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。  由于网易云音乐后端具有非常严格的反爬虫措施，对于非正常的请求一律会屏蔽掉，所以只能采取跨站请求伪造的方式来爬取，同时针对网易云音乐针对单个IP地址限制请求频率的问题，采用多个代理IP并行爬取的策略来进行爬取。这些措施需要爬虫程序针对性的做出相应的修改来使用这些措施以规避网易云音乐的反爬虫措施，对多个模块之间的整合具有一定的挑战。同时代理服务器集群的的搭建和调度也是具有一定的难度的。   1. **数据的分析**   数据爬取之后会放置在自己搭建的MongoDB数据库之中，在对数据进行分析的时候回直接从数据库之中提取数据。在数据的分析之中由于基于floyd算法的路径长度时间复杂度为O(n^3)，这个时间复杂度在数据量比较大的时候对计算的要求非常高。目前打算采取的策略是采取非易失性存储代替易失性存储同时租用一台计算能力非常强的云服务器来进行计算。同时从网易云音乐爬取的数据需要经过一定的处理之后才能进行分析和可视化。   1. **数据的可视化**   由于需要非常直观的进行数据分析结果的展示，而且展示需要基于地理位置进行优化，使得结果更加的直观明了。这个具有非常强的针对性，没有现成的库可以完成这些工作，需要调用非常底层的API来完成数据可视化图表的绘制。 |
| 五、研究方法与技术路线(重点论述技术方案)   1. 主要的编程语言   项目的主要编程语言是Node.js。  爬虫程序采用的是基于Node.js的request库来进行编写的，数据爬取后经过相应的处理存入搭建好的MongoDB数据库之中。  Node.js是在Chrome的V8 JavaScript引擎上构建的JavaScript运行时。 Node.js使用事件驱动的非阻塞I / O模型，使其轻量且高效。 Node.js的软件包生态系统npm是全球最大的开源库生态系统[9]。  同时用于协助爬虫的分布式代理服务器集群也是基于Node.js来编写的。   1. 数据库   数据库采用的是MongoDB  MongoDB 是一个高性能，开源，无模式的文档型数据库，是当前noSql数据库产品中最热门的一种。它在许多场景下用于替代传统的关系型数据库或键值对存储方式，MongoDB是用C++开发，MongoDB的官方网址为 <http://www.mongodb.org/>  NoSQL，全称是”Not Only Sql”,指的是非关系型的数据库。这类数据库主要有这些特点：非关系型的、分布式的、开源的、水平可扩展的。原始的目的是为了大规模 web 应用，这场全新的数据库革命运动早期就有人提出，发展至 2009 年趋势越发高涨。NoSQL 的拥护者们提倡运用非关系型的数据存储，通常的应用如：模式自由、支持简易复制、简单的 API、最终的一致性（非 ACID）、大容量数据等。NoSQL 被我们用得最多的当数 key-value 存储，当然还有其他的文档型的、列存储、图型数据库、xml 数据库等。相对于目前铺天盖地的关系型数据库运用，这一概念无疑是一种全新思维的注入[10]。   1. 服务器   MongoDB数据库，爬虫程序以及数据分析程序都是搭建在阿里云的云服务器上。   1. 数据的可视化   数据的可视化是基于当下最流行的D3.js，用Web技术构建，通过AJAX从后端获取分析好的数据，然后基于可视化技术在网页上呈现出来，并且可以在浏览器中进行交互。  D3.js（D3或Data-Driven Documents）是一个用动态图形显示数据的[JavaScript](https://zh.wikipedia.org/wiki/JavaScript)库，一个[数据可视化](https://zh.wikipedia.org/wiki/%E6%95%B0%E6%8D%AE%E5%8F%AF%E8%A7%86%E5%8C%96)的工具。兼容[W3C](https://zh.wikipedia.org/wiki/W3C)标准，并且利用广泛实现的[SVG](https://zh.wikipedia.org/wiki/SVG)，JavaScript，和[CSS](https://zh.wikipedia.org/wiki/CSS)标准。它是早期的[Protovis](https://zh.wikipedia.org/w/index.php?title=Protovis&action=edit&redlink=1)框架的继承者。与其他的类库相比，D3对视图结果有很大的可控性[11]。 |
| 1. 论文的进度安排   2018年1月28日 —— 2018年2月28日 初步定题，开始着手准备，搜集资料。  2018年3月1日 —— 2018年3月15日 确定论文大纲，进行关键功能的分析设计。  2018年3月15日 —— 2018年3月31日 完成系统整体设计以及相应模块的  详细设计，并进行编码实现。  2018年4月1日 —— 2018年4月10日 系统功能测试试验。  2018年4月11日 —— 2018年5月1日 系统进行整合，完成最终的效果  2018年5月2日以后 修改论文初稿，形成最终的毕业论文。填写毕业设计结题所需的相关材料，制作答辩PPT，准备答辩。 |
| 七、毕业设计研制报告或毕业论文撰写提纲（初步）  封面  [摘要](http://localhost:3000/#/?id=%e6%91%98%e8%a6%81)  [目录](http://localhost:3000/#/?id=%e7%9b%ae%e5%bd%95)   1. [数据的爬取](http://localhost:3000/#/?id=_1-%e6%95%b0%e6%8d%ae%e7%9a%84%e7%88%ac%e5%8f%96)    1. [原理](http://localhost:3000/#/?id=_1-%e5%8e%9f%e7%90%86)    2. [反爬虫策略](http://localhost:3000/#/?id=_2-%e5%8f%8d%e7%88%ac%e8%99%ab%e7%ad%96%e7%95%a5)    3. [代理程序](http://localhost:3000/#/?id=_3-%e4%bb%a3%e7%90%86%e7%a8%8b%e5%ba%8f)    4. [Node.js基于事件循环的异步非阻塞并发机制](http://localhost:3000/#/?id=_4-nodejs%e5%9f%ba%e4%ba%8e%e4%ba%8b%e4%bb%b6%e5%be%aa%e7%8e%af%e7%9a%84%e5%bc%82%e6%ad%a5%e9%9d%9e%e9%98%bb%e5%a1%9e%e5%b9%b6%e5%8f%91%e6%9c%ba%e5%88%b6) 2. [数据的存储](http://localhost:3000/#/?id=_2-%e6%95%b0%e6%8d%ae%e7%9a%84%e5%ad%98%e5%82%a8)    1. [数据库](http://localhost:3000/#/?id=_1-%e6%95%b0%e6%8d%ae%e5%ba%93) 3. [数据的结构](http://localhost:3000/#/?id=_3-%e6%95%b0%e6%8d%ae%e7%9a%84%e7%bb%93%e6%9e%84)    1. [user](http://localhost:3000/#/?id=_1-user)    2. [follow](http://localhost:3000/#/?id=_2-follow) 4. [数据分析](http://localhost:3000/#/?id=_4-%e6%95%b0%e6%8d%ae%e5%88%86%e6%9e%90)    1. [用户之间相互关注的情况的对比分析](http://localhost:3000/#/?id=_1-%e7%94%a8%e6%88%b7%e4%b9%8b%e9%97%b4%e7%9b%b8%e4%ba%92%e5%85%b3%e6%b3%a8%e7%9a%84%e6%83%85%e5%86%b5%e7%9a%84%e5%af%b9%e6%af%94%e5%88%86%e6%9e%90)       1. [采用floyd算法进行计算用户之间的距离](http://localhost:3000/#/?id=_2-%e9%87%87%e7%94%a8floyd%e7%ae%97%e6%b3%95%e8%bf%9b%e8%a1%8c%e8%ae%a1%e7%ae%97%e7%94%a8%e6%88%b7%e4%b9%8b%e9%97%b4%e7%9a%84%e8%b7%9d%e7%a6%bb)       2. [用有向图展示用户之间的关联关系](http://localhost:3000/#/?id=_1-%e7%94%a8%e6%9c%89%e5%90%91%e5%9b%be%e5%b1%95%e7%a4%ba%e7%94%a8%e6%88%b7%e4%b9%8b%e9%97%b4%e7%9a%84%e5%85%b3%e8%81%94%e5%85%b3%e7%b3%bb)       3. [计算平均经过多少人, 两个用户之间可以产生关联](http://localhost:3000/#/?id=_2-%e8%ae%a1%e7%ae%97%e5%b9%b3%e5%9d%87%e7%bb%8f%e8%bf%87%e5%a4%9a%e5%b0%91%e4%ba%ba-%e4%b8%a4%e4%b8%aa%e7%94%a8%e6%88%b7%e4%b9%8b%e9%97%b4%e5%8f%af%e4%bb%a5%e4%ba%a7%e7%94%9f%e5%85%b3%e8%81%94)       4. [矩阵的存储](http://localhost:3000/#/?id=_1-%e7%9f%a9%e9%98%b5%e7%9a%84%e5%ad%98%e5%82%a8)       5. [矩阵的运算](http://localhost:3000/#/?id=_2-%e7%9f%a9%e9%98%b5%e7%9a%84%e8%bf%90%e7%ae%97)    2. [用户在全国范围内的分布](http://localhost:3000/#/?id=_3-%e7%94%a8%e6%88%b7%e5%9c%a8%e5%85%a8%e5%9b%bd%e8%8c%83%e5%9b%b4%e5%86%85%e7%9a%84%e5%88%86%e5%b8%83)       1. [男女分布](http://localhost:3000/#/?id=_1-%e7%94%b7%e5%a5%b3%e5%88%86%e5%b8%83)       2. [年龄分布](http://localhost:3000/#/?id=_2-%e5%b9%b4%e9%be%84%e5%88%86%e5%b8%83) |
| 八、主要参考文献  [1] 李希娟. 大数据时代下的数据可视化研究[D]. 河北大学, 2014.  [2] 刘勘, 周晓峥, 周洞汝. 数据可视化的研究与发展[J]. 计算机工程, 2002, 28(8):1-2.  [3] 陈建军, 于志强, 朱昀. 数据可视化技术及其应用[J]. 红外与激光工程, 2001, 30(5):339-342.  [4] 石昊苏, 韩丽娜. 数据可视化技术及其应用展望[C]// 全国自动化新技术学术交流会会议论文集(一). 2005.  [5] 韩子良, 毕妤. 数据可视化在数据挖掘中的应用[J]. 计算机应用与软件, 2003, 20(11):71-73.  [6] 吴猛. 基于Web的数据可视化技术初探[J]. 福建电脑, 2007(12):58-59.  [7] 涂聪. 大数据时代背景下的数据可视化应用研究[J]. 电子制作, 2013, 47(5x):118-118.  [8] 赵聪. 可视化库D3.js的应用研究[J]. 信息技术与信息化, 2015(2):107-109.  [9] https://nodejs.org/en/  [10] http://www.cnblogs.com/shaosks/p/5644129.html  [11] <https://zh.wikipedia.org/wiki/D3.js> |
| 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）]