第5章

大数据系统输出

5.1 数据的查询

5.1.1 在常规数据库查询结构化数据

数据库是为有效地管理信息而创建的，人们希望数据库可以随时提供数据信息，因此，对用户来说，数据查询是数据库较重要的功能。在数据库中创建对象并且在基表中添加数据后，用户便可以从数据库中检索特定信息。

结构化查询语言(structured query language，SQL)是一种为特殊目的而设计的编程语言，是一种数据库查询和程序设计语言，用于存取数据以及查询、更新和管理关系数据库系统，其简写 sql 同时也是数据库脚本文件的扩展名。

结构化查询语言允许用户在高层数据结构上工作，它不要求用户指定数据的存放方法，也不需要用户了解具体的数据存放方式，所以具有完全不同底层结构的不同数据库系统也可以使用相同的结构化查询语言作为数据输入与管理的接口。

不过各种通行的数据库系统在实践过程中都对 SQL 规范做了某些编改和扩充。所以，在实践中不同数据库系统之间的 SQL 并不能完全通用。

5.1.2 大数据时代的数据搜索

在大数据时代，人们对于搜索引擎的要求不再是从海量信息里找到结果，而是在海量结果里找到唯一，快速找到准确的答案比找到更多的答案更重要。

1. 结构化数据对搜索的价值

结构化数据和网页数据相比更能满足“找准唯一答案”的需求。网页分析是靠文本匹配，而结构化数据的分析既支持内容提供者的主动接入，也支持搜索引擎的个性化精准分析。这两种方式都会增加内容提供者或者搜索引擎的成本，但是其带来的回报是用户能快速得到准确的唯一答案。

2. 大数据挖掘是搜索引擎的机会

搜索引擎经过 20 多年的发展，在文本分析、关系发掘、图谱构造和用户语义

理解等方面已有丰富的技术经验积累,这些技术是实现大数据挖掘的基本技术。

一般来说,搜索引擎提供非结构化文本的查询服务,数据库引擎提供结构化数据的查询服务。因此,结构化应用和利用数据库实现的数据挖掘过程难以被拓展到非结构化数据上。例如,搜索引擎对一个公开站点进行索引后,试图利用结构化数据分析方法来对该站点的注册用户行为进行分析通常是不太可能的,对BBS、博客、微博的顶帖人分析,哪些是假冒的明星粉丝,哪些人是托,但这些分析对一些商业化公司是有用的,特别是广告公司。

目前缺乏有效的手段来进行跨越站点的综合分析,一般只能针对特定网站设计分析。如果能够用搜索引擎提供结构化查询的方法,很多标准的结构化分析程序将可以派上用场。

如果说大数据是金矿,拥有大数据的垂直网站、社交网站、App服务商、云计算服务商、物联网拥有者、政府组织和企业组织就是矿山的老板。这些大数据的拥有者可以自行从金矿中掘金,也可以将金矿卖给搜索引擎或者大数据挖掘公司。搜索引擎为金矿埋单的同时,必须让自己从加速信息流动的管道转变为会淘金的人。

3. 互联网信息的特点

1) 面向显示与面向数据

从信息交换的角度看,目前互联网上的信息大多以HTML(hypertext markup language,超文本标记语言)文档的形式存在,用户与服务器之间信息的传递主要依赖超文本传输协议。HTML文档中的信息是面向显示的,其使用规范的标记定义文档的文本应如何显示。这些标记由浏览器负责处理,而信息的理解工作则由用户自己完成。

2) 半结构化与非结构化

在互联网上,数据被嵌在HTML文档的文本中,而数据的部分组织信息则被嵌在标记中。从文档标记的角度看,HTML是显示超链接的文档;从数据的角度看,文档所蕴含的数据也是半结构化的,这是因为:数据没有严格的结构模式、含有不同格式的数据(如文本、声音和图像等)、HTML文本无法区分数据类型、多个异质数据源中不同的站点给含义相同的信息起了不同的名字(如“级别”与“等级”等)。

3) 不同形式数据源的数据

除了保存在HTML文档中的信息外,互联网上还有大量信息被存储在文本文档、传统的关系或对象数据库中,这些不同形式的数据在互联网上需要通过集成并用HTML文档显示,以实现共享和交换。

如何有选择地从已有数据开始,生成供浏览的页面并建立站点是管理互联网站点必须仔细考虑的问题。

4) 静态与动态

互联网站点上的信息是随时间而动态变化的,信息内容的变化(增加、删除和修改)需要被及时地反映到互联网页面中。另一方面,站点的页面组织结构可能发生改变(如页面的增加、删除和修改)也要被及时反映到站点页面的目录层次结构中。

5）界面友好

Web 站点的信息主要面向一般的非计算机专业用户。因此，用户对界面的友好性、易用性提出了更高的要求。用户获取信息的渠道越来越多，方式越来越灵活。因此，提供给用户的服务应该适应多种形式的用户界面。

4. XML 成为数据组织和交换的事实标准

随着网上数据数量的不断激增，人们对网上信息的应用需求也不断提高，原有的对文本文件的链接浏览和关键词检索已无法满足一些复杂的应用需求。但是，将传统数据库技术直接应用于网上数据的最大困难在于：网上数据缺乏统一的、固定的模式，数据往往是不规则且经常发生变动的。因此，半结构化数据模型应运而生，其无固定模式及自描述的特点适宜描述网上数据。

事实上，日益普及的 XML 数据就是一种自描述的半结构化数据，它的出现推动了互联网技术在电子商务、电子数据交换和电子图书馆等多方面的应用。但对于如何有效地存储、管理和查询这类数据，目前却莫衷一是，已有的数据库技术如关系数据库、面向对象数据库等都不能完全适应新的应用需求，而专用的半结构化数据管理系统目前仍处于初步实验阶段。

可以预言，XML 将成为数据组织和交换事实上的标准，大量的 XML 数据将很快出现在 Web 上。XML 为 Web 的数据管理提供了新的数据模型，很多成熟的数据库技术将进入 Web 信息处理领域，将其变为一个巨大的数据库。

5.1.3　数据库与信息检索技术的比较

互联网目前还只是一个巨大的、分布式的信息检索系统，大多数搜索引擎仍基于信息检索技术。数据库技术与信息检索技术有很多不同，如表 5.1 所示。

表 5.1　数据库技术与信息检索技术的比较

比 较 项 目	数据库技术	信息检索技术
数据	有结构	无结构
模型	有确定的模型	基于概率
查询语言	人工的（如 SQL 等）	自然的
查询规范	完全的	不完全的
匹配	精确匹配	部分匹配、最佳匹配
所需条目	基于匹配	基于相关
出错报告	敏感的	不敏感
推理	演绎	归纳
类属	单向度	多向度
数据更新	完全支持	不支持

续表

比 较 项 目	数据库技术	信息检索技术
事务	支持	不支持
使用	面向应用	面向人

以上两者最重要的区别是数据库的数据结构性更强,比信息检索的数据包含更多的语义。在一定意义上,信息检索技术更适合处理无结构数据,数据库则是管理结构数据的最好途径。在本质上,信息检索使用近似法为用户的浏览需求查找相关信息。这个“近似”的含义包括近似的查询条件说明、近似匹配和近似结果。

数据库的查询语言通常是人工语言,有严格的语法和词汇表;在信息检索中,人们经常使用的是自然语言。

随着数据数量的激增和 Web 规模的快速增长,想要使用传统的信息检索方法在这样一个无限的信息海洋中准确、快速定位所需信息,将越来越力不从心,在未来的 Web 发展中,提高信息检索的准确性和效率将成为一个关键问题。

另一方面,目前出现了超越浏览方式而使信息面向应用访问的迫切需求,这种需求将为各种服务提供自主性、互操作性和 Web 意识。无结构的 HTML 文档及其相应的信息检索技术将不再适应下一代更复杂的 Web 应用。

因此,未来的 Web 信息将由更近似数据库的方式进行管理,而不是单一的信息检索方式,Web 资源需要以有结构的方式组织和访问。

5.2 网络数据索引与查询技术

5.2.1 搜索引擎技术概述

目前查询网络数据使用较多的是搜索引擎(search engine),搜索引擎可以根据一定的策略,运用特定的计算机程序搜集互联网上的信息,将信息进行组织和处理后显示给用户,是为用户提供检索服务的系统。

1. 搜索引擎的发展

1990 年,加拿大麦吉尔大学计算机学院的师生希望拥有一个可以用文件名查找文件的系统,于是他们开发出了 Archie。当时,万维网(world wide web,WWW)还没有出现,人们只能通过 FTP 共享交流资源。Archie 能定期搜集并分析 FTP 服务器上的文件名信息,查找分布在各个 FTP 主机中的文件。用户可以输入精确的文件名进行搜索,Archie 将告诉用户哪个 FTP 服务器能下载该文件。

虽然 Archie 搜集的信息资源不是网页(HTML 文件),但其和搜索引擎的基本工作方式是相同的:自动搜集信息资源、建立索引、提供检索服务。所以,Archie 被公认为是现代搜索引擎的鼻祖。由于 Archie 深受人们欢迎,受其启发,1993 年人们又开发了一个 Gopher 搜索工具。

2. 搜索引擎的分类

1）全文索引

全文索引是名副其实的搜索引擎，国外代表有谷歌，国内则有著名的百度搜索。它们从互联网提取各个网站的信息，建立数据库，并能检索与用户查询条件相匹配的记录，按一定的排列顺序返回结果。

根据搜索结果来源可将全文索引分为两类：一类拥有自己的检索程序，俗称“爬虫”程序或“机器人”(robot)程序，能自建网页数据库，直接从自身的数据库中调用搜索结果，上面提到的谷歌和百度搜索就属于此类；另一类则是租用其他搜索引擎的数据库，并按自定的格式排列搜索结果，如 Lycos 搜索引擎。

2）目录索引

目录索引虽然有搜索功能，但严格意义上不能被称为真正的搜索引擎，只是按目录分类的网站链接列表而已。用户完全可以按照分类目录找到所需要的信息，不依靠关键词(keyword)进行查询。目录索引中较具代表性的是新浪分类目录搜索。

3）元搜索引擎

元搜索引擎(meta search engine)在接受用户查询请求后可以同时在多个搜索引擎上搜索，并将结果返回给用户。著名的元搜索引擎有 InfoSpace、Dogpile 和 Vivisimo 等。

5.2.2　Web 搜索引擎的工作原理

Web 搜索引擎的工作原理：首先，用爬虫程序进行全网搜索，自动抓取网页；其次，为抓取的网页建立索引，同时也记录与检索有关的属性，中文搜索引擎还需要先对中文进行分词；最后，接受用户查询请求，检索索引文件并按照各种参数进行复杂的计算，将产生结果返回给用户。下文将基于上述原理简要介绍 Web 搜索引擎。

1. Web 搜索引擎的组成

Web 搜索引擎一般由搜索器、索引器、检索器和用户接口 4 部分组成，如图 5.1 所示。

(1) 搜索器：其功能是在互联网中漫游，发现和搜集信息。

(2) 索引器：其功能是分析搜索器所得信息，从中抽取索引项，以其表示文档以及生成文档库的索引表。

(3) 检索器：其功能是根据用户的查询在索引库中快速检索文档，进行相关度评价，对将要输出的结果排序，并按用户的查询需求合理反馈信息。

(4) 用户接口：其作用是接收用户查询信息，显示查询结果，提供个性化查询项。

2. Web 搜索引擎的工作模式

(1) 利用网络爬虫获取网络资源。这是一种半自动化的资源获取方式，所谓半自动化是指搜索器需要人工指定资源的统一资源定位符(uniform resource locator，URL)，然后获取该 URL 所指向的网络资源，并分析和获取该资源所指向的其他资源。

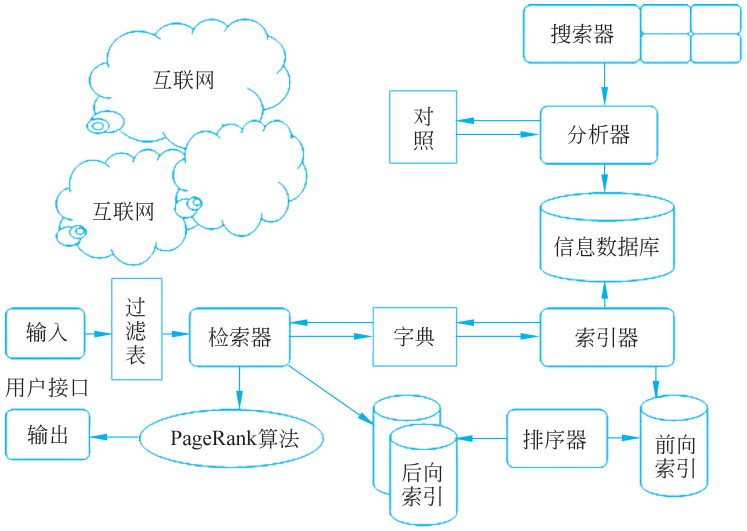

图 5.1 Web 搜索引擎的组成

网络爬虫访问资源的过程是遍历互联网上信息的过程。实际的爬虫程序为了保证信息收集的全面性、及时性以及多个爬虫程序的分工和合作问题,往往会有复杂的控制机制。如谷歌公司在利用爬虫程序获取网络资源时,会使用多个分布式的爬虫程序管理程序活动任务,然后将获取的资源作为结果返回,并重新获得任务。

(2) 利用索引器从搜索器获取的资源中抽取信息,并建立利于检索的索引表。在网络爬虫获取资源后,索引器需要对这些资源进行加工过滤,去掉控制代码及无用信息,提取出有用的信息,并把信息用一定的模型表示,使查询结果更为准确。

Web 上的信息一般表现为网页,每个网页都须生成一个摘要,此摘要将显示在查询结果的页面中,告诉查询用户各网页的内容概要。模型化的信息将被存放在临时数据库中,由于 Web 数据的量极为庞大,为了提高检索效率,索引器须按照一定规则建立索引。

索引的建立包括如下内容。

① 分析过程,处理文档中可能的错误。

② 索引文档,完成分析的文档被编码进存储单元,有些搜索引擎还会使用并行索引技术。

③ 排序,将存储单元按照一定的规则排序。

④ 生产全文存储单元。最终形成的索引一般按照倒排文件的格式存放。

(3) 检索及用户交互。前两部分属于搜索引擎的后台支持,本部分将在前面信息索引库的基础上接受用户查询请求,并到索引库检索相关内容以返回给用户,Web 搜索引擎的工作模式如图 5.2 所示。这部分的主要内容如下。

① 理解用户查询(query),即最大可能贴近地理解,用户通过查询串表达查询目的,并将用户查询转换为可供后台检索使用的信息模型。

② 根据用户查询的检索模型在索引库中检索结果集。

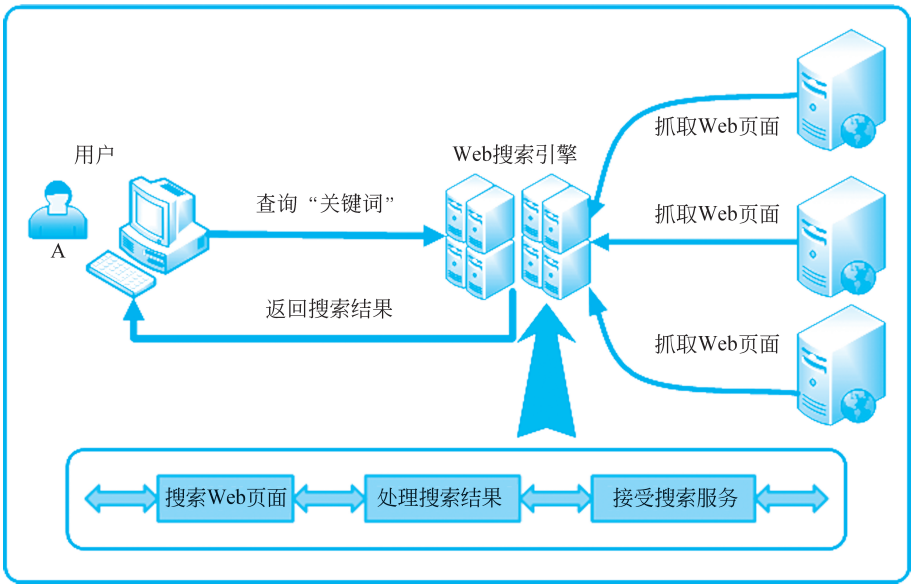

图 5.2　Web 搜索引擎的工作模式

③ 结果排序：通过特定的排序算法对检索结果集进行排序。

现在用的排序因素一般有查询相关度、谷歌公司发明的网页排名(PageRank)算法、百度公司的竞价技术等。由于 Web 数据的海量性和用户初始查询的模糊性，检索结果集一般很大，而用户一般不会有足够的耐心逐个查看所有的结果，所以设计结果集的排序算法时，把用户感兴趣的结果排在前面是十分重要的。

3. 搜索引擎的设计技术与算法

搜索引擎的评价指标有响应时间、查全率、查准率和用户满意度等。其中，响应时间是从用户提交查询请求到搜索引擎给出查询结果之间的时间间隔，其必须在用户可以接受的范围之内；查全率是指查询结果集信息的完备性；查准率是指查询结果集中符合用户要求的数目与结果总数之比；用户满意度是一个难以量化的概念，除了搜索引擎本身的服务质量外，它还和用户群体、网络环境有关系。在搜索引擎可以控制的范围内，核心问题是搜索结果的排序，即前文提到的把合适的结果排到前面。

总的来说，Web 搜索引擎的 3 个重要问题如下。

(1) 响应时间：一般来说合理的响应时间在秒数量级。

(2) 关键词搜索：得到合理的匹配结果。

(3) 搜索结果排序：如何对海量的结果数据排序。

所以在设计搜索引擎的体系结构时就需要考虑信息采集、索引技术和搜索服务 3 个模块的设计。

5.3 大数据索引和查询技术

5.3.1 大数据索引和查询

索引和查询技术是数据处理系统的重要入口之一,近年来随着数据量、数据处理速度和数据多样性的快速发展,大数据相关的索引和查询技术也变得更为重要。

传统的索引和查询技术虽然不能很好地解决大数据带来的挑战,然而其核心技术如数据库和数据挖掘系统中使用的经典索引、信息检索系统中的倒排索引等依然是大数据索引和查询系统的基石。

大数据带来的主要挑战是其庞大的数据量,单个结点不能或者无法有效地处理这种数量级的数据。此外数据增长速度非常快,这要求系统不但能处理已有的大数据,还要能快速地处理新数据。这些特征使得人们需要多考虑在大数据环境中独有的因素,并以此来开发和选择大数据索引和查询技术。

例如,设某搜索引擎每天新增1亿篇网文,考虑到网文中有些太平凡的字词不适合被作为关键字,如"的""地""得""不但"和"而且"等,每个网页平均按100个有效关键字估算,要做完一天新增网页的索引表,用传统方法需要扫描1亿网页,写出100亿词汇,然后记录下所有如下的对子:<关键字,所在页面> 再加以整理,去重、合并、压缩后,这需要用多长时间?需要多大的空间?

谷歌公司提出了一个从海量文档中做索引的方法——MapReduce,正是它协调若干台计算机,并行计算以完成索引表的构建与维护,使谷歌公司在求多求快的竞争中立于不败之地。

分布式是处理大数据的一个基本思路,这同样适用大数据索引和查询系统。分布式索引把全部索引数据水平切分后存储到多个结点上,这可以很好地解决两个问题。

(1) 单个结点无法存储庞大的索引数据。

(2) 单个结点构建索引的效率瓶颈。当业务增长,需要索引更多的数据或者更快的索引数据时,服务者可以通过水平扩展增加更多的结点来解决。

目前各大数据厂商都已经有支持分布式索引和查询的产品,很多NoSQL数据库如MongoDB、HBase也支持分布式索引和查询。

5.3.2 大数据处理索引工具 MapReduce

MapReduce是建立海量数据索引的方法,有人说它是像里程碑一样的技术。理解MapReduce,又是理解更时髦的Hadoop和Spark等大数据技术的基础。其实,过去人们就不知不觉地用了MapReduce技术,如机场分发登机牌、银行取号排队和流水作业阅卷等。

海量数据索引使用的是倒排索引(inverted index)算法。倒排索引源于实际应用中根据属性的值来查找记录的思路。这种索引表中的每一项都包括一个属性值和具有该属性值的记录地址。由于其不是由记录来确定属性值而是由属性值来确定记录的位置,因

而其被称为倒排索引。

例如，乘客在机场办理登机手续时，会经过三次映射和一次归约。

(1) 第一次映射，进入机场候机大厅，乘客会看到航班信息显示屏，被分散引导到相应的值机区域，如国航 CA1209 航班到 K 区办理。

(2) 第二次映射，在工作人员指引下把乘客分到各值机台办理登机牌。

(3) 第三次映射，把乘客映射到具体的航班、座号，柜台处理包括验看证件、发放登机牌、把乘客分到航班上以及给托运行李挂上航班标签。

(4) 归约成为倒排表。

把上述映射的结果按航班合并、约简，即可组成便于使用的倒排表，如表 5.2 所示。

表 5.2　归约成为倒排表

关　键　字	乘客证件号及其座号队列
…	…
航班 CA1209	1(1 排 A)，3(2 排 B)，5(3 排 C)…
航班 3U8882	2(5 排 A)，4(7 排 B)，6(2 排 C)…
…	…

登机牌在该航班起飞前半小时将会停办，对应倒排表停止变化，把乘客按某指标(通常为关注重要程度)排序并分发到该航班和机场、保险公司等相关部门。

此外，用多个单关键字的倒排索引作交集，可以得到多关键字的倒排索引。

(5) 倒排表帮助改善服务。上述倒排索引能帮助机组人员知道登机人数与座位并改善服务，如叫出头等舱客户和金卡客户的姓名且服务到座位。

如有突发事件发生，则倒排表将能作为处突依据，例如，马航官方能在突发事件后很快查出 MH370 的乘客信息。

综上所述，办理登机牌的全过程可以被表达为经典的 MapReduce 图，这个图大致反映了并行 MapReduce 的流向，但没有描述归约的细节，如图 5.3 所示。

现在的互联网搜索引擎使用的倒排表机理大致如上，但数量增大若干数量级则相当于在上图中的乘客组有几千万而值机台(CPU)有 100 万个，且航班(倒排索引项)有几万甚至几十万。

这只是为了说明 MapReduce 机制而编的例子，真实的机场工作机制要复杂得多。

大数据中的 MapReduce 有下列要点。

① 目标：完成某一类计算，典型实例就是生成某个关键字上的倒排索引。

② 对象：PB 级的数据，例如来自云端、分布式文件系统的文档。

③ 并行处理，多个处理单元。

④ 有序：在机场、车站，当客户增加，仅增加服务台来做归约(reduce)常常不够有序，那么增加一个映射(map)机制，把被处理对象分配到处理单元是必不可少的环节。春运中人们便体会到了这一条。

⑤ 多层映射，多层归约：根据实际情况，归约也可以是多层次的。

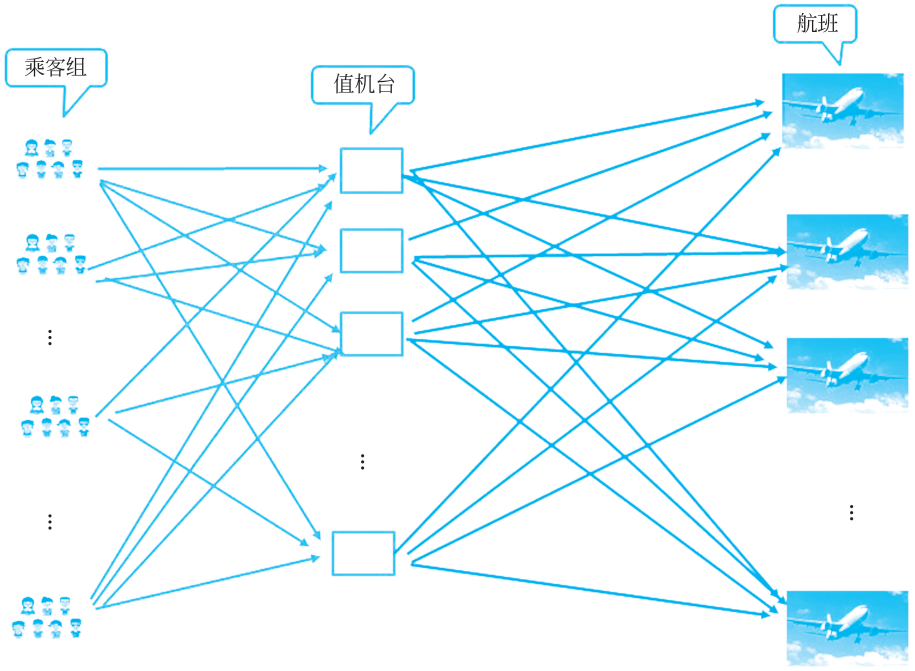

图 5.3 办理登机牌的全过程 MapReduce 图

5.3.3 相似性搜索工具

相似性搜索工具被用于识别与要匹配的一个或多个输入要素最相似(或最相异)的候选要素,相似性基于数值属性(感兴趣属性)的指定列表。如果指定了一个以上的要匹配的输入要素,那么相似性将基于每个感兴趣属性的平均值。输出要素类(输出要素)将包含要匹配的输入要素以及找到的所有匹配的候选要素,这些要素以相似程度排序(由最相似或最不相似参数指定),返回的匹配数基于结果数参数的值。

1. 可能的应用

可以用相似性搜索工具找出某城市在人口、教育以及临近特定娱乐机会方面相似的其他城市。

政府可能希望促进其城市的潜在业务,从而提高税收。相似性搜索工具有助于帮助政府找出与其城市相似的城市,以便他们可以找出自身比较有吸引力的属性(例如,低犯罪率和高成长率)。

政府也可能有兴趣查找比其城市大或小但位置相似的城市。找出与其城市相似但更小或更大且具有政府期望拥有的商业吸引力的地方,可以让政府指出相似性,同时可以强调小的优势(不拥堵、有小城镇韵味)或者大的好处(如更多顾客)。

政府还可能关注和其城市不相似的城市。如果任何不相似的地方表现出他们期望吸引的业务竞争优势,此分析可以为政府提供相对所需的信息。

人力资源经理可能希望能够证明公司的工资合理性。找出在大小、生活成本、市容建