30 | 推荐系统的后处理及日志回采

2023-06-23 黄鸿波 来自北京

《手把手带你搭建推荐系统》

你好，我是黄鸿波。

到现在，可以说我们已经把推荐系统从头到尾学习了一遍。这节课是最后一节正课内容，也就是推荐系统的后续处理和日志回采。

我把本节课分成了下面三个要点。

1. 推荐列表给到用户后的操作。
2. 如何进行推荐系统的后处理。
3. 如何进行日志回采。

现在正式开始本节课的内容。

推荐列表给到用户后的操作

到现在推荐系统从最开始的数据到最终给到用户的推荐列表策略，都已经完成了。按理来说，已经跑完了整个流程，但站在推荐系统的角度，我们还需要确认用户是否对推荐进行了进一步的了解和行动，以及反馈他们的反应。

在推荐的后续步骤中，往往需要建立日志系统，记录用户的行为以及系统的运行状态，以便于对系统进行优化和监控，并做好用户反馈等工作。具体来说，主要有下面五个方面。

1. **建立日志系统**：在推荐系统中建立日志系统来记录用户行为、推荐结果和系统运行状态，可以使用日志收集工具（如 Logstash、FluentD 等）。在建立日志系统时，需要考虑哪些数据需要记录，如用户 ID、访问时间、推荐结果、点击次数等。
2. **日志分析**：使用数据分析工具（如 Hadoop、Spark 等）对日志数据进行分析，提取有用的信息，例如用户兴趣、推荐结果点击率等。
3. **对结果影响评估**：结合分析结果，评估用户行为对推荐结果的影响，发现可能出现的问题，例如推荐结果展现不足、推荐结果与用户兴趣不匹配等。
4. **改进系统**：针对问题做出改进，例如优化推荐算法、调整推荐策略等，以提高推荐结果的准确性和满意度。
5. **监控系统**：对系统进行监控，发现问题及时解决，提高系统的稳定性和推荐准确性。在监控过程中，可以使用运维工具（如 Nagios、Zabbix 等）对系统的运行情况进行实时监控，快速发现和解决问题。

如何进行推荐系统的后处理

我们来举例说明一下，在推荐系统中如何利用日志系统来改进推荐算法。

利用日志系统可以收集用户的行为数据（包括用户的点击、浏览、收藏、购买等行为），详细记录用户与推荐系统交互的过程。当拿到这些数据之后，需要对用户行为数据进行多维度的分析和建模，例如根据用户的年龄、性别、地理位置、历史购买记录等建立用户画像。我们也可以使用数据挖掘和机器学习等技术对这些数据进行多维分析和建模，从而对数据进行整体拆解。

当然，我们还可以通过提取一些关键特征来描述用户的行为习惯（例如点击率、购买频率、购买金额等），从而训练和改进推荐算法。

在收集用户信息时，一般会在推荐系统中内置统计功能，记录用户的行为数据并上传到服务器。在实际开发中可以使用 Numpy、Pandas、Scikit-learn 等数据处理常用库，配合 TensorFlow 或者 PyTorch 等深度学习方法来进行模型构建。这个过程中，可以使用大数据技术来实现日志处理和分析，以下是五个常用的技术和库。

1. **高性能分布式文件系统**：可以使用分布式文件系统（如 HDFS、AWS S3 等），存储海量日志数据，方便高效地进行数据读取和处理。
2. **高性能分布式数据处理框架**：可以使用分布式计算框架（如 Apache Spark、Hadoop MapReduce 等），对海量的日志数据进行高效并行化处理。利用分布式计算框架可以通过并行计算的方式，高速安全地进行数据处理。
3. **分布式数据库**：使用分布式数据库（如 Apache HBase、Amazon DynamoDB 等），对处理后的日志信息存储和管理，方便进行数据查询、统计和分析。
4. **分布式 Web 日志收集器**：使用分布式日志采集技术（如 Flume、Logstash、Grok 等），将推荐系统的请求和响应日志收集到一个中心化的地方，并根据业务需求利用数据预处理和清洗技术将原始的日志转换为结构化数据，方便后续的自动化处理。
5. **分布式机器学习平台**：使用分布式机器学习平台（如 Spark MLlib、Amazon SageMaker 等），对海量的推荐系统日志数据进行机器学习建模和数据挖掘。通过并行计算和分布式训练模型，可以更快地实现算法的迭代和更新，达到更准确的结果。

通过使用上面技术和库，可以方便地对推荐系统的日志进行处理和分析，并实现数据存储、清洗、转换、机器学习建模等一系列操作。

如何进行日志回采

最后我们再来说说日志回采的问题。对于一个推荐系统而言，常见的日志可以分为原始日志、点击日志、会话日志、反馈行为日志四个类别。

原始日志是推荐系统最基础的日志，它包含了所有用户在系统中的行为记录，如浏览、搜索、购买等。

点击日志是用户在推荐系统中进行点击的记录，包括点击的物品、时间、位置、IP 地址等。

会话日志是指用户在推荐系统中的会话记录，一次会话包括多个行为，通常是用户在一段时间内的连续操作。

反馈行为日志包含用户对推荐结果的反馈记录，如喜欢、不喜欢、加入购物车等。

这些日志记录了用户的行为，可以用于优化推荐算法，提高系统的准确性和用户的参与度。同时对于推荐系统的评估和监控来说，这些日志也是非常重要的数据来源。

进行日志采集时，主要使用的方法是**埋点技术**和**服务端日志采集**。

埋点技术

埋点技术是指在应用程序中添加数据采集代码，以便收集和记录特定事件和活动的详细信息。每当应用程序执行特定操作时，采集代码会触发并生成一个日志记录，其中包含有关事件或活动的详细信息。

通过使用埋点技术，开发人员可以更轻松地诊断和解决问题，提高应用程序的稳定性和性能，同时还可以提供更好的用户体验。在数据分析和决策制定方面，通过分析采集的数据，企业可以更好地了解用户需求和行为，制定更有效的业务策略。

一般来讲，埋点通常被添加在客户端。首先，开发人员需要确定需要采集哪些数据，例如用户点击按钮、访问特定页面、触发某些事件等，然后在应用程序中添加代码，以便在触发特定操作时记录日志信息。此代码应该能够获取目标数据并将其存储到日志记录中。

当应用程序启动时，开发人员需要确保采集代码被正确注册，以确保在触发事件时，可以为其分配正确的代码。一旦采集代码被实现并注册，开发人员需要测试它是否正常工作。可以使用模拟数据以及开发人员工具进行测试，这一步我们称为埋点采集测试。开发人员需要收集输出的日志，以确保数据已正确采集。通过收集并分析日志，开发人员可以了解应用程序的运行情况，及时处理潜在的问题。

服务端日志采集

对于服务端来说，日志采集主要通过在 Controller 接口中进行埋点，然后通过 AOP 技术、Kafka 消息等对用户的行为信息进行采集。

之所以使用 AOP 技术，是因为可以在不修改原有业务代码的情况下，将日志采集逻辑统一添加到共同的切点中，这样可以提高代码的可维护性和可扩展性。

而使用 Kafka 消息系统可以实现日志数据的异步发送，减少日志对系统性能的影响。同时，Kafka 可以提供高可靠性的消息传输和消息存储功能，保证日志数据的安全性和完整性。

最后，使用 Logback 可以实现对日志数据的规范化和标准化输出，方便后续的日志分析和处理工作。同时，Logback 还支持日志级别的配置和动态调整，可以根据实际需要控制日志输出的详细程度和数量。

这里的 AOP 技术实际上是指面向切片编程（一种编程范式），旨在提高代码复用性、降低代码耦合度、增强代码可维护性和可扩展性。

AOP 技术通过将横切关注点（如日志、事务、权限等）从程序主逻辑中抽离出来，再将其统一使用一些特殊的语法或术语来描述和实现，从而使得程序主逻辑更加简洁、可读、可维护和可扩展。

AOP 技术在日志采集方面有非常大的优势。

1. 它是一种跨越多个对象和代码层次（例如业务逻辑层、持久化层）的编程思想和技术，可以使代码组织结构更加清晰。
2. 通过增强（Advice）、切点（Pointcut）、连接点（Joinpoint）等概念对代码进行修改，从而实现功能的重用和代码的复用，减少代码的重复性。
3. 可以将关注点与业务逻辑进行分离，将代码进行模块化划分，便于代码维护和扩展。
4. 可以在不修改原有代码的情况下，对程序进行拓展和增强，从而实现额外的功能需求。
5. 提供了更加灵活的扩展机制，可以方便地对功能进行切换、拓展和升级。

总之，AOP 技术是一种很有用的技术，我们可以将它运用到日志采集、事务管理、权限控制等场景中，提高代码的可读性、可维护性和可扩展性。

总结

本节课到这里就已经讲完了，我们来做一个简单的总结，学完本节课你应该知道下面五个要点。

1. 推荐系统根据用户的历史行为、个人信息、社交关系等计算出一组推荐列表，但这些推荐并不一定都能满足用户的需要或兴趣。因此推荐系统把排序列表给到用户后，还需要做其他流程来满足用户的需求。
2. 推荐系统后处理主要包括用户反馈、用户行为收集、日志分析、改进系统、监控系统等。
3. 推荐系统通常使用数据处理库和深度学习方法来构建模型，并使用分布式文件系统进行日志处理和存储。
4. 对于一个推荐系统而言，常见的日志可以分为原始日志、点击日志、会话日志、反馈行为四个类别。
5. 埋点技术是通过在应用程序中添加代码，记录特定事件和活动的详细信息，从而帮助开发人员更好地诊断和解决问题，提高应用程序的稳定性和性能。

思考题

学完本节课，给你留两个思考题。

1. 如果用户不愿意被追踪并收集个人信息，如何在保证推荐准确率的前提下最小化用户信息采集？
2. 埋点技术可以帮助企业收集用户需求和行为，以制定更有效的业务策略。可以分享一个大数据分析在提高用户体验和企业运营效率上起重要作用的例子吗？

期待你的分享，如果今天的内容让你有所收获，也欢迎你推荐给有需要的朋友！

精选留言（1）

peter

2023-06-24 来自北京

请教老师几个问题：

Q1：推荐系统在整个网站一般占多大比重？

一个网站，包含很多系统，推荐系统一般占多大分量？可以从人力投入角度来衡量，或者从硬件资源占用角度等方面来衡量。

Q2：用spark来分析日志，具体有什么方法？

Q3：会话日志和会话记录会和其他日志重复吗？

如果说会话记录是用户的操作，那么就会和“点击日志”和“反馈行为日志”重复啊，因为后两者就是用户的操作。

Q4：Java中用了AOP，Python中也有吗？

Q5：对于日志，不需要用ES(ElasticSearch)吗？