吉林大学本科毕业设计（论文）开题报告

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 学 院 | 计算机科学与技术学院 | | | | 专业 | 计算机科学与技术 |
| 学生姓名 | 孟政元 | | | | 学号 | 21140821 |
| 指导教师 | 王康平 | | | | 职称 | 讲师 |
| 合作导师 |  | | 职称 |  | 单位 | 计算智能 |
| 设计（论文）题目 | | 网易云音乐用户数据的分析和可视化技术研究与实现 | | | | |
| 一、课题研究的背景和意义（综述国内外相关研究现状，阐述课题的研究目的、意义）  网易云音乐（NetEase Cloud Music）是中国[网易](https://zh.wikipedia.org/wiki/%E7%BD%91%E6%98%93)公司自2013年4月23日基于网易云服务推出的一个[音乐](https://zh.wikipedia.org/wiki/%E9%9F%B3%E4%B9%90)平台，除了提供主要的音乐播放、下载与查找服务外，还提供音乐[社交](https://zh.wikipedia.org/wiki/%E7%A4%BE%E4%BA%A4)功能（歌曲评论、声学指纹（听歌识曲）、根据历史播放记录的歌单推荐、地理位置识别等）。现支持iOS、Android、Windows、 Linux在内的多个操作系统[[9]](https://zh.wikipedia.org/wiki/%E7%BD%91%E6%98%93%E4%BA%91%E9%9F%B3%E4%B9%90#cite_note-%E6%96%B0%E6%B5%AA%E7%BD%91%E7%BD%91%E6%98%93%E4%BA%91%E9%9F%B3%E4%B9%90%E6%AD%A3%E5%BC%8F%E5%8F%91%E5%B8%83-9)。截至2017年11月，网易云音乐用户数量超过4亿，拥有超过1000万首歌曲，用户创建歌单达8000万个。  网易云音乐中的用户大致可以分为两类, 一类是普通听众, 另一类是歌手, 音乐创作人这类发布音乐的人.在整个用户群体中, 存在着许多的关联关系. 这些用户可以随意的互相关注, 而每个用户又有着自己独特的特征, 比如性别, 地域, 年龄等等.用户之间相互关注的情况可以用一个很大的有向图来表示, 用户是有向图中的点, 每一个用户对另外一个用户的关注就是有向图的一条边. 这篇论文对部分用户信息(大概N用户)进行了采集,分析了用户的个体特征, 用户之间相互关注, 用户之间间接性的相互关联(基于floyd算法)等等在地域上的分布情况,并且基于分析结果进行回归预测, 从而估计整个用户群体的情况, 最后进行可视化展现, 使得结果直观明了. | | | | | | |
| 二、课题研究已有的工作基础（总结归纳本人的学习、科研、实习等成果，以及已掌握的前人资料，简述自己初步的学术见解，附证书、报告、外文文献）  喜欢钻研Web前端, 数据可视化, 网络安全方面的的技术, 时常关注相关的标准技术的最新进展和方向.喜欢自己动手实践, 通过所学习的知识做一些自己感兴趣的东西并在网络上与他人分享  作为吉林大学2016年大创国家级项目(YOYO笔记)负责人（附件1），负责相关的系统的代码的编写和系统的搭建，拥有一定的实践经验。  这个项目主要由数据采集模块，数据分析模块，数据可视化模块三个部分构成。其中，数据的采集由自己编写针对网易云音乐的爬虫来完成，数据分析模块基于MongoDB数据库和Node.JS，数据可视化模块基于当前最流行的D3.js来构建  其中，数据可视化作为当下一个热门的技术和研究方向，有着许多非常成熟的成果，在项目进行过程中可以提供许多的参考和借鉴。但是在数据采集和分析方面的资料和成果并不是很多。主要是因为这方面偏重于实践，具有比较强的针对性，很多的成果都没有相应的论文或者文献记录，反而是存在于各个项目的官方网站和文档上面。因此需要在项目进行的过程中广泛的查阅资料和结合具体的实践经验。  虽然存在一些比较具有挑战性的难题，但是我相信只要多付出努力去寻求解决的办法，这些困难是可以找到解决的方法的。 | | | | | | |

|  |
| --- |
| 三、研究的内容及可行性分析   1. **数据的爬取**   跨站请求伪造 (CSRF), 伪造请求头 , 调用官方 API. 爬虫通过提供与正常请求相同的参数, 通过post方法向网易云的后台发起请求, 从而获取相应的数据.  由于网易云的后台设置有反爬虫的机制, 所以不能爬取的太快, 否则IP会被封掉. 但是, 由于做分析需要的数据量较大,爬取的速度太慢无法再短时间内怕取到足够的数据. 因此必须采取一定的策略加快数据爬取的速度. 根据网易云的反爬虫机制, 单个IP地址一分钟最多发送20个请求. 因此, 这里采用多个代理并行爬取的方式. 同时, 在http请求头中, 随机化user-agent, 更好的把爬虫伪装成正常用户.  在搭建代理的时候, 由于我们的代理是为了隐藏爬虫, 所以必须使用http正向代理中的高匿代理, 否则会被网易云的后台发现异常. 这里使用的是自己使用Node.js编写的一个简单的匿名代理.  代理的原理是客户端先将请求发送到代理服务器, 由代理服务器向目标服务器发起请求, 并将请求结果发回客户端.  代码如下:  const PROXY\_PORT = 65535  const http = require('http')  const net = require('net')  const url = require('url')  function request(cReq, cRes)  {  const u = url.parse(cReq.url)  console.log(`[http.request] ${cReq.method} ${u.hostname}:${u.port || 80} ${u.path}`)  const options =  {  hostname : u.hostname,  port : u.port || 80,  path : u.path,  method : cReq.method,  headers : cReq.headers,  }  const pReq = http.request(options, function(pRes)  {  cRes.writeHead(pRes.statusCode, pRes.headers)  pRes.pipe(cRes);  }).on('error', function(e)  {  console.error(e)  cRes.end()  });  cReq.pipe(pReq)  }  function connect(cReq, cSock)  {  console.log(`[http.connect] ${cReq.url}`)  const u = url.parse('http://' + cReq.url);  const pSock = net.connect(u.port, u.hostname, function()  {  cSock.write('HTTP/1.1 200 Connection Established\r\n\r\n');  pSock.pipe(cSock);  }).on('error', function(e)  {  console.error(e)  cSock.end();  });  cSock.pipe(pSock);  }  http.createServer()  .on('request', request)  .on('connect', connect)  .on('error', function(err)  {  console.error(err)  })  .listen(PROXY\_PORT, function()  {  console.log(`proxy server online: http://localhost:${PROXY\_PORT}`)  });  由于需要爬取的数据量较大, 传统的单线程同步程序耗时非常长, 所以必须采取并发机制. 这里采用的是Node.js基于事件循环的异步非阻塞并发机制. 在使用并发机制的时候, 需要使用互斥锁来确保程序有序运行.   1. 数据的存储   数据存储在自己搭建的MongoDB数据库中, 共五个collection。 MongoDB数据库是一种结构自由, 无schema的文档数据库, 便于存储这种对象化的用户数据.   1. 数据的结构   用户的个人profile存储在user这个collection中, 主要的结构如下:  {  "\_id":"5a51ba300604d374f28af6fb",  "profile":  {  "userId":363516402,  "gender":0,  "birthday":-2209017600000,  "city":220100  },  "\_\_processing":  {  "follow":false,  "followed":false,  "playlist":false,  "detail":false  },  "\_\_processed":  {  "follow":true,  "followed":true,  "playlist":false,  "detail":true  }  }  各个字段的含义如下:   1. \_id(数据库主键) 2. profile.userId(用户id) 3. profile.gender(用户性别, 0为男性, 1为女性) 4. profile.birthday(用户的出生日期) 5. profile.city(用户所在的城市) 6. \_\_processing(互斥锁, 用于并发) 7. \_\_processed(表示这个用户是否已经处理过了)   用户的个人profile存储在follow这个collection中, 主要的结构如下:  {  "\_id":"5a4b488c954454ebecda66cb",  "from":"test",  "to":"test1",  "updatedAt":1515168662545  }  各个字段的含义如下:   1. \_id(数据库主键) 2. from(关注别人的那个人的userId) 3. to(被关注的那个人的userId) 4. updatedAt(这条记录的插入时间) 5. 数据的分析   1. 用户之间相互关注的情况的对比分析  2. 采用floyd算法进行计算用户之间的距离  3. 用户在全国范围内的分布   1. 数据可视化   通过D3对分析的结果进行可视化，形成可以交互，直观明了的数据图表和基于地理位置的可视化图表。  本项目基于现在已经比较成熟的网络爬虫，NOSQL数据库以及数据可视化等技术，具有较好的可行性。虽然在一些方面需要解决具有较强的针对性的问题，但是这些问题并不会对项目造成不可解决的影响，不会导致项目无法完成。 |

|  |
| --- |
| 四、论文拟解决的关键问题及难点  这个项目的难点主要有3个：   1. 数据的采集   由于是要采集网易云音乐官方未公开的用户数据，所以只能使用网络爬虫来进行自动化的采集。为了满足分析的需求，数据量的规模还是比较大的，其中用户个人信息需要几十万条，用户关注信息更是需要百万条，这些数据不可能通过手工的方式去完成，只能使用网络爬虫来进行爬取。 |
| 五、研究方法与技术路线(重点论述技术方案) |
| 1. 论文的进度安排 |
| 七、毕业设计研制报告或毕业论文撰写提纲（初步） |
| 八、主要参考文献 |
| 九、指导教师意见  签字： 20 年 月 日 |
| 十、开题审查小组意见  （要求具体意见，对前8项进行评价，结论：通过，不通过）  开题小组评分（满分为20分）：  开题小组组长签字： 20 年 月 日 |

**注：**专业一定要填写全称。[专业名称：计算机科学与技术、计算机科学与技术（网络与信息安全）、物联网工程、理科试验班（唐敖庆计算机班）]