doi:10.3969/j.issn.1672-4348.2019.04.010

基于 MapReduce 的快消品电商网站热搜品牌 TOP-N 计算

王晨阳

(福建工程学院 信息科学与工程学院,福建 福州 350118)

摘要:设计一个迭代的 MapReduce 并行计算工作流,用于分析快消品电商网站的搜索引擎日志。该工作流根据每次检索在商品品牌字段上的层面搜索结果,挖掘关键字检索和品牌检索热度之间的潜在相关性,为关键字检索计算出其对品牌层面搜索结果集中各品牌的检索热度贡献值,最后对品牌检索热度贡献值列表进行归并计算得到各个品牌的检索热度排名并取 Top-N。

关键词:热搜品牌;品牌推荐;层面检索;MapReduce

中图分类号:TP311　　**文献标志码**:A　　**文章编号**:1672-4348(2019)04-0365-06

Top-N calculation of hot search brands in FMCG e-commerce websites based on MapReduce

WANG Chenyang

(School of Information Science and Engineering, Fujian University of Technology, Fuzhou 350118, China)

Abstract: An iterative MapReduce parallel computing workflow was designed to analyze the log of search engine in FMCG e-commerce websites. According to the facet search results on brand field of each retrieval, the workflow mines potential relevance between each keyword retrieval and the retrieval popularity of the brand, and calculates each keyword retrieval's contribution to the retrieval popularity of each brand in brand facet search results. Finally, the list of popularity contribution values of brand retrieval is combined and calculated to get the retrieval popularity ranking of each brand and the Top-N.

Keywords: hot search brands; brand recommendation; facet search; MapReduce

快消品,快速消费品(FMCG, fast moving consumer goods)的简称,是指那些使用寿命较短,消费速度较快的消费品。快消品牌商在线下渠道面临困境,从线下转向线上来寻求额外的销售增长,是行业内最显著的趋势。近年来,互联网快消品市场迎来迅速发展,涌现出一大批快消品电商网站,如阿里 1688 零售通、京东掌柜宝、掌合天下、惠民网、便利宝、易酒批、万全速配、中国邮政邮乐网等[1]。

因为快消品使用寿命较短、价格较为便宜,消费者的购买决策往往相对简单,更加热衷于品牌化、大众化。然而快消品牌种类成千上万,如果能从电商网站的搜索日志里挖掘出热搜品牌并推荐给用户,将大大降低用户的时间成本。但是,用户在搜索时输入的关键字往往不总是商品完整的品牌名称,所以如何从搜索引擎的每一条搜索日志里挖掘出与品牌检索热度间的潜在相关性成为问题的关键。

1 相关知识介绍

1.1 层面检索

层面检索(facet search),也称层面导航,是

收稿日期:2019-07-30
基金项目:福建工程学院青年基金项目(GY-Z18168)
作者简介:王晨阳(1984-),男,福建莆田人,讲师,硕士,研究方向:大数据技术、电商推荐系统、搜索引擎技术。

一种按照分类法进行存储和检索信息的技术[2]。层面检索能够在搜索关键字的同时,按照 Facet 的字段进行分组并统计,主要用于导航实现渐进式精确搜索,从而给用户提供更加友好的搜索体验。比如在京东或淘宝的搜索栏输入“笔记本电脑”进行搜索,在搜索结果页面上方会展示品牌、内存容量、屏幕尺寸等不同类目的相关查询结果。用户可以通过点击这些类目里的结果进行渐进式精确搜索。这里的品牌、内存容量、屏幕尺寸就是一个个 Facet。目前绝大多数的电商网站搜索引擎都有提供层面搜索功能。而且常用的几大开源搜索引擎框架,如 Apache Solr、ElasticSearch 也都实现了层面搜索功能。层面搜索使得搜索引擎可对数据之间的内在联系进行挖掘,从而作为海量数据的统计工具。

1.2 MapReduce

MapReduce 技术是 Google 公司于 2004 年提出的一种分布式编程模型,主要用于大规模数据集的并行计算[3]。MapReduce 采用了分而治之的思想,将复杂的、运行于大规模集群上的并行计算过程高度地抽象到两个函数:Map 和 Reduce。谷歌的 MapReduce 运行在 GFS(google file system,谷歌文件系统)上,Hadoop MapReduce 是谷歌的 MapReduce 的开源实现,运行在 HDFS(hadoop distributed file system,Hadoop 分布式文件系统)上。如无特殊说明,本文所提的 MapReduce 均为 Hadoop MapReduce。

数据本地化是 MapReduce 的核心特征,即采用“计算向数据靠拢”的设计理念[4],因为在大数据集群环境下,移动数据需要大量的网络传输开销,而移动计算则比移动数据更加经济。本着这个理念,在一个集群中,MapReduce 框架会尽量让 Map 程序就近地在 HDFS 数据所在的节点运行,即尽量将计算节点与数据节点放在一起运行,从而大大减少了数据在节点间的移动开销,有效提升整体性能。

MapReduce 极大地方便了分布式编程工作,它把计算过程分解为两个阶段,即 Map 阶段和 Reduce 阶段。程序员只需实现 Map 函数和 Reduce 函数,而如分布式存储、集群任务调度、节点通信、负载均衡、容错处理等并行编程中的各种复杂问题则由 MapReduce 框架负责解决。Map 函数和 Reduce 函数都是以键值对<key,value>作为输入,按给定的映射规则生成另一个或一批键值对<key,value>进行输出,且 Reduce 函数的输入类型必须和 Map 函数的输出类型相同。这两个函数的输入和输出如表 1 所示。

表 1 Map 和 Reduce 函数[5]

Tab.1 Map and Reduce functions[5]

函数	输入	输出
Map	<k1,v1>	List(<k2,v2>)
Reduce	<k2,List(v2)>	<k3,v3>

具体的 MapReduce 工作流程如图 1 所示,详细描述如下[4,6-11]:

(1)在 Map 阶段,将存储在 HDFS 上的输入文件逻辑切分为多个等大小的分片。每个分片即为一个块(Block)的大小,默认为 128M。

(2)因为 HDFS 上每个块默认保存 3 个副本,Map 任务会尽量就近读取输入数据分片,并从中解析出一个键值对集合,作为 Map 任务的输入。

(3)Map 任务会根据用户自定义的映射规则,输出一系列的键值对<key,value>作为中间结果。

(4)为了让 Reduce 任务可以并行处理 Map 的输出结果,需要对 Map 的输出进行混洗(Shuffle),即进行分区(Partitioin)、排序(Sort)、合并(Combine)、归并(Merge)等操作,以得到<key,value-list>形式的中间结果,等待 Reduce 任务拉取。

(5)在 Reduce 阶段,Reduce 任务以一系列<key,value-list>形式的中间结果作为输入,执行用户定义的逻辑,输出一系列的键值对<key,value>最为最终结果并写入 HDFS。

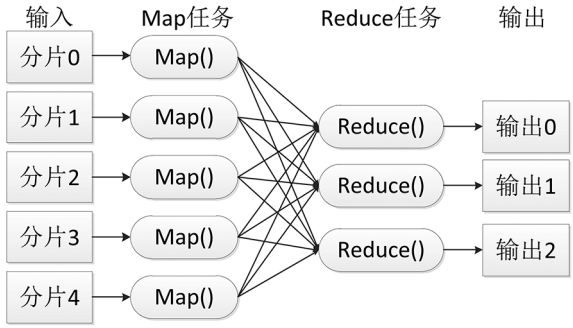

图 1 MapReduce 的工作流程

Fig.1 MapReduce workflow

实际应用中，很多复杂的问题很难用一轮 MapReduce 任务解决，需要将其拆分成多个 MapReduce 子任务去完成。由于后一个子任务要使用前一个子任务的输出结果，所以经常在一轮 MapReduce 任务执行完成之后，其输出并不合并成一个文件，而是直接作为下一轮 MapReduce 任务的输入，从而构成迭代的 MapReduce 工作流[12-13]。

2　热搜品牌计算方法

用户在电商网站检索商品时，输入的关键字往往并不是商品的品牌名称，为了从用户的每次检索行为统计出品牌检索热度，本文在搜索引擎的检索日志里增加记录每次用户检索的结果集在商品品牌字段上的层面统计，格式如图 2 所示。

```
2018-12-30 23:19:39|康师傅|brandFacet|00006:66
2018-12-30 23:19:43|可乐|brandFacet|00005:33 00028:13
```

图 2　品牌层面检索日志格式

Fig.2　Log format of facet search on brand

日志文件中每行为一条检索日志记录，每条检索日志记录包含多个属性，各个属性由符号‘|’分隔。图 2 中第一个属性表示检索时间；第二个属性表示检索关键字；第三个属性“brandFacet”是品牌层面统计的标识，说明下一个属性表示本次检索的结果集在品牌字段上的层面统计；第四个属性表示层面统计，由若干个键值对组成，一个键值对中的键和值用符号‘:’分隔，键值对间用空格分隔。例如图 2 中第一行的含义为用户在 2018-12-30 23:19:39 输入关键字“康师傅”进行检索，本次检索在品牌层面的统计结果为“00006:66”，即共检索出 66 个品牌编码为“00006”（“00006”是康师傅的品牌编码，品牌名称与品牌编码的对应关系存储于字典表中）的商品。图 2 所示的检索日志格式还可以扩展为每行同时记录多维层面统计结果。

接下来根据每次检索的品牌层面统计结果计算出本次检索对该结果集中各品牌热度的贡献值。具体计算过程如公式(1)所示。

$$H_i = \frac{N_i}{\sum_{j=1}^{n} N_j} \tag{1}$$

其中，H_i 表示某次检索对品牌 i 的热度贡献值，N_i 表示该次检索结果集中品牌 i 的商品数目，n 表示该次检索结果集中品牌的数目。可见 H_i 的取值范围是(0,1]，其值越大，表示本次检索对品牌 i 的热度贡献值越大。

最后，将某个时间段内的检索日志对各品牌的热度贡献值进行归并累加和排序即可得到该时间段内各品牌的搜索热度排行榜。

3　MapReduce 工作流设计

从搜索引擎的检索日志统计热搜品牌需要进行三轮的 MapReduce 作业。第一轮 MapReduce 作业完成各个品牌的搜索热度的计算，输出一系列键值对<品牌编码，搜索热度>。第二轮 MapReduce 作业将第一轮作业的输出与品牌字典表进行连接操作，将品牌编码信息转换为“品牌编码|品牌名称”。第三轮 MapReduce 作业对第二轮作业的输出按 value（即品牌搜索热度）的值进行降序排序输出。

3.1　第一轮 MapReduce 作业的算法设计

第一轮 MapReduce 作业的输入为搜索引擎的检索日志文件，读取日志里的每条品牌层面统计结果，在 Map 阶段按照公式(1)算出该次检索对检索结果集中各品牌的热度贡献值，在 Reduce 阶段对相同品牌的热度贡献值进行累加。第一轮 MapReduce 作业的 map 和 reduce 函数的具体实现代码如下：

```
map(LongWritable ikey, Text ivalue, Context context) {
String readline = ivalue.toString();
/*取每行的第四个属性(即品牌层面统计),利用空白符进行分割得到一个字符串数组,如读图 2 第二行则 brands 数组包含两个元素:"00005:33"和"00028:13" */
String[] brands = readline.split("\\|")[3].split("\\s+");
float sum = 0;
/*算出每次检索结果集中商品的总数量,如读图 2 第二行则 sum=46 */
for(int i=0;i<brands.length;i++)
{
if(brands[i].contains(":"))
sum += Float.parseFloat(brands[i].split(":")[1]);
```

```
else return;
}
/*算出每次检索对检索结果集中各品牌的热度贡献值,如读图 2 第二行则输出<"00005",33/46>,<"00028",13/46> */
for(int i=0;i<brands.length;i++)
{
String[] brand = brands[i].split(":");
context.write(new Text(brand[0]), new FloatWritable(Float.parseFloat(brand[1])/sum));
}
}
/* reduce 函数:接收<k,List(v)>形式参数,这里 k 表示品牌编码,List(v)表示搜索热度值的列表 */
reduce(Text _key,Iterable<FloatWritable> values, Context context) {
float sum =0;
/* 对品牌的搜索热度值进行累加 */
for (FloatWritable val : values) {
sum += val.get();
}
/* 输出键值对<品牌编码,搜索热度> */
context.write(_key, new FloatWritable(sum));
}
```

3.2 第二轮 MapReduce 作业的算法设计

第二轮 MapReduce 作业实现将第一轮作业的输出与品牌字典表的连接操作。因为品牌字典表仅存储品牌编码与品牌名称的映射,其数据集足够小到可以完全放到缓存中,所以这里采用 MapReduce 提供的复制连接(Replication join)策略。复制连接常用于大数据集与小数据集的连接操作,它是一种 Map 端连接,省去 Shuffle 和 Reduce 的过程,大大降低了作业运行时间。复制连接的基本思路如下:

(1)在 main 方法中调用 Job 对象的 addCacheFile(URI uri)方法将品牌字典表复制到所有运行 map 任务的节点的缓存中。其中 uri 为品牌字典表在 HDFS 上的地址。

(2)在各个 map 任务的 setup 方法中调用 context.getCacheFiles()从缓存中取出这个品牌字典表,装载到一个哈希表 brandMap 中。

(3)在 map 函数中遍历哈希表进行连接操作。

(4)输出结果(即没有 Reduce 阶段)。

第二轮 MapReduce 作业的 map 函数的具体实现代码如下:

```
map(LongWritable ikey, Text ivalue, Context context) {
Stringreadline = ivalue.toString();
/* 读取第一轮 MapReduce 作业的输出并用'\t'分隔得到的数组 reads 有两个元素,reads[0]表示品牌编码,reads[1]表示搜索热度 */
String[] reads = readline.split("\t");
//如品牌编码不在字典表中,则不统计
if(brandMap.get(reads[0])==null) return;
kout.set(reads[0]+"|"+brandMap.get(reads[0]));
/* 输出键值对<品牌编码|品牌名称,搜索热度> */
context.write(kout, new FloatWritable-
(Float.parseFloat(reads[1])));
}
```

3.3 第三轮 MapReduce 作业的算法设计

第三轮 MapReduce 作业实现将第二轮作业的输出按照 value(即品牌搜索热度)的值进行降序排序并取 TOP-N。因为在 Map 端的 Shuffle 过程中会对 map 函数的输出按照 key 做升序的默认排序。现要按照 value 进行排序,所以第三轮 MapReduce 作业的 map 函数要实现将 key 和 value 互换,即输入<k,v>,则输出<v,k>,这样在 Shuffle 阶段就可以按照品牌的搜索热度值进行排序,最后在 reduce 函数中再次对 key 和 value 互换即可得到排好序的键值对<品牌编码|品牌名称,搜索热度>。因为默认排序是升序排序,现要降序排序,而品牌搜索热度的类型为 FloatWritable,所以需定义一个比较器类,继承 FloatWritable.Comparator 并重写其中的 compare 方法,返回-super.compare()。第三轮 MapReduce 作业的 map 和 reduce 函数的具体实现代码如下:

```
map(LongWritable ikey, Text ivalue, Context context) {
String[] reads=ivalue.toString().split("\t");
FloatWritable kOut=new FloatWritable();
```

```
TextvOuT = new Text();
kOut.set(Float.parseFloat(reads[1]));
vOut.set(reads[0]);
/*输出键值对<搜索热度,品牌编码|品牌名称>*/
context.write(kOut, vOut);
}
reduce(FloatWritable _key, Iterable<Text> values, Context context){
/*输出键值对<品牌编码|品牌名称,搜索热度>*/
for(Text val : values){
        //搜索热度值保留两位小数
    context.write(val, new FloatWritable-
((float)(Math.round(_key.get() * 100))/100));
}
}
```

在上面的 reduce 函数中是把所有的键值对<品牌编码|品牌名称,搜索热度>都输出到 HDFS,当然如果只需输出 TOP-N,则可以定义一个变量充当循环变量,在 for 循环里输出 N 次即可。

4　实验

4.1　实验环境

在福建工程学院大数据教学服务器上虚拟化出 9 个节点,在这 9 个节点上搭建 Hadoop 分布式集群。实验放在该集群上运行,集群中每个节点的硬件配置为 2 核 CPU,8G 内存,操作系统为 Ubuntu 16.04,Hadoop 版本为原生的 Hadoop 2.7.5,JDK 版本为 1.8。

集群中每个节点的主机名和 IP 地址如表 2 所示。

4.2　实验数据

实验数据采用便利宝电商网站(www.wqblb.com)2018 年 12 月份的检索日志数据,当月的检索日志共有 3 327 413 条检索记录,日志文件总大小为 1 124MB。品牌字典表文件共含 7 748 条记录,文件大小为 124 kB。

4.3　响应时间

图 3 是分别在 3、5、7、9 个节点的集群环境下运行完整工作流的响应时间。从图 3 可以看出在 9 个节点的集群环境下响应时间只要 28.64 s,满足批处理的响应需求。

表 2　节点的主机名和 IP 地址

Tab.2　Host name and IP address of node

主机名	IP 地址	角色
brandMaster	172.17.0.2	主节点
slave1	172.17.0.3	从节点
slave2	172.17.0.4	从节点
slave3	172.17.0.5	从节点
slave4	172.17.0.6	从节点
slave5	172.17.0.7	从节点
slave6	172.17.0.8	从节点
slave7	172.17.0.9	从节点
slave8	172.17.0.10	从节点

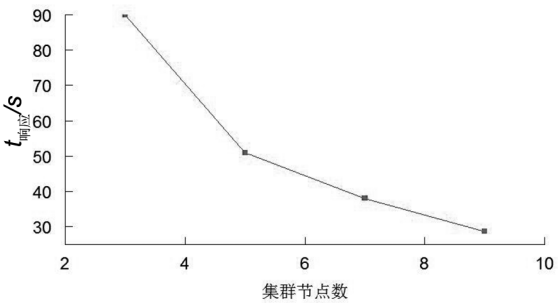

图 3　MapReduce 工作流的响应时间

Fig.3　Response time of MapReduce workflow

4.4　TOP-N 分析

在实验中,设置 N 为 10。获取到搜索热度 TOP-10 的品牌列表为{00006|康师傅,00606|悦巢,00005|可口可乐,00031|晨光,00037|心心相印,01450|保为康,00007|统一,00008|达利园,00290|伊利,00011|农夫山泉}。为了验证实验计算出的品牌搜索热度排名的准确性,可以和该期间各品牌产生的成交量排名对比,如图 4 所示。各品牌产生的成交量排名可以通过查询订单明细表,按品牌编码分组统计成交量排名。

从图 4 可看出,品牌搜索热度排名和成交量排名大致相当。从而验证了从搜索引擎日志里挖掘品牌搜索热度排名榜的可行性。

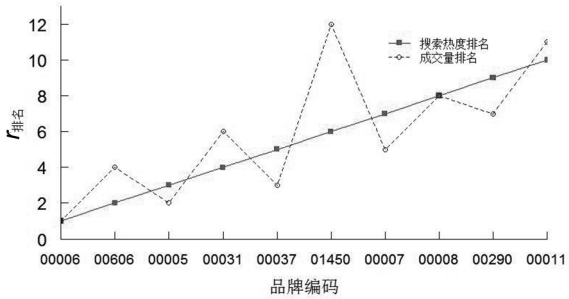

图 4 品牌搜索热度排名和成交量排名对比

Fig.4 Comparison of brand search popularing rankings and trading volume rankings

5 结语

快消品电商网站搜索引擎的检索日志记录着用户的检索偏好,本文根据检索日志里的品牌层面统计结果挖掘出用户每次检索对检索结果集中各品牌的搜索热度的贡献值,设计一个迭代的 MapReduce 工作流用于计算网站某时间段内的各品牌的搜索热度总值排名,从而可以在网站适当地方向用户推荐热搜品牌。在未来的研究中,将引入 Spark 技术,以实现热搜品牌的实时个性化推荐。

参考文献:

[1] 王晨阳, 刘垣, 郭李华, 等. 融合位置相似性度量的快消品电商网站推荐算法[J]. 福建工程学院学报, 2017, 15(6): 586-590.

[2] 姚晓娜, 祝忠明. 基于分面搜索引擎 Solr 的机构知识库访问统计[J]. 现代图书情报技术, 2011(Z1): 37-40.

[3] 杜江, 张铮, 张杰鑫, 等.MapReduce 并行编程模型研究综述[J]. 计算机科学, 2015, 42(S1): 537-541, 564.

[4] 林子雨. 大数据技术原理与应用[M]. 北京: 人民邮电出版社, 2015:132-139.

[5] 徐文涛, 刘锋, 朱二周. 基于 MapReduce 的新型微博用户影响力排名算法研究[J]. 计算机科学, 2016, 43(9): 66-70, 86.

[6] 高见文, 薛行贵, 罗杰, 等. 基于迭代式 MapReducede 的海量数据并行聚类算法研究[J]. 中国科技论文, 2016, 11(14): 1626-1631.

[7] 梁秋实, 吴一雷, 封磊. 基于 MapReduce 的微博用户搜索排名算法[J]. 计算机应用, 2012, 32(11): 2989-2993.

[8] 李锐, 王斌. 文本处理中的 MapReduce 技术[J]. 中文信息学报, 2012, 26(4): 9-20.

[9] 薛胜军, 潘吴斌. 基于 MapReduce 的气象数据并行 PK-means 算法[J]. 武汉理工大学学报, 2012, 34(12): 139-142.

[10] 李伟卫, 赵航, 张阳, 等. 基于 MapReduce 的海量数据挖掘技术研究[J]. 计算机工程与应用, 2013, 49(20): 112-117.

[11] 陈子军, 张娟娜, 刘文远.MapReduce 框架下基于范围的空间文本相似连接[J]. 小型微型计算机系统, 2015, 36(10): 2245-2251.

[12] WU R. Cyclic workflow execution mechanism on top of MapReduce framework[C]//Seventh International Conference on Semantics, Knowledge and Grids. Washington,DC:IEEE Computer Society,2011: 28-35.

[13] YOO D, SIM K M. A scheduling mechanism for multiple MapReduce jobs in a workflow application (position paper)[C]//Computing, Communications & Applications Conference: IEEE, 2012: 405-410.

[14] 林子雨, 李雨倩, 李粲, 等. PipelineJoin: 一种新的基于 MapReduce 的多表连接算法[J]. 中国科学技术大学学报, 2015, 45(10): 836-845.

(责任编辑: 方素华)