软件学报 ISSN 1000-9825, CODEN RUXUEW
Journal of Software,2011,22(2):258−268 [doi: 10.3724/SP.J.1001.2011.03771]

E-mail: jos@iscas.ac.cn
http://www.jos.org.cn
Tel/Fax: +86-10-62562563

Graph OLAPing 的建模、设计与实现*

李 川[1], 赵 磊[2], 唐常杰[1+], 陈 瑜[1], 李 靓[3], 赵小明[1], 刘小玲[1]

[1](四川大学 计算机学院,四川 成都 610065)

[2](中国科学技术大学 计算机科学与技术学院,安徽 合肥 230027)

[3](北京大学 信息科学技术学院,北京 100871)

Modeling, Design and Implementation of Graph OLAPing

LI Chuan[1], ZHAO Lei[2], TANG Chang-Jie[1+], CHEN Yu[1], LI Jing[3], ZHAO Xiao-Ming[1], LIU Xiao-Ling[1]

[1](College of Computer Science, Sichuan University, Chengdu 610065, China)

[2](School of Computer Science and Technology, University of Science and Technology of China, Hefei 230027, China)

[3](School of Electronics Engineering and Computer Science, Peking University, Beijing 100871, China)

+ Corresponding author: E-mail: cjtang@scu.edu.cn, http://cs.scu.edu.cn/~tangchangjie

Li C, Zhao L, Tang CJ, Chen Y, Li J, Zhao XM, Liu XL. Modeling, design and implementation of graph OLAPing. *Journal of Software*, 2011,22(2):258−268. http://www.jos.org.cn/1000-9825/3771.htm

Abstract: This paper presents a series of models and algorithms to implement OLAPing on graph data. The major contributions include (1) proposing a graph-oriented data warehouse model, called a double star model, (2) proposing the concept of graph data cube and its building algorithm, (3) designing an informational OLAPing algorithm, I-OLAPing, (4) designing topological dimensional OLAPing algorithm, T-OLAPing, and (5) building a Graph OLAPing prototype, Graph OLAPer1.0, based on the proposed approaches. Experimental results show that the Graph OLAPing algorithms designed and implemented in this paper, together with Graph OLAPing prototype, Graph OLAPer1.0 can work effectively on Co-Author Networks.

Key words: graph OLAP; graph warehouse; graph cube

摘 要: 提出了一系列 Graph 的 OLAP 模型和算法,实现了以 Graph 数据为中心度量的 OLAP 操作.主要贡献包括:(1) 提出了面向 Graph 的数据仓库概念模型——双星模型;(2) 提出了 Graph 的数据立方概念和创建过程;(3) 设计了信息维聚集算法 I-OLAPing;(4) 设计了拓扑维聚集算法 T-OLAPing;(5) 实现了 Graph OLAP 的原型系统 Graph OLAPer1.0.实验结果表明,设计和实现的 Graph OLAPing 算法及原型系统 Graph OLAPer1.0 能够有效地进行科研合作网分析.

关键词: 图在线分析处理;图数据仓库;图数据立方

中图法分类号: TP311 **文献标识码**: A

OLAP(on-line analytical process)能够提供用户从不同维度、不同粒度观察数据对象的视图,是数据仓库和数据挖掘领域的核心技术之一,近年来得到广泛的研究[1−6].文献[7,8]研究数据立方体计算有效性.在传统数据

* 基金项目: 国家自然科学基金(600773169); 国家科技支撑计划(2006BAI05A01); 高等学校博士学科点基金(20090181120064)
收稿时间: 2009-07-29; 定稿时间: 2009-11-04

立方中,一个数据记录是与一组维值相关的,不同数据记录是相互独立的.多元记录可通过一致定义的聚集函数如 COUNT,SUM 以及 AVERAGE 进行概要数据的描述.若一个概念层次能够与每个属性产生联系,即可通过上卷、下钻、切片、切块操作在不同的维度和概念层次上进行概要描述.传统 OLAP 的研究主要针对传统关系表数据,进行以分布式变量为度量的聚集值计算,如销量的总和、均值、计数等,不能有效地处理复杂主题数据.

近年来,越来越多的数据源超出传统电子表格模式,形式多种多样,如化学、生物信息学中化学组合物和蛋白质网络,模式识别中待处理的 2D/3D 物体,电脑辅助设计的电路,半结构化数据 XML,社会网络,信息网络等.上述应用中,已积累大量图数据,急需从不同角度和不同粒度进行分析.分析考察的度量不只是单一的实体,而且更侧重实体间的相互联系.图有很强的表达能力,日益广泛地应用于富含复杂结构的数据建模,适合于主题对象具有复杂结构时的情形,如交通网络、社会网络、犯罪网络等[9−12].

有关图数据的分析和挖掘近年来得到了深入探索.文献[13]提出了分析大规模图的有效的信息聚集的方法.在图的 OLAP 中,聚集图被认为是基于特定角度和粒度的潜在网络的概括,近年来的研究大多集中于在图形密集、抽象、复杂度等方面.文献[14,15]研究如何浓缩大图,如 Web 页间结构等.然而,这些工作仅讨论如何有效存储计算网页链接信息,如 PAGERANK 等,却未给出任何图形结构.文献[4]提出利用分析简单图的出入度分布和跳跃点进行统计概括,其在多维分析方面的实用性尚待改进.文献[16−18]研究通过图的拓扑信息来聚集大规模的网络.图的聚集、稠密子图检测和图的可视化研究可参见文献[19−21].文献[22]提出基于 Graph 进行 OLAP 的设想和概念,但未考虑 OLAP 算法的设计,未提出支持 Graph 在线分析处理新的数据仓库、数据立方技术,未深入考虑有效性以及性能问题.本研究的主要任务是设计 Graph OLAPing 模型、算法并实现原型,用 I-OLAPing 和 T-OLAPing 等方法获得任意组合维度及维内取值约束的聚集图,反映复杂结构主题数据的本质.

例 1:科研合作者网络(Co-Author Network)记录计算机领域科研人员合作发表论文的情况.图中每个点表示一个作者,若两人合作发表过文章,则两点间存在一条边.他们在特定时间、特定会议发表文章篇数记为 w,每个作者总合作的人数记为 d.合作关系如图 1 所示.假定用户想根据不同的时间、会议、背景来观察研究者的合作关系,由于传统数据立方的单元格仅存储分布式度量聚集值,现有技术难以有效地解决.

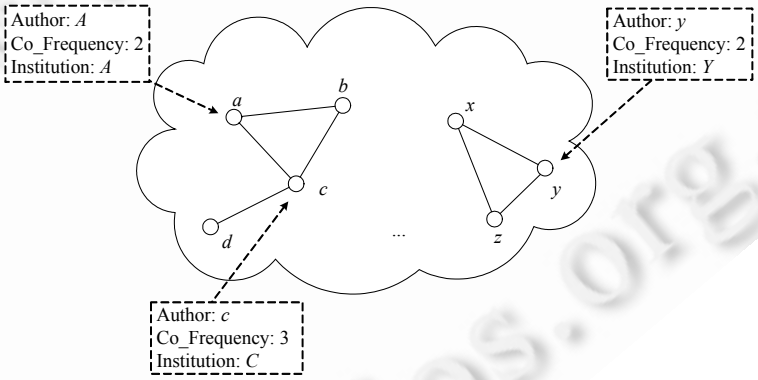

Fig.1 Co-Author Network

图 1 合作者网络

1 问题描述

为了准确地描述 Graph OLAP 的相关概念,仍以例 1 中的“科研合作者网络”为背景进行说明.假定在 Co-Author Networks 中有 3 个不同的维度:Time,Conference 和 Background.3 个维度分别具有如下的概念层次:

(1) Time: year(年份)→decade(年代)→all(所有)

(2) Conference: name(名称)→area(会议类型)→all(所有)

(3) Background: individual(个人)→institution(机构)→all(所有)

Graph OLAP 的度量是上述查询得到的 Co-Author Network.不同于传统 OLAP 框架,这里,用户关注的中心由一个度量值提升为一个图或网络,因为后者能够再现同一对象不同要素之间的复杂联系.Graph OLAP 应根据用户提交的维度和概念粒度给出 Co-Author Networks 的查询结果.例如:

$Q1$: 1980 年,SIGMOD 会议上所有合作者间的关系.

$Q2$: 在 20 世纪 80 年代,SIGMOD 会议上所有合作者间的关系.

$Q3$: 在 20 世纪 80 年代,所有 DB 会议上合作者间的关系.

$Q4$: 1980 年,SIGMOD 会议上所有合作单位间的关系.

如图 2 所示,Graph OLAP 应当支持两种查询类型,类似于 $Q1$~$Q3$ 的查询及类似 $Q4$ 的查询.$Q1$~$Q3$ 通过对合作者网络子图的选择、叠加与聚集计算得到.在此过程中,结果图边以及边的权值和相应的信息将发生变化,但图的拓扑结构不变.其计算方式与传统 OLAP 的技术路线相仿.另一种查询类似于 $Q4$,查询结果是一个新图,需依据点的信息重构图形的拓扑结构,包括新的图节点和新的边.这时,结果图的网络结构将发生较大变化.本文将这两种查询分别定义为基于信息维(informational dimensions,简称 ID)**的查询以及基于拓扑维(topological dimensions,简称 TD)的查询.

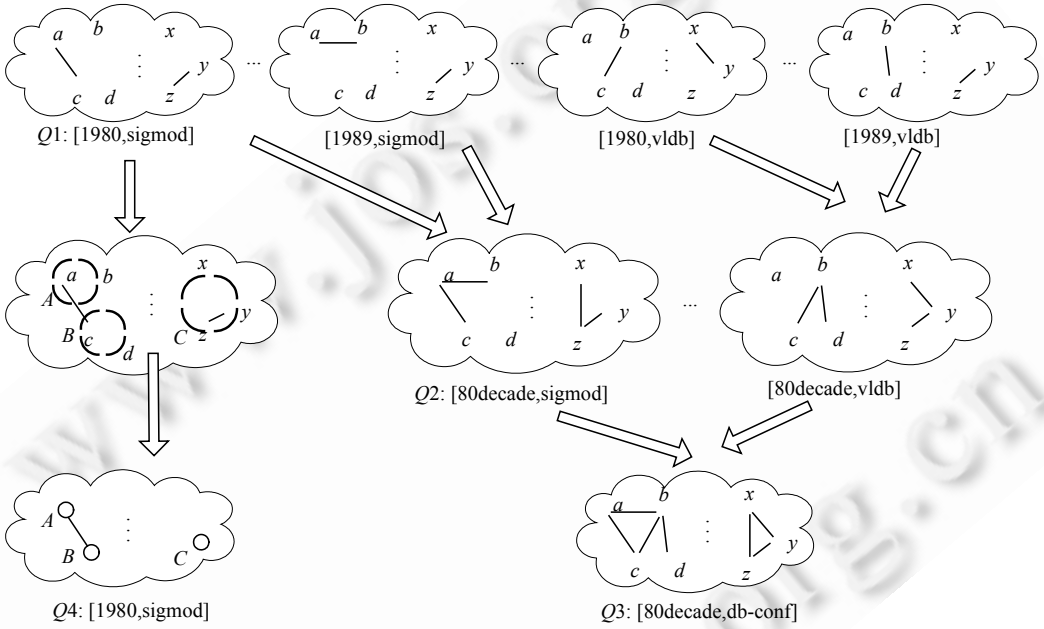

Fig.2 Queries of Co-Author Networks: $Q1$~$Q4$

图 2 基于 Co-Author Networks 的查询:$Q1$~$Q4$

定义 1(信息维). 设图数据库中待分析图的结构为 $G(V,E)=G(V,\theta(ID))$.其中,V 是图中点的集合,E 表示边的集合,函数 θ 为图 G 的边信息决定函数.设变量 $ID=\{I_1,I_2,\ldots,I_m\}$ 是 Graph OLAP 中待考察的维度集合,其中,$i=1,2,\ldots,m$.这 m 个信息属性构成的维度集合只能决定图的边集,不能改变图的拓扑结构,称 ID 为信息维集合.对于每个信息维度 I_i,存在一个概念层次集 $H_i=\{h_1,h_2,\ldots,h_m\}$.信息维及其概念层次的配置决定中心度量 Graph 的覆盖范围和内容.

定义 2(拓扑维). 设变量 $TD=\{T_1,T_2,\ldots,T_n\}$ 是刻画 Graph OLAP 中心度量拓扑结构的一个集合.一个图可表

** 本研究对文献[22]提出的信息维和拓扑维进行深度和广度扩展,引入边信息决定函数、点拓扑决定函数和边信息决定函数等,对传统信息维和拓扑维基于简单数据统计的汇总方式进行抽象,为基于同构子图集的存储压缩和查询性能的提升提供了空间.

示为 $G(V,E)=G(\varphi(TD),\delta(TD))$,其中,函数$\varphi$为点拓扑决定函数,函数$\delta$为边拓扑决定函数.这 n 个拓扑属性构成的拓扑维决定图的点集合和边集合,从而决定了图的拓扑结构,称 TD 为拓扑维集合.对其中每个拓扑维度 T_i,存在一概念层次集 $L_i=\{l_1,l_2,\ldots,l_m\}$.各拓扑维及概念层次的配置决定中心度量 Graph 的拓扑形态.

定义 3(Graph OLAP). Graph OLAP(G,TD,ID)是一个三元组,其中,$G=\{G_1,G_2,\ldots,G_k\}$是 $G_i(V_i,E_i)$子图的一个集合.根据定义 1、定义 2,$G_i(V_i,E_i)$记为 $G_i(\varphi(TD_i),\delta(TD_i),\theta(ID_i))$.对信息维进行 OLAP 操作,称为信息维 OLAP 操作,简称 I-OLAP.对拓扑维进行的 OLAP 操作称为拓扑维 OLAP 操作,简称 T-OLAP.

定义 1~定义 3 引入点、边的信息和拓扑决定函数$\theta(ID),\varphi(TD_i),\delta(TD_i),\theta(ID_i)$等扩展了传统图结构的表义范围.综合上述定义,面向 Graph 主题数据 OLAP 涉及的上卷、下钻、切片、切块、数据透视等操作均可转化为对信息维的图聚集计算和基于拓扑维的图聚集计算.因此,首先考虑图数据仓库概念建模问题,以期高效地实现图相关数据的计算和提取.

2 Graph OLAP 的数据仓库概念模型——双星模型(binary-star schema)

数据仓库建模的主要工作是选择和确定事实表、维表的结构.由于 Graph OLAP 的主题数据是 Graph 而非传统 OLAP 中的分布式度量值,而且 Graph OLAP 不仅涉及信息维,更涉及拓扑维,因此需要设计新的数据仓库实现方案.根据定义 3,Graph OLAP(G,TD,ID)是一个三元组.与传统的 OLAP 建模类似,Graph OLAP 的建模也有事实表和维表.Graph OLAP 的事实表记录三元组中的 G,即 Graph 数据集,而维表则包含 Graph OLAP 中的信息维表(informational dimensions table,简称 IDT)和拓扑维表(topological dimensions table,简称 TDT).为在 RDBMS 中高效组织图数据,把 Graph 数据分别存储在节点事实表(node fact table,简称 NFT)和边事实表(edge fact table,简称 EFT)中,NFT 与 EFT 通过外键进行连接.

Graph 的拓扑表示需要借助 NFT 和 EFT.但注意到:(1) 当 Graph OLAP 操作涉及拓扑维时,需要调整图形的拓扑结构,此时 NFT 和 EFT 都需要进行大量合并;(2) 当 Graph OLAP 仅设计信息维时,由于无须调整图形的拓扑结构,只需对 NFT 和 EFT 对应标量权值属性进行累加即可.进而,由于 NFT 和 EFT 是通过节点标识 *Node_ID* 进行关联的,EFT 中边的合并借助 NFT 中节点的合并完成.因此,维表的分布分别按 EFT 和 NFT 中心组织,最为高效合理.以 Co-Author Networks 的概念建模为例,Graph 主题数据由 EFT 和 NFT 联合表示,信息维表 IDT 围绕在 EFT 周围,拓扑维表 TDT 围绕在 NFT 周围.Graph 数据立方计算可以高效地由相应维表、事实表的特化、概化、聚集等来计算,由此得到本文提出的双星模型,如图 3 所示.

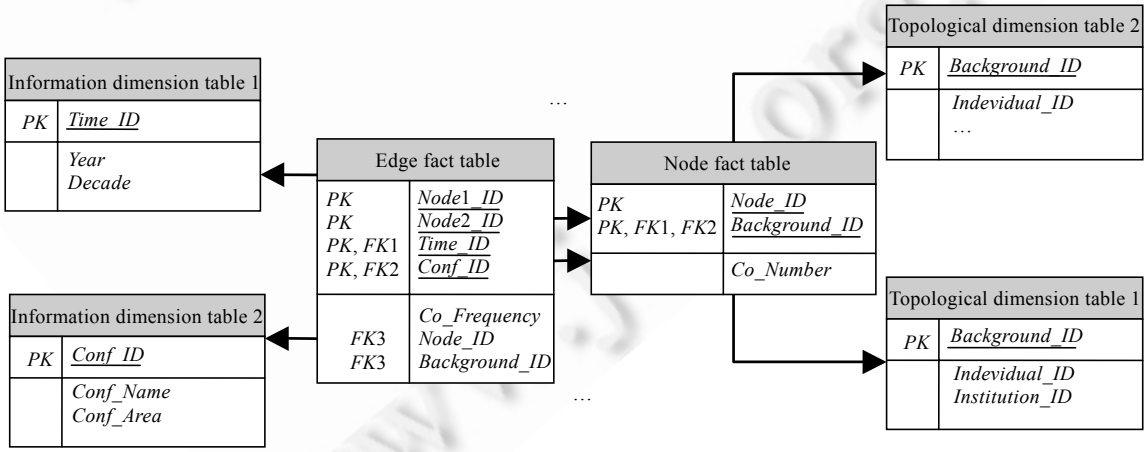

Fig.3 Double-Star model of Co-Author Networks

图 3 Co-Author Networks 的双星模型

定义 4(双星模型). 设 *IDT*,*TDT*,*NFT*,*EFT* 分别表示信息维表、拓扑维表、节点事实表、边事实表,则 Graph OLAP 的数据仓库概念模型是一个四元组(*IDT*,*TDT*,*NFT*,*EFT*).若干 *IDT* 围绕在 *EFT* 周围,构成一个中心星形;若干 *TDT* 围绕在 *NFT* 周围,构成另一个中心星形.*EFT* 和 *NFT* 分别成为两星内核.*EFT* 和 *NFT* 通过 *node_id* 键相互联系.该模型称为双星模型.

在图 3 中,*IDT* 包括 *Time* 维表、*Conference* 维表等,*TDT* 包括 *Background* 维表等,*NFT* 中存储节点标识 *Node_ID*,节点标量度量 *Node_Measure* 以及描述拓扑信息的 *Background_ID*.而 *EFT* 存储表示边的两个节点的 *Node_ID*1,*Node_ID*2,关于边的标量度量——权值,以及描述边信息的 IDT 的 Key.双星模型易于转化为关系数据库中的表,为 Graph OLAP 的逻辑设计和实现提供了便利.为便于进行 Graph OLAP 的聚集和图形重构等计算,首先应把落在信息维范围内的任务相关的图数据库抽取出来,进行计算并保存于 Graph 数据立方中.

3 Graph OLAP 的数据立方——对称方体格

Graph OLAP 需由多个维度、层次对图形结构进行观察,本研究提出以 Graph 数据立方体作为图形数据的组织方式.Graph 数据立方体与传统的数据立方体不同,*n-D* 数据立方体的每个单元格存储的不再是事实表的度量值,而是由特定信息维和拓扑维确定的子图快照 G_k.要得到不同的汇总聚集数据,需将对应的各个子图进行叠加.如图 4 所示为 Co-Author Networks 的一个 Graph 数据立方体示例.

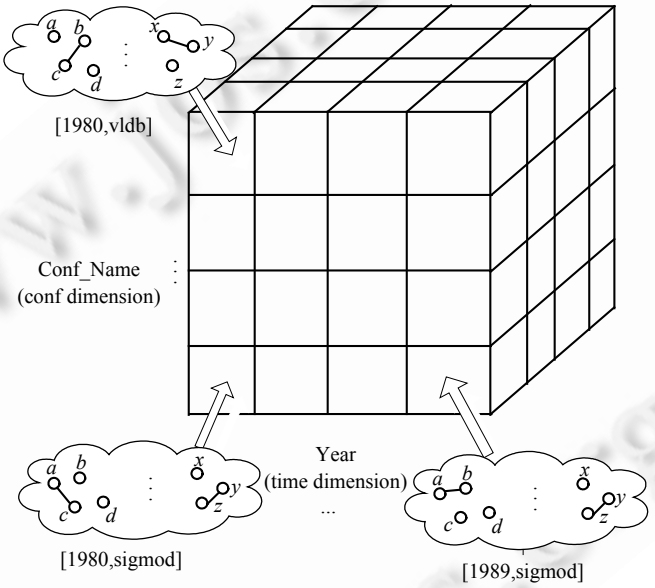

Fig.4 Graph data cube of Co-Author Networks

图 4 Co-Author Networks 的 Graph 数据立方体

Graph OLAP 的数据立方可构成方体的格.不同维度、不同层次对应特定条件的不同数据立方体.图 5 表示 Co-Author Networks 的 Graph 立方体格.数据立方间的实线表示两数据立方体可以通过 I-OLAP 操作进行相互转化.比如,左边的[all,all]的数据立方体可以通过对 *Time* 维进行下钻获得[decade,all]立方体.两个数据立方体间的虚线表示两个数据立方通过 T-OLAP 操作进行相互转化.比如,左侧 Background 维处于 individual 层次的数据立方体通过上卷聚集到 institution 层次,获得右侧的数据立方体.

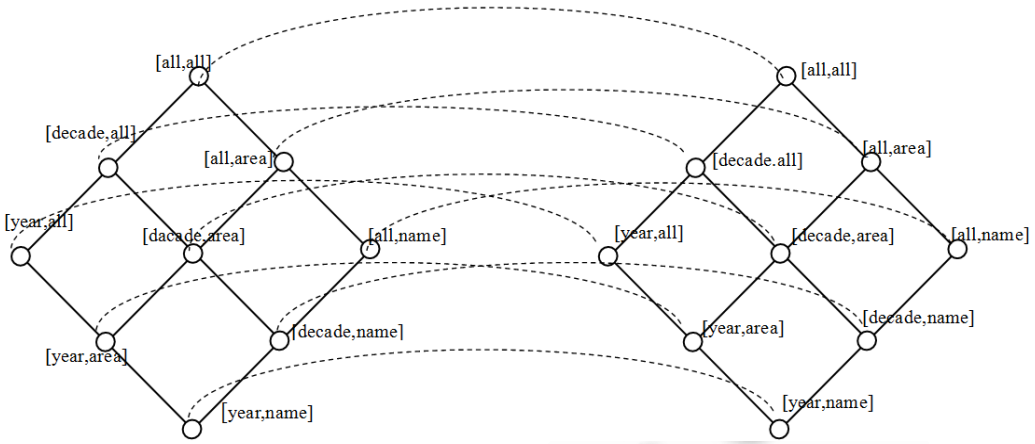

图 5 Graph OLAP symmetric lattice of Co-Author Networks

图 5 Co-Author Networks 的 Graph OLAP 对称方体格

4 基于 RDBMS 的 Graph OLAPing 算法设计

切片和切块是基于特定维度集合$\{I_1,\ldots,I_m\}$取值,如$\{I_1=A1,I_2=A2,\ldots,I_m=Am\}$时,对数据立方的子集计算.对基于 RDBMS 的 Graph OLAP 系统,切片和切块操作可以使用 SQL 语句 SELECT…FROM…WHERE…高效地实现.而上卷、下钻操作使用不同粒度的维度取值替代原维度,不仅涉及概念层次替换操作,而且必须进行大量数据立方中心度量的聚集计算.因此,对基于 RDBMS 的 Graph OLAP 系统,上卷下钻操作必须使用 GROUP BY 子句.由定义 1~定义 3,Graph OLAP 聚集操作可分为 I-OLAP 聚集和 T-OLAP 聚集,分别对应不同的执行机制.

4.1 I-OLAP聚集算法

I-OLAP 聚集计算的思路是基于信息维组合和取值约束进行 Graph 中心度量的计算.因为 I-OLAP 聚集仅对不同空间的 Graph 进行合并与聚集,不涉及图形拓扑结构的变化,因此主要以基本图数据库 $GDB_{base}(V,E)$的边集 E 作为处理对象.首先选取任务相关的边事实表,进而通过与 IDT 的选择-连接-投影和 GROUP BY 等操作完成结果 Graph Cube 的计算,如算法 1 所示.

算法 1. I-OLAPing.

输入:

(1) $GDB\_{base}(V,E)$:基本图数据库,包含 *EFT*,*NFT*,*IDT* 等;

(2) $IDS=\{I_1,I_2,\ldots,I_n\}$:用户指定的聚集操作涉及的信息维;

(3) $Constraints=\{C_1,C_2,\ldots,C_n\}$:用户指定的 Graph OLAP 的信息维概念层次和参数取值约束;

(4) F:聚集函数.

输出:$G\text{-}Cube_{Iagg}$:经信息维聚集后的 Graph Cube.

步骤:

(1) *WorkingEFT*=*GetEFT*(*EFT*,*IDS*); //获取工作边事实表

(2) **for** (**int** i=1;$i\leqslant n$;i++){

(3) IDT_n=*GetIDT*(I_n,C_i); //获得相关信息维表 IDT_n

(4) *WorkingEFT*=*WorkingEFT*$\bowtie IDT_n$; //*WorkingEFT* 与 IDT_n 作自然连接

(5) }

(6) *EdegeCube*=*EdgeCompute*(*WorkingEFT*,F,I_1,…,I_n); //对 *WorkingEFT* 进行基于 GROUPBY 的聚集计算

(7) $G\text{-}Cube_{Iagg}$=*RebuildG*(*NFT*,*EdegeCube*); //得到信息维聚集后的结果 Graph Cubiod

(8) Output G_{Iagg}; //输出 Graph Cubiod

由于大量图数据库以半结构化文本(XML)或文本(TXT)形式存储和交换,首先按照双星模型建立 Graph OLAP 的数据仓库,进而由预处理程序完成数据仓库导入.算法 1 首先根据相关信息维集合 *IDS* 与边事实表 *EFT* 提取和本次聚集操作相关的边事实表子集,称为工作边事实表 *WorkingEFT*(第 1 行).进而,根据 *WorkingEFT* 与用户指定的维度概念层次,通过连接、GROUPBY、聚集等操作得到边信息构成的数据立方体,*EdegeCube* (第 2 行~第 6 行).最后,经由 *Rebuild*()函数根据 *CubeEdge* 和 *NFT* 重建图,得到 Graph 数据立方体,*G-Cube*(第 8 行、第 9 行).

4.2 T-OLAP算法

T-OLAP 涉及结果 Graph Cube 中图形拓扑结构的变化,不但需要对基本图数据库 $GDB_{base}(V,E)$的边集 E 进行计算,而且需要对节点集 V 进行操作.由于双星模型将 IDT 和 TDT 分别分布于 EFT 和 NFT 周围,使拓扑维的计算免受 EFT 和 IDT 的干扰,简化了问题复杂程度.但根据定义 1~定义 3,拓扑维 OLAP 同时受制于点拓扑决定函数φ、边拓扑决定函数δ以及边信息决定函数θ,因而比 I-OLAP 聚集复杂得多.拓扑维的变化对 EFT 和 NFT 将同时发生影响.而且,T-OLAP 的计算同时涉及 NFT 与 TDT 的连接、EFT 与 IDT 的连接和 NFT 与 EFT 的连接.上述操作处理不慎,必然导致动态组合爆炸问题.

因此,T-OLAP 算法设计的核心问题是尽可能减少参与最终运算的 NFT,EFT,TDT,同时将概念层次和预定义参数尽可能早地融入核心处理过程.算法 2 描述了 T-OLAP 的执行步骤.

算法 2. T-OLAPing.

输入:

(1) $GDB\_{base}(V,E)$:基本图数据库,包含 EFT,NFT,TDT,IDT 等;

(2) $TDS=\{T_1,T_2,\ldots,T_m\}$:用户指定的聚集操作涉及的拓扑维;

(3) $Constraints=\{C_1,C_2,\ldots,C_n\}$:用户指定的 Graph OLAP 的概念层次和参数取值约束;

(4) F:聚集函数.

输出:$G\text{-}Cube_{Iagg}$:经拓扑维聚集后的 Graph Cube.

步骤:

(1) *WorkingNFT*=*GetNFT*(*NFT*,*Constraints*); //获得相关工作节点事实表

(2) **for** (**int** i=1;$i\leqslant m$;i++){

(3) TDT_m=*GetTDT*(T_m); //获得相关拓扑维表 TDT_m

(4) *WoringNFT*=*WorkingNFT*$\bowtie TDT_m$; //*WorkingNFT* 与 TDT_m 作自然连接

(5) }

(6) *WorkingEFT*=*EFT*;

(7) *WorkingEFT*=*WorkingEFT*$\bowtie$*WorkingNFT*; //获得相关工作边事实表

(8) *WorkingEFT*=*GetEFT*(*WorkingEFT*,*Constraints*); //过滤不相关边数据

(9) *EdgeCube*=*EdgeCompute*(*WorkingEFT*,*F*,$T_1,\ldots,T_m$); //对 *WorkingEFT* 进行聚集

(10) *VertexCube*=*VertexCompute*(*WorkingNFT*,*F*,$T_1,\ldots,T_m$); //对 *WorkingNFT* 进行聚集

(11) $G\text{-}Cube_{Tagg}$=*RebuildG*(*VertexCube*,*EdgeCube*); //得到拓扑维聚集后的 Graph Cubiod

(12) Output $G\text{-}Cube_{Tagg}$; //输出 Graph Cubiod

算法 2 首先根据 *Constraints* 给定的取值约束对 *NFT* 进行过滤,得到任务相关的工作 *NFT*,*WorkingNFT*(第 1 行).继而,对每个待处理维度 *Tm*,得到相关拓扑维表,通过 *WorkingNFT* 与 TDT_m 的自然连接,剪枝待处理的 *WorkingNFT*(第 2 行、第 3 行).然后,通过与 *WorkingNFT* 的自然连接和经 *Constraints* 的过滤,计算任务相关的 *WorkingEFT*(第 6 行~第 8 行).得到最终待处理的 *WorkingNFT* 和 *WorkingEFT* 后,通过基于 GROUPBY 的聚集函数 *F* 计算 *EdgeCube*,*VertexCube*,合并得到 $G\text{-}Cube_{Tagg}$ 并输出(第 9 行~第 12 行).

5 Graph OLAPing 性能分析

5.1 Co-Author Network数据生成器

通过对 Social Network,BioNetwork,Web 等的观察发现,这些网络具有某些共同性质:

(1) 整体稀疏、局部密集;

(2) 顶点度数服从幂率分布,即无标度(scale-free)特性;

(3) 整体分布高内聚、低平均路径长度(平均最短路径=$O(\log(\log n))$,符合小世界(small-world)特性.

具有上述性质的网络又称为复杂网络[14].

Co-Author Network除了满足上述复杂网络相关特性以外,还有其自身的规律性.例如,论文作者倾向于与同一单位的作者合作,合作关系呈现“强者愈强”的“富人俱乐部”现象,单位间的合作概率也有明显的区别.两作者是否合作发表论文主要受制于 3 个因素:个人因素、单位因素、随机因素.个人因素包括个人的努力程度、研究水平、时间和精力.由于每发表一篇论文都消耗大量的时间和精力,随着论文数目的增多,代表个人因素的因子将呈减小趋势.而随着环境改善和经费投入的增加,个人因素因子逐步增加.单位因素包括单位经费、影响力、设施等.如果两作者同处一个单位,研究课题相近,交流机会较多,则合作论文的概率比不同单位作者之间合作概率大.再有,较强的研究者或研究单位也容易产生合作.

根据以上特点,建立 Co-Author Network 模型,为使分析结果更加逼近真实而不受特定数据分布的影响,使用人工合成科研合作网实验数据集.定义两个作者 i,j 之间的合作关系的概率 $Pr(i,j)$,满足如下条件:

$$Pr(i,j)=\begin{cases}\alpha(P\_ind[i]\cdot P\_ind[j])+\beta(P\_ins[i]\cdot P\_ins[j])+\gamma\cdot random(0,1), & i.ins\neq j.ins\\ \alpha(P\_ind[i]\cdot P\_ind[j])+\beta+\gamma\cdot random(0,1), & i.ins=j.ins,\\ \alpha+\beta+\gamma=1\end{cases}$$

其中,α,β,γ是网络构形参数,通过调整α,β,γ的值可以改变 Co-Author Network 的合作关系分布.如,当α=0,β=0,γ=1时,合作关系网呈现随机分布的状态;当α=1,β=0,γ=0 时,忽略单位间的差异,合作的概率只与科研者自身水平相关;当α=0,β=1,γ=1 时,同单位内部合作概率加大,强强合作明显增加.设置α=0.55,β=0.3,γ=0.15,不同合作者数目 N,不同单位数目 M,得到 6 种不同规模的网络,见表 1.

Table 1 Co-Author Network dataset

表 1 Co-Author Network 数据集

Dataset number	Co-Authored papers N	Institution number M	Vertex number V	Edge number E
$G1$	50	5	50	108
$G2$	500	50	500	1 626
$G3$	1 000	100	1 000	4 334
$G4$	2 000	200	2 000	9 196
$G5$	4 000	400	4 000	26 167
$G6$	5 000	500	5 000	35 423

5.2 实验环境

本文实验环境是:(1) CPU:Intel Pentium 7370 2GHz;(2) 内存:2GB;(3) 操作系统:Windows XP Home;(4) 数据库:MS SQL Server 2000 Personal;(5) 编程语言:Eclipse Java 3.5.1;(6) 可视化模块:Java3D 1.4.

5.3 实验分析

为了系统地测试算法性能,同时消除具体操作路径的影响,本节分别对数据集 $G1$~$G6$ 执行以下两种 Graph OLAP 查询:

$Q7$:查询每个合作作者按照会议类型会议年份的聚集结果(I-OLAP);

$Q8$:查询每个合作单位按照会议类型会议年份的聚集结果(T-OLAP).

图 6、图 7 反映对于两种查询在 $G1$~$G6$ 数据集中的时间和内存消耗.

注意,图 6 时间维坐标是 ms,图 7 空间单位是 KB.由图 6 可知,$Q8$ 的时间开销略大于 $Q7$,其原因是,T-OLAP

比 I-OLAP 增加拓扑维表的聚集计算以及对结果的后续处理过程.当数据量较少时,T-OLAP 与 I-OLAP 聚集的时间消耗相当;随着数据量的增大,T-OLAP 的复杂性致使需要更多时间进行相关运算,算法 2 通过内外存的配合使这种差距尽可能地小.图 6 同时表明,T-OLAP 与 I-OLAP 的时间消耗随数据量的增长基本上呈线性增长态势.如图 7 所示,对于同样的查询,T-OLAP 的内存开销小于 I-OLAP,其原因是,T-OLAP 所处理的图是原始图的一种聚集、压缩.

图 8 表明,对于相同的维度个数,T-OLAP 的时间消耗大于 I-OLAP;同时,随着查询维度的增加,T-OLAP 与 I-OLAP 的时间基本上呈线性增长.

图 9 对比两种算法的存储消耗随维度数变化的规律,可以看出,对于相同维度个数,T-OLAP 的内存消耗小于 I-OLAP.T-OLAP 与 I-OLAP 的内存消耗随查询维度的增加而增长.其原因是,查询的维度越小,图数据的规模就越小,因此,查询所需的内存空间也就越小.

图 8、图 9 描述在图数据集 *G*6 中查询的维数对于 T-OLAP 以及 I-OLAP 算法时间和存储消耗的影响.

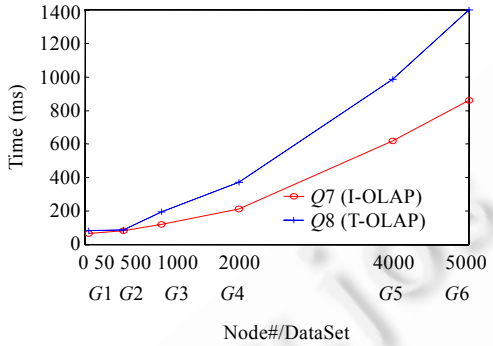

Fig.6 Time consumption of *Q*7, *Q*8 on *G*1~*G*60

图 6 *Q*7,*Q*8 在 *G*1~*G*6 的时间消耗

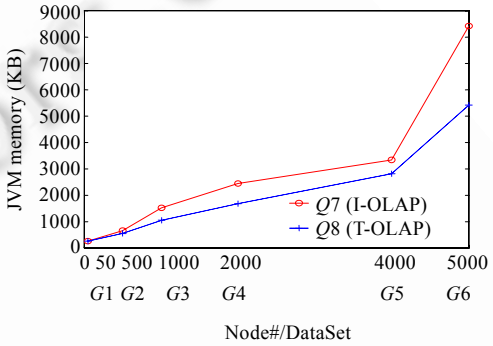

Fig.7 Memory consumption of *Q*7, *Q*8 on *G*1~*G*6

图 7 *Q*7,*Q*8 在 *G*1~*G*6 的内存消耗

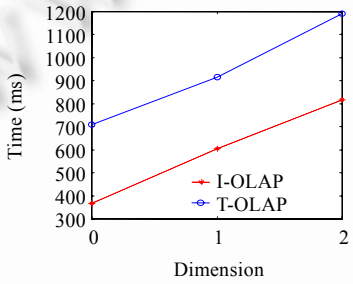

Fig.8 Effect of dimensionality to T-OLAP, I-OLAP time

图 8 维度数对 T-OLAP,I-OLAP 时间的影响

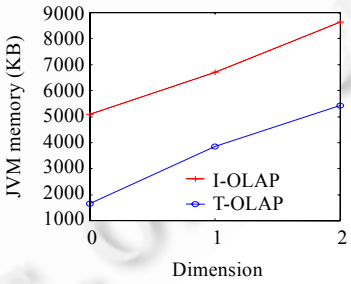

Fig.9 Effect of dimensionality to T-OLAP, I-OLAP memory

图 9 维度数对 T-OLAP,I-OLAP 内存的影响

5.4 Graph OLAPing原型系统

本文基于上述工作实现 Graph OLAPing 原型系统,可用于 Co-Author Networks 等图数据库的在线分析处理.系统演示版可在地址 http://cs.scu.edu.cn/~lichuan/GraphOLAP.zip 下载.系统界面如图 10 所示.

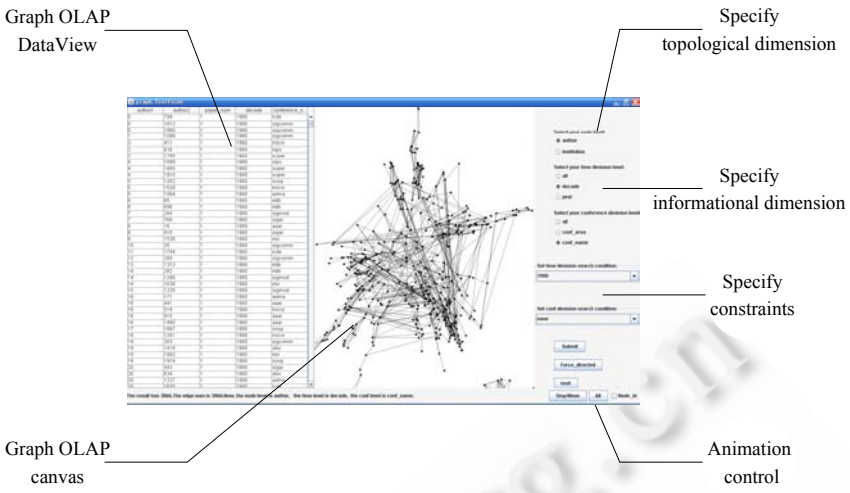

Fig.10 Prototype interface of Graph OLAP 1.0

图 10 Graph OLAPer1.0 原型系统界面

6 总结与展望

传统技术无法对大规模、复杂结构的 Graph 主题数据进行 OLAP 处理.针对此问题,本文提出 Graph OLAP 的概念建模、数据立方设计以及 Graph OLAP 的算法设计,提出双星模型、图数据立方、I-OLAPing 及 T-OLAPing 的设计思想和算法实现,并针对 Graph OLAPing 的数据特征设计两种 Graph OLAPing 优化方法,实现了基于关系数据库的 Graph OLAPing 原型系统.今后的工作重点包括:(1) 完善 Graph OLAP1.0 模型,将聚集函数扩展到一般的聚集函数,针对图数据特有的聚集函数提出相应的解决方法;(2) 改进 Graph OLAPing 算法,进一步优化图数据的处理,提升性能;(3) 在 Graph OLAPing 体系中引入缓存和图索引机制.

References:

[1] Harinarayan V, Rajaraman A, Ullman JD. Implementing data cubes efficiently. In: Jagadish HV, ed. Proc. of the SIGMOD. New York: ACM, 1996. 205−216. [doi: 10.1145/233269.233333]

[2] Gray J, Chaudhuri S, Bosworth A, Layman A, Reichart D, Venkatrao M, Pellow F, Pirahesh H. Data cube: A relational aggregation operator generalizing group-by, cross-by, cross-tab and sub-totals. In: Su SYW, ed. Proc. of the ICDE. New Orleans: IEEE Computer Society, 1996. 152−159. [doi: 10.1109/ICDE.1996. 492099]

[3] Chaudhuri S, Dayal U. An overview of data warehousing and OLAP technology. SIGMOD Record, 1997,26(1):65−74. [doi: 10.1145/248603.248616]

[4] Vassiliadis P, Sellis T. A survey of logical models for OLAP databases. SIGMOD Record, 1999,28(4):64−69. [doi: 10.1145/344816.344869]

[5] Beyer KS, Ramakrishnan R. Bottom-Up computation of sparse and iceberg cubes. In: Delis A, Faloutsos C, Ghandeharizadeh S, eds. Proc. of the SIGMOD Conf. Philadelphia: ACM Press, 1999. 359−370. [doi: 10.1145/304182.304214]

[6] Gupta A, Mumick IS. Materialized Views: Techniques, Implementations, and Applications. MIT Press, 1999.

[7] Zhao YH, Deshpande PM, Naughton JF. An array-based algorithm for simultaneous multidimensional aggregates. In: Peckham J, ed. Proc. of the SIGMOD Conf. Tucson: ACM Press, 1997. 159−170. [doi: 10.1145/253260.253288]

[8] Beyer KS, Ramakrishnan R. Bottom-Up computation of sparse and iceberg cubes. In: Delis A, Faloutsos C, Ghandeharizadeh S, eds. Proc. of the SIGMOD Conf. Philadelphia: ACM Press, 1999. 359−370. [doi: 10.1145/304182.304214]

[9] Lu HP, Shi Y. Complexity of public transport networks. Tsinghua Science and Technology, 2007,12(2):204−213. [doi: 10.1016/S1007-0214(07)70029-9]

[10] Xu J, Chen H. Criminal network analysis and visualization: A data mining perspective. Communications of the ACM, 2005,48(6):

100−107. [doi: 10.1145/1064830.1064834]

[11] Papadias D, Kalnis P, Zhang J, Tao Y. Efficient OLAP operations in spatial data warehouses. In: Jensen CS, Schneider M, Seeger B, Tsotras VJ, eds. Proc. of the 6th Int'l Symp. on Spatial and Temporal Databases. London: Springer-Verlag, 2001. 443−459.

[12] Chakrabarti D, Faloutsos C. Graph mining: Laws, generators, and algorithms. ACM Computing Surveys, 2006,38:Article 2.

[13] Tian Y, Hankins RA, Patel JM. Efficient aggregation for graph summarization. In: Lakshmanan LVS, ed. Proc. of the SIGMOD. New York: ACM, 2008. 567−580. [doi: 10.1145/1376616.1376675]

[14] Raghavan S Garcia-Molina H. Representing Web graphs. In: Dayal U, Ramamritham K, Vijayaraman TM, eds. Proc. of the ICDE. Bangalore: IEEE Computer Society, 2003. 405−416. [doi: 10.1109/ICDE.2003. 1260809]

[15] Boldi P, Vigna S. The WebGraph framework I: Compression techniques. In: Feldman SI, Uretsky M, Najork M, Wills CE, eds. Proc. of the WWW. New York: ACM, 2004. 595−602. [doi: 10.1145/ 988672.988752]

[16] Wu AY, Garland M, Han J. Mining scale-free networks using geodesic clustering. In: Kim W, Kohavi R, Gehrke J, DuMouchel W, eds. Proc. of the KDD. Seattle: ACM, 2004. 719−724. [doi: 10.1145/1014052.1014146]

[17] Archambault D, Munzner T, Auber D. Topolayout: Multilevel graph layout by topological features. IEEE Trans. on Visualization and Computer Graphics, 2007,13(2):305−317. [doi: 10.1109/TVCG.2007.46]

[18] Navlakha S, Rastogi R, Shrivastava N. Graph summarization with bounded error. In: Wang JTL, ed. Proc. of the SIGMOD Conf. Vancouver: ACM, 2008. 419−432. [doi: 10.1145/1376616.1376661]

[19] Ng AY, Jordan MI, Weiss Y. On spectral clustering: Analysis and an algorithm. In: Dietterich TG, Becker S, Ghahramani Z, eds. Proc. of the NIPS. Vancouver: MIT Press, 2001. 849−856.

[20] Gibson D, Kumar R, Tomkins A. Discovering large dense subgraphs in massive graphs. In: Böhm K, Jensen CS, Haas LM, Kersten ML, Larson PÅ, Ooi BC, eds. Proc. of the VLDB. Trondheim: ACM, 2005. 721−732.

[21] Herman I, Melancon G, Marshall MS. Graph visualization and navigation in information visualization: A survey. IEEE Trans. on Visualization and Computer Graphics, 2000,6(1):24−43.

[22] Chen C, Yan XF, Zhu FD, Han JW, Yu PS. Graph OLAP: Towards online analytical processing on graphs. In: Proc. of the ICDM Conf. 2008. 103−112. [doi: 10.1109/ICDM.2008.30]

李川(1977－),男,河南郑州人,博士,副教授,CCF 会员,主要研究领域为数据挖掘,人工社会.

李靓(1986－),女,硕士生,主要研究领域为智能机器人,计算机视觉,三维建模.

赵磊(1987－),男,硕士生,CCF 学生会员,主要研究领域为数据挖掘

赵小明(1986－),女,硕士生,主要研究领域为自然语言处理.

唐常杰(1946－),男,教授,博士生导师,CCF 高级会员,主要研究领域为数据挖掘.

刘小玲(1987－),女,硕士生,主要研究领域为数据挖掘.

陈瑜(1974－),男,博士,讲师,主要研究领域为数据挖掘,智能信息处理,进化计算.