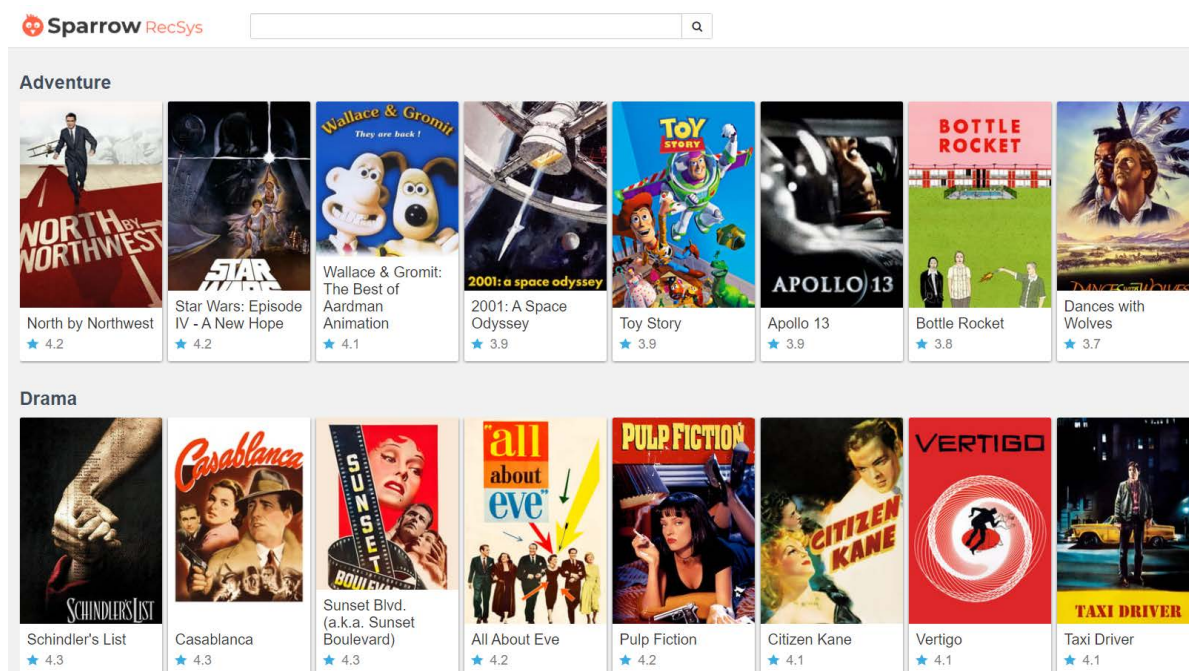


Figure 2. The homepage of Sparrow Recsys
图 2. Sparrow Recsys 的主页

Sparrow RecSys 的首页由不同类型的电影列表组成, 当用户首次访问首页时, 系统默认以历史用户的平均打分从高到低排序, 随着当前用户不断为电影打分, 系统会对首页的推荐结果进行个性化的调整, 比如电影类型的排名会进行个性化调整, 每个类型内部的影片也会进行个性化推荐。

其次, 是电影详情页, 如下图 3。

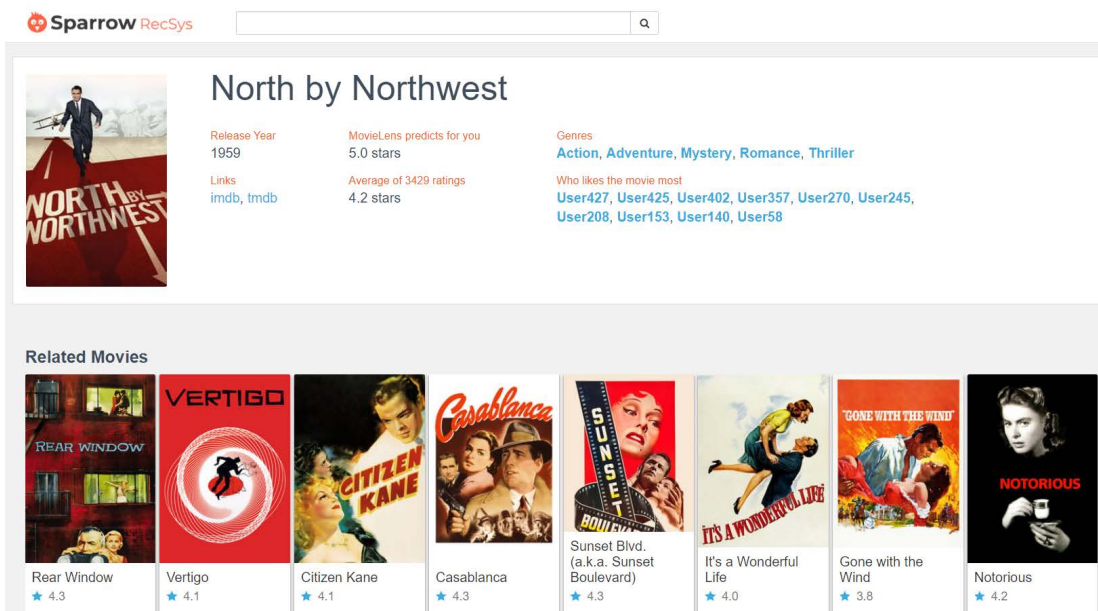


Figure 3. The movie detail page
图 3. 电影详情页

最后, 是为你推荐页, 如下图 4。

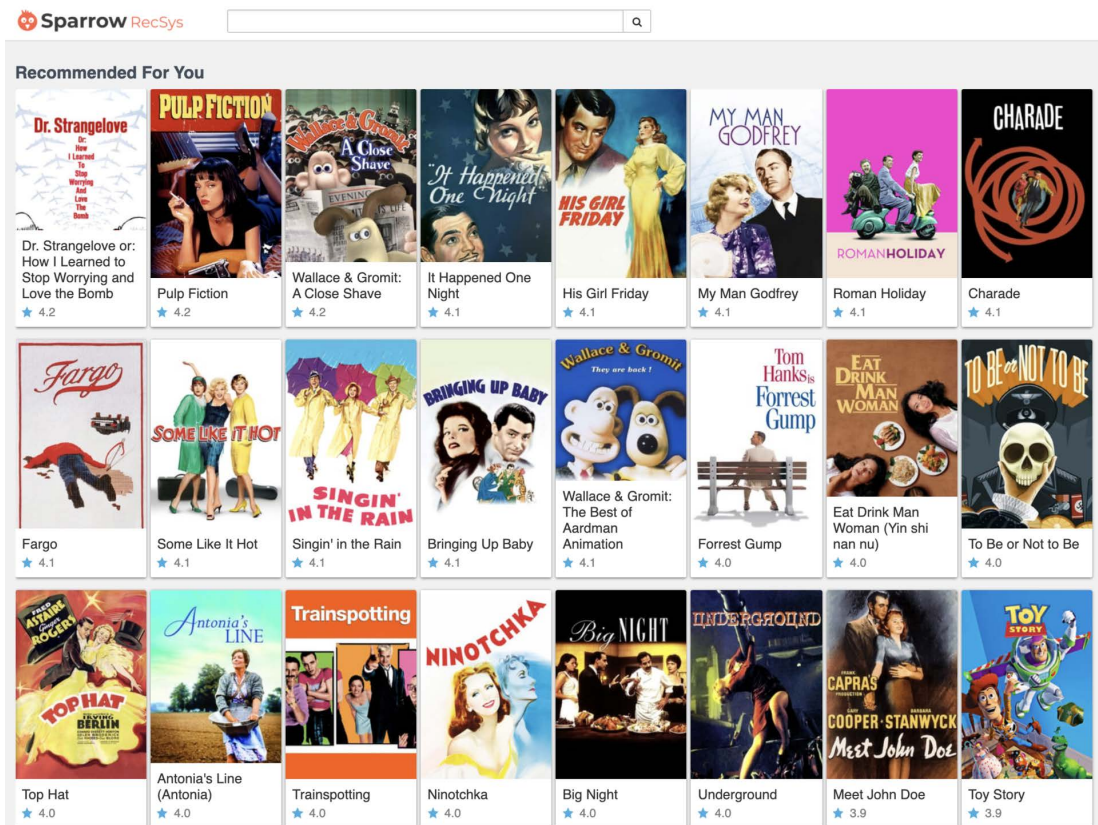


Figure 4. The page of recommended for you
图 4. 为你推荐页

3. 特征工程

3.1. 特征工程

推荐系统是利用“用户信息”、“物品信息”和“场景信息”这三大价值数据,通过构建推荐模型得出推荐列表的工程系统。在这个系统之中,特征工程就是利用工程手段从“用户信息”、“物品信息”和“场景信息”中提取特征的过程。特征其实是对某个行为过程相关信息的抽象表达。在构建推荐系统特征工程时,我们遵循尽可能地让特征工程抽取出一组特征,能够保留推荐环境及用户行为过程中的所有“有用”信息,并且尽量摒弃冗余信息的原则。

3.2. 推荐系统中的常用特征

用户行为数据:是人与物之间的“连接”日志。用户行为在推荐系统中一般分为显性反馈行为(Explicit Feedback)和隐性反馈行为(Implicit Feedback)两种,在不同的业务场景中,它们会以不同的形式体现[3],如下表 2。

Table 2. The user behavior data of different application system

表 2. 不同业务场景下的用户行为数据

| 业务场景 | 线性反馈行为 | 隐性反馈行为 |
|--------|--------------|--------------|
| 电子商务网站 | 对商品的评分 | 点击、加入购物车、购买等 |
| 视频网站 | 对视频的评分、点赞等 | 点击、播放、播放时长等 |
| 新闻类网站 | 赞、踩等行为 | 点击、评论等 |
| 音乐网站 | 对歌曲、歌手、专辑的评分 | 点击、播放、收藏等 |

用户关系数据:就是人与人之间连接的记录。也可以分为“显性”和“隐性”两种,或者称为“强关系”和“弱关系”。如下图 5。



Figure 5. The social networking relationships diagram

图 5. 社交网络关系图

利用用户关系数据的方式:

- 1) 将用户关系作为召回层的一种物品召回方式;
- 2) 也可以通过用户关系建立关系图,使用 Graph Embedding 的方法生成用户和物品的 Embedding;

- 3) 还可以直接利用关系数据, 通过“好友”的特征为用户添加新的属性特征;
4) 甚至可以利用用户关系数据直接建立社会化推荐系统。

属性、标签类数据: 直接描述用户或者物品的特征。属性和标签的主体可以是用户, 也可以是物品。它们的来源非常多样, 大体上包含下表 3 中的几类。

Table 3. The classification and source of property and label data
表 3. 属性、标签类数据的分类和来源

| 主体 | 类别 | 来源 |
|----|--|---|
| 用户 | 人口属性数据 (性别、年龄、住址等) | 用户注册信息、第三方 DMP (Data Management Platform、 数据管理平台) |
| | 用户兴趣标签 | 用户选择 |
| 物品 | 物品标签 | 用户或者系统管理员添加 |
| | 物品属性(例如, 商品的类别、价格; 电影的分类、年代、演员、导演等信息) | 后台录入、第三方数据库 |

内容类数据: 可以看作属性标签型特征的延伸, 同样是描述物品或用户的数据, 但相比标签类特征, 内容类数据往往是大段的描述型文字、图片, 甚至视频。如下图 6。

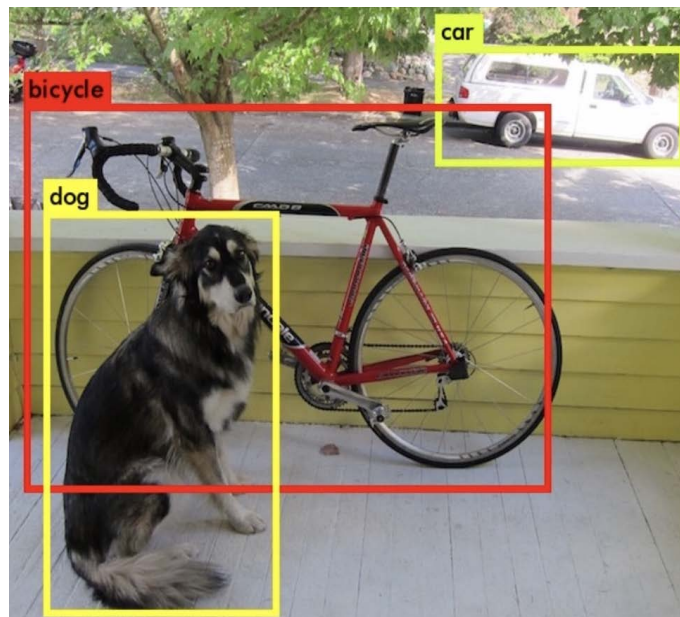


Figure 6. The object detection diagram using computer vision model
图 6. 利用计算机视觉模型进行目标检测

场景信息(上下文信息): 描述推荐行为产生的场景的信息。最常用的上下文信息是“时间”、“地点”、“当前所处推荐页面”、“季节”、“月份”、“是否节假日”、“天气”、“空气质量”、“社会大事件”等等。场景特征描述的是用户所处的客观的推荐环境, 广义上来讲, 任何影响用户决定的因

素都可以当作是场景特征的一部分[4]。

3.3. 利用 Spark 解决特征处理问题

Spark 是一个分布式计算平台。Spark 最典型的应用方式就是建立在大量廉价的计算节点上, 这些节点可以是廉价主机, 也可以是虚拟的 Docker Container (Docker 容器)。

3.3.1. Spark 架构

Spark 程序由 Manager Node (管理节点)进行调度组织, 由 Worker Node (工作节点)进行具体的计算任务执行, 最终将结果返回给 Drive Program (驱动程序)。在物理的 Worker Node 上, 数据还会分为不同的 partition (数据分片), 可以说 partition 是 Spark 的基础数据单元。如下图 7。

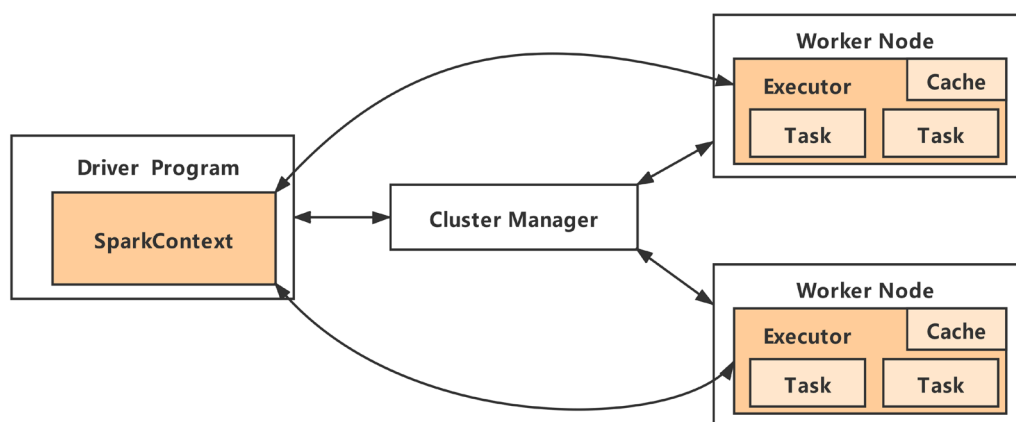


Figure 7. The spark structure diagram
图 7. Spark 架构图

3.3.2. Spark 工作原理

在 Spark 平台上处理这个任务的时候, 会将这个任务拆解成一个子任务 DAG (Directed Acyclic Graph, 有向无环图), 再根据 DAG 决定程序各步骤执行的方法。Spark 程序分别从 textFile 和 hadoopFile 读取文件, 再经过一系列 map、filter 等操作后进行 join, 最终得到了处理结果。如下图 8。

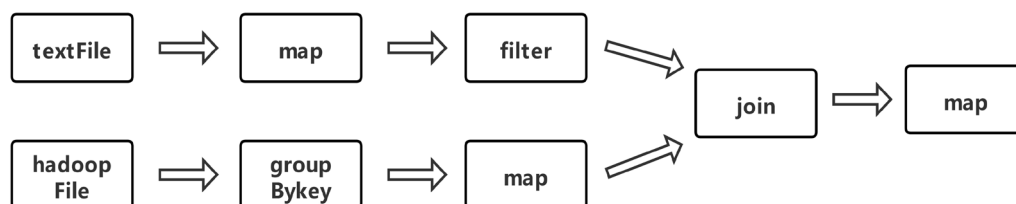


Figure 8. The directed acyclic diagram of Spark procedure
图 8. 某 Spark 程序的任务有向无环图

3.3.3. One-Hot 编码处理类别型特征

所有的特征都可以分为两大类。第一类是类别、ID 型特征(以下简称类别型特征)。第二类是数值型特征, 能用数字直接表示的特征就是数值型特征。进行特征处理的目的是, 把所有的特征全部转换成一个数值型的特征向量, 对于数值型特征, 直接把这个数值放到特征向量上相应的维度上就可以了。

我们使用 One-hot 编码(也被称为独热编码), 它是将类别、ID 型特征转换成数值向量的一种最典型

的编码方式。它通过把所有其他维度置为 0, 单独将当前类别或者 ID 对应的维度置为 1 的方式生成特征向量。

SparrowRecsys 使用 Spark 的机器学习库 MLlib 来完成 One-hot 特征的处理。

One-hot 编码也可以自然衍生生成 Multi-hot 编码(多热编码)。比如, 对于历史行为序列类、标签特征等数据来说, 用户往往会与多个物品产生交互行为, 或者一个物品被打上多个标签, 这时最常用的特征向量生成方式就是把其转换成 Multi-hot 编码。

3.3.4. 数值型特征的处理——归一化和分桶

当我们把特征的原始数值直接输入推荐模型, 就会导致这两个特征对于模型的影响程度有显著的区别, 归一化能够解决特征取值范围不统一的问题。“分桶(Bucketing)”, 就是将样本按照某特征的值从高到低排序, 然后按照桶的数量找到分位数, 将样本分到各自的桶中, 再用桶 ID 作为特征值, 用分桶的方式来解决特征值分布极不均匀的问题。

3.4. Embedding 技术

Embedding 就是用一个数值向量“表示”一个对象(Object)的方法。对象可以是一个词、一个物品, 也可以是一部电影等等。一个物品能被向量表示, 是因为这个向量跟其他物品向量之间的距离反映了这些物品的相似性。更进一步来说, 两个向量间的距离向量甚至能够反映它们之间的关系。

Embedding 技术对深度学习推荐系统的重要性

Embedding 是处理稀疏特征的利器。因为推荐场景中的类别、ID 型特征非常多, 大量使用 One-hot 编码会导致样本特征向量极度稀疏, 而深度学习的结构特点又不利于稀疏特征向量的处理, 因此几乎所有深度学习推荐模型都会由 Embedding 层负责将稀疏高维特征向量转换成稠密低维特征向量。

Embedding 可以融合大量有价值信息, 本身就是极其重要的特征向量。相比由原始信息直接处理得来的特征向量, Embedding 的表达能力更强, 特别是 Graph Embedding 技术被提出后, Embedding 几乎可以引入任何信息进行编码, 使其本身就包含大量有价值的信息, 所以通过预训练得到的 Embedding 向量本身就是极其重要的特征向量[5]。

3.5. 使用 Spark 生成 Item2vec 和 Graph Embedding

3.5.1. Item2vec: 序列数据的处理

Item2vec 是基于自然语言处理模型 Word2vec 提出的, 所以 Item2vec 要处理的是类似文本句子、观影序列之类的序列数据。在开始 Item2vec 的训练之前, 我们先为它准备好训练用的序列数据。通过处理 rating 表得到观影序列, rating (评分)的数据表, 里面包含了用户对看过电影的评分和评分的时间。如下表 4。

因为 MovieLens 这个 rating 表本质上只是一个评分的表, 不是真正的“观影序列”, 所以我们对评分做一个过滤, 只放用户打分比较高的电影。样本处理的思路, 对一个用户来说, 我们先过滤掉他评分低的电影, 再把他评论过的电影按照时间戳排序。

序列数据的处理在 Spark 上实现的步骤:

- 读取 ratings 原始数据到 Spark 平台;
- 用 where 语句过滤评分低的评分记录;
- 用 groupByuserId 操作聚合每个用户的评分记录, DataFrame 中每条记录是一个用户的评分序列;
- 定义一个自定义操作 sortUdf, 用它实现每个用户的评分记录按照时间戳进行排序;
- 把每个用户的评分记录处理成一个字符串的形式, 供后续训练过程使用。

Table 4. The rating table
表 4. 评级评分表

| userid | movieid | rating | timestamp |
|--------|---------|--------|------------|
| 1 | 2 | 3.5 | 1112486027 |
| 1 | 29 | 3.5 | 1112484676 |
| 1 | 32 | 3.5 | 1112484819 |
| 1 | 47 | 3.5 | 1112484727 |
| 1 | 50 | 3.5 | 1112484580 |
| 1 | 112 | 3.5 | 1094785740 |
| 1 | 151 | 4.0 | 1094785734 |
| 1 | 223 | 4.0 | 1112485573 |
| 1 | 253 | 4.0 | 1112484940 |
| 1 | 260 | 4.0 | 1112484826 |
| 1 | 293 | 4.0 | 1112484703 |
| 1 | 296 | 4.0 | 1112484767 |
| 1 | 318 | 4.0 | 1112484798 |
| 1 | 337 | 3.5 | 1094785709 |
| 1 | 367 | 3.5 | 1112485980 |

3.5.2. Item2vec: 模型训练

首先是创建 Word2vec 模型并设定模型参数。我们要清楚 Word2vec 模型的关键参数有 3 个, 分别是 `setVectorSize`、`setWindowSize` 和 `setNumIterations`。其中, `setVectorSize` 用于设定生成的 Embedding 向量的维度, `setWindowSize` 用于设定在序列数据上采样的滑动窗口大小, `setNumIterations` 用于设定训练时的迭代次数。这些超参数的具体选择就要根据实际的训练效果来做调整了。

其次, 模型的训练过程就是调用模型的 `fit` 接口。训练完成后, 模型会返回一个包含了所有模型参数的对象。

最后一步就是提取和保存 Embedding 向量, 通过调用 `getVectors` 接口就可以提取出某个电影 ID 对应的 Embedding 向量, 之后就可以把它们保存到文件或者其他数据库中, 供其他模块使用。

3.5.3. Graph Embedding: 数据准备

我们选择基于随机游走的 Graph Embedding 方法——Deep Walk 方法进行实现, 算法流程如下图 9。在 Deep Walk 方法中, 需要准备的最关键数据是物品之间的转移概率矩阵。下图是 Deep Walk 的算法流程图, 转移概率矩阵表达了图 3(b)中的物品关系图, 它定义了随机游走过程中, 从物品 A 到物品 B 的跳转概率。

生成转移概率矩阵的函数输入是在训练 Item2vec 时处理好的观影序列数据。输出的是转移概率矩阵, 由于转移概率矩阵比较稀疏, 因此采用了一个双层 Map 的结构去实现它。比如说, 要得到物品 A 到物品 B 的转移概率, 那么 `transfer Matrix (itemA) (itemB)` 就是这一转移概率。

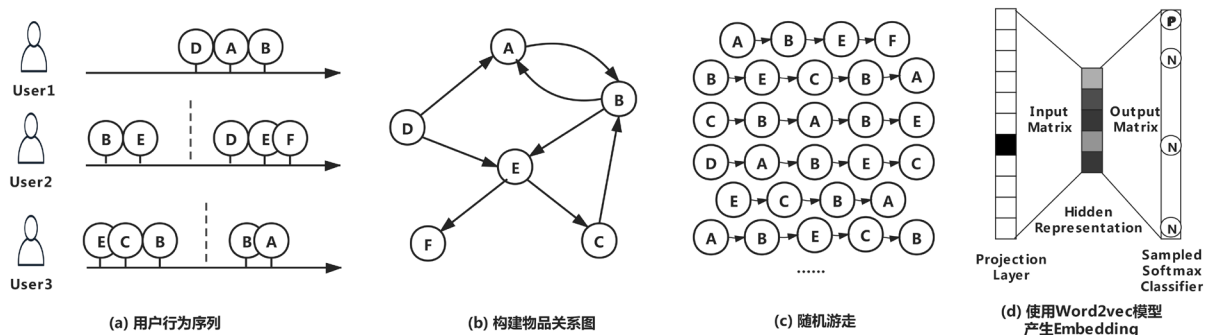


Figure 9. The algorithm process of Deep Walk

图 9. Deep Walk 的算法流程

在求取转移概率矩阵的过程中,我先利用 Spark 的 flatMap 操作把观影序列“打碎”成一个个影片对,再利用 countByValue 操作统计这些影片对的数量,最后根据这些影片对的数量求取每两个影片之间的转移概率。

在获得了物品之间的转移概率矩阵之后,进入图 3(c)的步骤,进行随机游走采样[6]。

3.5.4. Graph Embedding: 随机游走采样过程

随机游走采样的过程是利用转移概率矩阵生成新的序列样本的过程。我们要根据物品出现次数的分布随机选择一个起始物品,之后就进入随机游走的过程。在每次游走时,根据转移概率矩阵查找到两个物品之间的转移概率,然后根据这个概率进行跳转。比如当前的物品是 A,从转移概率矩阵中查找到 A 可能跳转到物品 B 或物品 C,转移概率分别是 0.4 和 0.6,那么我们就按照这个概率来随机游走到 B 或 C,依次进行下去,直到样本的长度达到了我们的要求。根据上面随机游走的过程,我们使用 Scala 进行了实现。

通过随机游走产生了训练所需的 sampleCount 个样本之后,后续过程和 Item2vec 的过程完全一致了,就是把这些训练样本输入到 Word2vec 模型中,完成最终 Graph Embedding 的生成。

4. 总结

近些年电商行业的火爆发展,在面对成千上万的商品时,用户想要得到自己希望的商品是比较有困难的。这时如何最大程度地给用户展示其最有可能看的商品显得十分重要,上述两点很大程度上催生出了推荐系统。通过深入调研后我们了解到,利用用户画像和协同过滤可以准确掌握用户的兴趣爱好,依据这些兴趣和爱好能够对用户进行精准的推荐。用户画像和协同过滤相结合可以使推荐结果不仅具有新颖性而且还准确。

本文通过使用基于协同过滤的算法来搭建电商推荐系统。我们通过三个方面进行推荐系统的设计:第一部分是设计推荐系统技术的架构。第二部分则是基于协同过滤算法:收集用户偏好计算相似度找到类似的用户或物品,并通过基于用户的 CF (User CF)和基于物品的 CF (Item CF)两种计算方法进行协同过滤推荐。Embedding 技术:把自然语言转变为一串数字,使得自然语言可以被用来计算;替代 OneHot,极大地降低了特征的维度;而且还替代了协同矩阵,极大可能性地降低了计算复杂度。第三部分,基于 Python 使用 flask 设计 API 服务接口设计。

同时我们也阐述了冷启动问题、A/B 测试、推荐系统与深度学习、推荐系统算法与知识图谱、推荐系统与强化学习、用户画像研究的现状和挑战以及推荐系统的可解释性,并得到了相关的解决以及优化方案。

电商的火热发展,使得大多数的人们都选择在网上进行购物,电商给人们带来了无尽的便利并改变着人们的生活方式。在面对电商平台上成千上万的商品时,推荐系统发挥着极大地作用,通过对用户的行为分析来找到其潜在喜欢的商品,推荐给他,不仅节省了用户寻找自己喜欢商品的时间,同时也为商家和电商平台增加了利润,由此可见推荐系统的重要性。大数据推荐系统是一个需要海量数据以及不断更新技术的行业,需要对所应用行业的物品属性有很抽象的理解。在享受大数据为企业和消费者带来便利的同时,也需要加强企业与消费者的信息安全。

在整个实验过程中,本文仅仅只是简单地叙述了基于协同过滤算法的基本原理、实现地步骤和搭建推荐系统所需要的技术,其实践过程还未实现,这是要在日后的时间内完成的。通过本次论文,我们了解了电商平台推荐系统的原理、实现过程以及所用算法和技术,让我们认识到了同时也让我们增加了信心。即使在遇到未知的困难也不会退缩。希望在未来的工作中,我们能在推荐系统这一领域上展露头角,并不断为之钻研,争取突破技术的瓶颈。

基金项目

衡阳市科技计划项目(2020211004); 新冠肺炎防治创新平台专项(大数据交通应急管理重点实验室)。

参考文献

- [1] 孙吉祥. 用户画像在推荐系统中的研究与应用[D]: [硕士学位论文]. 北京: 北方工业大学, 2020.
- [2] 谭浩, 郭雅婷. 基于大数据的用户画像构建方法与运用[J]. 包装工程, 2019, 40(22): 95-101.
- [3] 张宇, 阮雪灵. 大数据环境下移动用户画像的构建方法研究[J]. 中国信息化, 2020(4): 65-68.
- [4] 杨单, 刘启川. 基于大数据的跨境电商平台个性化推荐策略优化[J]. 对外经贸实务, 2020(11): 33-36.
- [5] 陈艺婷. 基于混合协同过滤的电商推荐系统的设计与实现[D]: [硕士学位论文]. 南京: 南京邮电大学, 2020.
- [6] 刘晓飞, 朱斐. 基于用户偏好特征挖掘的个性化推荐算法[J]. 计算机科学, 2020, 47(4): 50-53.