doi:10.3772/j.issn.1002-0470.2013.10.006

高效支持多维网络 OLAP 的数据立方体模型 CI-DCG①

古晓艳②* ** *** 王伟平③*** 孟 丹*** 杨秀峰* 周 江*

(*中国科学院计算技术研究所计算应用研究中心 北京 100190)
(**中国科学院研究生院 北京 100049)
(***中国科学院信息工程研究所 北京 100093)

摘 要 针对现有联机分析处理(OLAP)方法的空间开销随着数据维度增加呈指数级增长,因而不适用于维度较高的多维网络应用的问题,提出了一种新的多维网络数据立方体模型——封闭冰山双立方图(CI-DCG)。该模型通过引入邻接立方体的概念,将其实例化过程转化为两个计算传统数据立方体的阶段,从而可将传统数据立方体生成算法中较为成熟的空间优化技术引入到多维网络中。在保证多维网络上 OLAP 查询处理效率的同时,将多维网络数据立方体生成算法的空间复杂度降为多项式级别。理论分析和实验结果均表明,该模型在空间开销和查询性能方面均优于已有的多维网络 OLAP 模型,并且数据维度越高,这种优势就越明显。

关键词 多维网络,图立方体,邻接立方体,联机分析处理(OLAP)

0 引 言

图作为一种表达能力很强的结构,被广泛用来对各种应用领域中结构型关系建模,如交通网络、社会网络、DNA 分析等。为方便从大量的图数据中提取出有效信息,需要将图汇总成易于理解的简明形式。近年来,图汇总(graph summarization)算法相继被提出[1-6],但这些算法都只根据图的拓扑结构进行汇总。在实际应用中,除了图的拓扑结构外,与顶点关联的多维属性也很重要。图的拓扑结构和与顶点关联的多维属性一起,形成了一种新结构——多维网络[7]。

联机分析处理(on-line analytical process, OLAP)是数据仓库和数据挖掘领域的核心技术之一。在多维网络上支持 OLAP 查询具有重要的应用价值。近来,研究人员在多维网络上进行 OLAP 分析方面开展了很多研究[7-11]。其中,图立方体[7](Graph Cube, GC)是该领域的一个典型的研究成果,它成功地将数据立方体[12]技术应用到多维网络分析上。和数据立方体相似,图立方体根据所有可能的维度组合,对顶点和边进行聚集,重组多维网络,得到所有可能的聚集图,为用户提供不同维度、不同粒度观察数据对象的视图,具有重要的应用价值。然而,图立方体模型的空间复杂度随着维度呈指数级增长,维度较高时会导致该模型空间开销过大,限制了这项技术的应用。虽然图立方体的一些研究工作为降低空间开销采用了部分实例化技术[13,14],但这些方法在降低空间开销的同时会增大查询响应时间,降低查询效率。此外,虽然在数据立方体中还有许多成熟的技术来减少实例化立方体带来的空间开销,如封闭立方体(closed cube)[15-18]技术,可以在满足查询时间效率的基础上对数据立方体进行无损压缩,但对于图立方体来说,计算封闭立方体需涉及到其中的多个聚集网络,而这些聚集网络是根据不同的聚集独立生成的,造成彼此间关系丢失,因而,封闭立方体技术并不能直接应用到图立方体上。本文提出了一个新的多维网络数据立方体模型——封闭冰山双立方图(closed iceberg double cubed graph, CI-DCG),该模型通过引入邻接立方体的概念,将其实例化过程转化成两个计算传统数据立方体的阶段,进而可直接应用传统数

① 863 计划(2011AA01A203,2012AA011002),国家自然科学基金(60903047)和中国科学院先导专项(XDA06030200)资助项目。
② 女,1987 年生,博士;研究方向:海量数据处理;E-mail:guxiaoyan@ncic.ac.cn
③ 通讯作者:E-mail:wangweiping@nelmail.iie.ac.cn
(收稿日期:2012-09-25)

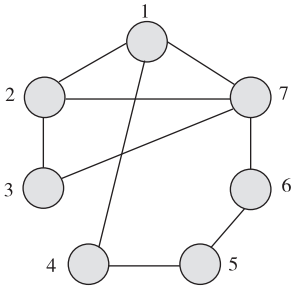
据立方体中较为成熟的空间优化技术,在保证多维网络 OLAP 查询效率的同时,将多维网络数据立方体生成算法的空间复杂度降至维度多项式级别。实验结果表明,同已有图立方体模型相比,该模型性能更优。

1 基本概念

关系 R 的模式表示为 $R(A_1,A_2,\cdots,A_n,M)$,其中 A_i 为维属性,$1\leqslant i\leqslant n$,M 为度量属性。R 上任意一种可能的聚集 $(A_1',A_2',\cdots,A_n')$ 对应一个方体(cuboid),其中 A_i' 等于 A_i 或者 $*$,$*$ 代表文献[12]提出的特殊值 ALL,是相应维度上的聚集。所有方体一起构成了由 R 产生的数据立方体。

定义 1:多维网络[7]是一个形式为 $N=(V,E,A)$ 的图,其中 V 是顶点的集合,$E\subseteq V\times V$ 是边的集合,$A=\{A_1,A_2,\cdots,A_n\}$ 是与顶点相关联的属性集合,对于 $\forall v\in V$,存在一个多维元组 $A(v)=(A_1(v),A_2(v),\cdots,A_n(v))$,其中 $A_i(v)$ 是顶点 v 上的第 i 个属性,即第 i 维,$1\leqslant i\leqslant n$。

为了更直观地理解多维网络的定义,下面给出一个社交网络中多维网络的示例。图 1 记录了社交网络中朋友喜欢玩的游戏的情况,图 1(a)表示社交网络图,图中有 7 个顶点,记作 $v_1,v_2,\cdots,v_7$,分别代表社交网络中不同的个体;9 条边,分别代表个体间的朋友关系。每个顶点均关联一个多维属性元组,记录该个体的基本信息,包括 ID、性别、爱玩的游戏和活跃度。所有顶点的多维属性元组的集合形成了多维属性表,如图 1(b)所示。



(a)社交网络图

ID	性别	游戏	活跃度
1	男	开心农场	中
2	女	开心农场	低
3	女	水果忍者	高
4	女	水果忍者	中
5	男	极品飞车	中
6	男	极品飞车	低
7	女	开心农场	低

(b)多维属性表

图 1 一个社交网络的多维网络图

定义 2:给定一个多维网络 $N=(V,E,A)$ 和 A 上一种可能的聚集 $A'=(A_1',A_2',\cdots,A_n')$,其中 A_i' 为 A_i 或者 $*$,在 A' 的作用下生成的 N 的聚集网络[7] G' 是一个权重图,表示形式为 $G'=(V',E',W_V',W_E')$,其中:

(1)对于 $\forall[v]$,$[v]=\{u\mid A_i'(u)=A_i'(v),u,v\in V,i=1,\cdots,n\}$,$\exists v'\in V'$ 代表 $[v]$。v' 的权重 $w(v')=F_v([v])$,其中 $F_v(\cdot)$ 为作用在顶点上的聚集函数。v' 称作聚集顶点。

(2)$\exists u',v'\in V'$,u' 代表 $[u]$,v' 代表 $[v]$,令 $E_{(u',v')}=\{(u,v)\mid u\in[u],v\in[v],(u,v)\in E\}$,若 $E_{(u',v')}$ 非空,则 $\exists e'\in E'$ 代表 $E_{(u',v')}$。边的权重 $w(e')=F_e(E_{(u',v')})$,其中 $F_e(\cdot)$ 为作用在边上的聚集函数。e' 称为聚集边。

仍以图 1 中的“社交网络”为背景,选取 A 的一种聚集 $A'=$(性别,$*$,$*$),以 COUNT($\cdot$)作为顶点和边上的聚集函数,则产生的聚集网络如图 2 所示。其中,灰色的顶点为聚集顶点,连接聚集顶点间的边为聚集边。此外,聚集网络中允许出现自环,且边的默认权重为 1。

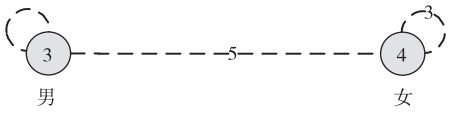


图 2 聚集网络示例

定义 3:给定多维网络 $N=(V,E,A)$,根据 A 的每种聚集 A',重组 N 产生一个聚集网络(如定义 2 所示),产生的所有聚集网络的集合构成了图立方体[7] GC。

定义 4:给定图立方体 GC 上两个不同的 cuboid S 和 T,crossboid 查询 $S\bowtie T$ 产生的结果是一个跨 cuboid 的聚集网络 $G_{\text{cross}}=(V_S'\cup V_T',E',W_V',W_E')$。$G_{\text{cross}}$ 中任意顶点 v 根据聚集 S 生成聚集顶点 $v_s'\in V_S'$,根据聚集 T 生成聚集顶点 $v_t'\in V_T'$,GC 中的边 $e(u,v)$ 则生成聚集边 $e(u_s',v_t')$ 和 $e(v_s',u_t')$,其中 u_s' 和 v_s' 是 V_S' 中的聚集顶点,v_t' 和 u_t' 是 V_T' 中的聚集顶点。W_V' 和 W_E' 的产生方法见定义 2。

在图 1 的例子中,给定 crossboid 查询语句“不同的性别和喜爱的游戏组合出的网络结构是什么?”,生成的查询结果如图 3 所示,它涉及到两个 *cuboid*,分别是(性别,$*$,$*$)和($*$,游戏,$*$)。

定义 5:给定数据立方体 C 和元组 $s\in C$,定义 dim(s)函数表示 s 中所有非 $*$ 的维度,$BV(s)$ 为 s 的基本元组集,表示在原始多维网络中 s 所覆盖的元组集合。如果不存在元组 $t\in C$,满足 $t\neq s$,$dim(s)\subseteq$

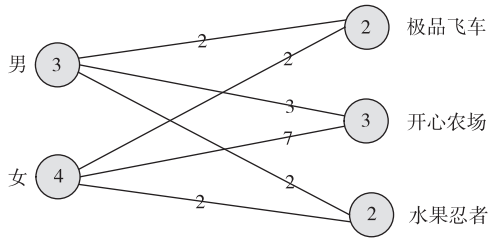


图 3　crossboid 查询产生的聚集网络示例

$dim(t)$, $BV(s) = BV(t)$, 则称元组 s 为封闭元组。如果 C 中所有元组均为封闭元组, 则称 C 为封闭立方体。

2　CI – DCG 模型

由上面的定义可以看出, Graph Cube 需要大量的存储空间, 空间开销随着维度以指数方式上升; 另外, Graph Cube 中的各个聚集网络间是独立的, 因而不能直接应用满足查询效率的无损压缩技术, 如封闭立方体等。基于这两点观察, 本文提出了一个新模型, 称之为 CI – DCG。该模型可以在多维网络上最大化 OLAP 查询效率的同时降低空间开销。

为了更清楚地阐述 CI – DCG 的实现过程, 本节首先给出一种基本模型(简称 DCG), 然后介绍其实现算法, 最后, 在该实现算法的基础上引入封闭立方体技术和冰山立方体(iceberg cube)[19,20] 技术对其优化, 即 CI – DCG 模型。

2.1　DCG 模型

定义 6: 对于一个多维网络 $N = (V, E, A)$, $A = \{A_1, A_2, \cdots, A_n\}$, 则 A 上有 2^n 种聚集, 对于每种聚集表示形式为: $A_k' = (A_{k1}', A_{k2}', \cdots, A_{kn}')$, $1 \leqslant k \leqslant 2^n$, A_{ki}' 等于 A_i 或者 $*$。生成的 DCG D 是一个权重图, 表示为 $D = (V', E', W_{V'}, W_{E'})$, 其中:

(1) 令 $[v]_k = \{u \mid A_{ki}'(u) = A_{ki}'(v), u, v \in V, i = 1, \cdots, n\}$, $\exists v' \in V'$ 代表 $[v]_k$。v' 的权重 $w(v') = F_v([v]_k)$, 其中 $F_v(\cdot)$ 是定义在顶点上的聚集函数。令 $[v] = \cup_{k=1}^{2^n} [v]_k$。

(2) $\forall u', v' \in V'$, u' 代表 $[u]_p$, v' 代表 $[v]_q$, 其中 $1 \leqslant p, q \leqslant 2^n$, $E(u', v') = \{(u, v) \mid u \in [u]_p, v \in [v]_q, (u, v) \in E\}$, 若 $E(u', v')$ 不为空, 则 $\exists e' \in E'$ 代表 $E(u', v')$。边的权重 $w(e') = F_e(E(u', v'))$, 其中 $F_e(\cdot)$ 是定义在边上的聚集函数。

由定义 6 可以看出, 在 DCG 上, 可以方便地支持 cuboid 查询和 crossboid 查询, 且任一查询的结果都对应着 DCG 中的一个子图。仍以图 1 中多维网络为例, 生成的 DCG 的一个子图如图 4 所示, 该图中左边虚线连接的子图代表 cuboid 查询“根据性别聚集出的网络结构”, 右边虚线连接的子图代表 cuboid 查询“根据游戏聚集出的网络结构”, 此外, 实线相连的子图则是 crossboid 查询“不同的性别和喜爱的游戏聚集产生的网络结构”的结果。且由于 DCG 实例化了整个图, 所以查询效率高。

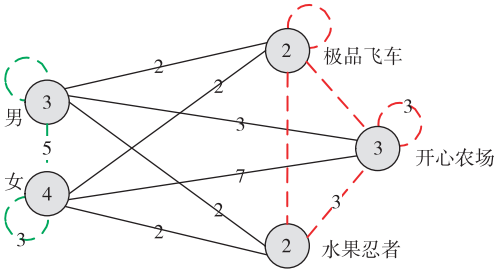


图 4　DCG 中的一个子图

2.2　实现算法

由于 DCG 完全实例化, 在维度增大时, 空间开销会急剧膨胀, 为了减少空间开销, 提出了一种优化的实现方法。本小节介绍给定一个多维网络 N, 如何高效地实现 DCGD。

在 DCG 中, 我们采用改进的邻接链表的方法来表示图。传统的邻接链表法对每个顶点建立一个链表来存放该顶点的相邻结点。本文对链表中结点的组织方式进行改进, 引入了一种新结构——邻接立方体(见定义 7)。

定义 7: 对于 DCG 中的任意顶点 v', 令 $BV(v')$ 表示被 v' 所代表的原始多维网络顶点集合, $N(v)$ 表示原始多维网络中顶点 v 的邻居顶点集合, 定义 $BN(v') = \cup_{v \in BV(v')} N(v)$, 表示 v' 所代表的原始多维网络中的顶点的邻居顶点集合。v' 的邻接立方体则是在 $BN(v')$ 上构建的数据立方体, 表示形式为 $AdjCube(v')$。

在本文的社交网络例子中, 顶点 $v' = (女, 水果忍者, *) \in DCG$, $BV(v') = \{v_3, v_4\}$, $BN(v') = \{v_1, v_2, v_5, v_7\}$, 根据 $BN(v')$ 构建数据立方体, 即可得到 v' 的邻接立方体 $AdjCube(v')$, 如表 1 所示。

之所以采用邻接立方体来替代邻接链表, 有三个原因: 首先, 相比于邻接链表, 邻接立方体更便于支持 *cuboid* 查询和 *crossboid* 查询; 其次, 由定义可知, 邻接立方体的生成很方便, 且有很多成熟的数据立方体生成算法; 再者, 通过采用邻接立方体, 将

DCG 模型的构建拆分成两个计算数据立方体的阶段,从而可方便地应用数据立方体中很多成熟的技术,来减少 DCG 的时间和空间开销。

表 1　DCG 顶点(女,水果忍者,*)的邻接立方体

性别	游戏	活跃度	COUNT
男	开心农场	中	1
男	极品飞车	中	1
女	开心农场	低	2
ALL	开心农场	中	1
ALL	极品飞车	中	1
ALL	开心农场	低	2
男	ALL	中	2
女	ALL	低	2
男	开心农场	ALL	1
男	极品飞车	ALL	1
女	开心农场	ALL	2
ALL	ALL	中	2
ALL	ALL	低	2
ALL	开心农场	ALL	3
ALL	极品飞车	ALL	1
男	ALL	ALL	2
女	ALL	ALL	2
ALL	ALL	ALL	4

算法 1 中介绍了实现 DCG 的基本算法,由算法 1 可以看出,计算 DCG 的过程通过借助邻接立方体的概念,分解成了两个计算数据立方体的阶段:(1)对多维网络 G 的顶点集合 V 计算数据立方体得到 DCG 中所有顶点的集合 V'(行 1 – 9);(2)对于 V'中每个顶点 v,计算其邻接立方体(行 10 – 16)。这两个阶段都是实现传统数据立方体的过程,也是 DCG(double cubed graph)名字的由来。

具体实现过程如下:首先,创建一个 hash 结构,根据聚集 A',产生元组到聚集顶点的映射(行 2),对于 G 中的每个顶点 v,如果对应的聚集顶点 $v' \in V'$从未被创建过,则创建 v',并将 v 添加到 $BV(v')$集合中(行 3 – 7),否则,只需更新 v 对应的聚集顶点的 BV 集合(行 8 – 9)。然后,对于 V'中的每个顶点 v' 计算其权重,获取 $BV(v')$中的每个顶点,找到其所有的相邻顶点,对每个相邻顶点计算数据立方体并根据边的聚集函数得到每个聚集边的权重(行 10 – 16)。

算法 1　DCG 实现算法

算法 DCG Materialize Algorithm

输入:一个多维网络 $N=(V,E,A)$,顶点的聚集函数 $F_v(\cdot)$ 和边的聚集函数 $F_e(\cdot)$

输出:DCG D

```
//阶段 1
1   for 任意一种聚集 A'
2   初始化一个 hash 结构 ζ:A'→V'
3   for v∈V do
4   if ζ(A'(v)) = NULL then
5   根据 A'(v)创建聚集节点 v'∈V'
6   ζ(A'(v))←v'
7   BV(v')←{v}
8   else
9           BV(ζ(A'(v)))←BV(ζ(A' (v)))∪{v}
//阶段 2
10  for v'∈V'
11  W_v' = F_v(BV(v'))
12     for p∈BV(v') do
13        for each (p,q)∈E do
14  BN(v')←BN(v')∪{q}
15     if(BN(v')! = NULL)
16  AdjCube(v') = Cube(BN(v'),F_e)
```

为了有效地减少 DCG 的空间开销,需要对其优化。由于 DCG 的实现过程转化成了两个计算传统数据立方体的阶段,因而可以很方便地应用数据立方体中较为成熟的减少空间开销的技术。

2.3　CI – DCG 模型

封闭立方体是将聚集单元分成封闭单元和非封闭单元,只考虑封闭单元的计算和输出,其他单元的度量值可以由相应的封闭单元得到。冰山立方体则通过过滤所有不满足用户定义的最小支持度的单元,尽可能避免不必要的聚集操作,来减少立方体的大小,从而缩小其占用空间。这两种技术的结合即封闭冰山立方体。通过计算满足冰山约束条件的封闭单元,进一步减少输出结果。

由算法 1 可知,DCG 的实现简化成了计算两个传统数据立方体的过程,所以封闭数据立方体算法和冰山数据立方体算法均可以直接应用到 DCG 上。在计算出的封闭邻接立方体的基础上过滤掉边的权重不满足约束条件的项,进而得到 CI – DCG(封闭冰山 DCG)。CI – DCG 的生成流程如下:首先,在算

法 1 的第一部分计算封闭冰山 DCG 的顶点集,具体操作为过滤掉所有非封闭顶点,得到封闭顶点的集合,在此基础上再过滤掉所有权重不满足约束条件的顶点(本文选取 C – Cubing(MM)算法[16],它是一个结合了冰山立方体算法 MM – Cubing[19],高效计算封闭冰山立方体的算法),从而得到封闭冰山顶点集合;然后在算法的第二部分对得到的所有封闭冰山顶点计算其封闭冰山邻接立方体(本文选取 C – Cubing(Star – Array)算法[16])。在本文的例子中,顶点(女,水果忍者,*)是步骤 1 中得到的封闭顶点,它的邻接立方体如表 1 所示,共有 18 个单元,其中封闭单元有 6 个,由表中的灰色单元表示,然后使用冰山立方体技术,在得到的 6 个封闭邻接立方体单元中过滤掉不满足约束条件(COUNT≥2)的单元,得到顶点(女,水果忍者,*)的封闭冰山邻接立方体,如表 1 中的深灰色单元所示。

通过引入封闭冰山立方体技术可大大降低 DCG 的空间开销,并且可以论证 CI – DCG 模型的空间开销随维度增大呈多项式增长。由于冰山立方体的空间复杂度和用户设定的最小支持度密切相关,具有不确定性,所以这里只考虑封闭立方体带来的影响。高维封闭立方体的空间复杂度为

$$O(T \cdot d^{\log_C T+1}/(\log_C T)!) \tag{1}$$

其中 T 表示表的记录数,C 表示每个维的基数(这里考虑各维的基数相等的简化情况),d 表示维度。由于实际应用中 $\log_C T$ 的取值通常较小,所以封闭立方体的空间大小近似为 d 的低阶多项式[16,21]。由于计算 CI – DCG 的顶点集和每个顶点的邻接立方体时均用到了封闭立方体技术,对应于式(1),算法的阶段 1 中 $T=|V|$;算法的阶段 2 中对于每个聚集点 v,$T=|\mathrm{BN}(v)|\leqslant|E|$,所以 CI – DCG 的空间复杂度为两个多项式相乘即为

$$O\left(\left(|V|\frac{d^{\log_C|V|+1}}{(\log_C|V|)!}\right)\cdot\left(|E|\frac{d^{\log_C|E|+1}}{(\log_C|E|)!}\right)\right)$$
$$=O\left(|V|\cdot|E|\cdot\frac{d^{\log_C|V|+\log_C|E|+2}}{(\log_C|V|)!\cdot(\log_C|E|)!}\right) \tag{2}$$

显然也近似为 d 的多项式。即 CI – DCG 的空间复杂度为维度的多项式。

3 实验结果与分析

实验从实例化模型的空间开销、实例化模型的时间开销、cuboid 查询响应时间及 crossboid 查询响应时间四个方面对模型的性能进行评价。将本文提出的模型 CI – DCG 与文献[7]提出的 Graph Cube 模型进行了比较,并对得到的实验结果进行了分析。

本文实验用的多维网络有 10^5 个顶点和 10^6 个边。在所有的实验中,聚集函数均为 COUNT(·),基数(cardinality)取 500,计算结果存放在文本文件中。为了测试结果的准确性,每种操作重复 5 次,取平均值作为最终结果。

Graph Cube 的实现方法有三种:完全实例化、部分实例化和不实例化[7]。本文选取了三种典型的情况进行实验:部分(50%)实例化 cuboid 查询且不实例化 crossboid 查询(下文标记为 PN – GC);完全实例化 cuboid 查询且不实例化 crossboid 查询(标记为 FN – GC);完全实例化 cuboid 查询和 crossboid 查询(标记为 FF – GC)。

3.1 实例化模型空间开销对比实验

实例化不同模型的空间开销实验结果如图 5 所示,横坐标轴代表维度,纵坐标轴代表模型实例化的空间开销,单位为 MB,采用了底为 10 的对数刻度。可以看到,Graph Cube 的三种实现方法的空间开销均随着维度的增大呈指数增长。其中,完全实例化 cuboid 和 crossboid 查询的情况空间开销最大,且随

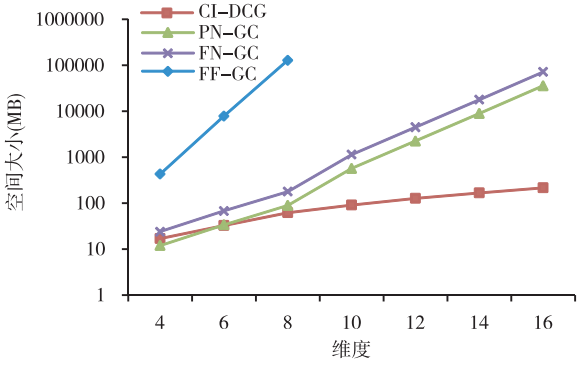


图 5 实例化模型的空间开销

维度增大而增长的速度最快,维度较高时,该模型的可用性会受到空间大小的限制;实例化的比例越小,空间开销越少。在维度较低时,CI – DCG 的空间开销和部分实例化 cuboid 查询的 Graph Cube 相近,这是因为在维度低的时候产生的非封闭单元比例小,封闭立方体技术效果不明显;当维度大于 6 时,CI – DCG 的空间开销明显低于 Graph Cube,且随着维度增大,空间优势越来越明显。

3.2 实例化模型时间开销对比实验

实例化模型的时间开销实验结果如图 6 所示。图中横坐标代表多维网络的维度,纵坐标代表实例化多维网络数据立方体的所用时间,单位为秒,也采用了底为 10 的对数刻度。实验结果表明,随着维度的增加,实例化各模型所需的时间也在增加,且 Graph Cube 的时间开销增长速度比 CI – DCG 快很多。在维度较低时,CI – DCG 比不实例化 crossboid 查询的 Graph Cube 的时间开销略大,维度大于 8 后,CI – DCG 的时间开销比 Graph Cube 的三种实现情况均低,且随着维度增大,前者的优势越明显。

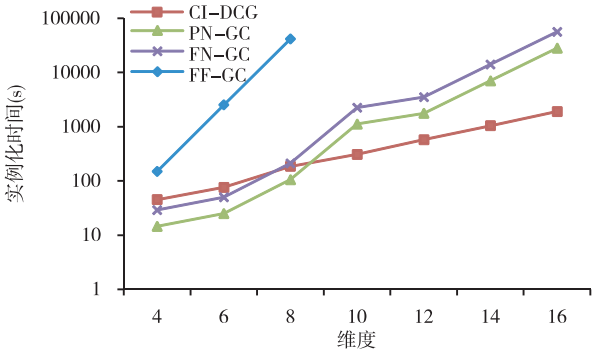


图 6 实例化模型的时间开销

3.3 Cuboid 查询响应时间对比实验

由于 Graph Cube 的三种实现方法中实例化比例不同,进而会影响到查询性能。为了更直观地对比 CI – DCG 模型和 Graph Cube 模型的查询性能及空间开销的关系,在本实验和下一个实验中,采取统一两模型空间开销的实验方法。具体操作方法为:在任意维度上,首先选取两个模型中完全实例化时空间开销较小的那个作为基准,将另一个模型部分实例化,以保证两模型实例化后占用的空间大小相等,然后再分别进行 OLAP 查询。

本小节对 Cuboid 查询进行实验,为保证实验结果的准确性,选取了 10 种不同的 cuboid 查询语句进行实验,计算这 10 种查询的平均响应时间作为最终实验结果。由图 7 可以看出,在空间开销相同的情况下,CI – DCG 比 Graph Cube 上的 cuboid 查询响应时间短,效率高。且随着维度的增大,CI – DCG 的查询响应时间变化不大,而 Graph Cube 的查询响应时间则迅速增加,这是因为随着维度的增加,Graph Cube 的空间开销增加比较快,为了保持和 CI – DCG的空间开销相等,Graph Cube 实例化的比例不断减少,导致其查询响应时间不断增加。本实验验证了 CI – DCG 上 cuboid 查询的高效性。

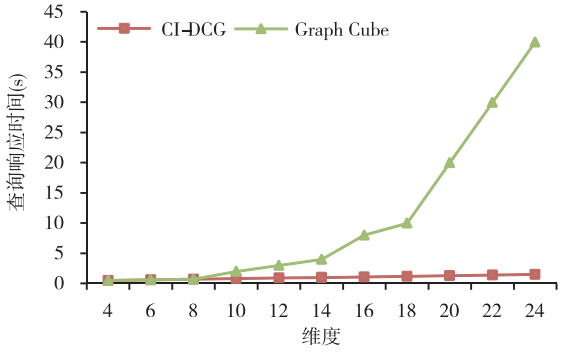


图 7 Cuboid 查询响应时间

3.4 Crossboid 查询响应时间对比实验

本小节对 CI – DCG 和 Graph Cube 的 crossboid 查询响应时间进行对比,实验选取了 10 种不同 crossboid 查询,并取所有查询响应时间的平均值作为输出结果。实验结果如图 8 所示,可以看出,在保证两模型实例化后占用空间大小相等的情况下,在 CI – DCG 上进行 crossboid 查询响应时间比在 Graph Cube 上低,而且维度越高,优势越明显。本实验结果验证了基于 CI – DCG 处理 crossboid 查询的高效性。

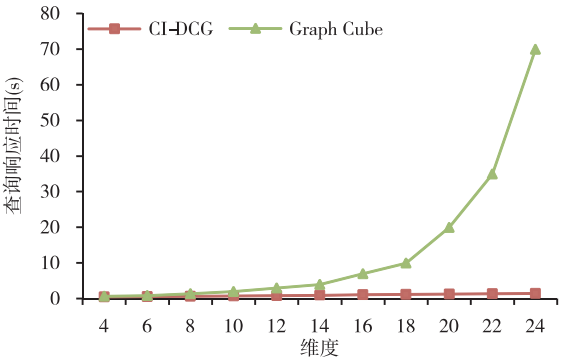


图 8 Crossboid 查询响应时间

4 结 论

针对多维网络数据立方体生成算法的空间开销随着数据维度增大呈指数级增长的问题,提出了一种新的多维网络数据立方体模型 CI – DCG。该模型首先定义了邻接立方体,将 CI – DCG 的实例化拆分成两个计算数据立方体的阶段,从而引入了传统数据立方体领域中成熟的降低空间开销技术,使得

空间开销降为维度的多项式级别。实验结果表明，在大规模多维网络上，实例化 CI－DCG 的时间和空间开销远远小于实例化 Graph Cube 的时间和空间开销。多维网络维度越高，CI－DCG 的空间优势就越明显。同时，在使用相同存储空间的情况下，基于 CI－DCG 处理 OLAP 查询的效率也要比基于 Graph Cube 处理 OLAP 查询的效率高。在未来的工作中，我们将研究如何基于 MapReduce 并行编程模型来实现 CI-DCG，以进一步提高 CI-DCG 模型实例化的效率。

参考文献

[1] Shi J, Malik J. Normalized cuts and image segmentation. *IEEE Trans on Pattern Analysis and Machine Intelligence*, 2000, 22(8): 888-905

[2] Newman M E J, Girvan M. Finding and evaluating community structure in networks. *Physical Review E*, 2004, 69(2): 026113

[3] Gibson D, Kumar R, Tomkins A. Discovering large dense subgraphs in massive graphs. In: Proceedings of the 31st International Conference on Very Large Data Bases, Trondheim, Norway, 2005. 721-732

[4] Xu X, Yuru k N, Feng Z, et al. Scan: a structural clustering algorithm for networks. In: Proceedings of the 13th ACM Special Interest Group on Knowledge Discovery and Data Mining, San Jose, USA, 2007. 824-833

[5] Navlakha S, Rastogi R, Shrivastava N. Graph summarization with bounded error. In: Proceedings of the 2008 ACM Special Interest Group on Management of Data, Vancouver, Canada, 2008. 419-432

[6] LeFevre K, Terzi E. GraSS: Graph structure summarization. In: Proceedings of 2010 International Conference on Society for Industrial and Applied Mathematics, Ohio, USA, 2010. 454-465

[7] Zhao P, Li X, Xin D, et al. Graph cube: on warehousing and olap multidimensional networks. In: Proceedings of the 2011 ACM Special Interest Group on Management of Data, Athens, Greece, 2011. 853-864

[8] Tian Y, Hankins R A, Patel J M. Efficient aggregation for graph summarization. In: Proceedings of the 2008 ACM Special Interest Group on Management of Data, Vancouver, BC, Canada, 2008. 567-580

[9] Zhang N, Tian Y, Patel J M. Discovery-driven graph summarization. In: Proceedings of IEEE 26th International Conference on Data Engineering, Long Beach, USA, 2010. 880-891

[10] Zhou Y, Cheng H, Yu J X. Graph clustering based on structural/attribute similarities. In: Proceedings of the Very Large Data Base Endowment, Lyon, France, 2009. 718-729

[11] Chen C, Yan X, Zhu F, et al. Graph olap: towards online analytical processing on graphs. In: Proceedings of the 8th IEEE International Conference on Data Mining, Pisa, Italy, 2008. 103-112

[12] Gray J, Chaudhuri S, Bosworth A, et al. Data cube: A relational aggregation operator generalizing group-by, cross-tab, and sub-totals. *Data Mining and Knowledge Discovery*, 1997, 1(1): 29-53

[13] Baralis E, Paraboschi S, Teniente E. Materialized view selection in a multidimensional database. In: Proceedings of the 23th International Conference on Very Large Data Bases, Athens, Greece, 1997. 156-165

[14] Li X L, Han J W, Gonzalez H. High-dimensional OLAP: aminimal cubing approach. In: Proceedings of the 30th International Conference on Very Large Data Bases, Toronto, Canada, 2004. 528-539

[15] Sismanis Y, Deligiannakis A, Roussopoulos N, et al. Dwarf: shrinking the petacube. In: Proceedings of the 2002 ACM Special Interest Group on Management of Data, Wisconsin, USA, 2002. 464-475

[16] Xin D, Shao Z, Han J W, et al. C-cubing: efficient computation of closed cubes by aggregation-based checking. In: Proceedings of the 22nd International Conference on Data Engineering, Rio de Janeiro, Brazil, 2006. 4-4

[17] Lakshmanan L, Pei J, Zhao Y. Qc-trees: an efficient summary structure for semantic olap. In: Proceedings of the 2003 ACM Special Interest Group on Management of Data, San Diego, USA, 2003. 64-75

[18] 李盛恩，王珊. 封闭数据立方体技术研究. 软件学报，2004，15(8)：1165-1171

[19] Beyer K, Ramakrishnan R. Bottom-up computation of sparse and iceberg cube. In: Proceedings of the 1999 ACM Special Interest Group on Management of Data, Philadelphia, USA, 1999. 359-370

[20] Shao Z, Han J W, Dong X. MM-Cubing: computing iceberg cubes by factorizing the lattice space. In: Proceedings of the 16th International Conference on Scientific and Statistical Database Management, Santorini Island, Greece, 2004. 213-222

[21] Sismanis Y, Roussopoulos N. The polynomial complexity of fully materialized coalesced cubes. In: Proceedings of the 30thInternational Conference on Very Large Data Bases, Toronto, Canada, 2004. 540-551

CI-DCG:an efficient data cube model for supporting OLAP on multidimensional networks

Gu Xiaoyan[*] [**] [***], Wang Weiping[***], Meng Dan[***], Yang Xiufeng[*], Zhou Jiang[*]

([*] Laboratory of Computing Applications, Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100190)

([**] Graduate University of Chinese Academy of Sciences, Beijing 100049)

([***] Institute of Information Engineering, Chinese Academy of Sciences, Beijing 100093)

Abstract

Considering that it is valuable to support efficient on-line analytical process(OLAP) query on multidimensional networks and the space overhead of the existing OLAP methods grows exponentially with the increase of the data dimensionality, which limits their use in multidimensional networks with high dimensionality, the closed iceberg double cubed graph(CI-DCG), a novel data cube model is proposed. By introducing the concept of adjacent cube, the model splits the process of materializing into two phases of data cube computing, which facilitates combining special characteristics of multidimensional networks with the existing well-studied data cube techniques, to gain high query performance with polynomialspace complexity. Both theoretical analysis and experimental results demonstrate the efficiency and effectiveness of the CI-DCG, especially in case of high dimensionality.

Key words: multidimensional network, graph cube, adjacency cube, on-line analytical process(OLAP)