修订日志：

|  |  |  |
| --- | --- | --- |
| 修改日期 | 修改人 | 备注 |
| 11.17 | 王文祥 | 完成数据采集 |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

**爬虫程序编写**

在本系统中，我们使用了豆瓣网中的电影数据及评论数据作为我们的数据源。为了保证推荐系统的电影数量、推荐算法的基础打分数据源规模和课堂要求的数据规模，我们爬取了1000部电影，每部电影选择约120条评论，共计十一万余条记录。

爬虫程序分为三个阶段。第一阶段，爬虫程序通过豆瓣的搜索接口“<https://movie.douban.com/j/new_search_subjects>”获取电影的简要信息。该网址可以通过指定数字来返回20条电影的简要信息，尽管这些简要信息远不够用于最终展示，但是每一条信息都提供了本电影在豆瓣中的链接和id，因此我们可以通过这些信息来进行进一步的爬取。第一阶段的爬虫只需要循环50次，其作用是确定1000部电影的id，即本系统最终会展示哪些电影。

第二阶段，爬虫程序根据第一阶段获取的每一部电影的豆瓣链接进行访问，提取其中当前电影的详细信息。这些信息就是本系统最终会在电影详情界面中展示出来的信息。第二阶段的爬虫将循环1000次，即获得1000部电影的每一部电影的详细信息。

第三阶段，爬虫程序通过访问豆瓣网中保存每一部电影的短评的url进行内容识别和保存。由于每一个短评页面（以网址<https://movie.douban.com/subject/26612258/comments?status=P>为例）都会展示20条评论，所以每一部电影需要访问6次便可以获得120条评论。经过实际运行，我们发现豆瓣网上的个别短评是没有打分的，而打分却是本系统在计算电影相似度时的一个重要依据，所以对于没有评分的短评，我们没有进行保存，故每部电影最终会保存的评论信息有略小于120条。第三阶段的爬虫将循环6000次，将获得1000部电影的前120条短评。

**html处理**

在对于从网页抓取到的内容进行处理时，我们使用了BeautifulSoup库进行网页内容的解析。我们使用了BeautifulSoup库配合lxml解析器对于抓取到的html内容直接进行解析，然后便可以通过其find\_all函数并指定要寻找的标签以获得所有此标签的内容的list。这样缩小范围后，便可以直接通过字符串处理和匹配的方法获得我们想要的各种电影属性及评论信息。

**防反爬虫机制**

根据网络资料，我们得知豆瓣网的反爬虫机制是同一ip限制日间访问次数不得超过40次/分钟，晚间访问次数不得超过60次/分钟。尽管这个数字不一定是准确数字，但是确实值得我们参考。经过理论计算，我们得出结论为按照上述频率可以在本项目的日程内获得我们所需的数据规模，故没有采用使用代理池或其他方法，而是只是每一次访问豆瓣网后就休眠3秒钟，然后继续循环访问。