一种改进的海量信息分类算法

戴立平,谭正华,张进修,张又文
(湘潭大学 信息工程学院,湖南 湘潭 411105)

摘 要:首先对信息分类过程中出现的问题进行了阐述,指出了传统信息分类方法的不足与缺陷,以及后续查询效率较低的问题等。接着分析了普通的信息分类方法,主要包括了分类方式与查询方式,然后针对企业海量信息的查询效率较低及信息分类层级较少的问题,提出一种改进的海量信息分类算法。采用 64 位二进制值作为信息分类编码,通过对分类编码的解析,能够准确并较为快速地分析出信息分类层次结构等信息。该编码机制可以实现 8.59×10^{18} 个分类。将优化算法与传统算法进行了比较,实验结果表明:在分类层级较多的情况下,改进的迭代算法相较于普通的迭代算法在耗时上要少,且当分类层级和分类总数增加时,优化的分类算法在耗时增长速率上优于普通的迭代算法,提高了查询效率。

关键词:海量信息;二进制;分类层级;改进迭代算法
中图分类号:TP301.6 **文献标识码**:A **文章编号**:1673-629X(2019)10-0201-04
doi:10.3969/j.issn.1673-629X.2019.10.039

An Improved Algorithm for Classification of Massive Information

DAI Li-ping,TAN Zheng-hua,ZHANG Jin-xiu,ZHANG You-wen
(School of Information Engineering,Xiangtan University,Xiangtan 411105,China)

Abstract:Firstly,the problems in the process of information classification are elaborated,and the shortcomings of traditional information classification methods,as well as the low efficiency of subsequent queries are also pointed out. Secondly,the common information classification methods are analyzed,including classification and query. Then,aiming at the problem of low query efficiency and less information classification hierarchy of enterprise massive information,an improved massive information classification algorithm is proposed. 64-bit binary value is used as information classification coding. The information like hierarchical structure of information classification can be worked out accurately and quickly by analyzing classification coding. The encoding mechanism can achieve $8.59*10^{18}$ categories. Comparing the optimized algorithm with traditional algorithm,the experiment shows that the improved iterative algorithm is less time-consuming than the ordinary iterative algorithm in the case of more classification hierarchies,and the optimized classification algorithm is better than the ordinary iterative algorithm in the time-consuming growth rate when the classification hierarchies and the total number of classifications increase. It also improves query efficiency.

Key words:massive information;binary;classification level;improved iterative algorithm

0 引 言

信息的分类常常表现为树状结构,在实际应用中,一般用 Table 表表示,而快速生成资讯的所在路径和实现某个分类的增删,是研究的关键和热点。

通常情况下,在数据库中对每一个分类新建一个数据表,增加一个分类就增加一张表。当需要将该分类的信息显示时,通过查找找到该分类的数据表,但操纵数据表比操纵数据表中的记录困难得多,因此数据表的多少直接关系到查询的效率。

Hao Chunmei 等[1]提取信息属性的特征,根据特征建立优化信息分类模型,实现高效的信息分类,为快速查询搜索目标提供支持;张成刚等[2]提出利用降噪编码自适应神经网络对采集的信息进行降噪处理并利用分类规则对海量信息进行智能分类,但该方法计算过程较为复杂,且分类耗时较长;曾劲松等[3]提出基于冲突博弈算法的海量信息智能分类,通过提取信息特征,确定信息分类策略。

以上方法难以很好地解决信息分类查询的效率问

收稿日期:2018-11-15 **修回日期**:2019-03-19 **网络出版时间**:2019-04-24
基金项目:湖南省自然科学基金(11JJ4051)
作者简介:戴立平(1993-),男,硕士,CCF 会员(65180G),研究方向为数据仓库与数据挖掘;谭正华,博士,研究方向为计算机图形学与数字矿山理论技术、数据库技术。
网络出版地址:http://kns.cnki.net/kcms/detail/61.1450.TP.20190424.1055.080.html

题,因此文中提出一种海量信息分类算法,包括信息分类编码和算法 API,实现了分类处理的各种应用需求,进一步提高了查询搜索效率。

1 普通的信息分类方法

在信息分类中,树形结构的每一个节点表示一个分类[4],定义一个 Catalog 表,分类节点中只保留一个 Name 信息。添加一个 CatalogId 自增主键作为节点编号。为了将该树形结构更为准确地描述出来,在节点中再增加一个字段,将这个字段命名为 FatherID。如果节点 ID2 是 ID1 的子节点,那么说明这个分类是该分类的子节点,即 ID1 就是 ID2 的 FatherID。约定 0 为第一层父类编码,作为一个在数据库中无任何记录的虚拟分类。

上面定义的 Catalog 表可以较为轻松地恢复出一棵分类树。当需要查找该父类的下级分类时,只需要使用 SQL 语句选择表中 FatherID = FID 的部分。通过调用寻找分类下子分类的 GetChildren 函数,然后把该子分类的 CatalogId 主键作为下一个分类的 FatherID,通过递归调用得到该分类下的所有子分类[5]。

假设要搜索某一个大类下的所有资讯,该大类下有较多的分类层级,而且资讯靠近根分类[6]。不仅需要搜索出大分类的信息,也要将子分类的信息搜索出来。一般情况下,将递归调用 GetChildren 函数查找得到所有分类及其子分类,通过获得的分类 ID,调用 GetProduct 函数查询该分类对应的产品。比如一个分类下其子分类,子子分类有 20 个,那么函数调用次数将达到 20 次,其查询效率较为低下。

2 分类编码的优化设计

2.1 文档分类存储表结构

在设计中,主要使用了两张表(见图 1)。

图 1 数据结构表

一张为信息分类基础 DocCatalog 表,主要用来存储文档的分类信息,里面主要有 4 个关键字段。其中 CatalogId 为分类节点 ID,FatherId 为分类父节点 ID,Name 为分类名称;另一张表为 Documents 表,主要用来存储文档数据,关键字段:DocId 为文档 ID,CatalogId 为文档所属分类 ID,这个字段作为外键与 DocCatalog 表中的主键 CatalogId 具有约束关系[7]。

2.2 信息分类编码结构

将 DocCatalog 表中的主键 CatalogId 定义为 bigint 数据类型[8],大小为 8 个字节,存储 64 位二进制。如图 2 所示,将 64 长度从右到左分割成 9 段,每一段为 7 位,最左边剩余的一位废弃不用,可以获得 ABC 至 I 个完整的二进制段,规定当 7 位编码全为 0 时不表示任何分类编码。A 段位第一层分类编码,每一层分类的编码最小值是从 0000001 开始表示的。全零的段值不表示分类编码。当创建第一层的第一个分类时,就用二进制 0000001 表示,如果该层级需要创建第二个分类,分类编码就为 0000010,本层级的最后一个分类用编码 1111111 表示。

图 2 信息分类编码

设计的分类编码结构,对于每一个层级,可以有 $2^7 = 128$ 个编码,除去全为 0 的编码,每层级可以表示 127 个分类。对于 64 位编码,一共有 127^9 个分类。这对于企业信息分类已经足够使用[9]。

当 A 段分类下无子分类的时候,那么其他段位值全为 0。当 A 段一层分类有子分类时,则第二层的子分类编码从 0000001 开始。从 A 至 I 段的编码总是从 0000001 开始到 1111111 结束。如图 3 所示,这是一个多层分类实例。

假如给定一个分类,其十进制的编码值为 4210818,将其转换成二进制编码:10000001000000010000010,按从右到左 7 位为一段,不足 7 位补 0,依次为:0000010 0000001 0000001 0000001,一共分割成了 4 段,该分类为第四层子分类。可以得到其上父节点的编码为 0000001 0000001 0000001,十进制编码值为 16 513;第二层级的编码为

0000001 0000001,十进制编码值是 129;第一层级的分类编码为 0000001,十进制编码值是 1。对于分类的分类层级和其所有的分类路径,只需要给定该分类的编码值,就能完整地分析出来[10]。

信息分类示例	层级	分类编码二进制表示						编码十进制值
		I段	……		C段	B段	A段	
˅1 分类一	第一层		……				0000001	1
˅分类 1.1	第二层		……			0000001	0000001	129
˅分类 1.1.1	第三层		……		0000001	0000001	0000001	16513
分类 1.1.1.1	第四层		……	0000001	0000001	0000001	0000001	2113665
分类 1.1.1.2	第四层		……	0000010	0000001	0000001	0000001	4210818
分类 1.1.2	第三层		……		0000010	0000001	0000001	32897
˅分类 1.2	第二层		……			0000010	0000001	257
分类 1.2.1	第三层		……		0000001	0000010	0000001	16641
分类 1.2.2	第三层		……		0000010	0000010	0000001	33025
分类 1.3	第二层		……			0000011	0000001	385
˅2 分类二	第一层		……				0000010	2
分类 2.1	第二层		……			0000001	0000010	130
分类 2.2	第二层		……			0000010	0000010	258
3 分类三	第三层		……				0000011	3
4 分类四	第四层		……				0000100	4

图 3　分类编码二进制位

3　算法实现

基于此分类编码,设计了一套使用简单、效率高效的算法。该算法基于微软 LINQ 技术进行实现[11]。在信息分类的程序集中定义一个 CatalogTreeBase 信息分类基础类,另外还定义了几个关键基础信息。

(1)MaxLevels,这是信息分类的总级数,由前面分类级数可知,其值为 9,是一个常量。

(2)LevelLengh,这是分类编码的二进制长度,也是一个常量,其值为 7。

(3)分类 K 码,这是算法中的关键信息,其值的定义为 $2^{LevelLengh}$。这是一个永恒不变的常量,由于上述 LevelLength 值为 7,所以 K 值为 128。

(4)LevelGrade 分类编码层,在计算机中,移位操作是非常便捷高效的。通过对编码值的右移操作,将编码值进行分割,那么 LevelGrade 值就是其分割后的段数。

(5)父类截取码,在程序中用 LevelK 表示。这是一个变量,用于将分类编码与父类截取码进行运算后取得上一级的父类编码,算法定义为 $K^{LevelGrade}$。按照定义,对于第一层的 LeveK 值即为 K 码,第二层为 K^2,依次类推,可以看出当父节点相同时,其他兄弟分类具有相同的 Level 值。

在实际应用中,如果给出一个分类编码的 Id[12],对于其父类的编码值只需要用分类 Id 的编码对上层的 LevelK 进行取模运算:FatherId = Id% GetFatherLevelK(Id)。当输入分类编码值 Id,即可求出上一级的 LevelK 值。在项目的实际应用中,为了能够便于随时进行调用,定义:public static ulong GetFatherId(ulong Id)。这是一个计算父节点编码的函数。而对于海量信息分类的解决方案,传统方法是通过对数据库进行迭代查询操作,改进的分类算法只需要进行取模运算,即能快速地得到分类的父分类编码[13]。

对于获取某一分类下的直接下一级子分类,或者根据指定的分类编码获取下级的所有分类,在信息方案中,普通方法是采用迭代循环的方式,调用查找子类的方法,来找到所有的子分类。对于优化的分类编码,所有子分类编码中就包括了父节点的编码。在运算的时候,如果子分类的编码中能够解析出父类编码,那么该子类就能够被视为找到。LINQ 的子类查询语句如下:

```
Public IQueryable<Catalog> GetSubCatalog(long id)
{
longlevelK = GetLevelK(id);
var query = from u dbContext. Catalog where u. CatalogId % levelK = = id select u;
return query. AsQueryable<Catalog>();
}
```

在上述语句中,对于父分类编码的获得,在查询时只需要分类编码对父分类 levelK 进行取模操作,u. catalogId% levelK = = id。在图 3 中,对于 3 层级分类 1.2.1,其分类编码值为 16 641,父类的 levelK 为 K^2 = 16 384,对其取模得到 16 641%16 384 = 257,而二层级分类 1.2 的编码值就是 257,可以判定为该分类是分类 1.2 的子分类,同理可判定 1.2.2 也是分类 1.2 的子分类。如果需要查找一个指定分类的下层级子分类,使用 LINQ 查询时,增加一个判定条件,下层级的比父分类层级大 1,其代码如下[14]:

```
Public IQueryable<Catalog> GetChildrenCata
{
int levelGrade = GetLevelGrade( id) +long levelK = GetLeveelK
( id) ;
var query = from u dbContext. Catalog where u. CatalogId % levelK select u;
return query. AsQueryable<Catalog>( ) ;
}
```

由于 LINQ 语句不能直接去调用外部函数 GetLevelGrade( ),这里使用的是伪代码,用来判定父分类比子分类的层级高 1 级。

在实际项目中,经常需要恢复出目录树。采用递归的方式,调用前面定义的 GetChildrenCatalog( ) 函数,就能较为简便轻松地实现完整的信息分类树。

4 算法测试

为了测试改进的海量信息分类查询方法和普通迭代信息分类方法的效率差异,将在不同测试条件下进行算法的性能测试,其测试环境相同。测试主要采用的实例为生成分类树。

测试一:查找的分类指定,其 Id 为 1,分类层级为 3,每层级有 8 个子分类。

测试二:查找的分类指定,其 Id 为 1,分类层级为 4,每层级有 8 个子分类。

测试三:查找的分类指定,Id 也为 1,分类层级为 4,每层级有 20 个子分类。

图 4 和图 5 为测试用时结果比对,可以得出以下几点结论:

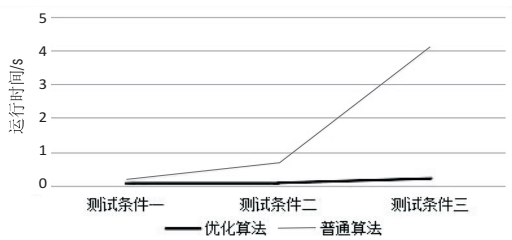

图 4 运行耗时结果比对

测试条件	测试条件一	测试条件二	测试条件三
普通迭代算法用时(s)	0.15	0.68	4.09
优化的分类算法(s)	0.08	0.09	0.22

图 5 测试用时

(1)当分类的层级较少且分类数目不多时,改进的海量信息分类算法和普通的迭代算法在耗时时间上差异较小,但改进的分类算法性能上较优。

(2)在分类层级增加,而且分类总数也增加的情况下,改进的分类算法要比普通的迭代算法少一个数量级的时间。

(3)当分类层级和分类的总数大量增加时,改进的分类算法在耗时上差异较小,在增长速率上更加优于普通的迭代算法。

5 结束语

提出了一种查询、检索海量信息的分类算法,其特点如下:优化编码设计,编码值包含了信息分类的层次结构,能提供最大 9 层级约 8.59×10^{18} 个分类,保证了信息分类结构的完整。通过基于微软的 LINQ 技术设计了一套简单、高效的算法,借助计算机的高速运算能力,能快速解析出信息分类的结构层次与分类编码。优化的算法为海量信息分类提供了参考,有利于提高信息数据检索的效率。

参考文献:

[1] HAO Chunmei. An information classification optimization model based on improved ant colony algorithm[J]. Advanced Materials Research,2014,989-994:1989-1992.

[2] 张成刚,宋佳智,姜静清,等. 一种改进的降噪自编码神经网络不平衡数据分类算法[J]. 计算机应用研究,2017,34(5):1329-1332.

[3] 曾劲松,饶云波. 基于冲突博弈算法的海量信息智能分类[J]. 计算机科学,2018,45(8):208-212.

[4] 路永和,彭燕虹. 融合实用性与科学性的互联网信息分类体系构建[J]. 图书与情报,2015(3):118-124.

[5] 张 晶. 基于软件编程的 SQL 语句读写操作研究[J]. 电子科学技术,2017,4(3):58-60.

[6] 石 微,王 欢. 信息编码体系在企业信息化中的作用[J]. 计算机光盘软件与应用,2014(21):136.

[7] KOUTRIS P,WIJSEN J. Consistent query answering for self-join-free conjunctive queries under primary key constraints[J]. ACM Transactions on Database Systems,2017,42(2):1-45.

[8] KARNOK D,KEMÉNY Z,ILIEZUDOR E,et al. Data type definition and handling for supporting interoperability across organizational borders[J]. Journal of Intelligent Manufacturing,2016,27(1):167-185.

[9] 李 兰,刘 洋,邵明文. 客户海量兴趣数据分类的推荐系统优化仿真[J]. 计算机仿真,2014,31(9):449-453.

[10] 李志虹. 基于遗传迭代优化的云计算下海量数据分类查询[J]. 科技通报,2015,31(6):34-36.

[11] 唐 磊. 浅析 LINQ 技术原理及应用[J]. 计算机光盘软件与应用,2014(2):134-136.

[12] 赵 琰. 主数据分类及其编码管理体系的信息化建设[J]. 浙江理工大学学报:自然科学版,2015,33(4):570-573.

[13] CHU Yanhua,YU Jinling. The research of database query optimization based on XML[J]. Advanced Materials Research,2012,546-547:519-525.

[14] 黄红伟,谭 鹏,李 俊,等. 基于 LINQ 及表达式树的组合查询设计[J]. 计算机与网络,2017,43(17):62-65.