总

结

文

档

|  |  |  |  |
| --- | --- | --- | --- |
| 版本 | 作者 | 版本描述 | 完成日期 |
| 1.0 | 安志斌 | 基本实现点评网数据抓取 | 2016.08.26 |

目录

[第一节 代理问题 3](#_Toc459814062)

[1.1 代理设置逻辑 3](#_Toc459814063)

[1.2 代理访问压力测试 5](#_Toc459814064)

[1.3 异常错误码处理 6](#_Toc459814065)

[第二节 记录相关问题 8](#_Toc459814066)

[2.1 遗漏记录分析 8](#_Toc459814067)

[2.2 重复记录分析 10](#_Toc459814068)

* + 1. 代理问题
  1. 代理设置逻辑

有些网站为了防止爬虫或者DDos攻击等，会记录每个IP的访问次数以及访问频率，比如允许一个IP在1s内访问10次等，对于这样一些反爬机制较强以及受法律保护的网站，往往需要通过代理IP的设置来对本地IP进行一定程度的隐匿。简单的说，代理就是换个身份去访问目标网站，而网络身份就是IP地址。

那么，这些代理从何而来呢？对于企业来讲可以直接购买代理IP，但是对于个人而言，这可能会造成资源浪费。好在网上有很多提供免费代理的IP网站，如<http://www.xicidaili.com/nn/1>，我们可以先编写一个简单的爬虫从这些网站上爬IP，然后再用这些代理去爬我们的目标网站。

1. 代理IP获取

本例中采用<http://gatherproxy.com/>作为代理获取网站，该网站还提供了可下载的程序来手动获取代理IP，链接<http://pan.baidu.com/s/1c1JFY9a>，该程序对于免费用户只提供了最大30个线程，大多数时候代理获取速度也是可以接受到。

下面以另一种方法——python脚本的形式获取此网站代理IP，代码见下网站：https://github.com/jerryan999/dianping/tree/master/dianping/utils

代码说明：

a).这里我们使用requests模块来请求url，它是python的一个非官方http库，其API明显优于urllib2，因此具有广泛的运用。

b).由于透明代理以及普通匿名代理会将本地的IP也发送出去，所以，为了最大限度保持匿名性，我们只获取高度匿名的代理IP

c). 使用网站<http://myip.dnsdynamic.org对代理IP>进行一定成功次数测试，如果成功通过测试就证明该代理可用度相对较高。

1. 爬虫代理中间件逻辑

很多时候，代理可能即使通过测试网站的测试，当真正部署到目标网站上却会出现问题；可能测试的时间段内某代理有效，而正使用它的时候却各种错误频出。所以，为了高效的运行爬虫程序，使用代理中间件对目标网站一边爬取页面信息一边同时对代理IP进行统计分析。

下面通过引进两个IP优质程度评价指标：绝对净成功次数和相对成功百分率，提供一种代理中间件设置策略。

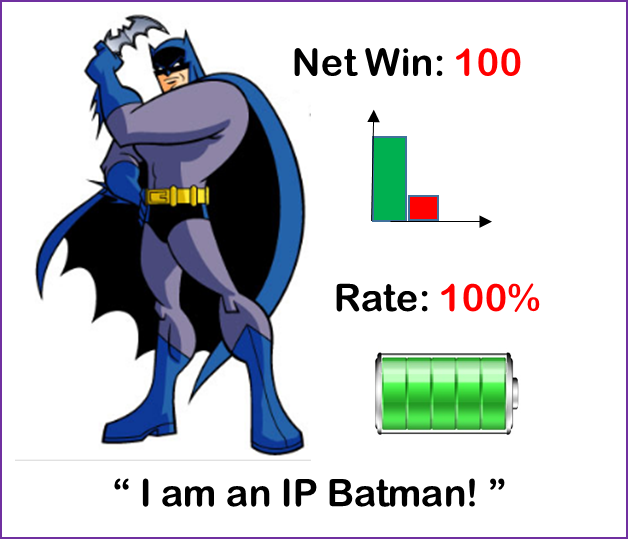
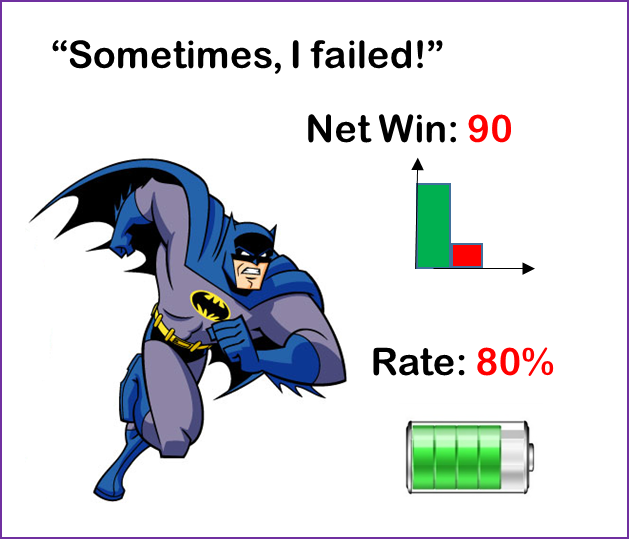
 

图1.1 代理IP类比

首先，很自然的我们可以想到一个评价指标——相对成功百分比，我们把一个IP比作是一位蝙蝠侠，如果他战斗过的每次战役（访问过的每个url）都可以赢得比赛（返回成功），那么他的成功百分比就是100%；而如果他偶尔输了某几次比赛（非成功返回如网络异常等），那么他的成功百分比必然就要小于100%。所以，我们可以设定当这一百分比低于某一值时（比如80%）时，这一蝙蝠侠需要退场休息（这一代理IP暂时停止使用）。

这一指标的计算方程如下：

式中：

—total request：某一IP总url请求次数

—net success request: 某一IP净请求成功次数

实际操作中，当我们获得一个新IP后，通过它对目标网站的请求成功与否进行统计就可以得到这一百分比。然而，这一算法存在一个较为明显的问题，假设某一IP是高质量的，但是如果第一次请求就失败了的话，其百分比就为0/1=0%,那么它立马就被出局了，而这样高质量的代理就被埋没了。

“为了解决这一问题，我们也许不应该从0开始计算净赢场次”，是的！这里，我们需要进行一个假设：获取来的代理是经过n次测试，测试成功百分比是100%，这样，当发生上述failed情况的时候，百分比就成了(。

下面以n=100来进行详细阐述某代理IP的运行期间各参数情况。

图1.2 某代理IP运行参数

运行期间可以分为三个阶段：

第一阶段，净成功返回请求计数和总请求数同步增加，两者的比值恒定为100%，体现优质代理连续成功返回的特性。

第二阶段，总请求数虽然增加，但是净成功返回请求计数由于这一阶段发生的失败请求而降低，百分比也降低

第三阶段，净成功返回请求计数达到下限（用户自己设定，比如100）或者百分比达到下限（比如80%），该代理暂时停止使用，即需要换别的代理继续请求目标网站。

* 1. 代理访问压力测试

针对目标网站对每个IP的访问次数的限制，如果我们能测试出这一门限值并在代理IP的使用中考虑到这一请求数的限制即可以提高爬虫的爬取效率。下面进行单个代理目标网站访问测试。

以代理IP：113.232.49.30:9529为例，表1.1列出了每次访问目标网站的时间及禁止前总访问成功次数统计：

表1.1代理113.232.49.30被禁止前历次访问最大成功次数表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | 访问开始时间 | 与初始时间差 | 访问次数 | 频率 次/s |
| 1 | 16:23:50 | 0 | 333 | 2.775 |
| 2 | 16:42:28 | 0:18:38 | 233 |  |
| 3 | 16:55:54 | 0:32:04 | 307 |  |
| 4 | 17:06:51 | 0:43:01 | 1438 | 1.1412 |
| 5 | 17:31:03 | 1:07:13 | 0 |  |
| 6 | 17:33:20 | 1:09:30 | 0 |  |
| 7 | 17:34:57 | 1:11:07 | 0 |  |
| 8 | 17:39:23 | 1:15:33 | 0 |  |
| 9 | 17:44:36 | 1:20:46 | 0 |  |

图1.3代理113.232.49.30被禁止前历次访问最大成功次数图

该代理前三次连续访问里，访问量达到300的时候就被禁止了；而第四次访问量达到1400才被禁止；并且第四次的访问频率比第一次的访问频率还小很多。由此可见，该目标网站反爬测试的门限值无法准确获取。

* 1. 异常错误码处理

1. 307异常

使用代理进行爬虫项目的时候，有时候会遇到重定向的问题，正常的重定向会将网址引导到新的url上，而出现问题的代理重定向则引导到一个莫名其妙的网址，比如图1.4显示的某个代理就重定向了一个陌生网页，而发生这一情况的时候，爬虫代理中间件并不认为该代理有问题，依然使用这一代理进行不休止的请求。

解决办法：将307错误排除在爬虫可允许遇到的返回码之外，即无论如何，当返回码为307时将告之此代理失效。

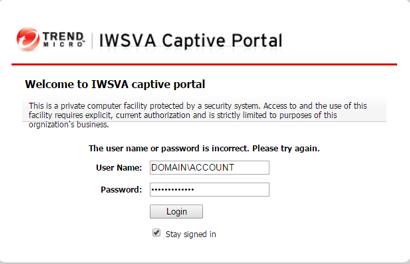


图1.4代理307错误.

1. 404，500，502异常

正常的404错误页面代表客户端在浏览网页时，所访问的页面不存在，即图1.5所展示的Not Found错误。在使用代理之后，有时候返回404时并非一定是页面没有找到，也有可能是页面正常存在而代理异常返回。如果不对这一情况进行考虑，必然导致目标网站很多url丢失。



图1.5 404错误

与此类似，500,502等服务器端发生的错误也存在如上假性错误的现象，也不能单纯的以为返回这些错误就一定是目标网站服务器的问题。

解决方法：针对以上两类问题， 我们增加一个对这些错误的真假判断：如果字符串“大众点评网”在返回的response中存在即真错误，反之为假错误，此时就需要切换另一代理。

3）403异常

403的处理比较简单，遇到此异常代表所用代理IP被目标网站服务器暂时禁止访问。

解决方法：立即切换代理IP

* + 1. 记录相关问题
  1. 遗漏记录分析

在大众点评网爬取过程中所遇到的主要记录遗漏可以分为两类：分类（二级或三级）标签未完整展示以及地区商圈分类问题，下面就这两类问题分别进行分析。

1. 分类标签未完整展示

大众点评网有十八个一级分类，如：美食、电影、休闲娱乐、酒店、丽人、爱车等。以美食为例，其又可以划分为火锅、咖啡厅、烧烤、面包甜点、其他等二级分类；其中其他一项又可以进一步细分为酒吧、茶馆、台湾菜、江西菜、创意菜等三级分类。

美食的二级分类-其他有25487个商铺，而其下所有的三级分类总和只有11430个商铺（图1.6、表1.2），可见漏掉了大约14000个店铺。这些漏掉的店铺由于找不到与之相比配的三级分类标签，所以没有办法抓取掉。



图1.6 商户二级、三级分类

表1.2 二级分类、三级分类商户总数对比表

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 三级分类 | 酒吧 | 茶馆 | 台湾菜 | 江西菜 | 创意菜 | 其他中餐 | 内蒙菜 | 西藏菜 | 徽菜 | 山西菜 | 总计 |
| 总数 | 1608 | 2392 | 201 | 142 | 661 | 5850 | 106 | 30 | 229 | 211 | 11430 |
| 二级分类 | / | / | / | / | / | / | / | / | / | / | 25487 |

1. 地区商圈问题

同样以一级分类美食，二级分类其他为例，如果我们不细分到三级分类而是以行政区进行分类展示，那么西城区的页面信息见图1.7，二级、三级分类商户总数见表1.3。



图1.7 商户二级、三级分类

表1.3 二级分类、三级分类商户总数对比表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 一级分类 | 二级分类 | 行政区 | 商圈 | 商户数 |
| 美食 | 美食其他不限 | 西城区 | 广外大街 | 151 |
| 美食 | 美食其他不限 | 西城区 | 右安门 | 33 |
| 美食 | 美食其他不限 | 西城区 | 月坛 | 47 |
| 美食 | 美食其他不限 | 西城区 | 广内大街 | 35 |
| 美食 | 美食其他不限 | 西城区 | 前门 | 90 |
| 美食 | 美食其他不限 | 西城区 | 新街口 | 38 |
| 美食 | 美食其他不限 | 西城区 | 虎坊桥 | 34 |
| 美食 | 美食其他不限 | 西城区 | 西四 | 44 |
| 美食 | 美食其他不限 | 西城区 | 复兴门 | 85 |
| 美食 | 美食其他不限 | 西城区 | 西直门/动物园 | 64 |
| 美食 | 美食其他不限 | 西城区 | 德外大街 | 53 |
| 美食 | 美食其他不限 | 西城区 | 南菜园/白纸坊 | 5 |
| 美食 | 美食其他不限 | 西城区 | 地安门 | 151 |
| 美食 | 美食其他不限 | 西城区 | 什刹海 | 232 |
| 美食 | 美食其他不限 | 西城区 | 西单 | 96 |
| 美食 | 美食其他不限 | 西城区 | 菜市口 | 14 |
| 美食 | 美食其他不限 | 西城区 | 陶然亭 | 3 |
| 美食 | 美食其他不限 | 西城区 | 牛街 | 23 |
| 美食 | 美食其他不限 | 西城区 | 阜成门 | 69 |
| 美食 | 美食其他不限 | 西城区 | 宣武门 | 45 |
|  |  |  |  | 1312 |

在此分类下，西城区显示总店铺数1972个，而实际各个商圈总合店铺数1312个，将其他行政区的店铺数同样方法统计见表1.4，可见针对二级分类美食其他，各个行政区总合12498个店铺，而理论店铺数23508个，漏掉的店铺很可能在位置划分上属于大行政区而不属于任何一个商圈，这样的店铺也是基本上没法爬取到的。

表1.4 各行政区二级分类、三级分类商户总数对比表

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 昌平区 | 朝阳区 | 大兴区 | 东城区 | 丰台区 | 海淀区 | 密云县 | 顺义区 | 通州区 | 西城区 | 总计 |
| 理论 | 2026 | 7422 | 1066 | 1878 | 2066 | 4151 | 787 | 840 | 1320 | 1952 | 23508 |
| 实际 | 694 | 4057 | 415 | 1281 | 1062 | 2268 | 775 | 213 | 421 | 1312 | 12498 |

* 1. 重复记录分析

很多时候同一商户会属于不同的一级分类，比如药店可以属于购物频道，也同时可以属于生活服务频道，两个频道爬取到的这一商户必然在数据库中存在重复，所以记录需要进行去重处理。

数据去重的逻辑如下：用SQL语句根据url筛选出重复的记录，然后通过这些url记录反查数据库并剔出掉重复的记录，保留信息最完整的一条记录。