大连海事大学

毕 业 论 文

二〇一七年六月

基于浏览器历史记录的上网

行为预测算法研究

专业班级： 软件工程3班

姓 名： 赵松辉

指导教师： 高健

信息科学技术学院

摘 要

随着互联网的技术的大规模普及和发展，互联网已经渗入到我们每个人日常生活的方方面面，尤其是我们国家在互联网领域近十年来的发展，日新月异，发展成就举世瞩目，网民数量也达到了空前的规模。由于互联网上的信息量爆炸式增长，也带来了数据冗余和垃圾信息等问题，用户在上网时可能很难找到自己真正需要的资料。作为计算机专业的学生，在日常的学习研究中，对此现象深有感触。因此，决定进行这方面的研究，以提高用户在上网过程中信息检索的效率。

本文进行研究的大概的逻辑过程描述如下：使用浏览器的扩展程序采集每个用户的上网历史记录，服务器端得到这些数据之后将其保存在数据库中；接着对历史记录的网页标题进行分词，分词之后去除其中的停止词（Stop words），得到每条记录的关键词；然后以用户为单位，将单个用户的所有历史记录的关键词保存在关键词文本文件中，一个用户对应一个关键词文本文件。

提取出每个用户的关键词文本文件的兴趣词——词频最高的前K个词，再进行用户之间兴趣词的横向比较。拥有相同兴趣词最多的两个用户，我们就认为他们的兴趣最相近，然后就给他们其中一个用户推荐另一个用户。

本文选择在两种场景下进行网址推荐，一种是当用户主动点击扩展程序按钮时，向用户呈现出推荐的结果，这种是非实时的推荐；另一种是在用户使用搜索引擎查找问题时进行推荐，推荐的结果以弹窗的形式呈现，这种是实时的推荐。

关键词：上网行为分析；数据挖掘；聚类；推荐网址；文本挖掘；上网历史记录；Chrome扩展程序；用户协作

ABSTRACT

With the large-scale popularization and rapid development of the technology of the Internet, the Internet has infiltrated every aspect of our daily life, especially our country in the Internet field in recent years, the development of the ever-changing achievements of the world's attention, the number of Internet users has reached an unprecedented The scale. As the amount of information on the Internet explosive growth, but also brought data redundancy and spam and other issues, users in the Internet may be difficult to find their real needs of the information. As a student majoring in software, in the daily study and research, I deeply feel this phenomenon. Therefore, the decision to conduct this research to improve the user in the process of Internet access to the efficiency of information retrieval.

The approximate logical process of this paper is described as follows: Use the browser's extension to collect the user's Internet history, and the server will save the data in the database; then the word page of the history of the word segmentation, word segm-entation After the removal of one of the stop words (Stop words), get each record of the keyword; and then the user as a unit, all the user's history of all the keywords stored in the keyword text file, a user corresponds to a keyword text file.

Extract the keyword of each user's keyword text file - the highest word of the former K words, and then the user between the words of interest in the horizontal comparison. Have the same interest in the two most users, we think that their interest is closest, and then give them one of the users recommend another user.

This article chooses to make a URL recommendation in two scenarios, one is to give the user a recommended result when the user clicks the Extensions button, which is a non-real-time recommendation; the other is when the user uses the search engine to find the problem, recommended results based on the key words in the problem in the form of a window, this is a real-time recommendation.

Keywords: Internet behavior analysis, Data mining, Clustering, Websites-recommendation, Text mining, Historic records in internet browser, Chrome extension, User collaboration

目 录

目 录 III

第1章 绪论 1

1.1 研究背景 1

1.2 研究内容 1

1.3 程序和方法 2

1.4 本章小结 2

第2章 收集历史数据 3

2.1 浏览器的选择 3

2.2 Chrome扩展程序 3

2.2.1 基础 3

2.2.2 文件 4

2.2.3 架构 5

2.2.4 跨域请求 7

2.3 用于采集数据的扩展程序的设计与实现 7

2.3.1 清单文件 7

2.3.2 Popup页面 9

2.3.3 选项页面 10

2.4 发送数据 10

2.5 数据库的设计 11

2.5.1 数据库的选择 11

2.5.2 数据库关系表的建立与ER图的设计 12

2.6 本章小结 15

第3章 历史数据预处理 16

3.1 数据清理 16

3.1.1 清理非网络数据 16

3.1.2 清理私人数据 17

3.1.3 清理公共网站 17

3.1.4 清理无效记录 17

3.1.5 具有校园特色的记录 18

3.2 数据变换 18

3.2.1 数据变换概述 18

3.2.2 分词工具的选择 18

3.2.3 提取网页标题中的关键词——分词 19

3.2.4 提取用户最感兴趣的关键词 21

3.3 本章小结 21

第4章 预测并推荐数据 22

4.1 预测感兴趣的用户 22

4.2 预测感兴趣的网址 23

4.2.1 基于多用户协作的推荐策略 23

4.2.2 基于搜索文本的推荐策略 24

4.3 本章小结 24

结论 25

参 考 文 献 26

致 谢 28

绪论

研究背景

随着互联网技术的发展和普及，网络已经渗入我们生活的方方面面，衣食住行等等，甚至可以说，日常生活的一切需求都能在网络上得到满足。中国互联网络信息中心（CNNIC）发布的第39次《中国互联网络发展状况统计报告》显示，截至2016年12月，我国网民规模达7.31亿，普及率达到53.2%，超过全球平均水平3.1个百分点，超过亚洲平均水平7.6个百分点。中国网民规模已经相当于欧洲人口总量。其中，手机网民6.95亿，连续三年增长率超过10％。

与此同时，互联网的信息量也正在爆炸式地增长，信息的种类日益繁多，用户在访问互联网的时候可能很难找到自己感兴趣或有价值的信息。对于从事学习研究工作的用户，这种现象显得尤为严重。特别是在查找开发中遇到的问题时，仅靠搜索引擎可能显得非常无力。

近年来，大量相关学者对于网络用户行为的分析都展开了各种层次的研究，他们研究的主要目的是为商业推广提供价值信息，从而吸引网络用户的点击量，具有很强的商业性质。通过采集用户上网行为数据，对其进行分析，获取用户的购物喜好，在用户浏览网页时弹出广告，引诱用户点击，从而达到客户的商业目的——让更多的网络用户访问到自己的网站。这种研究在互联网时代也无可厚非，无疑是互联网能够使商人更轻松地吸引客户、赚到钱最好的例证。然而，通常大多数网络用户对于浏览网页中所遇到的商业推广广告是比较反感的，甚至怀有抵触心理。因此，一些不良的商业推广对互联网用户是有害无利的，比如，去年比较火热的“魏则西事件”，就彰显出了百度商业推广的弊端。

当然，除了商业广告推广，也有一些学者研究了新闻的推荐模型，用来从互联网海量的信息资源中提取出符合用户兴趣的新闻，并将其推荐给用户。本文的研究灵感就来源于新闻推荐模型。

研究内容

本文要探讨的并非是如何向用户进行商业广告的推销，而是通过采集大量网络用户的上网行为数据，分析每个用户的上网兴趣，并对用户的上网行为数据进行比较，找出一些上网浏览数据相似度比较高的记录，从而帮助用户找到与他们兴趣相近的人，再推荐给用户。举个例子吧，比如现已收集到了A、B、C、D四个人的上网行为数据，经过对比发现A、B、C三人的数据相似度比较高，那么我们就会向A用户推荐B和C用户，同时也会向B用户推荐A和C用户，向C推荐A用户和B用户。

但是，具体的数据相似度的度量也是本文所要研究的重要内容，并非简单的直接比较每个用户的历史记录。在采集到用户的历史记录数据之后，首先对其进行预处理，已达到数据缩减和数据格式优化的效果，然后再进行相似度的分析。

由于本文所做的成果并不是一个社交软件，因此用户之间不能直接交流。我们会让用户自愿填写他们的社交账号资料（现在比较流行的社交软件的账户资料——微信号和QQ号），向用户推荐的也是其他用户所填写的社交账号，用户可以根据需要自行添加好友。

程序和方法

本文的实践成果大体分为两个部分：数据收集采用基于Google Chrome浏览器的Chrome扩展程序，我开发出了自己的扩展程序——FindX；服务器端采用Java Web + Tomcat的结构处理采集到的数据以及负责与数据库的访问，使用Python语言进行数据分析和推荐算法。

本章小结

本章主要探讨了当前各界学者在用户上网行为领域的研究现状，以及本文即将研究的内容，也对所用到的程序和研究的方法做了简要的介绍。

收集历史数据

浏览器的选择

收集网络用户的数据，在众多的网络浏览器中，我选择谷歌浏览器（Google Chrome）。谷歌浏览器是谷歌公司基于Webkit开发的一款功能强大的Web浏览器。从某些方面来看，谷歌浏览器已经远远超越了一般的Web浏览器，它成为了一个广阔的平台，甚至成为了一个操作系统。谷歌浏览器集成了Webkit内核对HTML的高速渲染能力，而且谷歌公司自行开发的V8引擎使得JavaScript在谷歌浏览器中的执行效率大幅提升，这使得更加高级复杂的JavaScript程序在谷歌浏览器中运行成为可能。

谷歌浏览器除了页面渲染的速度快、JavaScript执行速度快以外，另一大特点就是支持开发者为其编写各种各样的扩展程序来扩充其功能，支持用HTML5编写桌面程序，这使得谷歌浏览器变得更加强大，也吸引了更多开发者的兴趣。

另外，谷歌公司也提供了扩展程序开发的API，并且相当丰富，关键是有人把谷歌官方提供的API文档翻译成了中文供开发者学习参考。这些文档从方方面面详细地讲述了如何开发谷歌浏览器的扩展程序，这让开发人员很容易上手入门，也大大降低了扩展程序开发的门槛。

综上所述，在众多浏览器中，谷歌浏览器成为我进行毕业设计的不二选择。

Chrome扩展程序

2.2.1 基础

扩展程序其实在本质上还是网页，在开发过程中也能调用谷歌浏览器为HTML网页提供的所有API，例如XMLHttpRequest、JSON、HTML 5 等等。但是，扩展程序与网页还不太一样，它不仅仅是网页。扩展程序是由一些文件，包括HTML、CSS、JavaScript、图片以及其他任何开发人员所需要的文件，经过zip压缩打包得到的crx格式的文件，用来为谷歌浏览器增加功能。扩展程序可以通过内容脚本或跨站XMLHttpRequest与网页或者服务器交互，也可以以编程的方式与浏览器功能（例如书签、标签页、cookie等）交互。

许多扩展程序都能以浏览器按钮或页面按钮的形式为 Google Chrome 浏览器添加用户界面，但是每个扩展程序最多只能有一个浏览器按钮或页面按钮。当扩展程序与大部分网页相关时选择使用浏览器按钮，当扩展程序的图标显示还是消失需要取决于具体网页时应该选择使用页面按钮。下图2.1展示了扩展程序在浏览器中三种不同的呈现方式：

图2.1 扩展程序在浏览器中三种不同的呈现方式

扩展程序也可以以其他形式呈现用户界面，例如在谷歌浏览器的右键菜单中添加内容，提供选项页面，或者利用内容脚本动态更改页面的显示方式。

2.2.2 文件

每一个扩展程序包含以下文件：

一个清单文件

一个或多个 HTML 文件（除非扩展程序是一个主题背景）

可选：一个或多个 JavaScript 文件

可选：扩展程序需要的任何其他文件，例如图片

当编写扩展程序时，需要将所有的文件放在一个文件夹中。当发布扩展程序时，该文件夹的内容将被压缩成一个特殊的 ZIP 文件，以 .crx 为后缀。如果通过 Chrome 开发者信息中心上传扩展程序，该 .crx 文件会自动创建。如果无需上传至应用商店时，可以开启谷歌浏览器的开发人员模式。

引用文件

可以在扩展程序中放置您需要的任何文件，但是如何使用它们呢？通常，您可以通过相对 URL 引用文件，就像在普通的 HTML 页面中那样。下面是一个例子，引用位于子文件夹 images 中名为 myimage.png 的文件。

<img src="images/myimage.png">

清单文件

清单文件名为 manifest.json ，提供有关扩展程序的各种信息，比如版本信息、扩展程序的描述、脚本文件的位置、权限信息等等。以下是一个典型的颇具学习参考价值的清单文件，用于一个浏览器按钮，它将会访问来自 google.com的信息：

{

"name": "我的扩展程序",

"version": "2.1",

"description": "从 Google 获取信息。",

"icons": { "128": "icon\_128.png" },

"background": {

"persistent": false,

"scripts": ["bg.js"]

},

"permissions": ["http://\*.google.com/", "https://\*.google.com/"],

"browser\_action": {

"default\_title": "",

"default\_icon": "icon\_19.png",

"default\_popup": "popup.html"

}

}

2.2.3 架构

许多扩展程序有一个后台网页，它是一个包含扩展程序主要逻辑的不可见页面。扩展程序也可以包含其他页面，展现扩展程序的用户界面。如果扩展程序需要与用户加载的网页交互（相对于包含在扩展程序中的页面），扩展程序必须使用内容脚本。

后台网页

下图所示的浏览器至少安装了两个扩展程序：一个浏览器按钮（黄色图标）和一个页面按钮（蓝色图标）。浏览器按钮和页面按钮都有后台页面。下图显示了浏览器按钮的后台页面，由 background.html 定义，并且包含在这两个窗口中控制浏览器按钮的 JavaScript 代码。

图2.2 浏览器按钮的后台页面

后台网页分两种：持续运行的后台网页与事件页面。正如它们的名称所述，持续运行的后台网页保持打开状态，事件页面根据需要打开与关闭。除非您绝对需要您的后台网页一直运行，请首选事件页面。

用户界面网页

扩展程序可以包含普通的 HTML 网页，用来显示扩展程序的用户界面。例如，浏览器按钮可以包含弹出菜单，通过 HTML 文件实现。任何一个扩展程序都可以有选项页面，让用户自定义扩展程序的工作方式。另外一种特殊页面是替代页面。最后，您可以使用 tabs.create 或 window.open() 来显示扩展程序中的任何其他 HTML 文件。

扩展程序中的 HTML 网页可以互相访问其他页面的全部 DOM，并且可以互相调用函数。

下图2.3显示了浏览器按钮弹出菜单的架构。弹出菜单是由一个 HTML 文件（popup.html）定义的网页，该扩展程序也正好有一个后台网页（background.html）。弹出窗口不用重复后台网页中的代码，因为弹出窗口可以调用后台网页上的函数。

图2.3 浏览器按钮弹出菜单的架构

内容脚本

如果您的扩展程序需要与网页交互，您就需要使用内容脚本。内容脚本是一些 JavaScript 代码，它们在浏览器中已加载页面的上下文中执行。您应该将内容脚本视为已加载页面的一部分，而不是打包在一起的扩展程序（它所属的扩展程序）的一部分。

内容脚本可以读取浏览器访问的网页的细节，并且可以修改页面。在下图2.4中，内容脚本可以读取并且修改显示的网页的 DOM。然而，它不能修改所属扩展程序后台网页的 DOM。

内容脚本并不是完全与所属扩展程序隔离的。内容脚本可以与所属扩展程序交换消息，如下图2.5箭头所示。例如，每当在浏览器页面中发现 RSS 供稿时，内容脚本可以发送消息，反过来后台页面也可以发送消息要求内容脚本更改浏览器页面的外观。

图2.4 内容脚本读取并修改显示网页的DOM

图2.5 内容脚本与所属扩展程序交换消息

2.2.4 跨域请求

跨域指的是JavaScript通过XMLHttpRequest请求数据时，调用JavaScript的页面所在的域和被请求的页面的域不一致。对于网站来说，浏览器出于安全考虑是不允许跨域的。另外，当域相同而端口或协议不同时，浏览器也是禁止的。

但是，这个规则如果同样限制Chrome扩展应用，就会使其能力大打折扣，所以谷歌允许Chrome扩展应用不必受限于跨域规则。但是出于安全考虑，需要在Manifest.json文件的permissions属性中声明需要跨域的权限。

用于采集数据的扩展程序的设计与实现

接下来，开始介绍我的扩展程序的开发过程。

清单文件

首先是扩展程序的用户界面的选择，我选择浏览器按钮的形式，即让扩展程序显示在浏览器的工具栏里，点击弹出一个页面窗口，页面窗口中的内容应该显示推荐的用户和推荐访问的网页。

然后是确定是否添加选项页。选项页的作用可以是完善进一步扩展程序的功能，或者是让用户自定义扩展程序的设置，或者是让用户添加自己的个人信息。因此，有必要添加选项页。

接着是声明我的扩展程序所需要的浏览器权限。

获取浏览器的历史记录，就要声明历史记录的权限；

用户成功登录或注册之后，将用户名和密码保存在浏览器的cookies中，方便用户下次使用扩展程序。其中涉及到对用户浏览器cookies的增加和修改，这就需要声明cookies的权限；

跨域请求。由于在我的扩展程序中将来一定会用到AJAX请求，以便与我的服务器通信，因此需要声明跨域请求的权限。” http://\*/\* ”, ”https://\*/\* ”表示允许扩展程序和任何域通信。

以下就是清单文件Manifest.json的大致内容：

Manifest.json

{

"manifest\_version": 2,

"name": "FindX",

"version": "1.0",

"description": "Help to find someone who has common interest with you.(Based on your explorer history)",

"browser\_action": {

"default\_icon": "icon19.png",

"default\_popup": "popup.html"

},

"icons": {

"38":"icon38.png",

"16": "icon16.png",

"48": "icon48.png"

},

"background":{

"scripts":["background.js"],

"persistent":true

},

"options\_page":"options.html",

"update\_url": "http://172.23.206.189:8080/myextensions/FindX/updates.xml",

"permissions":[

"history",

"cookies",

"http://\*/\*", "https://\*/\*"

]

}

Popup页面

为了能使扩展程序直接与用户交互，我给它指定了一个Popup页面popup.html。该页面可以实现用户的登录、注册，以及将来的推荐功能（显示推荐的结果）。因为我选择的扩展程序的用户界面是浏览器按钮的形式，所以它会在用户点击扩展程序按钮时弹出显示。

Popup页面是实现数据采集及用户交互的主要界面，它具有用户注册、用户登录、发送历史数据以及获取推荐数据的功能。

用户注册

新用户注册需要用户提交用户名、密码、确认密码。用户名可以是汉字、字母、字母+数字组合，长度小于20，不能与已存在的用户名相同。密码长度必须在6~15位之间，可以是字母、数字、标点符号，不能是空格。

注册成功时，服务器返回新用户的用户ID，然后扩展程序会将用户名和用户ID储存在浏览器的cookie中，并自动登录；同时，使用AJAX将用户名和密码发送至服务器，经服务器处理（对密码进行MD5加密）之后保存到数据库中。

注册失败时，返回错误代码或错误提示。部分错误代码或错误提示如下表2.1所示：

表 2.1 错误代码及错误提示

错误代码 错误提示

404 找不到服务器

500 服务器错误

— 用户名已存在

— 用户名长度太长

— 密码长度必须是6~15位

— 密码中不能有空格

用户登录

已注册用户输入注册时的用户名和密码实现登录，登录成功时，自动将用户名和用户ID储存在浏览器的cookie中，并设定永久不过期，以便今后可以免密码自动登录。

发送数据

用户登录成功后，扩展程序会自动将用户浏览器中的历史记录以JSON字符串的形式发送至服务器，并在服务器以TXT文档的格式保存，等待后续的分析处理。发送数据的过程用户是看不到的。

获取推荐结果

当服务器端对某个用户的历史记录分析完毕后，会将推荐的结果返回给用户端，并在Popup页面显示。推荐的结果包括：已注册用户中与当前用户兴趣相近的用户的社交资料，以及当前用户可能感兴趣的网站。

选项页面

为了完善扩展程序的功能，我给扩展程序添加了选项页面——options.html。在选项页面，用户可以选择修改昵称或者完善社交资料，提交之后即发送至服务器端，并且更新数据库。

选项页在加载的时候也会检查浏览器的cookies以判断用户是否登录，若cookies为空，即不包含FindX的用户名和用户ID，则判断当前浏览器中用户未登录；否则，扩展程序会利用cookie中保存的当前用户的用户名和用户ID来进行用户的修改资料操作。具体操作流程如下所述：

(1) 当用户点击“修改昵称”时，页面会根据浏览器cookie中username的值，显示旧昵称。

(2) 当用户输入新昵称并提交时，扩展程序将cookie中userid的值与新昵称一起发送至服务器端，userid是数据库中用户的唯一标识，因此，可以作为修改数据库记录的条件。

(3) 当用户点击“完善社交资料”时，扩展程序会将cookie中userid的值发送至服务器，作为查询用户已提交的社交资料的条件；修改社交资料并提交时，原理同（2）。

发送数据

当用户第一次登录成功或已登录用户点击扩展程序图标时，FindX会向服务器发送一次数据，数据将转化为JSON的数据格式。

发送的用户历史记录数据的格式是这样的：

[{"id":"1","lastVisitTime":1494466172281.013,"title":"YouTube","typedCount":0,"url":"https://www.youtube.com/","visitCount":29},…]

发送一次数据可能包含多条历史记录，全部记录由“[]”包括，其中“{}”表示一条记录，多条记录之间用逗号隔开；每条记录具有的信息如下表2.2所示：

使用JavaScript提供的Ajax进行前端与服务器端的交互，

服务器端接收到用户发送的历史数据之后，会对其进行相应的处理，然后保存在数据库中。

表2.2 历史记录项目的字段及其描述

字段名 字段描述

id 对同一网址访问的历史记录的唯一标识

lastVisitTime 用户上次访问该网址的时间，该处以毫秒计

title 访问的页面标题

typedCount 用户以地址栏输入的方式访问此网址的次数

url 该历史记录的网址

visitCount 用户访问此网址的次数

数据库的设计

数据库的选择

本系统的数据库采用MySQL数据库。选择MySQL的主要原因是它足够轻，SQL Server数据库和Oracle数据库等在它面前都显得过于庞大。首先，从安装包的体积来看，MySQL for Windows通用版（Generally Available Releases）5.7.18（最新版）安装包才四百零五点八兆，相较于SQL Server 2016版本2.85GB的安装包，轻的不是一点半点；而且SQL Server的管理工具——SSMS的安装包就有将近1个G。其次，从安装的过程来看，在同样一台计算机上，安装MySQL数据库在五分钟以内就能完成，而且步骤简单，但是安装SQL Server数据库我必须需要半个小时的时间，而且安装的步骤也比较复杂；况且，MySQL数据库还有免安装版的，下载完压缩包之后直接解压就能使用。

MySQL数据库除了轻以外，还有其他优点，比如简单、高效、开源、可靠。

简单——从它的安装过程就可以看得出来，除此以外，从数据库的创建来看，MySQL数据库仅仅需要一个简单的CREATE DATABASE命令就能在瞬间完成创建一个数据库的动作，然而Oracle数据库与之相比，创建一个数据库简直就是一个恢宏浩大的工程。

高效——高性能一直是MySQL数据库引以为傲的一个优点。在权威的第三方评测机构多次测试比较各种数据库TPCC值的过程中，MySQL数据库一直都有非常优异的表现，而且在其他所有商用的通用数据库管理系统中，只有Oracle数据库能够与其一较高下。

开源——许多中小型网站都是为了降低网站总体拥有成本而选择了MySQL数据库作为网站数据库。当然，对于开发者人员来讲，开源的软件自然会有更强大的吸引力。

可靠——可能对大多数人来讲开源软件就意味着不够稳定、不够健壮，虽然目前并没有详细的关于MySQL数据库可靠性的评测记录，但是世界最大的社交平台Facebook网站也在用它做数据库，可以看出MySQL在稳定性和可靠性方面，并不会比商业厂商的产品逊色太多。除此之外，雅虎公司和谷歌公司也有部分业务是运行在MySQL数据库环境上的。

数据库关系表的建立与ER图的设计

经过系统的分析以及反复多次的修改规划，最终确定了以下6个表，分别是用户信息表、历史文件表、历史记录项目信息表、推荐用户表、推荐网站表、用户关键词表。这些表的关系如下图2.6所示。

现将各个表的功能结构以及表中各个字段分别做一个简要的描述。

下表2.3所示为用户信息表，保存了用户的基本信息，如用户编号、用户昵称、用户的登录密码、用户的社交账号等信息。从图2.6也可以看出用户信息表与其余各表均有联系。

下表2.4所示为用户历史记录文件信息表。当用户点击浏览器中的扩展程序按钮时，扩展程序会将用户浏览器中的前1000条历史记录数据以JSON串的形式发送至服务器，服务器直接将该JSON串写入TXT文本文件中，以用户ID命名，保存在该Web项目根目录下的history文件夹里，并将该文件的相关信息保存在数据库的历史记录文件信息表中，以便今后调用读取。

图2.6 数据库关系图

表2.3 用户信息表

字段名 字段描述 数据类型 是否主键 是否外键

Id 用户编号，用户的唯一标识 Int 是 否

Name 用户昵称 Varchar(20) 否 否

Password 用户的登录密码，

以MD5加密的形式保存 Varchar(255) 否 否

Create\_date 用户注册时间 Datetime 否 否

Wechat 用户的微信号 Varchar(20) 否 否

QQ 用户的QQ号 Varchar(10) 否 否

下表2.5所示为历史记录项目信息表，该表中每一行的数据表示一条历史记录信息，包括网页标题、网页的URL、用户访问该网页的次数，以及用户最近一次访问该网页的日期。

下表2.6所示为推荐用户信息表，经过预测分析得到的与用户兴趣最相近的人的ID保存在这个表中。当用户点击扩展程序按钮时，该用户的ID将通过Ajax发送至服务器（Django），服务器根据收到的用户ID执行“推荐用户”模块的代

表2.4 历史记录文件信息表

字段名 字段描述 数据类型 是否主键 是否外键

Id 用户上传的历史文件的编号，

历史文件的唯一标识 Int 是 否

History\_list 历史文件的保存位置：

文件路径+文件名 Varchar(255) 否 否

User\_id 该历史文件对应的用户的ID Int 否 是

Upload\_date 该历史文件的创建时间 Datetime 否 否

表2.5 历史记录项目信息表

字段名 字段描述 数据类型 是否主键 是否外键

Id 历史记录项目编号，

项目的唯一标识 Int 是 否

Hid 历史记录在浏览器中的编号。

在浏览器中作为区分用户访问的网址的唯一标识 Int 否 否

User\_id 该历史记录项目对应的用户的ID Int 否 是

Web\_title 该历史记录项目的网页标题 Varchar(255) 否 否

Web\_url 该历史记录项目的网页URL Varchar(512) 否 否

Visit\_count 用户访问该项目的次数 Int 否 否

Last\_visit\_date 用户最近一次访问该项目的时间 Datetime 否 否

words 从网页标题中提取的关键词 Varchar(255) 否 否

码，并将执行的结果（可能有不止一个结果）插入到推荐用户信息表中。接着，服务器从表中提取出该用户的所有推荐信息，以JSON字符串的形式将数据返回给popup.html，呈现给用户；用户可以对每一条推荐的结果进行评分，并将评价的结果发送至这个表中的对应数据项的rate字段中。

下表2.7所示为推荐网站信息表。由于推荐的结果全部来自使用我的扩展程序的所有用户的上网历史记录，因此表中的网站信息用它在历史记录项目信息表中的ID表示。该表与推荐用户信息表类似，也包含了用户反馈的对于推荐结果的评价，以0~5分表示。

表2.6 推荐用户信息表

字段名 字段描述 数据类型 是否主键 是否外键

Id 推荐用户记录编号

推荐用户记录的唯一标识 Int 是 否

Userid 接收推荐的用户ID Int 否 是

Rec\_userid 被推荐的用户ID Int 否 是

Rate 用户对推荐结果的评价，0~5分 Int 否 否

表2.7 推荐网站信息表

字段名 字段描述 数据类型 是否主键 是否外键

Id 推荐记录编号，推荐记录的唯一标识 Int 是 否

Userid 接收推荐的用户ID Int 否 是

Rec\_website 被推荐的网站在历史记录项目表中的ID Int 否 是

Rate 用户对推荐结果的评价，0~5分 Int 否 否

表2.8 用户关键词表

字段名 字段描述 数据类型 是否主键 是否外键

Id 用户关键词记录编号 Int 是 否

Userid 接收推荐的用户ID Int 否 是

Topwords 用户历史记录的网页标题中出现次数最多的前N个词 Varchar(255) 否 否

上表2.8所示为用户关键词表，每行记录表示一个用户的兴趣关键词，以用户ID作为外键。用户关键词计算的方法是：从历史记录项目信息表中提取出每一条记录中的words字段的数据，按照用户ID分别保存在以用户ID命名的文本文件中；然后使用分词工具提取出每个文本文件中词频最高的前K个词，作为用户关键词，插入到用户关键词表中。

本章小结

本章主要介绍了扩展程序的基础知识、用户历史数据的采集过程、服务器端数据库的选择及数据库表的设计。用户历史数据的采集由本人开发的Chrome扩展程序——FindX完成。FindX不仅是收集数据的工具，也是向用户进行推荐的窗口。

历史数据预处理

我的扩展程序FindX会在用户每次点击图标时，通过Ajax向服务器发送一次数据，该数据包含用户的ID，以及浏览器中当前的历史记录数据；当服务器接收到这些数据时，会根据用户ID，更新数据库中的历史记录表，将来推荐给用户的网站全部来自这些历史记录数据。

然而，服务器端直接收到的这些数据，总会有一些数据是不能作为推荐的，这些数据就是无意义的、冗余的。比如下面这些记录：

当然，选择清理这些数据，是经过分析数据库表中已经存在的数据之后，才采取的策略。在有了这样的策略之后，我们可以根据这样的规律，在准备插入数据库表的时候，直接就把这些无意义的数据拦在“门外”，即在插入数据之前，先做过滤的工作。

从用户浏览器中获取的第一手历史记录数据，下文称作原生数据（raw data）。

数据清理

清理非网络数据

删除URL不是以’http’开头的数据，有一些历史数据可能是用户在本地打开的网页文件，这些记录是以’file’开头的；对于高级用户（开发人员），历史数据中可能还包含本地开发测试的WEB项目，这些历史数据的URL地址一般包含“localhost”或者“127.0.0.1”。

SQL语句

DELETE FROM `history\_items`

WHERE web\_url NOT LIKE "http%" OR web\_url LIKE "%localhost%" OR web\_url LIKE "%127.0.0.1%";

清理私人数据

比如用户个人邮箱，这些记录的url的域名一般是以 “mail”开头的。以及个人的QQ空间或者个人微博首页。

SQL语句

DELETE FROM `history\_items` WHERE web\_url LIKE "%://mail%"

OR web\_url LIKE "%user.qzone.qq.com/%" OR web\_title LIKE "%我的首页%";

清理公共网站

公共网站通常是指每个人都可能访问过的网站，比如大型门户网站、搜索引擎、社交网站、大型购物网站等等。经过在同学间的调查，总结出一些大家普遍频繁访问的网站。而且经过对采集的数据进行分析，发现这些网站的首页一般是以“/”结尾的。

SQL语句

DELETE FROM `history\_items`

WHERE web\_url LIKE "%www.baidu.com/" OR web\_url LIKE "%www.taobao.com/"

OR web\_title="Google" OR web\_title="首页%" OR web\_url LIKE '%bilibili.com/'

OR web\_url LIKE "%www.zhihu.com/" OR web\_url LIKE "%bing.com/"

OR web\_url LIKE "%www.jd.com/" OR web\_url LIKE "%qq.com/"

OR web\_title='爱辽宁' OR web\_url LIKE '%account%';

清理无效记录

观察发现历史记录表中有一些记录的网页标题是空的，这些记录不能作为推荐的结果，因此是毫无意义的数据，应该剔除掉。这些记录的清理比较简单，可以直接在数据库中操作，一条SQL语句就能完成。还有一些记录是某网站的登录页面，经过调查发现，登录页面的URL中一般包含“login”。

SQL语句

DELETE FROM `history\_items` WHERE web\_title="" OR web\_url LIKE "%login%";

具有校园特色的记录

由于我们将在校园内进行实验，针对同学们经常访问的和校园网有关的记录也需要单独清理。比如校园网主页、一些IPv6资源站的主页。

SQL语句

DELETE FROM `history\_items`

WHERE web\_url LIKE "%dlmu.edu.cn%" OR web\_url LIKE "%bt.neu6.edu.cn%";

数据变换

数据变换概述

在进行完数据清理的工作之后，我发现当前的数据依然不适合用于挖掘，要想让数据变得易于挖掘，还需要进行一些数据的变换工作，主要是对网页的标题展开变换。于是，项目的研究过程就进入了文本挖掘的研究。

其实，在项目开始进行推荐策略分析时，一直在尝试聚类挖掘，看了大量的文献资料，几乎也都是关于聚类分析的。在我看过的那些论文或期刊里，几乎都提到了特征向量。于是，我也尝试从我收集的历史数据中也提取出特征向量，但是无果。直到后来发现文本挖掘。

传统的数据挖掘，比如聚类挖掘，它们所处理的数据都是结构化的，划分出的特征通常不超过几百个；而非结构化或半结构化的文本数据转化成特征向量之后，特征数可能高达几万甚至几十万。由于历史记录中这些网页的标题数据是非结构化的，因此不适合使用传统的数据挖掘方法。

要使用文本挖掘，首先是要解决文本数据如何在数据库中合理地表示，然后才是对这些文本数据的挖掘分析。“合理地表示”是指处理后的文本信息要足以描述原有数据的特征，但是又不能过于庞大使分析算法的效率受到影响。目前，相关研究人员的普遍策略是对原始文本数据进行分词。

分词就意味着可以使用一条记录中出现的字或词的集合来代表这条记录，这样就能使文本数据的表示和处理形式化，并且可以在文本处理，如分类、检索等领域里面取得较好的效果，还能提高文本数据分析的效率。

分词工具的选择

数据挖掘在中国已经发展了将近二十年的时间了，也已经有大量的学者在文本挖掘领域作出了很多有意义的探索，现在的中文分词工具也比比皆是。中科院的ICTCLAS、哈工大的LTP、东北大学的NIU Parser都是学术界著名的分词器，这些分词器毋庸置疑地在中文分词领域都很强大。但是他们都是独立的软件，调用起来很不方便，因此我没有选择他们，而选用了jieba分词工具。

Jieba分词工具的优点

jieba分词是用Python写成的一个算是工业界的分词开源库，并且已经被整合在Python的扩展库中了。jieba分词虽然效果上不如ICTCLAS和LTP，但是胜在Python编写，代码清晰，扩展性好，调用简单，上手快。由于分析数据部分都采用Python编程，因此选择jieba分词工具简直是如虎添翼。

jieba分词的功能包括支持三种分词模式（精确模式、全模式、搜索引擎模式），支持繁体分词，支持自定义词典等，是目前一个非常不错的Python中文分词解决方案。

jieba分词对已收录词和未收录词都有相应的算法进行处理，其处理的思路很简单，当然，过于简单的算法也是制约其召回率的原因之一。

其主要的处理思路如下：

（1）加载词典dict.txt

（2）从内存的词典中构建该句子的DAG（有向无环图）

（3）对于词典中未收录词，使用HMM模型的viterbi算法尝试分词处理

（4）已收录词和未收录词全部分词完毕后，使用dp寻找DAG的最大概率路径

（5）输出分词结果

提取网页标题中的关键词——分词

这一环节使用Python，由于Python十分强大，一个pip命令就能安装所有你想要的并且在库中存在的扩展包，十分方便。使用jieba分词工具对记录的网页标题进行分词，并过滤掉没有意义的词（StopWordList.txt中存在的词）。

StopWordList.txt文件来自于开源社区，其中包括中英文停止词。

分词过程中要进行的工作：

从数据库表中取出记录，逐行分析；

去除标题中的标点符号，将所有的标点符号统一用空格代替；

使用jieba.cut进行初步分词，得到一个字典（dict）对象；

为方便后续操作，将（2）中得到的字典对象转化为列表（list）对象；

去除分词列表中的空格以及无意义的词语，即在stop\_list中列出的中英文词语；

将得到的结果插入到表中words字段。

该算法的Python代码实现如下所示：

Fenci.py

import pymysql

import jieba

jieba.load\_userdict('userdict.txt')#加载自定义词库

import jieba.analyse

from optparse import OptionParser

import string

import re

def insertW(iid,words):

cur2 = conn.cursor()

cur2.execute("update history\_items set words=%s where id=%s",(words,iid))

cur2.connection.commit()

print('ok')

conn = pymysql.connect(host='localhost',user='root',

passwd='',db='FindX',port=3306,charset = 'utf8')

cur = conn.cursor()

cur.execute("use FindX")

count = cur.execute("select id,web\_title from history\_items where web\_title <>''")

stopwords = {}.fromkeys([ line.rstrip() for line in open('stopword.txt') ])

#定义中英文标点符号的集合

punctuation='[.\[\]\/\/:·,“”\"，「『』」\'。——|\-\_？+【】»、!！()《》：；<\>（）?]'

for i in range(0,count):#不含上界

item=cur.fetchone()

title = item[1]#获取一条记录中的标题字段

iid = item[0] #获取一条记录中的用户ID字段

#去除标题中的中英文标点符号，全部用空格替换

title = re.sub(punctuation,' ',title)

words=jieba.cut(title)#调用jieba分词，函数返回的是字典对象

words=list(words)#将字典对象转为list对象，以便去除空格元素

#去除list中的空格

words = [word for word in words if word!=' ' and word not in stopwords]

words = "|".join(words)#将list中的元素用“|”分隔开，形成字符串

insertW(iid,words)#插入数据库

执行结果：

原标题：安装和管理扩展程序 - Chrome 网上应用店帮助

分词结果：安装|管理|扩展|程序|Chrome|网上|店

提取用户最感兴趣的关键词

分析认为，用户浏览的记录中，某一个词出现的次数越多，表明该用户对这个词越感兴趣，即某词的词频越大，说明这个词对这个用户越重要。词频，在数据挖掘的文本挖掘领域中也是一个很重要的概念。词频可分为绝对词频和相对词频。绝对词频是指特征词在文本中出现的次数；相对词频是指特征词在文本中的权重。本文主要是指绝对词频。

Jieba分词工具不仅能分词，还自带了分析功能，能够直接从一段文本或字符串中提取出词频最高的前K个词。

现在要做的是提取出某个用户最感兴趣的前K个关键词。姑且我们把K值暂定为20；经过上一步的处理，每条记录的关键词已经被提取出来了，接着可以根据每条记录的用户ID，分别将每个用户所有记录的关键词取出来，单独存放在文本文件中，文件中每一行的内容就是每条记录的关键词，文本文件以用户ID命名。这样处理之后的结果是，每个用户都生成了自己的一个记录关键词文档，这个文档的文件名就是用户的ID。

得到文本文件之后，使用jieba分词工具分析文本数据就很简单了，使用Python三四行代码就能搞定。

提取出用户关键词文档中出现次数最多的前K个词

file\_name = str(UserID)+".txt"

content = open(file\_name, 'rb').read()

tags = jieba.analyse.extract\_tags(content, topK=40)

print(",".join(tags))

以上代码得到的tags是一个list类型的对象，每一个词即是list中的一个元素。

在获取到了用户最感兴趣的前K个词之后，为了后续处理方便，我将这个结果保存在数据库User\_words表中。

本章小结

在进行正式的推荐工作之前，对用户的历史数据进行适当的预处理是十分必要的，本章就数据的清理和变换作了详细的介绍。在数据清理方面，本章详细分析了需要清理的用户历史记录类型，并总结出了这种数据中存在的一些规律，用于后续对数据过滤的研究；在数据变换阶段，首先简要描述了数据变换在数据挖掘领域中的意义，然后介绍了如何使用分词工具一步步提取出用户最感兴趣的关键词。

预测并推荐数据

预测感兴趣的用户

在第三章已经获取到了每个用户最感兴趣的前K个关键词，为方便表述，将其简称为用户兴趣词。我们的推荐方法就是基于每个用户的兴趣词来进行的。

推荐感兴趣的用户，就是指向用户A推荐和他兴趣相近的用户B，那么怎样找到这个用户B呢？

本文采用的策略表述如下：

以用户A的兴趣词为基准，和所有用户的兴趣词一一进行比较，得到其他每个用户与用户A的兴趣词一致的词的数量D，D值最大的用户，就认为他和用户A的兴趣最相近。

用集合的思想可以这样表述：

设用户A的兴趣词为集合A={w\_A1,w\_A2,w\_A3,…,w\_An}，w\_Ai表示用户A的第i个兴趣词。用户B的兴趣词集合为B={w\_B1,w\_B2,w\_B3,…,w\_Bn}，使集合B与集合A进行比较，得到他们的交集P\_AB=A∩B。同理，比较用户A和其他所有用户C~Z的兴趣词集合，并得到集合A和其他所有集合的交集P\_AC~P\_AZ。当然不止A、B、C到Z这26个用户。现在获取到这些交集了，然后再分别比较这些交集P的大小，就自然能够得到最大的交集P\_AX，这个时候就可以得到结论：用户X就是与用户A兴趣最相近的用户。

用Python代码实现算法的过程如下：

#获取与目标用户兴趣最相近的人

Def getXs(uTags,otherUsers):#目标用户的兴趣词与其他用户集合作为参数

resultUsers=[]#算法得到的结果——一个用户ID集合（list对象）

d={}#定义一个字典对象，保存其他用户的ID和对应的交集的大小

max=0#初始化交集的最大size值

for user in otherUsers:#遍历其他用户集合

oTags=getTopWords(user.id+'.txt')#其他用户的兴趣词集合

com=[tag for tag in oTags if tag in uTags]#或取oTags和uTags的交集

n = len(com)

d[user.id]=com#保存在字典中，以用户ID为键，交集为值

m=max(d,key=d.get)#字典d中值最大的第一个键

max=d[m]#字典中最大的value值，即用户兴趣词交集的最大size

for id in d:#遍历字典对象d的键，寻找交集大小等于最大值的用户ID

if len(d[id])==max:

resultUsers.append(id)#将交集大小等于最大值的用户ID添加到结果

#集合中

return resultUsers,d

预测感兴趣的网址

推荐用户感兴趣的网址可以有两种策略：一是根据其他用户的历史记录来推荐，当然，“其他用户”应该是用户感兴趣的人，这种推荐的结果在用户主动点击扩展程序按钮时呈现给用户；二是按照用户在搜索引擎中输入的关键词或句子来推荐，这种可以实现实时的推荐。

基于多用户协作的推荐策略

目前，多用户协作经常用于信息共享和信息的有效利用。比如，用户A在Python语言的学习中有比较深入的研究，他在上网过程中获取到了大量的关于Python的有用的信息，而这些信息会随着他学习的继续深入而失去已有的学习参考价值；另外，有个用户B刚开始学习Python语言，对Python还仅有基础的了解，那么用户A在上网过程中获取到的关于Python的信息对于用户B来讲就有重要的参考价值。

基于多用户协作的推荐策略是在获取到感兴趣的用户的基础上进行的，简而言之，就是将用户A感兴趣的用户B的历史记录共享给用户A，以达到推荐的效果。

当我们获取到与用户A兴趣最相近的用户B之后，就从用户B的历史记录中找到一些记录，这些记录满足以下任一条件：

网页标题中要包含他们共同的兴趣词，而且该记录是用户A未曾访问过的；

网页访问次数大于一个标准值，而且该记录是用户A未曾访问过的；

该算法使用Python代码描述如下：

预测用户感兴趣的网页

def getResultWebsiteList(resultUsers,targetUser,d):

resultWebsiteList=[]#保存符合推荐条件的历史记录的ID

sql = 'select id,words,visit\_count from history\_items where userid=%s'+

'and web\_title not in(select web\_title from history\_items where userid=%s)'

for userid in resultUsers:

count=cur.execute(sql,(userid,targetUser.id))

com=d[userid] #表示兴趣词交集

for i in range(0,count):

item = cur.fetchone()

wid = item[0]

words=item[1]

words=words.split('|')#字符串转为list

visit = item[2]

if visit>5:#满足第二个条件

resultWebsiteList.append(wid)

continue

for w in words:

if w in com:#满足第一个条件

resultWebsiteList.append(wid)

break

return resultWebsiteList

基于搜索文本的推荐策略

扩展程序会一直监听用户正在访问的网址，但是不会一直往服务器发送数据。当监听到用户正在使用搜索引擎查找关键词时，扩展程序能够获取到用户想要查找的关键词，这个时候，扩展程序会把获取到的关键词发送给服务器寻求帮助；服务器根据收到的关键词在用户浏览历史记录的数据库表中查找相关的记录，然后将查找的结果再发送至前端扩展程序；扩展程序就把得到的网址以弹窗的形式呈现给用户，以达到实时推荐的效果。

本章小结

本章分别介绍了推荐用户以及推荐网址的不同策略，对推荐策略的原理和过程都进行了详细的描述。同时也展示了推荐算法的Python代码实现，希望对读者有借鉴参考的价值。

结论

本文对用户上网行为预测进行了初步的研究和探索，详细介绍了使用Chrome扩展程序建立推荐模型的主要开发过程和方法，并提出了一种推荐上网用户的策略和两种推荐网址的策略。本文所建立的推荐模型主要包括四个功能模块，分别是用户信息的采集、用户浏览数据的采集、用户浏览数据的预处理，以及用户数据和用户浏览数据的推荐策略。在网址推荐方面，本文分别介绍了非实时的基于多用户协作的推荐策略，以及实时的基于用户搜索文本的推荐策略。

参 考 文 献

[1] 文本聚类算法Java实现- yiihsia的学习阵地-博客频道-CSDN.NET．[EB/OL]

http://blog.csdn.net/java2king/article/details/5156075

[2] 刘梦超，肖基毅，陈荣等．数据挖掘在用户上网行为分析中的应用研究[J]．

电脑知识与技术，2012．8(31)：7409-7412．

[3] Syota Suzuuchi，Mineichi Kudo． Location-associated indoor behavior analysis of multiple persons[C]．2016 23rd International Conference on Pattern Recognition (ICPR)．IEEE Conference Publications，2016：2079 – 2084．

[4] Yufei Chen，Chao Shen．Performance Analysis of Smartphone-Sensor Behavior for Human Activity Recognition[J]．IEEE Access，2017 (5)：3095-3110．

[5] 维基百科中的词条：先验算法．https://zh.wikipedia.org/wiki/先验算法

[6] 维基百科Apriori algorithm词条．[EB/OL]

https://en.wikipedia.org/wiki/Apriori\_algorithm

[7] Apriori | 数据挖掘十大算法详解 - wizardforcel – GitBook．[EB/OL]

https://wizardforcel.gitbooks.io/dm-algo-top10/content/apriori.html

[8] Overview – Google Chrome．https://developer.chrome.com/extensions/overview

[9] 马力,焦李成,董富强. 一种Internet的网络用户行为分析方法的研究[J]. 微电子学与计算机,2005,(07):124-126.

[10] 张波,巫莉莉,周敏. 基于Web使用挖掘的用户行为分析[J]. 计算机科学,2006,(08):213-214+239.

[11] 吕佳. 基于兴趣度的Web用户访问模式分析[J]. 计算机工程与设计,2007,(10):2403-2404+2407.

[12] 张宁. 群体兴趣网的统计特性研究[J]. 上海理工大学学报,2008,(03):243-248.

[13] 朱明．数据挖掘[M]．中国科学技术大学出版社，2002．

[14] 徐科,崔志明. 基于搜索历史的用户兴趣模型的研究[J]. 计算机技术与发展, 2006,(05):18-20.

[15] 单京晶. 基于内容的个性化推荐系统研究[D].东北师范大学,2015.

[16] 冯大为, 朱锐, 王怀民. 基于用户协作的新闻共享模型[J]. 计算机工程与科学, 2011, 33(5):165-170.

[17] Xuan J, Jiang H, Zhang H, et al. Developer recommendation on bug commenting: a ranking approach for the developer crowd[J]. Science China Information Sciences, 2017, 60(7):072105.

[18] 中国互联网络信息中心. 第39次中国互联网络发展状况统计报告[EB/OL]. 2017-01-22. www.cnnic.net.cn/hlwfzyj/hlwxzbg/hlwtjbg/201701/t20170122\_66437.htm

[19] 王超,黄建海,李俊,潘金贵. 多用户协作的Web知识获取[J]. 模式识别与人工智能, 2002,(03):285-289.

致 谢

首先，我衷心地感谢我的指导老师——高健老师，感谢他提供的这个课题，并且从一开始就帮我指明了一个总体的研究方向和可以使用的研究方法，从而使我有机会学习了解扩展程序的开发；在研究过程中，当我遇到问题时，他也都耐心地为我答疑解惑；为了方便实验，高健老师还给我提供了服务器，再次对他表示衷心的感谢！

我还要感谢大学本科四年来遇到的每一位老师！感谢你们课堂上的辛勤耕耘以及期末考试前的悉心指导，使我具备扎实的专业知识，掌握熟练的专业技能。

感谢吴锶导员，四年如一日地陪伴我们信息科学技术学院2013级2中队，在生活、学习的方方面面给予我们帮助，使我们能够愉快、顺利地完成四年的学业。

感谢我的同学、我的室友，四年来为我提供浓厚的学习氛围、和谐的生活环境，在我遇到困难时帮助我、关心我。

最后，祝愿母校大连海事大学早日建成世界一流海事大学！祝愿信息科学技术学院的学子在大型赛事中有更加精彩出色的表现！也衷心祝愿四年来与我同行的朋友在未来创造出各自辉煌灿烂的人生！

附录