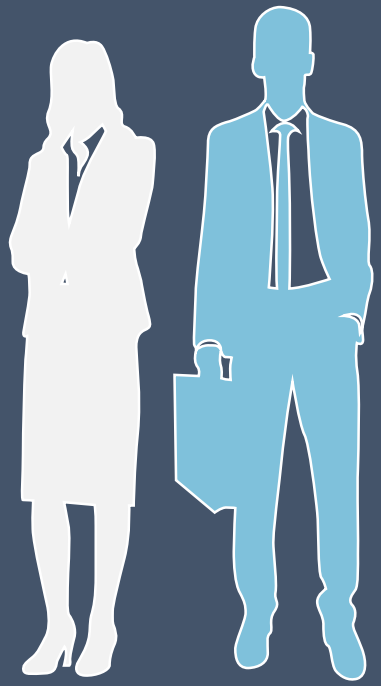
AQF Analyst of Quantitative Finance



Python网络爬虫实战介绍

朱映秋

CONTENTS

AQF Analyst of Quantitative Finance

爬虫的概念

概念

- 爬虫（Web爬虫或网络爬虫）是一种自动化程序，用于在互联网上浏览和抓取信息。它模拟人类用户的行为，通过HTTP或其他网络协议访问网页，并提取感兴趣的数据。
- 网络爬虫就像一只蜘蛛一样在互联网上沿着URL的丝线爬行，下载每一个URL所指向的网页，分析页面内容。
- 爬虫在各种应用场景中发挥重要作用。搜索引擎使用爬虫来收集互联网上的网页内容，并建立索引以供搜索。新闻聚合网站使用爬虫来获取各种新闻来源的信息。电子商务网站使用爬虫来收集竞争对手的产品信息和价格等。

爬虫基本原理

- **基本原理**
  - 获取网页并提取和保存信息的自动化程序。

通用网络爬虫

- **通用网络爬虫又称为全网爬虫，其爬行对象由一批种子URL扩充至整个Web，该类爬虫比较适合为搜索引擎搜索广泛的主题，主要由搜索引擎或大型Web服务提供商使用。**
  - 深度优先策略：按照深度由低到高的顺序，依次访问下一级网页链接，直到无法再深入为止。
  - 广度优先策略：按照网页内容目录层次的深浅来爬行，优先爬取较浅层次的页面。当同一层中的页面全部爬行完毕后，爬虫再深入下一层。

聚焦网络爬虫

- **聚焦网络爬虫又被称作主题网络爬虫，其最大的特点是只选择性地爬行与预设的主题相关的页面。**
  - 基于内容评价的爬行策略：该种策略将用户输入的查询词作为主题，包含查询词的页面被视为与主题相关的页面。
  - 基于链接结构评价的爬行策略：该种策略将包含很多结构信息的半结构化文档Web页面用来评价链接的重要性，其中一种广泛使用的算法为PageRank算法。
  - 基于增强学习的爬行策略：该种策略将增强学习引入聚焦爬虫，利用贝叶斯分类器对超链接进行分类，计算出每个链接的重要性，按照重要性决定链接的访问顺序。
  - 基于语境图的爬行策略：该种策略通过建立语境图学习网页之间的相关度，计算当前页面到相关页面的距离，距离越近的页面中的链接优先访问。

增量式网络爬虫

- **增量式网络爬虫只对已下载网页采取增量式更新或只爬行新产生的及已经发生变化的网页，需要通过重新访问网页对本地页面进行更新，从而保持本地集中存储的页面为最新页面。**
- **常用的更新方法如下。**
  - 统一更新法：以相同的频率访问所有网页，不受网页本身的改变频率的影响。
  - 个体更新法：根据个体网页的改变频率来决定重新访问各页面的频率。
  - 基于分类的更新法：爬虫按照网页变化频率分为更新较快和更新较慢的网页类别，分别设定不同的频率来访问这两类网页。

深层网络爬虫

- Web页面按照存在方式可以分为表层页面和深层页面两类。表层页面指以传统搜索引擎可以索引到的页面，深层页面为大部分内容无法通过静态链接获取，隐藏在搜索表单后的，需要用户提交关键词后才能获得的Web页面。
- 深层爬虫的核心部分为表单填写，包含以下两种类型。
  - 基于领域知识的表单填写：该种方法一般会维持一个本体库，通过语义分析来选取合适的关键词填写表单。
  - 基于网页结构分析的表单填写：这种方法一般无领域知识或仅有有限的领域知识，将HTML网页表示为DOM树形式，将表单区分为单属性表单和多属性表单，分别进行处理，从中提取表单各字段值。

爬虫的合法性与robot.txt协议

- **爬虫的合法性**
  - 目前，多数网站允许将爬虫爬取的数据用于个人使用或者科学研究。但如果将爬取的数据用于其他用途，尤其是转载或者商业用途，严重的将会触犯法律或者引起民事纠纷。
- **以下两种数据是不能爬取的，更不能用于商业用途。**
  - 个人隐私数据：如姓名、手机号码、年龄、血型、婚姻情况等，爬取此类数据将会触犯个人信息保护法。
  - 明确禁止他人访问的数据：例如用户设置了账号密码等权限控制，进行了加密的内容。
  - 此外，还需注意版权相关问题，有作者署名的受版权保护的内容不允许爬取后随意转载或用于商业用途。

爬虫的合法性与robot.txt协议

- robot.txt
  - 当使用一个爬虫爬取一个网站的数据时，需要遵守网站所有者针对所有爬虫所制定的协议，这便是robot.txt协议。
  - robots.txt是一种用于控制网络爬虫行为的标准。它是一个文本文件，位于网站的根目录下，用于告诉搜索引擎和其他爬虫哪些页面可以被访问，哪些页面应该被忽略。
  - robots.txt协议的作用是指导爬虫在访问网站时遵守特定规则。网站所有者可以在robots.txt文件中定义一些规则，以控制爬虫的访问行为。这些规则通常用于限制爬虫访问敏感页面、排除无关内容或避免对服务器造成过大的负载。

获取网页

- 源代码：包含了网页的部分有用信息，可以从中提取想要的信息。

提取信息

- **提取信息**
  - 即分析网页源代码，从中提取我们想要的数据，以便我们后续处理和分析数据。
- **相关方法**
  - 通用方法：正则表达式（构造时较复杂且易出错）
  - 有规则的网页结构（网页节点属性、CSS选择器或XPath） ：Beautiful Soup库、pyquery库、lxml库等。使用这些库，我们可以高效快速地从中提取网页信息，如节点的属性、文本值等。

保存数据

- **保存数据**
  - 提取信息后，我们一般会将提取到的数据保存到某处以便后续使用。
- **保存形式**
  - 简单保存：TXT文本、JSON文本。
  - 数据库： MySQL、 MongoDB等。
  - 远程服务器：借助SFTP进行操作等。

自动化程序

- 即爬虫可以代替人来完成这些操作。
- 当量特别大或者想快速获取大量数据的话，我们肯定还是要借助程序。
- 而爬虫就是代替我们来完成这份爬取工作的自动化程序，它可以在抓取过程中进行各种异常处理、错误 重试等操作，确保爬取持续高效地运行。

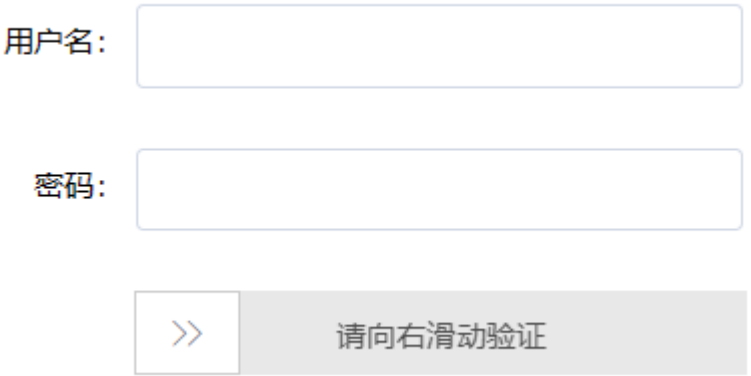
CONTENTS

AQF Analyst of Quantitative Finance

- **1. 通过User-Agent校验反爬**
  - 通过User-Agent校验是一种常见的反爬虫技术之一。User-Agent是HTTP请求头的一部分，用于标识发送请求的客户端或用户代理。在爬虫中，可以通过修改User-Agent字段的值来模拟不同的浏览器或爬虫代理。
  - 网站可以通过检查请求中的User-Agent字段来判断请求的来源是否为合法的浏览器或特定的爬虫代理。如果User-Agent与预期的值不匹配，网站可能会拒绝服务或采取其他反爬虫措施。

2. 通过访问频度反爬

- 普通用户通过浏览器访问网站的速度相对爬虫而言要慢的多，所以不少网站会利用这一点对访问频度设定一个阈值，如果一个IP单位时间内访问频度超过了预设的阈值，将会对该IP做出访问限制。
- 验证码是一种人机验证机制，旨在区分真实用户和自动化程序（如爬虫）。通常，验证码以图像、文字、数字或拼图等形式呈现给用户，要求用户根据要求进行正确的识别或操作。这样可以有效阻止自动化程序的访问，因为爬虫很难自动解析和正确处理验证码。。



网站反爬虫的目的与手段

- **3. 通过验证码校验反爬**
  - 有部分网站不论访问频度如何，一定要来访者进行验证才能继续操作。例如12306网站，不管是登陆还是购票，全部需要进行验证，与访问频度无关。
- **4. 通过变换网页结构反爬**
  - 一些社交网站常常会更换网页结构，而爬虫大部分情况下都需要通过网页结构来解析需要的数据，所以这种做法也能起到反爬虫的作用。在网页结构变换后，爬虫往往无法在原本的网页位置找到原本需要的内容。

网站反爬虫的目的与手段

- 5. 通过账号权限反爬
  - 部分网站需要登录才能继续操作，这部分网站虽然并不是为了反爬虫才要求登录操作，但确实起到了反爬虫的作用。
  - 例如，CSDN等网站查看评论就需要登录账号。

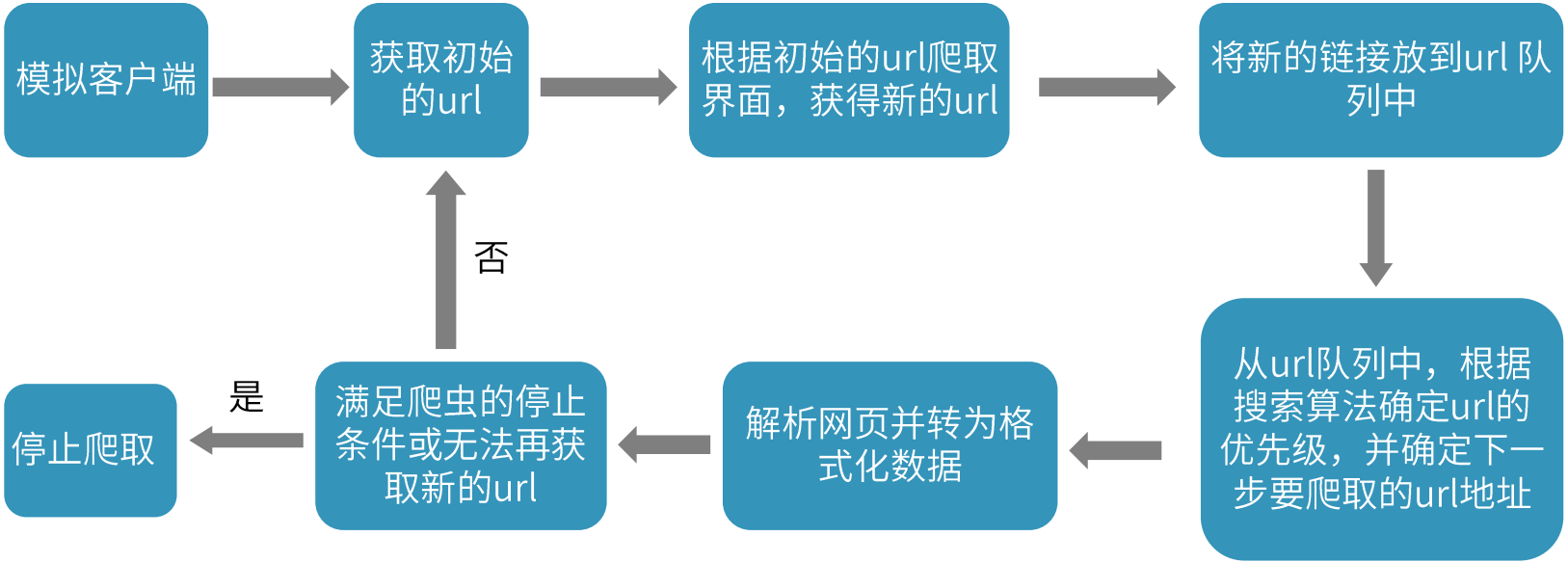
爬取策略制定

- **针对之前介绍的常见的反爬虫手段，可以制定对应的爬取策略如下。**
  - 发送模拟User-Agent：通过设置请求的User-Agent头部信息，将其伪装成一般用户使用的User-Agent，以绕过User-Agent校验。这可以增加请求的合法性，使其看起来更像是真实用户的请求。
  - 调整访问频度：通过备用IP测试网站的访问频率阈值，然后设置访问频率比阈值略低。这种方法既能保证爬取的稳定性，又能使效率又不至于过于低下。
  - 通过验证码校验：使用IP代理，更换爬虫IP；通过算法识别验证码；使用cookie绕过验证码。
  - 应对网站结构变化：只爬取一次时，在其网站结构调整之前，将需要的数据全部爬取下来；使用脚本对网站结构进行监测，结构变化时，发出告警并及时停止爬虫。
  - 通过账号权限限制：通过模拟登录的方法进行规避，往往也需要通过验证码检验。
  - 通过代理IP规避：使用代理IP池，定期更换爬虫的IP地址，以减少被网站监测和封禁的风险。使用高质量的代理IP可以提高匿名性和稳定性。

CONTENTS

爬虫流程

- 模拟客户端正常访问，快速自动化爬取大量数据的程序。
- 模拟是为了避免被服务器识别为爬虫程序被禁止。
- 客户端指的是：浏览器、app等。



Thank you!