

OLAP 分析中一种有效的下钻路径裁减技术^{*})

张德辉 唐世渭 杨冬青 马秀丽 姜力争

(北京大学信息技术学院 北京 100871)

摘 要 OLAP 技术为企业数据分析提供了极大的便利。然而,复杂的多维结构导致了复杂的下钻路径组合,从而使得用户的数据分析效率低下。解决 OLAP 分析中用户的探查路径过于复杂冗长是 OLAP 面临的主要问题之一,但是当前的研究成果大多由于与特定的分析任务相关而不能完全解决该问题。本文提出了一种与分析任务无关的下钻路径裁剪方法。该方法从多维数据结构本身出发将无效的下钻从分析过程中裁剪掉,从而达到了简化分析过程的目标。本文采用向量夹角法评估下钻操作的有效性,并给出了对应的高效下钻路径裁剪算法。该算法以有序的实事表为输入,只需一次扫描即可完成裁剪过程。实验结果证明了本文方法的可行性、高效性、抗稀疏性和抗偏斜性。

关键词 OLAP, 向量夹角, 下钻路径

An Effective Drill-down Paths Pruning Method in OLAP

ZHANG De-Hui TANG Shi-Wei YANG Dong-Qing MA Xiu-Li JIANG Li-Zheng

(Department of EECS, Peking University, Beijing 100871)

Abstract The OLAP provides a convenient way for data analysis. But the complexity of multi-dimensional data structure affects the efficiency of OLAP, because there are too many drill-down paths to be chosen from when analysis. How to prune the complex and tedious analysis paths is one of the major issues in this research field. While most methods in the literature are associated to some specific analysis tasks, so they cannot get reasonable effect. In this paper, we proposed a new method that is irrelevant to analysis task. In order to simplify the OLAP analysis process, we prune the invalid drill-down operations. The vectorial angle method is employed to evaluate the validness of every drill-down operation. We give the corresponding path pruning algorithm, and it is effective that it takes the fact table as the input in only one pass scanning. The experiments show that our method is feasible, effective, sparsity-proof and skewness-proof.

Keywords OLAP, Vectorial angle, Drill-down paths

1 引言

OLAP(OnLine Analytical Processing)技术为企业的数据分析提供了极大的便利。然而,在为用户提供灵活的分析功能的同时,OLAP 分析也存在一定的不足。在当前的 OLAP 技术条件下,用户的分析基本上是通过手动执行 OLAP 操作来完成的。为了能够发现隐藏在数据中的有用知识,用户必须执行一系列的 OLAP 操作。从汇总级别较高的数据集开始,用户可能需要经过几十次,甚至上百次的钻取和回退才能发现有价值的知识。这种分析模式效率比较低,甚至有可能导致用户迷失在复杂的分析过程中。

为了达到分析目标而必须探查的分析路径过于复杂冗长是影响 OLAP 分析效果的主要原因之一。这个问题已经得到了充分认识,并且研究人员从不同的角度出发已经取得了一定的研究成果,并大致可以分成三类:提供自动化的操作,划分立方体单元的等价类和提取数据立方体的数据特征。

自动化的操作是从用户的分析任务出发,将某些分析任务所需要执行的一系列操作抽象成为一个单独的操作,使得用户的某些分析任务变成一个单一操作,从而简化了用户的分析探查路径。目前这方面的研究主要包括:智能上卷^[1],智能下钻^[2]和 INFORM^[3]。

另外一些研究人员则从立方体数据的多维特点出发,采

用不同的方法将立方体单元划分成一些等价类,然后用这些等价类来回答用户的分析查询。由于等价类的数目远小于单元数目,因此 OLAP 分析过程中的可选路径将得到缩减。这类研究成果主要包括:Range Cube^[4]和 Quotient Cube^[5]等。

由于 OLAP 分析的目标是要发现隐藏在数据中的规律,有些研究人员尝试了数据挖掘与 OLAP 的结合技术来提取数据立方体的特征。目前,提取立方体数据特征的主要方法包括:频繁特征^[6]、关联特征^[7]和聚类特征^[8]等。

这三类方法从不同的角度出发,提高了 OLAP 技术的可用性,然而它们并没有完全解决问题。OLAP 操作的自动化和数据立方体特征的提取都是与分析任务相关的,并非对所有的分析任务所涉及的操作组合都可自动化。而将数据立方体单元划分成等价类取决于多维数据本身的分布状态,以 Quotient Cube^[5]为例,等价类存在的前提是数据中存在一种被称为单一元组的特殊数据分布,而并非所有聚集值相等的单元都属于同一个等价类。

事实上,决定 OLAP 分析路径复杂程度的因素是下钻操作。基于这个事实,本文从数据的多维结构本身出发使得某些下钻路径得到裁剪,从而大大减少了用户分析过程中的可选路径数目。本文对于每个下钻操作的结果向量与对应的全局向量相比较,如果该向量与全局向量包含的信息相同,则认为这次下钻无效。如果对于一个无效的下钻,它的结果单元

^{*})本课题受到国家自然科学基金的资助,项目批准号:60473072。

的任何下钻都是无效的,那么我们将在分析路径中将该次下钻裁剪掉。通过下钻路径的裁剪,用户的任何查询分析都可以在更少的操作次数内实现,并且在裁剪过程中,通过与全局向量的对比保证了信息的无损。

本文采用向量夹角法来衡量钻取的有效性,如果某次下钻的结果向量与对应的全局向量的夹角小于某个阈值,我们可以认为两个向量线性相关,即包含相同的信息。

我们给出了用于路径裁剪的高效算法,该算法以有序的事实表为输入,只需一次扫描即可完成裁剪。实验结果证明了本文方法的可行性、高效性、抗稀疏性和抗偏斜性。

2 相关概念

为了便于表述,本文用不包含层次的维属性来描述路径的裁剪方法。虽然引入层次结构后,裁剪的方法会有所不同,但是二者的基本思路完全相同,只需对相关定义和处理方法稍加修改即可实现。

定义 2.1(多维数据集) 设多维数据集 R 形为 $R: (D_1, D_2, \dots, D_n; Measure)$, 其中 D_i 表示第 i 个维属性, n 为维属性的个数, $Measure$ 表示度量属性。记函数 $Mem(D_i)$ 表示维属性 D_i 的维成员集合。

定义 2.2(立方体单元) 与 R 对应的数据立方体 C_R 中的一个单元 x 表示为: $t: (t_1, t_2, \dots, t_n; value)$ 。其中 $value$ 表示该单元的聚集值,而 t_i 的取值分为如下不同情况:

1) $t_i = *$, 表示第 i 维上的取值为“ All ”, 特殊地, 我们称元组 $t: (*, *, \dots, *, value)$ 为 C_R 的根单元, 记作 Θ ;

2) $t_i = y, y \in Mem(D_i)$;

3) $t_i = ?$, 表示 $t_i = \forall y \in Mem(D_i)$;

4) 如果 $t \in R$, 称 t 为一个基本单元。

如果单元 t 的所有维属性值都不等于“?”, 那么 t 表示一个确定的单元; 否则 t 将代表一个单元的集合, 特殊地, 如果只有一个维属性值 t_i 等于“?”, 那么所表示的立方体单元将只在第 i 维上取不同值, 此时我们称 t 为一个关于 D_i 的向量。

OLAP 中的下钻操作是将立方体单元 t 的维成员值中某个取值为“*”的成员展开为“?”, 同时保持其他 $n-1$ 个维成员值不变的一种操作。

定义 2.3(下钻操作) 下钻操作(drill-down)操作在一个立方体单元的某个维属性值上, 得到的结果为一个更细粒度立方体单元的集合。其形为: $drill-down(t: (t_1, t_2, \dots, t_n; value), D_i) = s: (t_1, t_2, \dots, t_{i-1}, ?, t_{i+1}, \dots, t_n; value)$, 其中 $t_i = *$, 我们称 t 为操作单元, D_i 为操作维。易见 s 是一个关于 D_i 的向量。

对于一个立方体单元, 可选的下钻操作可能有多个, 在维属性不包含层次的情况下, 其个数等于所有 t 的维成员值中取值为“*”的个数。对于数据立方体 C_R 的根单元 Θ 来说, 可选的下钻操作共有 n 个, 并分别与多维数据集 R 的维属性 D_i 对应。

定义 2.4(全局向量) 给定多维数据集 R 和对应的数据立方体 C_R , 对于向量 $t: (t_1, t_2, \dots, t_{i-1}, ?, t_{i+1}, \dots, t_n; value)$, 如果对于任意 $j \neq i$ 都有 $t_j = *$, 我们称 t 为关于 D_i 的全局向量。 R 共包含 n 个全局向量。

定义 2.5(向量夹角) 给定两个 m 元向量 $X = (x_1, x_2, \dots, x_m)$ 和 $Y = (y_1, y_2, \dots, y_m)$, 向量 X 和向量 Y 的长度分别为 $L_X = \sqrt{x_1^2 + x_2^2 + \dots + x_m^2}$ 和 $L_Y = \sqrt{y_1^2 + y_2^2 + \dots + y_m^2}$, 则向

量 X 与向量 Y 的夹角可以用如下公式表示:

$$\theta_{XY} = \cos^{-1} \left(\frac{x_1 y_1 + x_2 y_2 + \dots + x_m y_m}{L_X L_Y} \right)$$

其中 $0 \leq \theta \leq \pi/2$ 。

3 下钻路径的裁剪算法

本节首先介绍下钻路径裁剪的形式化描述, 然后给出裁剪的算法。

3.1 下钻路径的形式化描述

数据立方体中长度为 k 的下钻路径定义为以根单元 Θ 为起点的连续 k 次下钻操作。对于一个包含 n 个非层次维属性的数据立方体来说, 下钻路径的最大长度为 n 。从第二次下钻开始, 每次都是以上一次下钻操作结果中的某一个单元作为本次操作的操作单元。而对于本次下钻操作的操作单元来说, 与上一次下钻操作的操作单元相比, 二者仅在一个维属性上的取值不同, 分别是某个具体的维成员值和“*”。

定义 3.1(下钻路径) 一条长度为 k 的下钻路径 p 形为: $p: ((t_1, D_{i_1}), (t_2, D_{i_2}), \dots, (t_k, D_{i_k}))$, 其中 $t_1 = \Theta$, t_j 表示该路径上第 j 次下钻操作的操作单元, 而 D_{i_j} 为本次操作的操作维。

给定一个 n 维数据集 R , 假设每个维属性的基数同为 λ , 那么 R 的一个饱和实例包含 λ^n 个元组, 与 R 对应的数据立方体 C_R 包含 $(\lambda+1)^n$ 个单元。在 OLAP 分析过程中, 从根单元 Θ 开始的分析路径最大长度为 n , 而路径总数为: $\sum_{k=1}^n k! \lambda^{\frac{k(k-1)}{2}}$ 。这些路径将组成一颗以根单元 Θ 为根结点的树结构, 其中操作维表示树的边, 用以区别两个连续操作单元维属性值的维成员 α_{ij} 表示树的结点。

定义 3.2(下钻路径树) 给定数据集 R 和对应的数据立方体 C_R , 所有从的根单元 Θ 开始的下钻路径组成一棵树, 称为下钻路径树。树的根结点是 Θ , 其余结点表示一个单元在某个维属性上的下钻结果单元集合, 形为 $(\alpha_1, \alpha_2, \dots, \alpha_s; pointer[s]; vector)$, 其中 s 为操作维的基数, α_i 用以区别结点中的不同单元, $pointer[s]$ 表示指向多个子节点的指针, $vector$ 表示结点中所有单元的度量属性构成的向量。树的每条边表示一个下钻操作, 并用该下钻操作的操作维来标识。

定义 3.3(无效下钻) 对于下钻操作 $drill-down(t: (t_1, t_2, \dots, t_n; value), D_i) = s: (t_1, t_2, \dots, t_{i-1}, ?, t_{i+1}, \dots, t_n; value)$, 并设其度量向量为 S, T 表示关于 D_i 的全局向量, 而参数 δ 表示预设的向量夹角阈值。

1) 如果 $t = \Theta$, 则下钻操作 $drill-down(t, D_i)$ 有效;

2) 如果 s 是基本单元集合, 并且 $\theta_{ST} \leq \delta$, 则下钻操作 $drill-down(t, D_i)$ 无效;

3) 如果 $\theta_{ST} \leq \delta$, 并且对于任意的 $D_j \neq D_i$ 和任意的 $z \in Mem(D_j)$, 都有 $drill-down((t_1, t_2, \dots, t_{i-1}, z, t_{i+1}, \dots, t_n; value), D_j)$ 无效, 则称下钻操作 $drill-down(t, D_i)$ 无效。

由定义 3.3 容易得到如下性质: 对于一个有效下钻 $drill-down(t, D_i)$, 任意一个以之为最后一次下钻操作的路径 p 中的其它下钻操作都是有效的。

本文所述的下钻路径裁剪策略实际上就是从下钻路径树中裁剪掉无效下钻操作的过程。

3.2 裁剪算法

事实上, 与数据立方体对应的下钻路径树只依赖于多维数据的结构本身, 因此, 无效路径的裁剪可以通过从完整的下钻路径树中删除有效路径的方法实现。

算法-1:下钻路径裁剪算法

输入:事实表 R ; $(D_1, D_2, \dots, D_n; \text{Measure})$, 夹角阈值 δ

输出:下钻路径树 T_{drill}

```

{  $T_{\text{drill}} = \text{NULL}$ ;
  对于  $R$  中的每个元组  $t$ 
  { 计算从  $\Theta$  到  $t$  的所有可达路径并插入  $T_{\text{drill}}$  中;
    for(路径上的每个节点)
    {if(新插入结点的 vector 域构成完整向量  $X$ )
      {if( $X$  与全局向量的夹角小于等于  $\delta$ )
        {删除该节点;
         令其父接点对应的指针域指向空结点;
        }
      }
    }
  }
  else
  {删除该节点;
   令其祖先结点对应指针域等于空值;
  }
}
```

设事实表包含元组数为 N , 每读取一个元组 t 需要创建对应的多个下钻路径, 这个过程的时间复杂性为 $O(n \log n)$, 空间复杂性为 $O(n^2)$ 。处理无效下钻的时间和空间复杂性均为 $O(1)$, 而处理有效下钻的时间和空间复杂性则均为 $O(n)$, 处理下钻操作的次数最多为 N 。因此, 算法的总体时间复杂性为 $O(N(n \log n + n))$, 空间复杂性为 $O(N(n^2 + n))$ 。在实际应用中 $n \ll N$, 因此本文算法的时间和空间复杂性均接近 $O(N)$ 。

4 实验结果

为了验证下钻路径裁剪的效果, 我们使用合成数据和实际数据做了一组实验。所有实验中的聚集函数为 SUM。我们用立方体中裁减前和裁减后始于根单元 Θ 的有效下钻路径数的比例来衡量裁剪的效果, 比例越小, 裁剪效果越好。

实验 1 验证了数据稀疏对算法的影响。我们生成了符合均匀分布的 5 个数据集合, 每个数据集合包含 4 个维属性, 并具有相同的基数 100, 元组的个数分别为 $10^4, 10^5, 10^6, 10^7$ 和 10^8 。结果如图 1 所示, 随着元组个数的增加, 不同向量夹角阈值下的裁剪效果都有上升趋势。这是因为当数据比较稀疏的时候, 聚集单元中存在大量的空值, 因此裁剪前的下钻路径就远少于饱和数据集的下钻路径数。这个实验说明了本文所述方法的抗稀疏性。

实验 2 验证了数据偏斜对算法的影响。我们产生了 7 个符合 Zipf 分布的数据集, 每个数据集有 4 个维, 并具有相同的基数 100, 元组的个数为 10^7 , Zipf 因子从 0 到 3。结果如图 2 所示, 随着偏斜程度的增加, 不同向量夹角阈值下的裁剪效果基本不变。这个实验说明本文所述方法的抗偏斜性。

实验 3 采用了湖北地税数据仓库系统的数据, 包含 7 个维, 一个度量, 共 25048308 个元组, 存储数据立方体约需 7.63GB。我们采用不同维数来测试裁剪的效果。图 3 表示不同维数的裁剪效果, 随着维数的增加, 裁剪前的下钻路径呈基数上涨, 因此裁剪效果也明显提高。图 4 表示用于存储裁减后的下钻路径树所需的硬盘空间, 随着维数的增加, 无效的下钻操作也将增加, 因此存储裁剪结果的空间也略有增加。

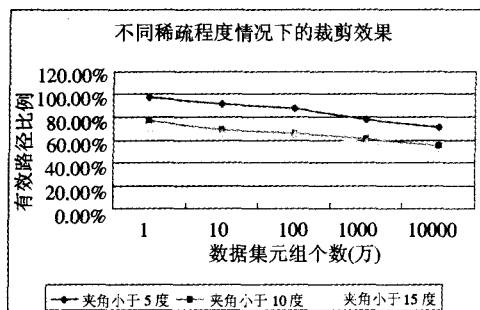


图2 不同稀疏程度

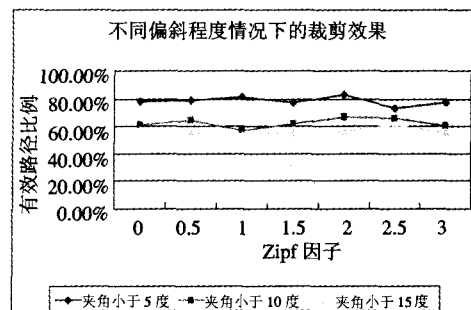


图3 不同偏斜程度情况

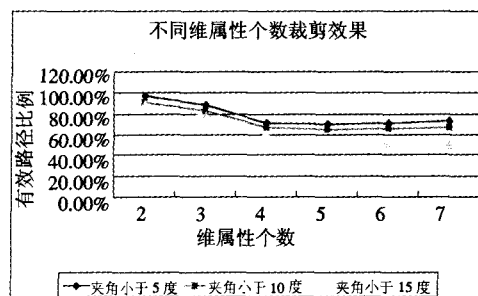


图4 不同维属性个数

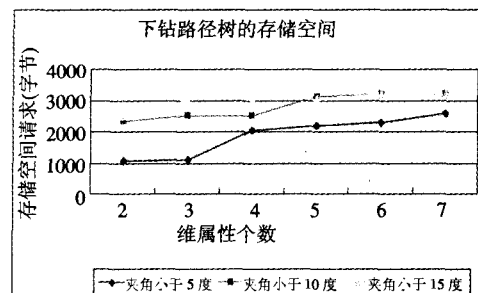


图5 下钻路径树的存储空间

参考文献

- 1 Sathe G, Sarawagi S. Intelligent Rollups in Multidimensional OLAP Data. VLDB', 2001. 531~540
- 2 Sarawagi S. Explaining differences in multidimensional aggregates. VLDB'99, 1999
- 3 Sarawagi S. User-adaptive exploration of multidimensional data. VLDB'00, 2000
- 4 Feng Y, Agrawal D, El Abbadi A, et al. Range CUBE: Efficient Cube Computation by Exploiting Data Correlation. ICDE 2004. 658~670
- 5 Lakshmanan L V S, Pei J, Han Jiawei. Quotient Cube: How to Summarize the Semantics of a Data Cube. VLDB, 2002. 778~789
- 6 Wang W, Feng J, Lu H, Yu J. Condensed Cube: An Effective Approach to Reducing Data Cube Size. ICDE, 2002
- 7 Cormode G, Korn F, Muthukrishnan S, et al. Diamond in the Rough: Finding Hierarchical Heavy Hitters in Multi-Dimensional Data. ACM SIGMOD, 2004
- 8 Shi Y, Song Y, Zhang A. A Stripping-Based Approach for Multi-Dimensional data Analysis. VLDB, 2003