**电 子 科 技 大 学**

UNIVERSITY OF ELECTRONIC SCIENCE AND TECHNOLOGY OF CHINA

**学士学位论文**

**BACHELOR THESIS**



论文题目  **基于机器学习的问答推荐算法设计**

专 业 **计算机科学与技术**

学 号 **2014060108020**

作者姓名 **伍峰**

指导教师 **俸志刚**

摘 要

当今互联网发展如此迅速,日新月异的各种技术层出不穷,什么大数据,云计算,人工智能等正潜移默化的进入了人们的生活,这里面我们都能看到一个技术的身影,那就是机器学习,面对互联网信息爆炸的今天,信息量爆炸,用传统的信息处理方式已经不足以满足现在的数据量,也就有了机器学习的用武之地.本文以问题答案匹配为机器学习的对象,以爬虫爬取数据,索引构建,模型构建,机器学习等为主要内容.

**关键词：**机器学习,问题答案推荐,模型构建,文本处理,关键词提取,爬虫,搜索引擎,索引

ABSTRACT

With the development of the Internet so quickly, so many new technology are born just like an endless stream, like big data, cloud computing, artificial intelligence and so on. Those technology are imperceptible to enter our daily life. Above all those technology, we can see, these is a core technology, which is machine learning. With internet information explosion so quickly today, traditional way of information processing is not sufficient to satisfy the big data today. That's the reason why we should learn to use machine learning to solve those complex problems. The main content of this paper is based on machine learning for question-answer recommendation, crawling data, index building, model building and so on.,

**Keywords:** Machine learning, question answer recommendation, model building, text processing, keyword extraction, crawler, search engine, index

目录

[摘 要 II](#_Toc513320417)

[ABSTRACT 3](#_Toc513320418)

[第一章 绪 论 5](#_Toc513320419)

[1.1 研究工作背景和意义 5](#_Toc513320420)

[1.2 机器学习国内外研究历史和现状 5](#_Toc513320421)

[1.3 LTR研究背景和现状 6](#_Toc513320422)

[1.4 本论文结构安排 6](#_Toc513320423)

[第二章 机器学习基础 6](#_Toc513320424)

[2.1 机器学习基础 6](#_Toc513320425)

[2.2 机器学习分类 6](#_Toc513320426)

[2.3 机器学习基本原理 7](#_Toc513320427)

[第三章 爬虫基础 7](#_Toc513320428)

[3.1 爬虫基本原理 7](#_Toc513320429)

[3.2 爬虫搜索策略分类 7](#_Toc513320430)

[3.3 爬虫常见问题 8](#_Toc513320431)

[第四章 搜索引擎基础 8](#_Toc513320432)

[4.1 搜索引擎基本原理 8](#_Toc513320433)

[4.2 数据爬取 8](#_Toc513320434)

[4.3 文本特征值处理 9](#_Toc513320435)

[4.3.1提取文本 9](#_Toc513320436)

[4.3.2分词 9](#_Toc513320437)

[4.3.3去停用词 9](#_Toc513320438)

[4.3.4除噪 9](#_Toc513320439)

[4.4 索引 9](#_Toc513320440)

[4.2.1索引组织方式 9](#_Toc513320441)

[4.2.2建立索引 10](#_Toc513320442)

[4.2.3搜索引擎中的索引 11](#_Toc513320443)

[4.5 检索模型和搜索排序 12](#_Toc513320444)

[4.5.1初始子集的选取 12](#_Toc513320445)

[4.5.1检索模型 12](#_Toc513320446)

[4.6 LTR 14](#_Toc513320447)

[4.6.1 单文档方法(PointWise) 14](#_Toc513320448)

[4.6.2 文档对方法(PairWise) 14](#_Toc513320449)

[4.6.3 文档列表方法(ListWise) 14](#_Toc513320450)

[第五章 问答推荐系统研究与设计 15](#_Toc513320451)

[5.1 数据的爬取 15](#_Toc513320452)

[5.1.1 爬虫框架选取 15](#_Toc513320453)

[5.1.2 数据结构设计 15](#_Toc513320454)

[5.1.3 爬虫代码设计 15](#_Toc513320455)

[5.1.4 爬虫中间件设计 15](#_Toc513320456)

[5.2 数据预处理 15](#_Toc513320457)

[5.2.1 文本特征值提取 15](#_Toc513320458)

[5.2.2 建立正向索引 15](#_Toc513320459)

[5.2.3 建立逆向索引 15](#_Toc513320460)

[5.2.4 相关度标记 15](#_Toc513320461)

[5.3 机器学习模型 15](#_Toc513320462)

[5.3.1 模型的选取 15](#_Toc513320463)

[5.3.2 模型的训练 15](#_Toc513320464)

[5.4 优化 15](#_Toc513320465)

[5.4.1 爬虫优化 15](#_Toc513320466)

[5.4.2 数据库优化 16](#_Toc513320467)

[第六章 全文总结和展望 16](#_Toc513320468)

[6.1 总结 16](#_Toc513320469)

[6.2 后续工作展望 16](#_Toc513320470)

[致谢 16](#_Toc513320471)

[参考文献 16](#_Toc513320472)

[https://blog.csdn.net/huagong\_adu/article/details/40710305 16](#_Toc513320473)

# 第一章 绪 论

## 1.1 研究工作背景和意义

当今互联网发展如此迅猛,互联网的数据量越来越大,于是便产生了大数据,而大数据是为了人工只能所服务的,而人工智能一个重要的理论基础和分支,就是机器学习.这几年来,人工智能领域重新崛起,特别是当谷歌的alphago出现在世人面前并且击败了九段围棋高手李世石之后,机器学习算是正式进入了大众的视野,各大高校也开始纷纷开始了有关课题的学习和研究工作.

那究竟什么是机器学习呢,我们先从学习的定义说起. 按照人工智能大师西蒙的说法,学习就是系统在不断重复的工作中对本身能力的增强或者改进,使得下一次的执行相同任务或类似的任务的时候,会比之前做的更好更出色或者效率更高.

顾名思义，机器学习是研究如何使用机器来模拟人类学习活动的一门学科。稍为严格的提法是：机器学习是一门研究机器获取新知识和新技能，并识别现有知识的学问。这里所说的“机器”，指的就是计算机；现在是电子计算机，以后还可能是中子计算机、光子计算机或神经计算机等等。

## 1.2 机器学习国内外研究历史和现状

机器学习属于人工智能领域,人工智能是从20世纪40年代和50年代开始,逐渐走上历史舞台.1956年正式被确立为一门学科.而之后人工智能领域的发展,多多少少可以看到机器学习的身影.机器学习是人工智能发展到一定阶段的必然产物.

从20世纪50年代到70年代,当时人工智能领域的学者们普遍认为,只要给机器赋予逻辑推理能力,那么机器就有了智能,当时具有代表性的成果主要有”逻辑推理家”程序以及之后的”通用问题求解”程序等.其中”逻辑推理家”程序在1952年就证明了<数学原理>的38条定理,又过了11年,就证明了全部52条定理,而且定理2.85甚至比罗素和怀特海证明的证明更为巧妙,这个程序也给作者带来了1975年的图灵奖.

然而随着研究的深入,学者们发现,光有逻辑推理能力,是完全不够的.又掀起了一股新的认知狂潮,认为要让机器智能,必须得先让机器有知识,于是20世纪70年代开始,学者们的研究方向转向了赋予机器人类的知识,大量的专家系统出现,但是人们也逐渐发现,由人来把知识总结出来再教授给计算机的这个过程,是相当困难的,于是有的学者就想:机器要是能自己学习就好了.这也就是机器学习的必然性.

## 1.3 LTR研究背景和现状

在互联网刚刚诞生的时候,最开始出现的是门户网站,后来才有了搜索引擎,而当时的搜索引擎采用的检索模型是十分简单的,依赖的特征值往往很少.如今互联网发展至今

## 1.4 本论文结构安排

# 第二章 机器学习基础

## 2.1 机器学习基础

机器学习,即是一种从数据中获取想要的信息的一种手段,或者说,从数据中学习一些新的知识的手段.机器学习的定义多种多样,但有一个广泛被认可的一个定义就是: 对于某类任务T和性能度量P，如果一个计算机程序在T上以P衡量的性能随着经验E而自我完善，那么我们称这个计算机程序在从经验E学习.这简短的定义,是Mitchell在1998年给出的,一直在业界被广泛流传.

## 2.2 机器学习分类

机器学习分为四大类:监督式学习,非监督式学习,半监督式学习,增强学习

监督式学习,基本上实现的就是分类的功能, 它从有标签的训练数据中学习，然后给定某个新数据，预测它的标签。这里的标签，其实就是某个事物的分类。在监督学习中,输入的数据被称作训练样本,每组样本都会有一个明确的标识或者结果.

非监督学习,与监督学习恰恰相反,非监督学习面对的训练数据都是没有标签的,给定数据,从数据中学习,能学到什么,这取决于数据本身具有什么样的特征.

半监督式学习,这种学习方式借鉴了监督学习和非监督学习,去其糟粕取其精华,半监督学习通过对部分标签数据的学习,从而对之后的数据进行分类,但不像监督学习一样,一旦学习完毕,就不再更新自身,而半监督学习会根据之前的经验继续对数据进行学习

增强学习,计算机通过观察来学习做什么样的动作.每个动作对环境都会产生影响,计算机会根据观察到的周围环境的影响当作反馈来做出判断

## 2.3 机器学习基本原理

# 第三章 爬虫基础

## 3.1 爬虫基本原理

互联网上的信息储存在成千上万的服务器之中,早期,人们在做搜索引擎的时候,网站的信息都是通过人工输入的,但随着互联网的发展,这些人工输入的数据的量,早已不能满足人们,于是便有了爬虫的诞生,爬虫诞生之初是为了搜索引擎所服务的,而不是现在大家常见的盗取他人网站等不良用途.

爬虫是一个自动提取网页信息的程序,它为搜索引擎提供搜索数据的来源,是搜索引擎的重要组成部分.传统意义上的爬虫,是由一个或者若干的起始网页开始,按照一定规则,不断从当前的链接中过滤筛选出所需要的链接放入爬取队列,再根据一定的规则,选择队列中的链接继续爬取,直到队列为空,从而爬取到海量的数据,为搜索引擎提供的数据保证.



## 3.2 爬虫搜索策略分类

搜索策略,即选取什么样的网页爬取的策略.根据爬虫在爬取数据过程中采取的搜索策略,一般分为以下三种策略:

1. 广度优先搜索:广度优先搜索,和其他广度优先的算法一样,优先保证算法广度,即在爬取过程中,先将当前层次的所有网页都爬取完毕,再进行下一层次的网页内容爬取.这个算法的核心思想是,认为约靠近初始网页的网页,相关程度越高,所以根据广度优先算法获取到的网页,整体上相关程度会高一些,但实则随着网页层次的深入,大量的不相关网页也会被爬取,降低爬取速度
2. 深度优先搜索:深度优先搜索,和其他深度优先算法一样,优先保证算法深度,即在爬取过程中,先从初始页面,获取到一个链接后,直接爬取该链接内容,继续再从这个链接获取下一个链接,这样处理完一条路线后,再次处理下一条路线.这个算法也并不复杂,但是由于深度优先算法,优先爬取离初始页面远的页面,这样爬取到的页面相关程度很低,价值不高,一般很少采用.
3. 最佳优先搜索:这个搜索策略下,会有一个分析算法,用于分析相应链接与主题相关程度,用于评判一个链接的好坏程度,该搜索策略下,会从队列中选取分析算法筛选出来的有效链接开始爬取网页,每次都只爬取分析算法认为有效的链接,减少了爬取无用网页带来的爬取速度减慢问题.算法核心在于如何设计分析算法.最佳优先搜索策略本质上是一个局部最优算法.

## 3.3 爬虫常见问题

爬虫基本原理十分简单,但由于爬虫的辨识度太高,即爬虫的机械重复程度很高,很容易识别出爬虫,一些网站就会对一些来历不明或者用意不明的爬虫进行识别和限制,一旦识别出爬虫,就会采用各种各样的手段限制你的动作,比如限制请求数目,暂时封禁相应IP访问,暂时封禁相关帐号,暂时封禁相关网段等.

# 第四章 搜索引擎基础

## 4.1 搜索引擎基本原理

互联网发展如此迅速,很大程度上是搜索引擎的功劳,如今的搜索引擎已经不是从前的简单只是根据网页的几个简单特征值来计算网页内容的相似度,也已经不是一起那个数据量少的可怜的检索时代了.现在的搜索引擎十分强大,里面涉及的原理十分复杂,但我们可以通过了解搜索引擎的基本原理,来在浅层逐渐了解一下搜索引擎内部的工作原理.

搜索引擎一般分为以下四个部分:

1. 数据爬取
2. 建立索引
3. 内容检索
4. 搜索排序

## 4.2 数据爬取

搜索引擎的数据爬取,即网站资源的获取,早期互联网刚刚发展起来的时候,网站资源的获取,完全靠人工获取并输入,像早期的各大门户网站等.随着互联网发展,如何在互联网中搜集想要的信息就成了一大挑战,于是就有了爬虫.网络爬虫技术是搜索引擎最根本的技术.通过爬虫,我们可以将网络上数以百亿计的网页信息存到本地,为搜索引擎提供数据支持.

## 4.3 文本特征值处理

爬虫爬取数据完毕后,下一步工作就是对爬取到的文档进行一定程度上的文本特征值处理,一般包括:提取文本,分词,去停用词,除噪等

### 4.3.1提取文本

提取文本,即从源代码中分析提取出所需文本的过程,爬虫爬去到的都是网站的源代码表示形式(如HTML),包含大量的不可见格式标签,这些标签都只是用于给浏览器解析,而展现给用户特定的样式,无实际意义,所以需要从中剔除掉,提取出用户可见的文本.除了用户可见的文本需要提取外,一般搜索引擎还会提取一些特殊的文字信息,如meta标签等

### 4.3.2分词

分词,即把文本中的所有词语分离出来,分词根据语种的不同,有不一样的分词算法.中文分词指的是将汉字序列切分出一个个的词语.中文分词比英文分词要难许多,我们知道,在英文中,单词之间是按照空格分割的,而中文只有字,句,段有明显的自然分隔符,而词语没有.中文分词的算法主要有两类: 基于词典分词算法,基于统计的机器学习算法.

### 4.3.3去停用词

停用词,即一些在语言中大量出现,而又对实际语言内容没有任何影响的词语,比如中文中的”的”,”得”,”地”等助词,或者是”啊”,”呀”等感叹词,包括一些转折介词等,都被称为停用词,而英文中也有类似的词语,如”the”,”to”,”of”等,在分词处理后,我们还需要去除停用词,进一步减少无用词带来的效率损失问题,减少无意义的计算和检索时间.

### 4.3.4除噪

除燥,即进一步去除无意义内容,这些内容被称作噪声,或者噪点,如网页中的导航文字,版权信息,广告栏等,这些内容对于排名计算毫无意义,只能增加计算量,所以在更进一步处理之前,需要除去这些无意义内容.搜索引擎会在这个阶段将这些内容识别并去除,这个过程就被称为除噪.

## 4.4 索引

### 4.2.1索引组织方式

建立索引,即如何把爬虫爬取到了网页信息保存到本地,这里主要涉及到数据结构的设计,索引的分类和建立索引的方式也有很多种,下面介绍一种简单的索引组织方式

* + - 1. 索引组织方式

a)倒排索引,称逆向索引:

先简单介绍一下基本概念:

文档:文本形式的待检索对象

文档编号:搜索引擎内部,标记每个文档的唯一ID编号

单词编号:搜索引擎内部,用于标记每个单词的唯一ID编号

单词词典:文档集合中出现过的所有单词构成的字符串集合,单词词典内每条索引都记载了单词本身和相关信息等,常用的存储方式有:哈希链表形式,树形结构等

倒排列表:出现了某个单词的所有文档的文档集合以及相关信息,倒排列表可以记录那些文档包含了某个单词,倒排列表由倒排索引项组成,每个倒排索引项由文档ID,单词出现次数等信息组成.

由于倒排比较简单,下面给出一个简单例子便于直观理解:

|  |  |
| --- | --- |
| 文档编号 | 文档内容 |
| 1 | 今天小明从学校回家 |
| 2 | 今天小兰从学校出发 |
| 3 | 学校小卖铺有辣条卖 |
| 4 | 小明回家路上买辣条 |
| 5 | 小兰不喜欢吃辣条 |

文档内容

根据以上的文档内容,即可生成对应的倒排索引,索引内容如下:

|  |  |  |  |
| --- | --- | --- | --- |
| 单词ID | 单词 | 文档频率 | 倒排列表(文档ID,出现次数) |
| 1 | 今天 | 2 | (1,1),(2,1) |
| 2 | 小明 | 2 | (1,1),(4,1) |
| 3 | 小兰 | 2 | (2,1),(5,1) |
| 4 | 辣条 | 2 | (3,1),(5.1) |
| 5 | 学校 | 3 | (1,1),(2,1),(3,1) |
| 6 | 回家 | 2 | (1,1),(4,1) |
| 7 | 从 | 2 | (1,1),(2,1) |
| 8 | 卖 | 1 | (3,1) |
| 9 | 路上 | 1 | (4,1) |
| 10 | 不喜欢 | 1 | (5,1) |
| 11 | 出发 | 1 | (2,1) |

倒排列表内容

从例子中,我们可以看出,倒排索引将文档中的每个单词和文档都对应生成一个唯一ID,并存储下对应关系,接着记录下每个单词在所有文档中出现的次数,并记录为文档频率,并把每个单词出现的文档ID和出现次数都记录下来

### 4.2.2建立索引

理解了简单的倒排索引的组成和结构,下面介绍索引生成的几种方式:

* + - 1. 索引生成方式

1. 两遍文档遍历法

生成索引步骤:

1. 遍历第一遍文档,收集全局信息,包括:文档集合的文档总数N,文档集合包含的所有单词数M,每个单词的DF.把所有单词对应的DF值相加,即可计算出最终索引的所需内存大小,分配所需内存大小,以供使用.
2. 遍历第二遍文档,逐个单词建立倒排索引.每扫描到一个单词,查询对应ID,如没有则生成唯一ID并保存对应关系,接着修改该单词在当前文档ID下出现次数+1,即最终会得到这个单词在文档出现的次数TF.最后第二遍遍历扫描结束后,第一步分配的内存将会全部占满,每个单词对应两个指针,分别指向了内存区域的片段,即为对应单词的索引项.

缺点:一次性需要大量的内存,对机器性能要求高

1. 排序法

生成索引步骤:

1. 读入一个文档,并对读入文档进行编号,标记唯一ID,解析文档内容
2. 将单词对应相应单词ID.
3. 建立相应的索引项(单词ID,文档ID,单词频率)
4. 将索引项存放到中间结果存储区
5. 处理下一个文档,重复1步骤
6. 如果中间结果存储区占用完毕,则对中间结果存储区的所有索引项依据单词ID为主项,文档ID为辅助项进行排序
7. 将排序好的中间结果存储区合并写入最终索引文件

### 4.2.3搜索引擎中的索引

前面简单介绍了什么是索引,而在搜索引擎中,我们使用的索引有所不同,一般的搜索引擎的索引部分,都是由正向索引和反向索引(倒排索引)构成.

正向索引,这个索引所记录的内容是每个文档对应的信息,比如关键词ID,关键词出现的次数,格式,位置,TF值等,具体保存什么信息,根据实际情况有所不同.实际搜索引擎中,索引中的关键词全部是以ID形式保存,而关键词ID和关键词对应关系,会保存在另外一个索引中.这种每个文档ID对应一串关键词ID的索引,被称为正向索引.

下图为一个正向索引的例子:

|  |  |
| --- | --- |
| 文档ID | 索引项 |
| 1 | 今天,小明,学校,回家 |
| 2 | 今天,小兰,学校,出发 |
| 3 | 学校,小卖部,辣条,卖 |
| 4 | 小明,回家,路上,辣条 |
| 5 | 小兰,不喜欢,辣条 |

反向索引,因为正向索引无法直接应用到搜索中,假设用户搜索了关键词3,那么搜索引擎需要检索库中所有文档,得到所有相关文档,这样的时间消耗对于用户来说根本不切实际,不可取,所以我们必须提前建立好反向索引,供用户搜索使用,可以大幅度减少搜索时间.这里的反向索引保存的信息一般包括:关键词ID,包含对应关键词的文档ID序列.

下图为一个反向索引的例子:

|  |  |
| --- | --- |
| 关键词 | 文档 |
| 今天 | 1,2 |
| 小明 | 1,4 |
| 小兰 | 2,5 |
| 辣条 | 3,5 |
| 学校 | 1,2,3 |
| 回家 | 1,4 |
| 从 | 1,2 |
| 卖 | 3 |
| 路上 | 4 |
| 不喜欢 | 5 |
| 出发 | 2 |

## 4.5 检索模型和搜索排序

索引建立完毕后,搜索引擎的前期工作基本已经完成,接下来已经可以向外提供搜索服务了,当用户输入要搜索的内容时候,搜索引擎会根据之前生成的反向索引迅速的获得相关文档,接下来的工作就是搜索排序,据相关度排序文档,以便用户优先看到自己想要的内容.

### 4.5.1初始子集的选取

当粗略获得了所有相关的文档后,往往相关文档的数量十分的庞大,甚至几十万,几百万,这么大的数据量进行匹配计算的话,所需要的时间无疑会更长,而且用户并不关心也不可能关心排名十分靠后的文档,显然十分的浪费计算机的性能.为了进一步提升搜索速度,进一步满足用户需求,一般在实际搜索引擎中只计算权值较高的文档返回给用户.

### 4.5.1检索模型

经过了初始子集的选取后,剩下数量不是很庞大的文档序列,接下来要做的是利用检索模型,计算出文档相关值,根据相关程度排序搜索结果.检索模型种类很多,判断好的模型的标准是.把文档集合划分为4个部分,第一部分是文档出现了用户查询的关键词也同时是被用户认为是相关的,第二部分是文档出现了用户查询的关键词却不被用户认为是相关的,第三部分是文档没有出现用户查询的关键词但却被用户认为是相关的,第四部分是文档没有出现用户查询的关键词,也不被用户认为是相关的.有个好的检索模型,应该尽量提升第一部分和第三部分的文档的排名,降低第二部分和第四部分的排名.

下面介绍几种检索模型:

#### 4.5.1.1 布尔模型

布尔模型是检索模型中最简单的,文档和用户搜索的内容的相似度计算是通过布尔代数计算得出的.

用户的搜索被视为逻辑表达式,比如用户要查询计算机硬件或者软件相关的内容,那么可以用以下逻辑表达式表达搜索:

计算机 AND ( 硬件 OR 软件 )

根据这个逻辑表达式,包含计算机关键词的文档中,还包含硬件或者还包含软件关键词,即被认为和用户查询相关,否则不相关.

从上面可以看出,布尔模型十分简单直观,但是存在明显的缺陷,即:无法排序,检索结果是二元的,文档只被划分为相关和不相关两个部分,其检索结果过于的粗糙.

#### 4.5.1.2 空间向量模型

空间向量模型目前属于较为成熟的模型之一,被广泛应用到各个领域的各个方面.

##### 4.5.1.2.1 文档表示方法

在空间向量模型中,文档有自己特殊的表示方法,会把每个文档表示为一个由N维特征值组成的一个向量,特征值的选取没有具体要求,但都会根据一定依据计算各自特征值的权重,由这N维向量就组成了文档.

下面是空间向量模型的文档表示例子:

|  |  |  |  |
| --- | --- | --- | --- |
| 文档ID | 文本长度 | 搜索关键词TF\*IDF | 包含搜索关键词的最小窗口大小 |
| 文档1 | 56 | 1.54 | 20 |
| 文档2 | 74 | 1.22 | 60 |
| 文档3 | 34 | 0.57 | 7 |
| 文档4 | 22 | 0.36 | 4 |

通过特征转换,所有文档和用户的查询都可以转换为对应的特征值向量表示

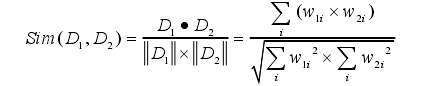
##### 4.5.1.2.2 相关性计算

给定用户的搜索向量和文档向量,即可计算出之间的相关性了.余弦距离是一种较为合适的指标来当作文本之间的相似度.

设文本D1和文本D2的空间向量表示分别为:

D1 = (w11,w12,w13..w1n)  
D2 = (w21,w22,w23..w2n)

则余弦距离计算公式如下:



余弦公式本身已经对特征值向量长度做了规范化,主要是为了惩罚长文档.但是存在明显的过分惩罚的现象,如果同时存在长文档和短文档,短文档的相关系数会大大高于长文档.

## 4.6 LTR

LTR,全称Learning To Rank,前面介绍了一些检索模型,他们的功能都是用来实现对文档的的排序,使得用户更关心的内容排名更加的靠前,而LTR也是一种可以实现对搜索结果进行排序的方法,这是几年机器学习兴起之后,变的热门的一种排序方法.这是机器学习和搜索领域结合下产生的全新排序方法.

从前面介绍的检索模型可以看出,前面的检索模型大部分考虑的特征值比较少,而且一般都是人工拟合排序公式,在搜索引擎刚刚发展的时代考虑的因素不多,还适用,而如今互联网发展如此迅猛,所谓连接互联网重要的枢纽的搜索引擎,之前单一的特征值和人工拟合,已经不能满足人们的搜索需求.随着搜索排序要考虑的因素越来越多,人工拟合已经变的不切实际,而这种类型的工作又很适合用机器学习来处理,于是便有了机器学习的勇武之地,这是LTR诞生的原因之一.

还有一方面的原因是,机器学习分为监督学习和非监督学习,而我们的排序只能通过监督学习来实现,而监督学习需要大量的训练数据,才能完成模型的训练.而对于搜索引擎来说,有一个很天然而又直接的指标来完成对搜索结果的标注工作,那就是用户的点击记录,用户会倾向于点击相关的搜索结果,所以我们可以认为,用户点击量越高的结果,应该就是相关度越高的搜索结果.依次为依据,我们就可以标注我们的搜索结果,从而作为训练数据的来源.

机器学习的排序系统由4个步骤组成:标注训练数据,文档特征值提取,训练模型,采用模型排序.

从目前研究的情况来说,LTR一般被分为3大类:单文档方法(PointWise),文档对方法(PairWise),文档列表方法(ListWise)

### 4.6.1 单文档方法(PointWise)

单文档方法中,训练数据是以单个文档为处理对象的,首先将文档转换为为特征值向量后,机器学习模型会根据从训练数据中学习到的回归(或者分类)分类函数对单一文档进行打分,打分结果即为搜索结果相关系数,下面是一个简单的例子:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 文档ID | 查询ID | PageRank值 | 余弦值 | 文档长度 | 相关值 |
| 1 | 2 | 3 | 0.6 | 30 | 0.8 |
| 3 | 2 | 4 | 0.13 | 352 | 0.5 |
| 2 | 3 | 5 | 0.24 | 35 | 0.7 |
| 3 | 3 | 1 | 0.36 | 45 | 0.5 |
| 2 | 4 | 5 | 0.3 | 87 | 0.3 |
| 4 | 4 | 8 | 0.4 | 12 | 0.8 |

例子中,采用了3个特征值,单文档方法中,模型会根据每一条训练数据,尽可能的拟合一个回归函数来接近训练数据,模型的输入是特征值向量(例子中是3维的特征值向量),输出是相关值.

### 4.6.2 文档对方法(PairWise)

文档对方法则是换了一个角度对搜索结果进行分析.对于搜索排序来说,最关键的是确认文档的先后顺序,即两个文档谁在前谁在后的问题.只要能确定任意两个文档的先后顺序,我们即可以完成对搜索结果的排序.总的来说,文档对方法相对于单文档方法来说,将重点转移到了文档顺序是否合理进行判断.

这种机器学习排序方式,最终的训练目标是判断任意两个文档的组成的文档对是否已经满足顺序关系.具体的学习过程如下,首先还是人工标定好训练数据集.将文档转换为特征值向量,根据人工标定的相关度,即可得到多对文档顺序对,将文档对作为训练数据,训练模型.

下面是一个简单的文档对方法的例子:

原始人工标定数据如上,则我们可以得到以下的文档顺序对:

|  |  |  |  |
| --- | --- | --- | --- |
| 查询ID | 文档ID | 顺序关系 | 文档ID |
| 2 | 1 | > | 3 |
| 3 | 2 | > | 3 |
| 4 | 4 | > | 2 |

这就是我们要作为训练数据的文档对,机器学习的模型有很多,不管选择何种模型,最终训练的效果就是输入一个文档对,模型可以判断该顺序是否合理,如果合理,则文档A在文档B前面,否则相关.通过这样的方式,我们就可以完成对搜索结果的排序工作.

文档对方法摒弃了单文档方法的只对单一文档打分的办法,做出了改进,但依旧存在一些明显的问题,比如说对每个文档的位置考虑都是同样的权重,或者说对每个位置的文档都是一样的待遇,这样就会出现一个问题,实际应用中,用户往往会更多的关注搜索结果的前几名的情况,即搜索结果的前列文档的顺序十分重要,如果前面文档的顺序出错,代价应该明显高于后面的文档顺序出错才对,而简单的文档对方法并没有体现这一点.

### 4.6.3 文档列表方法(ListWise)

文档列表方法与前面两种方法不同,单文档方法每次只是只是单独考虑一个文档,文档对方法则是对一次搜索结果的任意两个文档作为训练对象.文档列表方法摒弃了前面两种方法的角度,把一次搜索结果的所有文档看作一个整体,用于作为训练数据.

文档对方法要做的是,根据N个训练数据(每个训练数据就是一次查询结果的所有文档),训练得到一个最优评分函数,对于一个新的用户搜索,用评分函数对每个文档进行打分,即为相关度,最后根据相关度排序即为最终排序结果.

文档列表方法具体的实现算法有很多种,就不一一介绍了.

# 第五章 问答推荐系统研究与设计

## 5.1 数据的爬取

### 5.1.1 爬虫框架选取

### 5.1.2 数据结构设计

### 5.1.3 爬虫代码设计

### 5.1.4 爬虫中间件设计

## 5.2 数据预处理

### 5.2.1 文本特征值提取

### 5.2.2 建立正向索引

### 5.2.3 建立逆向索引

### 5.2.4 相关度标记

## 5.3 机器学习模型

### 5.3.1 模型的选取

### 5.3.2 模型的训练

## 5.4 优化

### 5.4.1 爬虫优化

### 5.4.2 数据库优化

# 第六章 全文总结和展望

## 6.1 总结

## 6.2 后续工作展望

# 致谢

# 参考文献

<https://baijiahao.baidu.com/s?id=1563025901009674&wfr=spider&for=pc>

<http://www.52ml.net/11881.html>

<https://blog.csdn.net/huagong_adu/article/details/40710305>

<http://blog.jobbole.com/101792/>