

路径-维度 GraphOLAP 大规模多维网络并行分析框架*

张子兴, 吴斌, 吴心宇, 张有杰, 孙思瑞, 彭程程, 刘昱彤



(智能通信软件与多媒体北京市重点实验室(北京邮电大学), 北京 100876)

通讯作者: 吴斌, E-mail: wubin@bupt.edu.cn

摘要: 现实生活中,大量数据都可以使用多维网络进行建模.如何更好地对多维网络进行分析,是研究人员关注的重点.OLAP(联机分析处理)技术已被证实是对多维关系数据进行分析的有效工具,但应用 OLAP 技术管理与分析多维网络数据以支持有效决策,仍是一项巨大的挑战.设计并提出了一种图立方体模型:路径-维度立方体,并针对提出的立方体模型将物化过程划分为关系路径物化与关联维度物化两部分,分别提出了物化策略,并基于 Spark 框架设计了相关算法.在此基础上,针对网络数据设计并细化了相关的 GraphOLAP(图联机分析处理)操作,丰富了框架的分析角度,提高了对多维网络的分析能力.最后,在 Spark 上实现了相关算法,通过对多个真实应用场景中的数据构建多维网络,在分析框架上进行了分析,实验结果表明,所提出的图立方体模型和物化算法具有一定的有效性和可扩展性.

关键词: 图立方体;立方体物化;关系路径;图联机分析处理

中图法分类号: TP311

中文引用格式: 张子兴,吴斌,吴心宇,张有杰,孙思瑞,彭程程,刘昱彤.路径-维度 GraphOLAP 大规模多维网络并行分析框架.软件学报,2018,29(3):545-568. <http://www.jos.org.cn/1000-9825/5443.htm>

英文引用格式: Zhang ZX, Wu B, Wu XY, Zhang YJ, Sun SR, Peng CC, Liu YT. P&D GraphOLAP: Parallel framework for large-scale multidimensional network analysis. Ruan Jian Xue Bao/Journal of Software, 2018,29(3):545-568 (in Chinese). <http://www.jos.org.cn/1000-9825/5443.htm>

P&D GraphOLAP: Parallel Framework for Large-Scale Multidimensional Network Analysis

ZHANG Zi-Xing, WU Bin, WU Xin-Yu, ZHANG You-Jie, SUN Si-Rui, PENG Cheng-Cheng, LIU Yu-Tong

(Beijing Key Laboratory of Intelligent Telecommunications Software and Multimedia (Beijing University of Posts and Telecommunications), Beijing 100876, China)

Abstract: Most data in real life can be described as multidimensional networks. How to process the analysis on multidimensional networks from multiple views and multiple granularities is still the focus of current research. Meanwhile, OLAP (online analytical processing) technology has been proven to be an effective tool on relational data. However, it is an enormous challenge to manage and analyze multidimensional heterogeneous networks via OLAP technology to support effective decision making. In this paper, a P&D (path and dimension) graph cube model is proposed. Based on this model, the graph cube materialization is divided into two parts, termed as path related materialization and dimension related materialization, and the corresponding materialization algorithms are designed. Some GraphOLAP operations are also refined to improve the ability of analyzing multidimensional networks. Finally, the algorithms are implemented on Spark and the multidimensional networks are constructed through real datasets. These networks are then analyzed using

* 基金项目: 国家重点基础研究发展计划(973)(2013CB329606); 国家自然科学基金(61772082)

Foundation item: National Program on Key Basic Research Project (973) (2013CB329606); National Natural Science Foundation of China (61772082)

本文由基于图结构的大数据分析与管理技术专刊特约编辑林学民教授、杜小勇教授、李翠平教授推荐.

收稿时间: 2017-07-31; 修改时间: 2017-09-05; 采用时间: 2017-11-07; jos 在线出版时间: 2017-12-05

CNKI 网络优先出版: 2017-12-06 15:23:18, <http://kns.cnki.net/kcms/detail/11.2560.TP.20171206.1522.008.html>

the framework. The results of experiments validate the effectiveness and scalability of P&D graph cube model and the materialization algorithms.

Key words: graph data cube; materialization; relation path; GraphOLAP

随着信息技术的不断发展,各个应用领域逐渐积累了海量数据.网络结构由于其强大的表达能力,能够对现实生活中大量的数据进行建模,例如社交网络、文本网络、科研合作网络、知识图谱、生物网络等.如何针对多维网络进行分析,进而挖掘出有价值的信息以支持有效决策,已成为迫切需要解决的问题.

传统的多维网络分析方法^[1-5]只能对网络当前层次下的各种实体间的关联关系进行分析,不能很好地对网络从不同角度、不同粒度、不同层次进行分析.例如,图 1(a)所示是一个音乐社交软件的社交关系网络.网络中的每个节点都代表一个实体,共包含歌曲、专辑、歌手和用户 4 种不同类型的实体,见表 1,每个实体都包含一组与其对应的多维属性;网络中的每条边代表两个实体之间的关系,歌手和歌曲之间的边表示这首歌曲是由该歌手演唱,用户与歌曲之间的边表示用户听过这首歌,歌曲与专辑之间的边表示这首歌曲属于此张专辑,用户的自环表示两个用户之间互为好友.

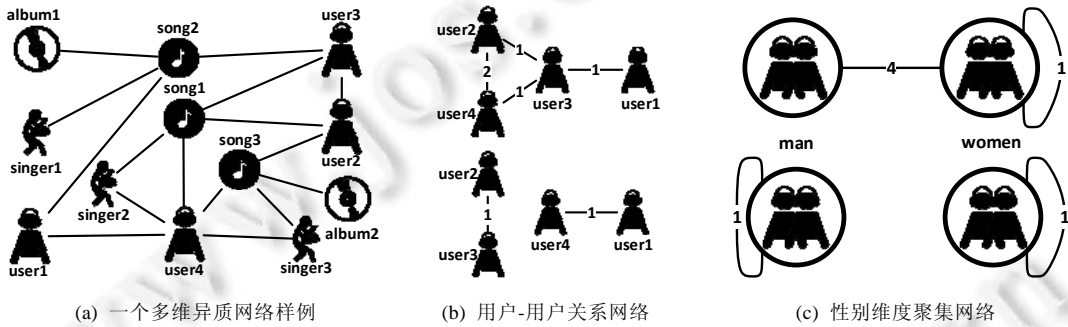


Fig.1 A sample of multidimensional network analysis

图 1 多维网络分析样例

Table 1 Entity Attribute Table

表 1 实体属性表

UserID	Gender	Age	SingerID	Gender	Country	Company
user1	man	20	singer1	man	China	Co1
user2	women	18	singer2	man	Korea	Co2
user3	women	23	singer3	women	America	
user4	man	36				
SongID	Date	Tag	Album	Price	Type	
song1	20161020	blue	album1	100	EP	
song2	20160101	jazz	album2	100	Mini	
song3	20160730	pop				

传统的分析方法针对原始网络进行建模,然而现实生活中的网络往往拥有大量不同类型的节点,且节点包含丰富的维度属性,我们对网络的分析需求也更为复杂.例如针对图 1(a)所示的网络,我们想探究网络中用户与用户之间的关系,显然不能从原始网络中直接得到希望的结果,还需要对原始网络进行一定的转化.图 1(b)中,下部分的网络为从图 1(a)中提取的原始用户关系网络,其中每条边是由原始网络中用户与用户之间的朋友关系产生的,我们可以从中探究出用户与用户之间的关联关系.进一步思考,两个用户可能共同听过某首歌,那么这两个用户之间也存在着由共同听过同一首歌曲建立起的联系.图 1(b)中,上部分的网络展示了用户之间共同听过同一首歌曲的聚合网络,其中,边的权重是由用户之间共同听过的歌曲数量来计算的.在这个例子中,我们选用 COUNT(*)作为聚合函数.例如,user2 和 user4 共同听过 song1 和 song3,因此 user2 和 user4 之间边的权重就是 2.更进一步,若希望探究用户与用户性别之间的关系,则在得到用户关系网络后,还需要将网络沿性别维度对节点进行聚合(如图 1(c)所示),从而得到用户与用户的性别之间的关系.

上述复杂的分析显然是传统分析方法所不擅长的.尽管我们能够通过一定的预处理先将网络进行转化,再对转化后的网络进行分析,但现实情况下的分析需求更为复杂,需要频繁地使网络在不同角度、不同粒度、不同层次上进行转化,单纯通过预处理的方式显然无法支持如此复杂的分析.

GraphOLAP(图联机分析处理)技术将传统的 OLAP(联机分析处理)技术与图分析技术相结合,以支持图数据的多维分析.传统的 OLAP 技术是数据仓库与数据挖掘领域中的有力工具,能够解决传统的多维关系数据分析问题,在其 20 余年的研究与发展中,已成为一项成熟的技术^[6-8].但由于网络数据除了包含传统关系数据中的实体信息与维度信息外,还包含不同实体间错综复杂的拓扑结构,传统的模型显然不能很好地解决网络的多维分析问题.尽管近年来已有学者针对 GraphOLAP 进行了一定的研究,但无论针对图立方体模型还是对应的 GraphOLAP 操作的研究都稍显不足.因此,本文关注的重点在于设计一种图立方体模型,提出物化策略,实现网络数据的建模和物化,并在此基础上设计并细化相关操作,丰富其应用场景.

文献[9]中最早提出了图立方体的概念,但这仅仅是从传统的数据立方体迁移而来,并没有过多地考虑网络中的拓扑结构.文献[10]针对多维图数据提出了图的超立方体模型,并将方体分为 V-Agg、E-Agg 和 VE-Agg 这 3 个部分,但该模型主要针对同构网络,无法对具有多种实体类型的网络建模.文献[11]中提出了一种名为 TSMH Graph cube 的图立方体模型,它以元路径作为指导构造实体超立方体和维度立方体,并探讨了立方体的物化过程.尽管该模型解决了异质网络分析问题,但该模型将两个实体间的不同元路径组织在一个方体中且只能应用同样的聚合函数,倘若我们需要针对不同的关系路径应用不同的聚合函数,这个模型显然是不能够实现的;同时,该模型将每条关系路径都存储成一个完整的聚合网络,且针对每个聚合网络都维护一个维度立方体,这样的设计策略消耗了大量的存储空间且并不具有很好的可扩展性.为了弥补上述模型的种种不足,我们提出了一种新的基于关系路径集的图立方体模型.

同时,随着多元信息相互融合,现实生活中的网络逐渐变得更为异质性,规模越来越大.在单机上处理如此大规模的网络显然不能满足我们的需求.为了能够更好地对这些海量图数据进行计算,众多的分布式图计算框架逐渐被提出,例如 Pregel^[12],Powergraph^[13],GraphX^[14]和 Hama^[15].这些计算框架遵循大规模并行计算模型的设计模式,对于需要大量迭代计算的图处理算法,如 page-rank、最短路径、无监督聚类^[12]等是非常有效的.然而,这些图计算框架在图立方体的物化与 OLAP 操作的计算方面并没有提出很好的解决方案.文献[16]中证明了在合理设计算法的基础上,在 Spark 和 MapReduce 模型上进行大规模图挖掘是可行且高效的.因此,我们将图立方体计算分解为关系路径物化和关联维度物化两部分,分别在 Spark 框架上实现了相关算法.

在 GraphOLAP 操作方面,Chen 等人^[17]首次提出了 GraphOLAP 概念,并定义了信息维、拓扑维以及这两个维度的上卷、下钻、切片、切块的操作.文献[18]提出了一种双星模型,并在此基础上设计了信息维度聚集算法与拓扑维度聚集算法.Zhao 等人^[9]针对图数据立方体模型,将方体查询引入图数据分析中,将 OLAP 操作转化为 crossboid 查询.上述研究都是针对同质网络,不能将操作应用于异质网络.文献[11,19]针对异质网络设计了模型,探究了针对异质网络的上卷、下钻操作,并提出了旋转、属性转化等操作.尽管上述工作对 GraphOLAP 操作进行了一定研究,但大部分操作都是从传统针对关系型数据的 OLAP 系统中继承而来,操作种类较为单一.且已有的研究大多针对科研合作网络与演员合作网络,缺乏更多真实应用场景的拓展.

因此,本文提出了新的图立方体模型以及相关的物化策略,细化了针对多维网络的 GraphOLAP 操作,在此基础上,基于并行计算框架设计了大规模多维网络分析框架,并基于多种真实应用场景中的数据集进行了广泛的实验.我们的贡献可概括为如下几点.

- (1) 在多维异质网络的基础上重新定义了聚合网络和关系路径集的概念,提出了一种以关系路径集为指导的新的图立方体模型:路径-维度立方体;
- (2) 在立方体物化方面,我们在路径-维度立方体模型的基础上将物化过程分为关系路径物化与关联维度物化,分别提出了相应的物化策略,并基于 Spark 设计并实现了高效的立方体物化算法;
- (3) 在路径-维度立方体的基础上,针对实际的分析需求,我们细化且丰富了 GraphOLAP 中的相关操作,以满足更为复杂的分析.并通过相关实验,丰富了 GraphOLAP 操作的应用场景.实验结果表明:提出的框

架能够以一些新的角度从网络中挖掘有价值的信息,且提出的物化算法是高效、可扩展的,能够对大规模网络进行处理。

本文第1节提出路径-维度立方体模型,并探讨路径-维度立方体的物化策略,设计物化算法.第2节针对网络数据,细化并设计基于路径-维度立方体模型的 GraphOLAP 操作.第3节介绍基于路径-维度立方体模型的 GraphOLAP 多维网络分析框架.第4节在 Spark 框架上实现相关算法并在多个真实场景中的数据集上进行丰富的实验,分析实验结果.第5节介绍相关工作.第6节总结本文的研究内容和研究成果,并讨论进一步的研究方向。

1 路径-维度立方体与物化策略

1.1 路径-维度立方体

在提出该模型之前,首先介绍基本的定义,便于更加清晰地阐述相关概念。

定义1(多维异质网络). 一个多维异质网络可以表示为 $N=(V,E,T,R,A)$,其中, V 表示顶点的集合; $E\subseteq V\times V$ 是所有边的集合; $T=\{T_1,T_2,\dots,T_n\}$ 表示网络中所有实体类型的集合且 $n\geq 1$; $R\subseteq T\times T$ 为不同类型实体间关系类型的集合,表示为 $R(T_1,T_2)$; $A=\{A_1,A_2,\dots,A_n\}$ 表示实体的维度集合,对于每种实体类型 T_i 都有一个对应的维度集合:

$$A_i=\{A_{i1},A_{i2},\dots,A_{in}\}.$$

例如,针对表1所示的多维网络,用 $N=(V,E,T,R,A)$ 表示该网络, V 为所有节点集合, E 为所有边集合,网络中分别有歌曲、专辑、歌手和用户这4种不同类型的节点,因此 $T=\{song,album,singer,user\}$.网络中有4种类型的边, $R=\{R(singer,song),R(user,song),R(song,album),R(user,user)\}$,每种节点对应一个维度集合, $A=\{A_{song},A_{album},A_{singer},A_{user}\}$,其中, $A_{song}=\{A_{songDate},A_{songTag}\}$,因此, $song1$ 的属性可以表示为 $A_{song}(song1)=\{20161020,blue\}$.

定义2(关系路径). 给定一个多维异质网络 $N=(V,E,T,R,A)$,一个关系路径可以表示为

$$P(S,D,R')=S \xrightarrow{R_0} T_1 \xrightarrow{R_1} T_2 \rightarrow \dots \xrightarrow{R_{k-1}} T_k \xrightarrow{R_k} D,$$

其中, S 是源实体类型; D 是目的实体类型; $R_i\in R$ 且 R_i 是路径 P 上的第 i 个关系, R' 是实体 S 和实体 D 之间的复合的关系,表示为 $R_0\circ R_1\circ\dots\circ R_k$,路径 P 的长度就是 R' 的长度.简单起见,我们可以将关系路径表示为

$$P(S,D,R')=S-T_1-T_2-\dots-T_k-D.$$

例如,针对图1所示的网络,用户与专辑之间的关系路径可以为

$$P(user,album,R')=user-song-album,R'=R(user,song)\circ R(song,album).$$

对于每个关系路径 $P(S,D,R')$,我们可以通过它计算出其对应的无向边 $E=S\times D$,它能够表示实体 S 和实体 D 之间的关系,因此,关系路径的引入,能够丰富网络中节点之间的联系.通常情况下,一个网络中实体 S 和实体 D 之间的关系路径可能存在多条.例如图1的网络中,我们观察用户与歌曲之间的关系路径,可表示为

$$P(user,song,R(user,song))=user-song$$

$$P(user,song,R(user,singer)\circ R(singer,song))=user-singer-song$$

$$P(user,song,R(user,singer)\circ R(singer,song)\circ R(song,user)\circ R(user,song))=user-singer-song-user-song$$

...

由于网络的传播性,理论上关系路径可以很长,但关系路径越长,起点与终点的关联性越弱,该关系路径对应的边在现实场景中的可解释性越差。

基于关系路径的概念,我们可以根据其性质进一步定义一些相关概念:若给定一个关系路径,该路径的长度为1或该路径中经过的所有实体中没有相同类型的实体,则称其为简单关系路径.若一对关系路径 $P_1(S_1,D_1,R'_1)$, $P_2(S_2,D_2,R'_2)$ 满足 $S_1=D_2,S_2=D_1$,且 $R'_1^{-1}=R'_2$,那么 P_1 与 P_2 互为对方的逆路径.更进一步,关系路径可以归结为对称路径和非对称路径,当一个路径 P 等于其逆路径 P^{-1} ,我们称 P 为对称路径;反之, P 为非对称路径。

在异质网络中,元路径的定义已经在文献[20-22]中讨论过.文献[11]给出了关系路径集的定义,它将所有起点为 S 且终点为 D 的关系路径表示为一个关系路径集,并证明了一个长度为 n 的关系路径 P 可以被分解成两个子关系路径.这种定义方法虽然能够描述实体 S 和实体 D 之间的所有关联关系,但不能有效地指引聚合网络的形成,因此,本文对关系路径集进行了重新定义,并以此指导关系路径聚合网络的生成。

定义 3(关系路径集). 关系路径集是一组关系路径的集合:

$$PSet(T_p, R_p) = \{P_1(S_1, D_1, R'_1), P_2(S_2, D_2, R'_2), \dots, P_n(S_n, D_n, R'_n)\},$$

其中, $T_p = \{S_1, \dots, S_n, D_1, \dots, D_n\}$, $R_p = \{R'_1, R'_2, \dots, R'_n\}$, 并且满足如下条件:

- (1) 对于集合中任意的关系路径 $P_i(S_i, D_i, R'_i)$, 至少存在另一个关系路径 $P_j(S_j, D_j, R'_j)$, 使得 S_i 与 D_i 中的任意一个等于 S_j 或 D_j ;
- (2) 对于集合中任意两条关系路径 $P_i(S_i, D_i, R'_i)$ 与 $P_j(S_j, D_j, R'_j)$, 若 $S_i=S_j$, 则 $D_i \neq D_j$; 若 $S_i=D_j$, 则 $D_i \neq S_j$.

条件(1)保证了在给定关系路径集下的路径聚合网络的连通性,条件(2)限制了聚合网络中两个实体之间的关系(边)的唯一性.因此,任意一个关系路径集都能生成一个与其对应的聚合网络.在关系路径集的基础上,我们可以给出路径聚合网络的定义.

定义 4(路径聚合网络). 给定一个多维异质网络 $N=(V,E,T,R,A)$,在给定 $Pset(T_p,R_p)$ 的情况下,路径聚合网络可以被表示为 $N'=(V',E',T',R',A',W_{E'})$,其中: $T'=T_p$; $R'=R_p$; $V' \subseteq V$,且 $\forall v \in V', T(v) \in T$; E' 是 R_p 下边的集合; $A' = \{A'_1, \dots, A'_k\}$,对于 T' 中的每个实体类型 T_i 都有一个对应的维度集 $A'_i \in A'$; $W(e') = \Gamma_{E'}(E(u,v))$ 为 e' 的权重,其中, $\Gamma_{E'}(E(u,v))$ 为聚合函数.

例如,针对图 1 所示的网络,给定一个关系路径集 $Pset(\{album, singer, user\}, \{R(album, song) \circ R(song, singer), R(singer, user)\})$,该关系路径集中包含两个关系路径 $P_1=album\text{-}song\text{-}singer$ 与 $P_2=singer\text{-}user$.根据定义 4,可以将该路径集聚合为一个新的网络,聚合函数为 $count(*)$,如图 2(a)所示.

定义 5(维度聚合网络). 给定一个多维异质网络 $N=(V,E,T,R,A)$ 以及一个可以表示为 $A' = (A'_{i_1}, A'_{i_2}, \dots, A'_{i_j}, \dots, A'_{m_1}, \dots, A'_{m_m})$ 的维度约束,其中, A'_{j_i} 表示实体 T_i 对应的第 j_i 个维度, $1 \leq j_i \leq |A_i|$.每个 A'_{j_i} 的值等于 A_{j_i} 或者 *,表示对 A_{j_i} 维度进行约束或不对该维度进行约束.

约束 A' 对应的聚合网络是一个有权网络,可以表示为 $N'_A = (V'_A, E'_A, W_{V'_A}, W_{E'_A})$.其中,

- (1) 对于 $\forall v \in V$,在 A' 的约束下,我们能够得到一个新的聚合点集:

$$[v] = \{v \mid A'_{j_i}(u) = A_{j_i}(v), u, v \in V, i = 1, \dots, |T|, j_i = 1, \dots, |A_i|\}.$$

将得到的点集 $[v]$ 对应 V'_A 中的一个节点 v' ,表示 v 在 A' 的约束下获得的属性聚合节点,且 $W_{V'_A}(v') = \Gamma_{V'}(v')$.其中, $\Gamma_{V'}(*)$ 是一个节点聚合函数;

- (2) 对于 $\exists u', v' \in V'_A$, u' 对应了 N 中的一个顶点集 $[u]$, v' 对应了 N 中的一个顶点集 $[v]$,我们可以得到边集 $E(u', v') = \{(u, v) \mid u \in [u], v \in [v], (u, v) \in E\}$.如果 $E(u', v')$ 不为空,则有 $\exists e' = E(u', v') \in E'_A$. e' 的权重 $W_{E'}(e') = \Gamma_E(E(u', v'))$.其中, $\Gamma_E(*)$ 是一个定义在边上的聚合函数.

例如给定的网络 N 为图 2(a)所示的网络,给定的维度约束为 $A' = (A_{AlbumPrice}, *, A_{SingerGender}, *, A_{UserGender}, *)$,节点集合函数与边聚合函数都为 $count(*)$,得到的维度聚合网络如图 2(b)所示.

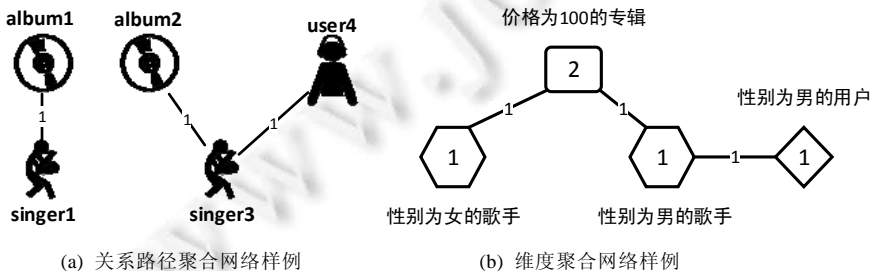


Fig.2 A sample of aggregate network

图 2 聚合网络样例

路径聚合网络反映了网络中实体节点之间的关系,通过路径聚合网络,我们能够挖掘出原始网络中并不能直接观察到的实体关系,例如图 2(a)所示的网络中专辑与歌手之间的联系.相应的维度聚合网络反映了网络中在属性约束下节点之间的联系,通过维度聚合网络,我们能够获得具有不同属性的节点间的关联情况,例如图

2(b)所示的不同性别的歌手与发行专辑价格之间的联系.路径聚合网络与维度聚合网络能够从一些新的角度探究原始网络中隐含的关联关系,很好地解决了引言中提出的问题.通常由于异质网络中往往具有大量的节点类型与维度属性,使得通过原始网络能够衍生出大量的路径聚合网络与维度聚合网络,因此,我们需要设计一个合理的图数据立方体来对这些聚合网络进行有效的管理.

传统的数据立方体将数据组织为维度表和事实表.一般而言,维度是数据的透视,而事实是各个维度的度量.针对网络数据,我们除了要考虑节点的维度信息,还需要考虑网络的拓扑结构.文献[11]中提出了一种名为 TSMH 的图立方体,它将网络数据组织为实体超立方体与维度立方体两种类型的立方体.每个路径聚合网络被组织为实体超立方体中的每个 cuboid,且针对每个关系路径聚合网络都有一个维度立方体与之对应.由于聚合网络的数量是不确定的,因此维度立方体的数量也是不确定的.这种设计方案是一种利用空间换二次查询时间的设计策略,尽管保证了二次查询的速度,但只适用于网络规模较小、结构简单的数据,一旦网络规模增大,模型将十分复杂,且带来的物化与存储压力将呈指数级增长.

针对这些问题,我们提出了一种路径-维度立方体的图立方体模型.路径-维度立方体尽管也将网络数据组织为两种类型的立方体,但与 TSMH 图立方体不同的是:TSMH 图立方体包含的是两种完整的立方体,立方体间通过聚合网络连接到一起;而路径-维度立方体中的两种立方体是一个完整的高维立方体在低维上的不同层次的映射.其中,路径立方体的每个边保存着网络中所有可能的简单关系路径,每个节点对应着一个实体的维度立方体.

路径-维度立方体只物化简单关系路径聚合网络,不再将解释意义较差的较长关系路径物化存储,若分析中需要对较长的关系路径分析,可以通过对简单关系路径的聚合计算得到.同时,由于不再针对聚合网络维护维度立方体,因此立方体的数量是固定的,更便于物化存储.这样的设计相较于 TSMH 图立方体,极大地节省了存储空间,更好地平衡了存储与计算.

图 3 给出了与表 1 样例对应的路径-维度立方体示例.

- 图 3(a)为样例对应的模式图;
- 图 3(b)为其对应的路径立方体,若给定的网络数据为强连通图,显然其对应的路径立方体为 n -完全图,其中, n 代表网络中实体类型的数量;
- 图 3(c)为每个用户实体与歌手实体对应的维度立方体,每个相连的 cuboid 可通过一定的计算进行转化.

在现实分析中,若希望得到一个聚合网络,首先在路径立方体中通过查询和聚合得到聚合网络中的边,再在维度立方体中根据维度约束得到节点聚合,进而得到完整的网络.例如,欲得到图 2(a)所示的网络,我们首先将给定的关系路径集中的每条路径进行物化, P_1 与 P_2 均为简单路径,可以直接在路径立方体得到对应的聚合网络,得到的聚合网络中包含 *album*,*singer*,*user* 这 3 种实体,各个实体不对维度属性进行限制,因此,在每个实体对应的维度立方体中直接查询节点与属性,即可得到待分析的网络.

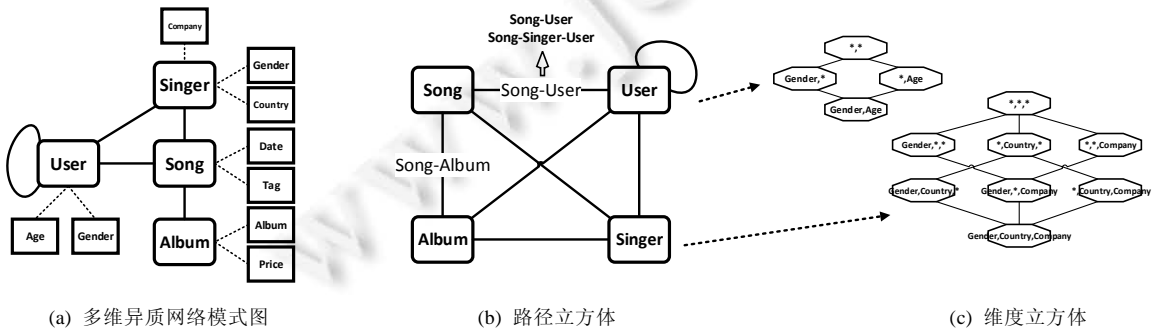


Fig.3 A sample of path-dimension cube

图 3 路径-维度立方体样例

1.2 路径-维度立方体物化

数据立方体的物化指数据立方体以及每个子立方体的存储与计算过程.本节将介绍路径-维度立方体物化与存储策略,并介绍相关物化算法.

1.2.1 立方体物化策略

立方体的物化可以减少重复计算,通常情况下,根据物化的程度,我们可以将物化分为不物化、部分物化以及完全物化.不物化即每次查询都需要对立方体根据查询需求进行计算,完全物化是将数据立方体中的所有数据全部计算好存储起来,这两种方法是物化策略的两个极端,无法应用于大规模数据集中.部分物化是将立方体中的部分数据计算存储起来,其余数据通过已物化的部分进行进一步计算得到.多维网络的数据量往往规模巨大,每一个维度有着复杂的层次,存储整个立方体需要海量的存储空间.研究与制定合理的物化与存储策略,可以使模型在处理大规模数据时,有效平衡存储空间与计算时间.

文献[11,19]中提出了图立方体的顺带物化策略,即:在物化过程中,首先在立方体中检查相关数据是否已经被物化过,如果物化过则直接利用,否则对该部分进行物化.应用顺带物化策略的立方体,初始状态下立方体全部数据都没有被物化,这样会导致初始查询的过程中计算时间较长.为了更好地平衡计算与存储,路径-维度立方体的物化策略选用顺带物化与部分物化相结合的策略.路径-维度立方体包含两种类型的立方体:路径立方体与维度立方体,针对不同类型的立方体,选择不同的物化策略.

路径立方体的物化过程主要是关系路径的物化过程,由于较长的关系路径可以分解成若干子关系路径,路径立方体的物化选择顺带物化的策略,即:在计算一个较长关系路径的过程中,将计算的中间过程中物化过的简单路径进行保存,其他计算过程中若再次应用到该段简单路径则可直接使用.例如,我们若物化一条关系路径 $P(user, user, R') = user-song-singer-song-user$, 可以将计算过程进行分解,将计算中得到的 $P(user, singer, R'') = user-song-singer$ 存储起来.事实上,对路径立方体中每条关系路径计算后的结果进行存储,并不需要保存一个完整的网络,只需要将其中连接情况即每条边存储起来,即:存储一张生成的边表,节点的维度数据可以在维度立方体中得到,避免数据的冗余存储.因此,路径立方体中的每条边对应着一组边表,每个边表为其对应的简单关系路径物化的结果.

维度立方体中的每个 cuboid 对应的是节点在当前维度约束下的聚合情况,即:每个 cuboid 保存的是若干组节点集合,每个集合对应的是在当前维度约束下的所有具有相同属性取值的节点集合.由于维度立方体的数量是不随物化的聚合网络的数量而改变的,且维度立方体的物化计算量相对于路径立方体物化的计算量较小,因此我们可以应用部分物化的策略来对维度立方体进行物化,即:通过预先计算出部分节点集合,后续相关的聚合计算在已得到的节点集合的基础上进一步计算得到.

路径立方体与维度立方体的物化过程中的相关算法在第 1.2.2 节与第 1.2.3 节中介绍.

1.2.2 关系路径物化

路径立方体的物化主要需要解决关系路径的物化问题,关系路径的物化过程为聚合关系路径形成新的边的过程.例如,假设存在如图 4 所示的异质网络,我们需要分析其中用户与用户之间的关联.观察如下的关系路径, $P(user, user, R') = user-song-singer-song-user$, 其中, $R = R_1 \circ R_2 \circ R_2^{-1} \circ R_1^{-1}$, R_1 和 R_1^{-1} 表示听过和被听的关系, R_2 和 R_2^{-1} 表示演唱和被演唱的关系.在 $user-song-singer-song-user$ 路径下,给定不同的聚合函数时,我们会得到不同的度量值.通常情况下,我们使用聚合函数 $count(*)$ 来计算不同路径的总数,用户之间在当前路径下的度量值可以表示为

$$P(user1-user1):count()=2, P(user1-user2):count()=0, P(user1-user3):count()=3, P(user2-user3):count()=4.$$

一条关系路径可以分解为多个子关系路径进行计算,子关系路径计算结束之后,我们对所有的子路径进行连接来获得完整的关系路径.因此,计算一个长度为 n 的简单关系路径,我们可以递归地将其划分为长度为 $n-1$ 的路径和长度为 2 的路径,并持续重复这一操作,直到 $n=2$ 时停止.对于复杂关系路径,我们可以将其划分为若干简单关系路径分别计算.

路径间的连接操作即边表间的 join 操作,这是相当耗费计算资源的过程.若不对其进行优化,每条路径的计

算时间会随着边表的规模增大而呈指数增长.对关系路径计算物化的优化,一方面要尽可能减少路径间的连接操作,另一方面要优化 join 操作的效率.因此,我们从两个方面提出了优化策略:

- 首先,从关系路径的定义出发,我们根据路径的对称性减少路径间的连接操作次数.

当一条路径可以被表示为 $T_1 \dots T_{k-1} T_k T_{k-1} \dots T_1$ 或 $T_1 \dots T_{k-1} T_k T_k T_{k-1} \dots T_1$ 时,那么这条路径为对称路径.对上述两种路径形式进行分析:第 1 种形式的路径长度为偶数,第 2 种形式的路径长度为奇数.对于偶数路径,我们可将其分解为两个对称的子路径.例如,路径 $A-B-C-B-A$,对其直接计算需要 3 次连接操作.若将其分为两个子路径 $A-B-C$ 和 $C-B-A$,且 P_1^{-1} 与 P_2 相等,因此,我们可以只计算 P_1 ,然后通过逆转 P_1 的源端和目的端来得到 P_2 ,这样只需要进行 2 次连接操作就可以完成整条路径的计算.对于偶数路径,例如路径 $A-B-C-C-B-A$,这种情况只发生在给定网络中的实体类型 C 存在自环关系的情况下,计算该路径需要 4 次连接操作.若将其分成 3 部分: $A-B-C, C-C$ 和 $C-B-A$, P_1^{-1} 与 P_3 相等,同理,逆转 P_1 的源端和目的端来得到 P_3 ,这样仅需 3 次连接操作就可以得到完整的关系路径.显然,若对称路径分解得到的子路径也为对称路径时,时间消耗更少.针对关系路径的时间消耗进行分析:当对称路径长度为 n 时,直接计算需要进行连接的次数为 $(n-2)$;但根据对称性进行优化后,最坏的情况下连接次数为 $(n-n\%2)/2$,因此,这种方法可以有效减少连接操作的次数,进而缩短计算时间.

- 其次,对 join 操作的效率进行优化.

join 操作可以通过 Map-Side Join 和 Reduce-Side Join 来实现.Map-Side Join 将一个远小于被连接数据集的数据设定为广播变量发送到每个节点^[14],接下来直接在 map 阶段进行连接操作.相反,Reduce-Side Join 在 reduce 这一阶段做连接操作,这需要一个相当耗费时间的 shuffle 过程.显然,Map-Side Join 能够有效减少 Join 操作的时间消耗.但由于集群中的每个计算节点都需要保存一份广播变量,因此该方法仅适用于选定数据集足够小、小到可以作为变量广播的情况下.为了有效降低路径连接所带来的时间消耗,我们可以根据连接操作中的数据大小去决定使用哪种连接策略.

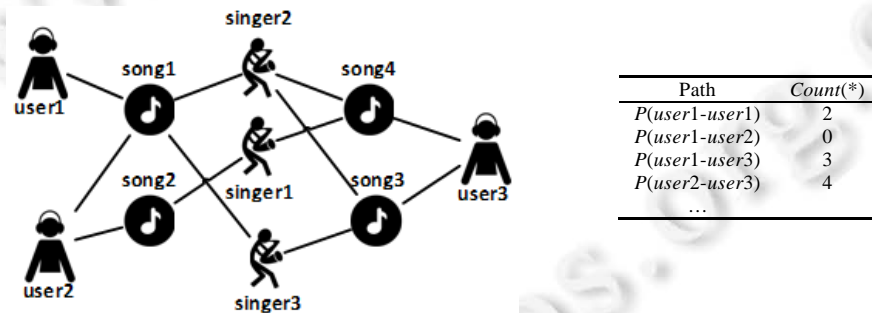


Fig.4 Path-Related materialization

图 4 关系路径物化

基于上述两方面优化策略,我们可以实现关系路径物化算法,见表 2.

- 对于每条关系路径,首先判断该关系路径是否是对称的:如果满足这个条件,我们就可以将路径的长度缩减为一半(第 2 行~第 4 行);
- 接着,递归地将长度为 n 的关系路径分为长度为 $n/2$ 和 $n-n/2$ 两部分,直到 $n=2$ 为止,然后将结果合成一批(第 5 行);
- 遍历过所有的关系路径与子关系路径之后,移除其中所有重复的关系路径,并且重新组织它们(第 7 行);
- 最后,对每个关系路径物化计算任务并行执行,并根据数据大小决定连接策略.新的边将作为关系路径聚合网络中的边加入其中(第 9 行~第 15 行).

Table 2 Path-Related materialization algorithm

表 2 关系路径物化算法

算法 1. Path-Related materialization algorithm.	
Input:	Multidimension Heterogeneous Network $G=(V,E,T,R,A)$;
Output:	Path-Related Aggregate Network.
1:	for each path p in relation path set do
2:	if $isSymmetric(p)$ then
3:	$p=p.half$
4:	end if
5:	$Batches.add(DecomposePathIntoBatch(p))$
6:	end for
7:	$batches=IntegrateAndRemoveDupBatches(batches)$
8:	for batch in batches do
9:	Do parallel execution for each $job(E_1,E_2)$ in batch as
10:	if $ E_1 \ll E_2 $ then
11:	$E_{12}=MapSideWeightedJoin(E_1,E_2)$
12:	else
13:	$E_{12}=ReduceSideWeightedJoin(E_1,E_2)$
14:	end if
15:	$E'.add(E_{12})$
16:	end for
17:	$V'=E.v, T'=PathSet.T, R'=PathSet.R$
18:	$A'=V'.dimensions$
19:	return $G'=(V',E',T',R',A')$

1.2.3 关联维度物化

关联维度物化基于维度聚合网络的定义,给定多维异质网络 $N=(V,E,T,R,A)$ 和关系路径集 $PSet(T)=(P_1, P_2, \dots, P_k)$,在关系路径集的基础上得到关系路径聚合网络后,我们需要列举出每个实体的所有可能的维度组合来得到所有的维度聚合网络集合。

维度立方体类似于传统的数据立方体结构,在多维异质网络中,每个节点类型都包含一组维度,所以需要对每个节点类型构建维度立方体,不同节点类型之间的维度立方体通过 Path Cube 中的关系路径集进行关联。但传统数据立方体物化和关联维度物化具有很大的区别:首先,我们不仅要根据维度约束合并实体节点得到属性聚合节点,还要根据聚合前实体节点间的关系对属性聚合节点进行连接,属性聚合节点之间的边是通过属性聚合节点内部的实体节点之间的联系继承得到的。其次,由于网络中往往包含丰富的实体类型与复杂的拓扑结构,因此相比于传统数据立方体,维度立方体往往呈现出更高的维度,对其完全物化的存储开销太过巨大。在对高维空间中的 OLAP 操作进行观察后发现,OLAP 操作只在一小部分的维度上进行。因此,我们的目标是设计一种具有一定预处理功能的半在线计算模型,以平衡空间和操作效率,即:在给定一个路径聚合网络的基础上,首先做一定的预计算,在此基础上就可以使用预计算得到的数据进行完全的在线查询。

传统的维度立方体构建方式大多围绕冰山立方体^[23]展开研究。冰山立方体是立方体部分物化技术中的一种,通过设置度量函数的阈值来避免不符合条件的物化。然而当涉及到高维数据时,冰山立方体的计算和存储开销仍然很大。另一种方法是去计算一个立方体外壳(shell),它预先计算给定大小下的所有维度的组合,其他的维度组合在立方体外壳的基础上实时计算获取。然而,它不能很好地支持高维度的 OLAP 操作。除此之外,外壳片段(shell fragment)^[11]方法也是一种有效的半在线计算策略,这种方法包括了两个部分:计算立方体外壳片段;用立方体片段做查询处理。外壳片段方法使用倒排索引的数据结构,这种数据结构在信息检索系统中非常普遍,该方法能够有效地处理高维数据,并且可以高效地对立方体进行在线计算。它的基本思路如下:给定一个高维度的数据集,我们把这些维度划分成多个不相交的维度片段集合,把每一个片段转化成相应的倒排索引,然后在倒排索引与 cuboid 之间建立联系,构建立方体片段。基于这些预先计算好的立方体片段,我们就可以动态地在线计算所需要的数据立方体中的每个 cuboid。

将外壳片段算法应用于异质网络,主要优势有:(1) 原始的方法是针对高维空间的,与多维异质网络的情况(有多种类别的实体与维度)一致。(2) 把所有维度划分为不相交的维度片段集合做各自的计算,这种方法能够把原先的串行计算过程转化成并行计算过程,以便于在分布式计算平台上计算,提高效率。

但传统的外壳片段算法针对的是关系数据,仅考虑了在不同维度约束下实体对应的度量值.直接将外壳片段算法应用于异质网络数据将不能解决网络中边的问题,不能生成维度聚合网络.因此,我们在针对关系型数据的外壳片段算法的启发下,不考虑实体的度量,而是针对维度约束后集合间的边,设计了一种针对多维异质网络的算法.

为了更好地解释这种方法,我们使用表 3 与表 4 展示的小数据集作为一个示例.表 3 列出了实体 P 、实体 V 及连接这两个实体的边.实体 P 有 3 个维度: A,B,C ,且共有 5 个节点,编号如 PID 列中所示.实体 V 有 2 个维度: D,E ,且共有 5 个节点,编号如 VID 中所示.表格中给出了 5 条边,编号如 EID 中所示.

首先,我们探讨如何对上述给定的数据构建倒排索引.对每个维度组合具体的属性值,列出节点属性中具有该值的所有节点编号.倒排索引的结果在表 4 中,左边的部分给出了实体 P 约束两个维度组合的元组,中间的部分给出了实体 V 的约束一个维度的元组.若我们设置外壳片段的阈值(一个片段中限定维度的个数的最大值)为 3,且将实体 P 和实体 V 的维度组合划分成一个片段,因此在针对维度片段(A,B,C)中,我们只计算 7 个方格 A,B,C,AB,AC,BC,ABC .在维度片段(D,E)中,我们只计算 3 个方格 D,E,DE .对于方体中的每个方格,都保留一个倒排索引,将相关联的边和实体记录下来.表 4 右边的内容展示了所有维度组合(根据上述集合的交集结果)和相关联的边.

Table 3 A sample entity P , V and relation in P - V

表 3 实体 P,V 与 P - V 关系表

EID	VID	PID	Weight	VID	D	E	PID	A	B	C
1	6	1	2	6	$d1$	$e1$	1	$a1$	$b1$	$c1$
2	9	3	5	7	$d2$	$e1$	2	$a1$	$b2$	$c2$
3	6	4	1	8	$d2$	$e3$	3	$a1$	$b1$	$c2$
4	7	2	1	9	$d2$	$e2$	4	$a2$	$b2$	$c1$
5	8	5	2	10	$d1$	$e3$	5	$a2$	$b2$	$c1$

Table 4 A sample inverted index for entity ID and edge ID

表 4 节点与边的倒排索引

Tuple(2-dim)	PID	EID	Tuple(1-dim)	VID	EID	Tuple1	Tuple2	Intersection	EID
$(a1,b1,*)$	{1,3}	{1,2}	$(d1,*)$	{6,10}	{1,3}	$(a1,b1,*)$	$(d1,*)$	$\{1,2\} \cap \{1,3\}$	{1}
$(a2,b2,*)$	{4,5}	{3,5}	$(d2,*)$	{7,8,9}	{2,4,5}	$(a1,*,c1)$	$(d1,*)$	$\{1\} \cap \{1,3\}$	{1}
$(a1,b2,*)$	{2}	{4}	$(*,e1)$	{6,7}	{1,3,4}	$(a2,b2,*)$	$(*,e3)$	$\{3,5\} \cap \{5\}$	{5}
$(a1,*,c1)$	{1}	{1}	$(*,e2)$	{9}	{2}				
$(a2,*,c1)$	{4,5}	{3,5}	$(*,e3)$	{8,10}	{5}				
$(a1,*,c2)$	{2,3}	{2,4}							

关联维度物化算法的伪码见表 5.我们首先得到所有的实体类型,对于每个实体类型,如果维度的总数比我们之前设置的阈值要高,那么就把它划分成若干独立的维度组,我们将这些维度组称为片段(第 1 行);否则,就把这个实体类型的所有维度直接放进一个组.然后,对于每个片段,扫描元组晶格(tuple lattice)^[24]并且从下向上构建每一种维度组合的倒排索引(第 7 行~第 9 行).事实上,从下向上和从上向下的策略都可以构建出所有的倒排索引,但考虑到计算效率和存储空间,我们更倾向于使用从下向上的策略.例如,我们想得到所有元组 $(a,b,*)$, $(a,*)$, $(b,*)$.如果我们从上向下建立,即先产生 $(a,*)$ 和 $(b,*)$,然后通过笛卡尔积获得 $(a,b,*)$,这意味着我们需要把每种可能的元组对组合起来.显然,在数据量很大、值很多的情况下,这种方式很不理想.假设维度 a,b 各有 10^4 个值,那么我们需要去计算 10^8 个 $(a,b,*)$ 对,并且计算出的这些 $(a,b,*)$ 并不一定都在数据中出现,因此大量的计算都是无意义的.相反,若先产生 $(a,b,*)$,元组 $(a,*)$, $(b,*)$ 可以由迭代所有元组与一些简单的键值变换而获得.接下来,对关系路径集中的每条路径,我们把边聚集成两个键值对的集合,每个键值对中键为这条边的实体标识符,值包含着所有相关边的标识符(第 13 行).对于每个倒排索引项,对其取并集操作,把每项中相关联的边的实体标识符组合起来(第 14 行~第 16 行).最终,我们把每个元组相关联的实体和边组合在一起,构成路径-维度图立方的片段(第 18 行).

Table 5 Graph Frag-Shell algorithm
表 5 针对图立方体的 Frag-Shell 算法

算法 2. Graph Frag-Shell algorithm.	
Input:	Path Aggregate Network $G=(V,E,T,R,A)$; $PathSet=\{p_1,p_2,\dots,p_k\}$;
Output:	Fragment P&D Graph Cube $S(S_1,S_2,\dots,S_n)$.
1:	for each type T_i in T do
2:	if $num(A_i)>threshold$ then
3:	partition dimensions $(A_i^1,A_i^2,\dots,A_i^n)$ into a set of k fragments $F_{i1},F_{i2},\dots,F_{ik}$
4:	else
5:	put dimensions $(A_i^1,A_i^2,\dots,A_i^n)$ into one fragment F_i
6:	end if
7:	for each fragment F_{ij} in F_i do
8:	scan the tuple lattice of F_{ij}
9:	build an inverted index entry $(tuple_t,list(V))$ for each tuple in Bottom-To-Up order
10:	end for
11:	end for
12:	for each path $P_i(S_i,D_i)$ in Path Set do
13:	aggregate edges into $(s_{ij},list(E))$ and $(d_{ij},list(E))$
14:	for each inverted index entry $(tuple_t,list(V))$ do
15:	union each v in $list(V)$ corresponding $list(E)$ as $(tuple_t,list(V))$
16:	end for
17:	end for
18:	combine $(tuple_t,list(V))$ with $(tuple_t,list(E))$

2 基于路径-维度立方体的 GraphOLAP 操作

给定一个路径-维度立方体,GraphOLAP 操作通过图立方体的指引,将网络在不同层次、不同粒度、不同角度之间进行转化.传统的 OLAP 操作有上卷、下钻、切片、切块等,但针对网络数据,GraphOLAP 的各种操作有着更为丰富的语义信息.传统的针对 GraphOLAP 操作的研究主要针对网络的上卷、下钻操作进行研究:文献[9,10,17,25]中,将网络划分为拓扑维与信息维分别进行上卷、下钻操作,但不支持异质网络.文献[19]引入了实体维的概念,并针对该维度设计了上卷、下钻操作.文献[11]中,针对实体立方体与维度立方体分别设计了上卷、下钻操作.尽管已有的研究中对上卷、下钻有着各自的定义,但分析目的大同小异,只是针对各自提出的立方体模型设计相关的操作.已有的研究中,对 GraphOLAP 新操作的提出研究还不足,不能满足日益丰富的分析需求.因此,本节针对路径-维度立方体设计并细化了 GraphOLAP 中的基本操作,并在此基础上设计了切片/切块与部分维度上卷/下钻等操作,丰富了对网络数据分析的角度.

2.1 上卷/下钻

上卷/下钻互为对方的逆操作,在这里,我们将二者共同讨论.在传统的 OLAP 中,上卷是指数据在维度的不同层次间变化,从细粒度向高层聚合.下钻是指数据从高层次向细粒度拆分.针对网络数据,网络中除了有实体的维度信息外,还包含着实体间的联系.单纯地在不同维度层次间进行聚集、拆分,显然不能满足我们对网络数据的分析需求,因此,我们针对实体关系与实体维度将上卷/下钻划分为关系路径上卷/下钻与关联维度上卷/下钻两种不同的类型.

2.1.1 关系路径上卷/下钻

关系路径上卷/下钻操作是针对网络中的实体关系.异质网络由于包含多种实体类型,不同实体之间相互连接.在对网络进行分析时,我们往往希望探究若干实体类型的节点间的关联,关系路径上卷/下钻操作将关系路径集进行聚集或拆分,进而得到一个关系路径聚合网络.关系路径上卷/下钻操作可表示为 $\langle PSet, M() \rangle$, 其中, $PSet$ 为关系路径集合, $M()$ 为聚合函数.关系路径上卷/下钻操作会导致网络的实体种类的数量与实体之间的关系发生变化,新的聚合网络不会产生新的节点,但聚合网络中的边为聚合函数聚集得到,因此会产生新的边,且边具有权重.关系路径上卷/下钻操作中对边进行查询可表示为 $\langle v_1, v_2, PSet, M() \rangle$, 其中, v_1, v_2 为待查询边的起始节点.关系路径上卷/下钻操作可通过对路径立方体的查询获得.

以图 1 所示的网络为例,图 1(b)展示的是网络在 $\langle\{user\text{-}song\text{-}user\},count(*)\rangle$ 与 $\langle user\text{-}user,count(*)\rangle$ 操作获得的聚合网络.且 $\langle user2,user4,\{user\text{-}song\text{-}user\},count(*)\rangle=2,\langle user2,user4,\{user\text{-}user\},count(*)\rangle=0$.

2.1.2 关联维度上卷/下钻

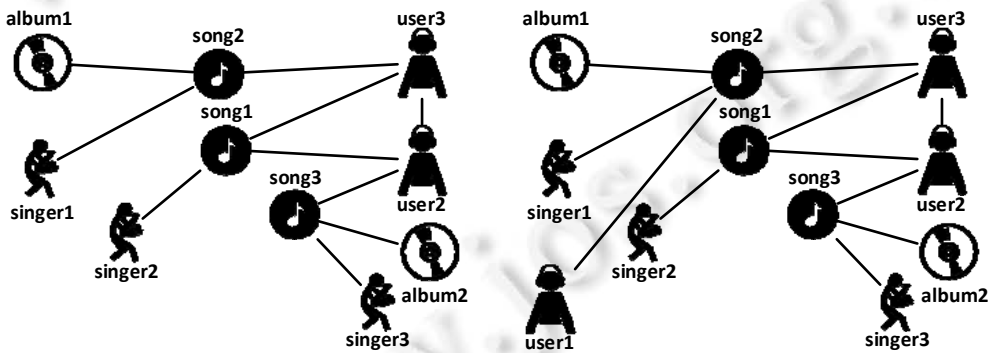
关联维度上卷/下钻与传统 OLAP 操作中的上卷/下钻类似,都是将数据按照维度进行聚集与拆分.不同的是,传统 OLAP 中的上卷/下钻得到的是聚集后的度量值,而 GraphOLAP 中的维度上卷/下钻得到的是聚集或拆分后的节点集合以及节点集合之间的关系,即关联维度聚合网络.关联维度上卷/下钻可以表示为 $\langle(\alpha_1,\alpha_2,\dots,\alpha_k),PSet,M()\rangle$,其中: $PSet$ 为关系路径集合且可为空,表示对原始网络进行维度上卷/下钻; $M()$ 为聚合函数; $(\alpha_1,\alpha_2,\dots,\alpha_k)$ 为维度集合,表示对这些维度进行约束.关联维度上卷/下钻会导致网络中的实体类型发生变化,即形成新的类型的实体,对这些实体进行查询可以表示为 $\langle A_i,(\alpha_1,\alpha_2,\dots,\alpha_k),PSet,M()\rangle$,其中, A_i 表示待查询的属性值(对应聚合网络中的一个节点);同时,聚合网络中也会通过聚合函数获得新的边,对这些边进行查询可以表示为 $\langle A_i,A_j,(\alpha_1,\alpha_2,\dots,\alpha_k),PSet,M()\rangle$,其中, A_i,A_j 为某一维度的某一个属性值, A_i 与 A_j 之间的边即待查询边.对应路径-维度立方体,关联维度上卷/下钻操作可通过对维度立方体的查询获得.

同样以图 1 所示的网络为例,图 1(c)所示的是 $\langle(Gender),\{user\text{-}song\text{-}user\},count(*)\rangle$ 与 $\langle(Gender),\{user\text{-}user\},count(*)\rangle$ 操作得到的聚合网络,其中,

$$\langle man,(Gender),\{user\text{-}song\text{-}user\},count(*)\rangle=2,\langle man,women,(Gender),\{user\text{-}song\text{-}user\},count(*)\rangle=4.$$

2.2 切片/切块

在传统的 OLAP 中,切片操作选择维中特定的值进行分析,即:通过对特定维度进行约束,对得到的数据进行分析.与之类似,切块操作选择维中特定区间的的数据或者某批特定值进行分析.在 GraphOLAP 中,这种通过约束维度对数据进行切片或切块的方法可以得到网络中的一个子图.例如,针对图 1 所示的网络,对用户的性别维度进行切片,我们切取性别属性为女的用户节点,则删除网络中所有性别不为女的用户节点以及与其相连的边,如图 5(a)所示;类似对于切块操作,我们切取网络中年龄段低于 30 岁的用户节点,则删除网络中所有年龄不在该区间的用户节点以及与其相连的边,如图 5(b)所示.



(a) 性别为女的用户网络关系(切片) (b) 30岁以下的用户关系网络(切块)

Fig.5 A sample of slice/dlice operation

图 5 切片/切块操作样例

2.3 部分维度上卷/下钻

第 2.1 节中的上卷/下钻操作得到的聚合网络,相同类型的节点都在相同的层次中.现实的分析中,我们通常会关注网络中某一个或某一类节点,探究与之相连的节点之间的关系以及与之相连的节点属性关系.以图 1 所示的网络为例,在通过 $user\text{-}song\text{-}user$ 关系路径得到图 1(b)后,我们想探究 $user3$ 与不同性别的人的连接情况.我们选择与 $user3$ 相连的节点进行维度上卷,可得到图 6 示的网络.

为了满足这种分析,我们提出了部分维度上卷/下钻操作.不同于普通的上卷/下钻,部分维度上卷/下钻允许

对网络中的部分节点按照维度约束进行聚集,从而得到一个新的聚合网络.该聚合网络中,允许相同类型的实体节点处于不同的层次中.部分维度上卷/下钻操作可以表示为 $((\alpha_1, \alpha_2, \dots, \alpha_k), VSet, PSet, M())$,其中, $VSet$ 为待操作的节点集合.网络中的节点集合 V 中,除 $VSet$ 外的节点不进行聚合.

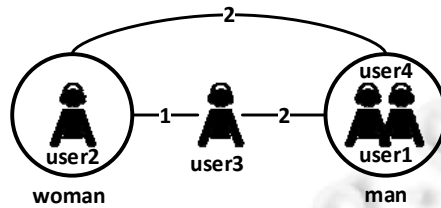


Fig.6 A sample of part-roll-up/part-drill-down operation

图 6 部分维度上卷/下钻操作样例

3 基于路径-维度立方体的大规模多维网络分析框架

为了支持对大规模多维网络的分析,我们基于 HDFS(hadoop distributed file system)与并行计算框架设计了基于路径-维度立方体的 GraphOLAP 大规模多维网络分析框架,整体架构如图 7 所示.

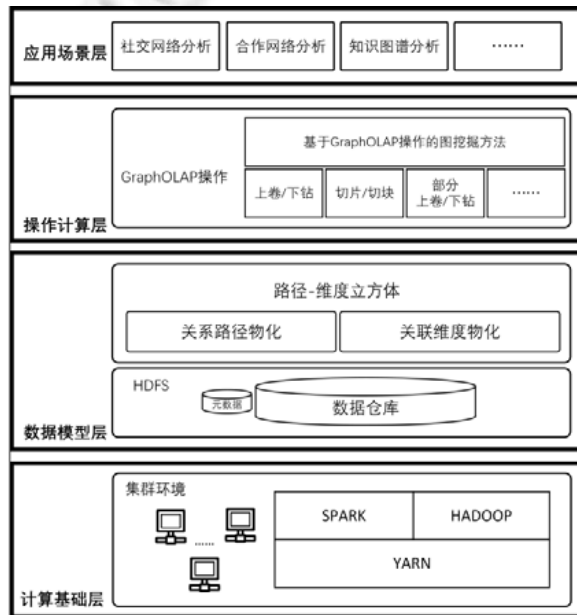


Fig.7 Framework for large-scale multidimensional network analysis architecture

图 7 大规模多维网络分析框架架构图

框架可以分为 4 个层次.

- 计算基础层

该层主要为上层相关算法的运行提供计算资源,并对资源进行管理和调度.Yarn 在 Hadoop 的生态系统中担任了资源管理和任务调度的角色,在该层中,我们使用 Yarn 对集群资源进行管理,并在此基础上搭建了 Spark 与 Hadoop 框架.上层的物化、聚合、操作、查询等相关算法将被提交到该层中运行,Yarn 对资源进行统筹管理与调度.

- 数据模型层

该层主要对网络数据进行存储与建模,并维护一个路径-维度立方体.该层次中,我们在 HDFS 上构建了一个

数据仓库对数据进行管理,其中,元数据维护了网络中的实体关系与维度信息.在数据仓库之上,我们构建了一个路径-维度立方体.在立方体中,将数据组织为边表与维度表,每个表在 HDFS 上存储为单独的文件,立方体间的关联关系以文件的形式存储在 HDFS 中,通过文件路径映射将各个表关联起来.立方体的物化策略在第 1.2 节中已详细介绍,物化过程分为两类:关系路径物化与关联维度物化,分别对应路径-维度立方体中的路径立方体与维度立方体.物化的相关算法需要并行计算框架的支持,因此,物化中的任务将提交给计算基础层进行计算.物化后的聚合网络通过数据仓库存储在 HDFS 上,路径-维度立方体对各个聚合网络之间的关联关系进行维护.

- 操作计算层

该层主要实现了 GraphOLAP 中的操作,并在此基础上实现更为复杂的图挖掘功能.该层次中,我们基于路径-维度立方体模型实现第 2 节中提出的 GraphOLAP 操作.对网络进行 GraphOLAP 操作,一方面需要对路径-维度立方体进行检索,另一方面也需要对网络进行一定的聚合计算.因此,GraphOLAP 操作首先对路径-维度立方体进行检索,再将聚合计算提交到计算基础层进行计算,以提高计算效率.于此同时,该层次也在 GraphOLAP 基本操作的基础上将相关操作进行结合,以实现更为复杂的图挖掘功能.

- 应用场景层

该层主要实现对真实场景中的多维网络数据分析.现实场景中的数据,应用网络构建技术构建为多维网络,得到的网络针对不同场景中的分析需求,应用 GraphOLAP 操作,从不同角度、不同维度、不同粒度对多维网络进行分析,进而挖掘出深层次的信息,以满足应用场景中的分析需求.

4 实验

现实世界中,大量的数据都可以组织成多维网络,但近年来针对 GraphOLAP 的研究工作大多针对科研合作网络与电影合作网络进行分析,应用场景较为单一.本节针对多个真实的数据集从多个角度进行分析,在证明我们提出的框架具有一定分析能力的同时,也借此丰富 GraphOLAP 的应用场景.同时,我们也应用了大规模数据集测试了图立方模型和相关算法.

实验的环境为 32 个节点组成的本地集群,每一个节点由 1 个 Intel® Xeon E5-2620, 2.3GHZ 的 CPU、64GB 内存和 2 个 500GB 的 SATA 硬盘组成,操作系统为 Centos 6.0.相关算法基于 Hadoop 2.6.0 和 Spark 1.5.1 实现.

4.1 视频社交网络分析

随着多媒体技术的发展,现实生活中存在着丰富的视频数据,其中包含着大量可挖掘的信息.本实验中分析的网络,从电视剧《甄嬛传》中构建得到^[26].构建方法如下:我们首先对该剧中出现的人物角色进行识别,作为网络中的节点;若多个人物同时出现在一个场景中,在他们之间构建节点间的边,人物之间共同出现次数作为边的权重;为了防止网络过于稠密,我们设置一个阈值过滤掉权重过低的边.经过上述步骤后,我们得到一个完整的网络,该网络为同质网络,即,网络中只有人物一种实体类型.对网络中的每个节点,我们抓取了百度百科中的词条,补充了每个人物的派系与民族两个维度,其中,派系维度用 D_p 来表示,民族维度用 D_n 来表示.处理后的网络中包含 29 个节点,98 条边,每个节点包含 2 个维度 D_p 与 D_n .

实验中,首先针对原始网络构建路径-维度立方体,由于网络为同质网络,即只有一种实体类型,因此,Path Cube 中只有一个 cuboid 且不存在边,我们针对这个网络主要从多维度进行分析.图 8 左边所示的是原始网络.我们首先分析了与“皇上”相连的派系之间的关系.根据分析目标,我们对网络进行了部分维度上卷,保留了“皇上”节点并将周边节点沿派系维度进行上卷.图 8 右侧上方的图展示了部分上卷后的结果,聚合函数为 $count(*)$,其中,边的权重为上卷过程中聚合后的结果.从图中我们可以清楚地看出:“甄嬛派”、“皇后派”与“皇上”之间的联系较为紧密;同时,“甄嬛派”与“皇后派”之间存在着错综复杂的关系.

接着,我们分析了不同派系、不同民族之间的关系.根据分析目标,我们对网络沿派系与民族两个维度进行上卷,即 $\langle(D_p, D_n), \{\}, count(*)\rangle$.上卷后的关系图如图 8 右侧下方的网络所示,从图中,我们可以看出约束派系与民族维度后的聚合节点之间的关系.

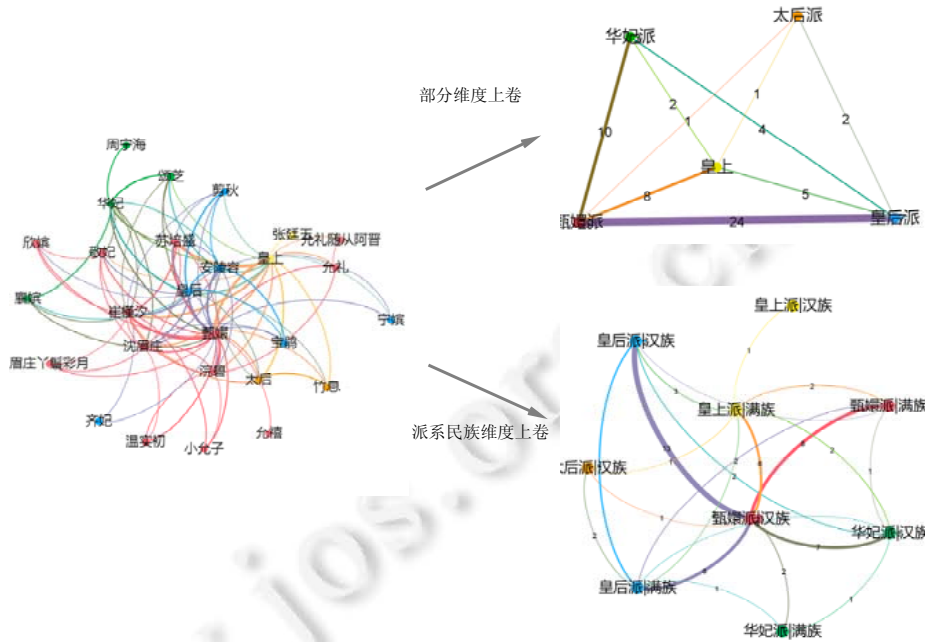


Fig.8 A sample of video network analysis

图 8 视频社交网络分析样例

4.2 知识图谱网络分析

知识图谱^[27]是一种特殊的多维异质网络,其初衷是为了提高搜索引擎的搜索能力,增强用户的搜索体验.随着智能信息服务应用的不断发展,知识图谱已被广泛应用于搜索与问答系统^[28]、个性化推荐系统^[29]等.在一个知识图谱网络中,节点代表一个实体,且通常具有多维属性,节点与节点之间的边代表实体间的某种关系.传统的针对知识图谱的查询通过网络中直接相联的关系进行,并不能对网络进行变换.本实验中,我们应用 GraphOLAP 的相关技术对知识图谱网络进行分析,以挖掘更为复杂的实体关系.

实验的数据集为面向电影领域构建的知识图谱网络,数据来自于在线的电影数据库^[30]中 2007 年~2017 年间上映的电影信息以及编剧、导演、主要演员的个人信息.在该知识图谱中,实体包含电影(M)、编剧(S)、导演(D)以及演员(A)这 4 种类型,其中,电影包含电影名称、上映年份、发行公司、类型等多个维度的属性,演员包含演员姓名、出生年月、籍贯等多个维度的属性,其余两种类型的实体节点也包含与其相关的多维属性.4 种实体之间的关系如图 9 所示,电影与演员之间的边表示该演员参演了这部电影,电影与导演之间的边表示该导演指导了这部电影,编剧与电影之间的边表示该编剧为这部电影编写了剧本.

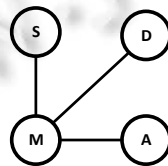


Fig.9 The schema of knowledge network

图 9 知识图谱网络实体模式图

该网络中共包含 24 005 个节点,其中电影与演员之间存在 28 543 条边,电影与编剧之间存在 3 752 条边,电影与导演之间存在 4 741 条边.对该网络构建路径-维度立方体模型,在此基础上应用 GraphOLAP 的相关操作,可以满足传统查询所不能解决的分析需求,以挖掘出一些潜在的有价值信息.

例如,若分析导演与演员之间的合作关系,探究哪些导演与演员之间有着频繁的合作.原始网络中不存在导

演与演员直接相连的边,显然,上述分析需求的结果不能直接从当前网络中获得.但我们可以构建出路径 $A-M-D$ 关系,该关系直接反映了导演与演员之间的合作情况.为得到该条关系路径对应的边,首先需要构建关系路径 $P=A-M-D$,在这条关系路径的基础上对网络进行关系路径上卷操作,即 $\langle\{A-M-D\},\text{Count}(\ast)\rangle$.关系路径上卷后我们将得到一个新的聚合网络,该网络反映了导演与演员之间的合作情况,边的权重表示了合作的次数.通过对网络进行查询,我们可以得到 2007 年~2017 年间导演与演员之间的合作关系,见表 6.类似的分析需求,例如探究演员与演员之间的合作关系,我们也可以通过关系路径上卷操作得到演员与演员之间的关系网络,图 10 展示了沿关系路径 $P=A-M-A$ 上卷得到的聚合网络中的一部分,不同的节点表示了不同的演员,节点间的边表示两名演员间发生过合作的关系,边的权重表示合作的次数.

Table6 Relationship between director and actor
表 6 演员与导演关系表

Director	Actor	Weight	Director	Actor	Weight
邓衍成	释小龙	10	冯小刚	李晨	4
贾樟柯	赵涛	6	吴宇森	金城武	4
郭敬明	杨幂	5	陈木胜	古天乐	4
宁浩	黄渤	5	冯小刚	张涵予	3
...

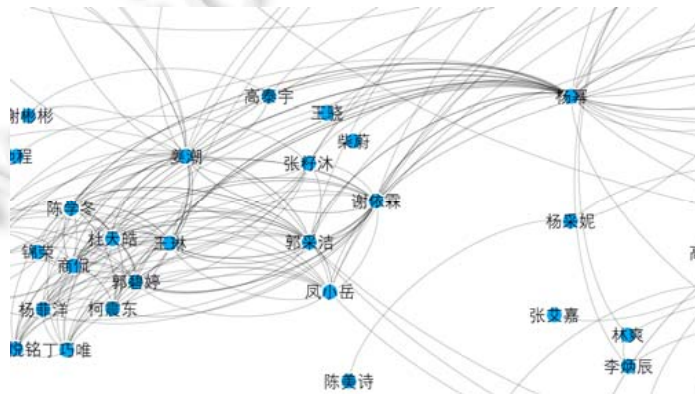


Fig.10 Network obtained through entity-roll-up operation
图 10 关系路径上卷操作得到聚合网络

在图 10 所示的聚合网络的基础上,我们还可以通过进一步的操作挖掘更深层次的信息.例如探究不同籍贯的演员之间的合作情况,可以对图 10 所示的聚合网络,约束每个节点的籍贯维度进行关联维度上卷,聚合函数选择 $\text{Sum}(\ast)$,即 $\langle(\text{Homeplace}),\{A-M-A\},\text{sum}(\ast)\rangle$,得到一个新的聚合网络.图 11(a)中展示了聚合网络的一部分,其中每个节点为籍贯是该地区的演员节点聚合形成的节点,节点间的边则表示两个地区之间演员的合作次数,通过对生成的聚合网络进行查询即可得到不同籍贯的演员间的合作情况.

上述的这些分析方法与分析角度是传统知识图谱所不擅长的,基于 GraphOLAP 的多维网络分析技术丰富了原始知识图谱网络中的实体类型与关系类型.因此结合多种操作,我们可以针对知识图谱中的某一特定实体进行关系刻画,从不同的角度挖掘其关联情况.例如图 11(b)展示演员黄渤的人物关系画像.

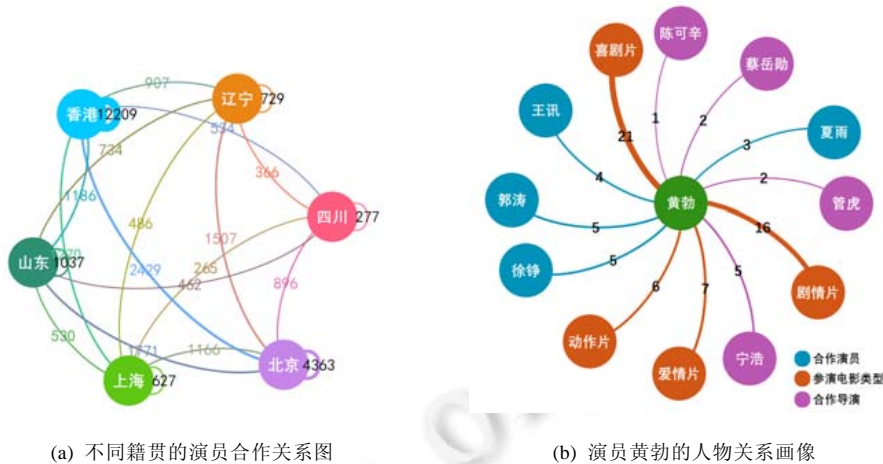


Fig.11 Samples of knowledge network analysis

图 11 知识图谱分析样例

4.3 物流网络分析

近些年,电子商务的飞速发展促进了物流行业的飞速发展.每天有大量的货物在全国各地运输,这其中蕴含着海量的物流数据.物流数据中主要为运单数据,运单数据中包含每个快递的寄件地址与收件地址,通过收寄件地址,我们可以构建出一个物流网络,通过对物流网络的分析,可以帮助快递企业调整路由、规划网点选址、合理分配运力、降低运输成本.运单数据通常涉及个人隐私以及商业机密,不宜公开.因此,我们根据国家邮政局公开的邮政行业运行情况中各省份各季度的邮件量百分比,模拟生成了两万条运单数据,并对其进行分析.本实验中分析的网络通过对这两万条运单数据进行提取以及构建得到.首先,提取运单数据中所有的收/寄件人地址中的区/县作为网络中的节点,该区/县所属城市与所属省份作为其维度.再将每一条有效的运单数据构建为网络中的边,其收/寄件人地址分别是边的端节点.

经过上述步骤后,我们得到了一个同质网络,网络中共有 482 个节点、20 000 条有效边,其中,每个节点包含两个维度,即,其所属省份及城市.在实验中,我们首先按照寄件时间进行切片,得到 4 个季度的网络,分别构建路径-维度立方体;接着,对网络中的节点按照其所属的省份维度进行上卷操作,图 12 为按照省份维度上卷操作后的结果,聚合函数为 $count(*)$,每条边的权重是不同的省份之间的物流量,用边的粗细进行表示.

从图中我们可以看出全年各个季度省份间的物流变化:每年在春、秋两季物流量最小;而由于京东“618”、淘宝“双十一”等网购节日的原因,夏季与冬季的物流量明显增大;同时,北京作为重要的物流中心,各个季度物流量都十分大.

物流数据类似人流迁徙网络,应用同样的方法也可以对迁徙数据进行分析,从不同的层次下分析迁徙网络的变化.

4.4 科研合作网络分析

科研合作网络的构建,我们使用了 MAG(微软学术图)数据集^[31].MAG 数据集是典型的多维异质网络,包括科研文献的出版发行信息、文献之间的引用关系、文献的作者、机构、会议、研究领域,包括了 126 909 021 篇论文(paper)、114 698 044 位作者(author)、53 834 个机构(institution)、19 483 个会议(venue)和 113 644 个关键词(keyword).实体间的关系模式图如图 13 所示.

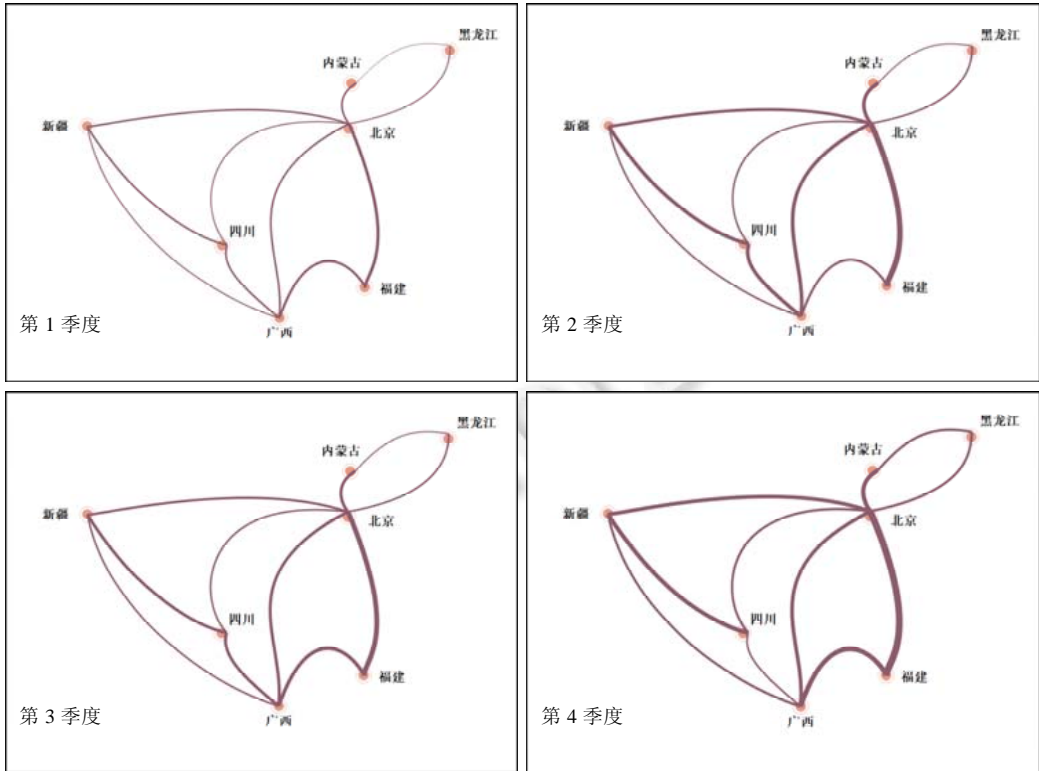


Fig.12 Samples of logistics network analysis

图 12 物流网络分析样例

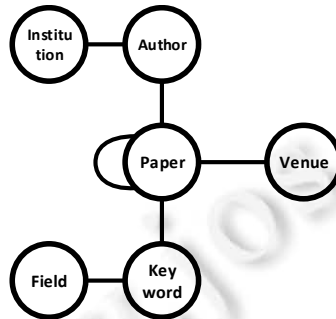
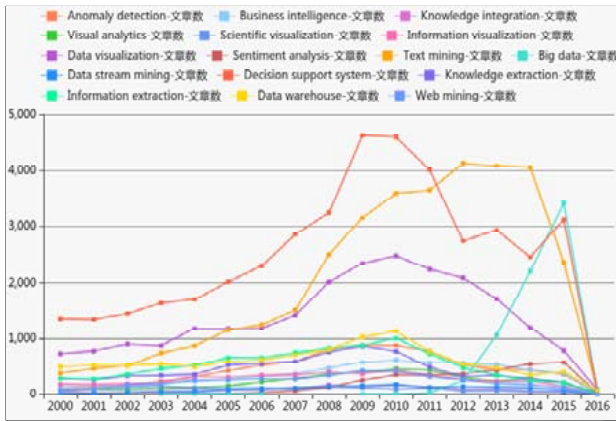


Fig.13 Schema of MAG dataset

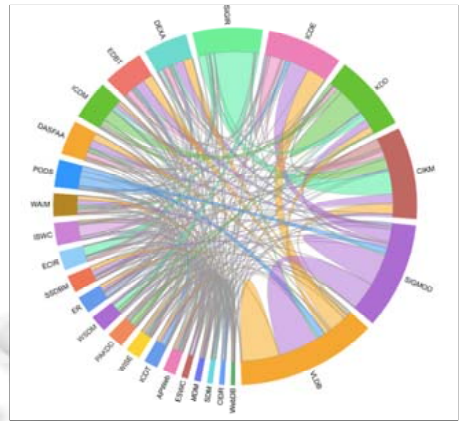
图 13 MAG 数据集关系模式图

针对 MAG 数据集,我们将每个实体作为网络中的节点,实体之间的关系作为网络中的边构建网络,并针对该网络构建路径-维度立方体,通过一定的 GraphOLAP 操作,能够从中挖掘出有价值的信息.

首先,探究各个领域近年来的发文情况.选择关系路径 $P=Paper-Keyword-Field$,并进行关系路径上卷得到 Paper-Field 聚合网络,聚合函数为 $Count(*)$;然后,再对 Paper 实体按照年份维度进行关联维度上卷,聚合函数为 $Count(*)$,上卷的过程中相同年份的 Paper 节点会聚集在一起;接下来对每个 Field 与 Paper 的年份聚集节点进行查询,进而得到每个领域的发文数量.图 14(a)展示了各个领域 2000 年~2016 年的发文数量变化.



(a) 2000 年~2016 年各领域发文情况



(b) 各个会议之间的关系图

Fig.14 Schema of MAG dataset

图 14 MAG 数据集关系模式图

我们可以看出:大数据领域自 2012 年起呈爆发式增长;文本挖掘、信息提取、数据可视化领域在 2007 年后开始迅速增长;其他领域的增长趋势相对稳定。

接着,我们分析了会议与会议之间的论文引用情况.首先选择关系路径 $P=Venue-Paper-Paper-Venue$ 进行关系路径上卷,得到 $Venue-Venue$ 关系网络,聚合函数为 $Count(*)$;接着,对网络中的任意两个节点对进行查询,得到二者之间的关系.图 14(b)展示了会议之间的关系图,我们选择了被引用次数较多的会议,将他们之间的关系绘制为弦图.从中我们清晰地看出:会议的等级越高,会议中的论文被引用的次数越多,越高等级的会议之间发生论文引用的情况越高。

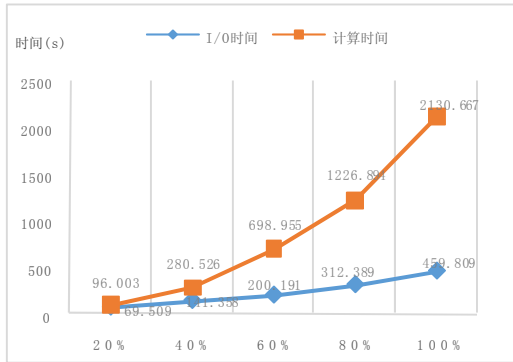
4.5 规模与性能实验

本节评估了关系路径物化算法与关联维度物化算法的性能和效率,算法的伪码在第 1 节中说明.在这个实验中,我们使用了第 4.3 节中提到的 MAG 数据集.原始网络中,机构与作者之间有 16 746 514 个边、作者与论文之间有 231 817 035 个边、论文与地点之间有 30 724 776 个边、论文与关键词之间有 104 830 261 个边、关键词与领域之间有 111 400 个边,数据集大小约 70G.

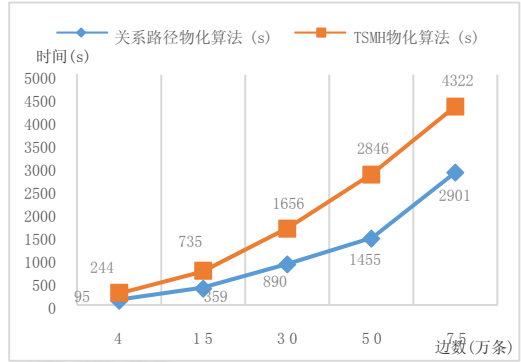
我们选择两个关系路径进行物化,并针对这两条关系路径组成的关系路径集生成聚合网络.关系路径集为 $PSet=\{Institution-Author-Paper, Institution-Author-Paper-Author-Institution\}$.我们分别从原始异构网络中随机抽取 20%,40%,60%,80%,100%的边作为关系路径物化算法的输入.图 15(a)展示了在集群中运行的节点大小为 4 时,不同规模的网络中关系路径物化算法的 I/O 时间和计算时间。

我们可以观察到:随着网络规模的增大,I/O 时间几乎呈线性增长,计算时间近似 $O(n^2)$ 增长.出现这种现象的原因是,计算时间主要取决于进行连接操作的边规模的大小.当被连接的两个边集的规模都加倍时,消耗的时间将是未加倍时时间消耗的 4 倍。

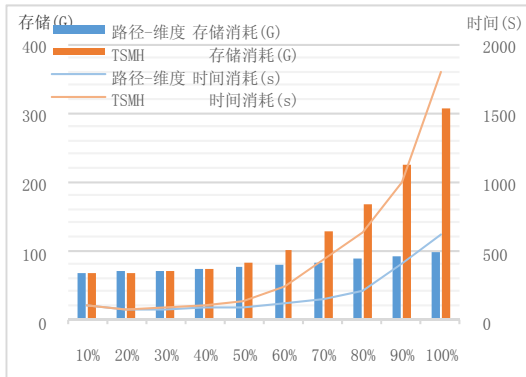
文献[11]中,针对关系路径的物化也提出了相应的物化算法,因此,我们比较了提出的路径相关物化算法与 TSMH 中的路径物化算法在生成相同数量的边时的时间消耗差异.TSMH 中的关系路径物化算法较为朴素,其仅仅将关系路径进行分解为长度为 1 的多条子关系路径并进行连接,对已物化的较长路径片段不可复用.图 15(b)显示了在数据不同数量的边时,两种物化算法的时间消耗.实验结果表明,我们的算法平均比 TSMH 中的路径物化算法快 30%左右.尽管当数据规模足够大时,关联操作将会退化为 Reduce-Side Join,但由于对称路径的优化,时间消耗依旧低于 TSMH 中的路径物化算法。



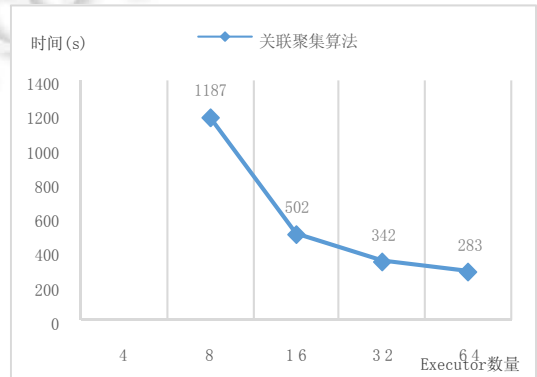
(a) 关系路径物化算法 I/O 时间与计算时间对比



(b) 关系路径聚集算法与 TSMH 路径物化算法计算时间对比



(c) 路径-维度立方体与 TSMH 立方体物化时间与存储消耗对比



(d) 关联维度聚集算法时间与并行程度关系

Fig.15 Result of efficiency experient

图 15 性能实验结果

同时,我们也比较了 TSMH 立方体与路径-维度立方体的物化过程在存储消耗上的差异.假设我们的分析需求会议与会议之间的关联关系,关系路径为 Venue-Paper-Author-Paper-Venue.我们对该分析需求进行处理,即:物化 Venue-Paper-Author-Paper-Venue 关系路径,探究在物化后立方体的存储空间上的占用情况与时间消耗.我们分别选择了不同规模的数据集进行了测试,实验结果如图 15(c)所示.可以看出:随着数据规模的增大,TSMH 立方体的存储占用近指数增长,而路径-维度立方体的存储占用增长缓慢.这是因为 TSMH 立方体将整条关系路径都进行了存储,而路径-维度立方体只存储计算中的简单路径,即 Venue-Paper-Author;同时,路径-维度立方体的物化时间也比 TSMH 的时间快很多,其原因是路径-维度立方体的物化过程中对关系路径物化计算的优化提高了物化的性能.

接下来,我们测试了关联聚集算法的性能,实验主要研究并行程度对物化时间的影响.

我们在 Spark 的 yarn-cluster 模式上运行算法.Spark 相关参数见表 7.

Table 7 Parameters of Spark

表 7 Spark 相关参数

Spark 参数	值
executor-core	6
executor-meory	20G
driver-meory	4G
spark.default.parallelism	100

在 Spark 框架中,任务将分配给每个 executor 来执行,每个 executor 可以调用指定规模的处理核心与内存资

源.因此,我们可以通过改变参数 `num-executor` 来改变程序的并行程度.在其他参数不变的情况下,该参数越大,参与计算的处理器核心数目与内存资源越多.图 15(d)展示了在不同个数的计算单元下,MAG 数据集上维度立方体物化的时间.从实验结果我们可以看出:

- (1) 当 `executor` 数目等于 4 时,由于内存不足的错误,作业失败,因此没有时间显示;
- (2) 当 `executor` 数目增加时,物化时间会减少;
- (3) 随着 `executor` 的增加,并行带来的性能提高逐渐减少.

对上述结果进行分析:当 `executor` 数目减少时,程序的并行程度降低且分配到的计算资源随之减少;反之,程序的并行程度增大,分配到的资源增加.并行程度增大会加快数据的处理速度,但随着并行程度的增大,作业调度时间和 I/O 时间会极大地降低并行程度增大所带来的性能增益.因此在实际应用中,我们需要根据数据集规模大小来确定集群的规模,以避免计算资源浪费.

5 相关工作

5.1 图立方体模型

在传统的数据库和 OLAP 技术^[6-8,32,33]中,数据立方体是一种被普遍使用的强大工具.然而,由于传统的数据立方模型无法很好地处理网络数据中的拓扑结构,因此不能很好地对多维网络数据进行分析.

Zhao 等人^[9]提出了一种针对多维网络的图立方模型,该模型是从传统数据立方体的概念迁移而来,仅仅考虑网络中的边,无法处理网络中不同实体间的联系.

Wang 等人^[10]提出了一种超图立方模型,该模型将节点和边在不同层次进行聚集,使得针对同质网络的图立方体模型研究更进了一步.

Beheshti 等人^[25]提出了图数据模型 GOLAP.

上述提到的立方模型仅针对同质网络,不能适用于异质网络.

Yin 等人^[19]提出了 HMGraph Cube,该模型在拓扑维、信息维的基础上提出了实体维的概念,解决了针对异质网络的分析问题,然而这种立方体模型仅仅把不同实体的属性组合在一起.

Wang 等人^[11]针对异质网络提出了 TSMH 图立方体模型,它以元路径作为指导,将图立方体分为实体超立方体和维度立方体两部分.该模型将两个实体间的不同关系路径组织在一个方体中,应用同样的聚合函数,但无法处理复杂的网络聚集操作.

除此之外,Hannachi 等人^[34]与 Rehman 等人^[35]也提出了一些针对社交网络的立方体模型.

5.2 立方体模型的物化技术

传统的立方体物化技术中,大量研究在 Hadoop 上实现相关算法,应用 MapReduce 计算框架来提高立方体物化算法的效率^[36].但由于针对传统立方体,这些物化算法并不能直接运用于图立方的计算.

Li 等人^[18]提出了一种适合 GraphOLAP 的双星模型,探讨了信息维与拓扑维的聚集算法,但并没有深入讨论立方体的详细物化过程.

Zhao 等人^[9]在传统立方体物化技术的基础上,基于提出的模型探讨了图立方体的物化策略.

Yin 等人^[19]针对 HMGraph Cube 设计了相关物化策略,并应用维度编码技术对物化过程进行优化.

Wang 等人^[11]针对提出的 TSMH 图立方体模型,采用随用随物化的策略,提出了一种节点类型与维度的编码技术,减少了维度立方体物化过程中重复的连接操作.

在数据立方体的部分物化技术方面已存在一定的研究.

Gupta 等人^[7,8]最早对传统立方体的部分物化技术进行了探讨.

Fang 等人^[23]在研究中提出了冰山立方体的概念,并探讨了其对应的计算策略.

Li 等人^[37]将倒排索引应用于立方体物化中,设计了立方体外壳片段的部分物化算法.

5.3 GraphOLAP相关操作

针对传统的 OLAP 研究已有近 20 年的历史,技术相对较为成熟,但针对网络数据的 Graph OLAP 技术的研究尚处于起步阶段.

Chen 等人^[17]首次提出了 GraphOLAP 的概念,并定义了信息维、拓扑维以及这两个维度的上卷、下钻、切片、切片的操作,但并没有对整个 GraphOLAP 的系统框架进行概述.

Li 等人^[18]在提出的双星模型的基础上,基于 RDBMS 实现了信息维与拓扑维的上卷下钻操作.

Zhao 等人^[9]将方体查询引入到图数据立方体上,将 GraphOLAP 操作转化为 crossboid 查询.

Yin 等人^[19]在 HMGraph Cube 的基础上提出了旋转、拉伸等操作,丰富了 GraphOLAP 操作的类型.

Wang 等人^[10]提出了边维度属性的上卷、下钻操作,并在 Hadoop 框架上实现了相关算法.

Wang 等人^[11]基于提出的 TSMH 图立方体模型,提出了实体立方体的上卷、下钻操作以及维度属性转换操作,并在 Spark 框架中实现了相关算法.

Beheshti 等人^[38]基于 Hadoop 设计了 P-OLAP 框架,并应用 BP-SPARQL 查询来描述 GraphOLAP 操作.

5.4 分布式图并行计算框架

在大规模数据挖掘与机器学习应用的推动下,基于图结构来描述数据及数据之间的关系分布式图并行计算框架逐渐得到研究人员的关注,研究人员设计并提出了针对图数据的通用计算平台.Google 公司提出了 Pregel^[12]系统,首个实现了基于整体同步并行计算模型^[39]的分布式图并行计算平台,并提出了以顶点为中心的图计算模型.由于 Google 公司并没有开源 Pregel 系统,因此在该平台的启发下,Apache 社区提出了 Giraph^[40]项目与 Hama^[15]项目,分别在 Hadoop 上实现相关算法.与同步并行计算模型相对的,卡耐基梅隆大学提出了基于异步计算模式的 GraphLab^[41]系统以及扩展实现版本 PowerGraph^[13]系统,该系统将图并行处理模型的计算逻辑抽象与调度层分离开来,提供了一套统一的图结构抽象与用户编程接口的定义.于此同时,随着 Spark 技术的发展,基于 Spark 的图计算框架 GraphX^[14]平台也被提出.从某种意义上说,GraphX 平台可以认为是 GraphLap 和 Pregel 在 Spark 上的重写与优化,它在 Spark 上提供了一站式数据解决方案.

上述通用分布式图并行计算框架强调在线计算能力,实现了许多需要大量迭代计算的图处理算法,但在网络数据的多维分析方面并没有提供很好的解决方案.因此,大量针对多维网络分析技术的研究都基于并行计算框架以及上述图并行计算框架进行进一步开发,文献[11,10,38]都在并行计算框架的基础上针对 GraphOLAP 的相关技术进行了研究.

6 总结

本文针对多维网络分析,基于异质网络中关系路径集的概念,提出了路径-维度立方模型,以解决现有的 Graph OLAP 模型对多维网络分析能力不足的问题.在路径-维度立方体的基础上,我们设计了关系路径物化与关联维度物化两种物化算法.

- 在关系路径物化中,主要通过物化算法在关系路径集的基础上生成关系路径聚合网络,我们根据关系路径的可扩展性与对称性对生成过程进行了优化,并基于 Spark 计算框架根据参与连接操作的子路径网络的规模,选择不同的连接策略;
- 在关联维度的物化中,我们把通过关系路径集得到的关系路径聚合网络作为关联维度物化算法的输入,在 Spark 计算框架上设计了外壳片段算法去生成路径-维度聚合网络.

在实验部分,我们基于多个真实的应用场景构建了多维网络,应用 GraphOLAP 操作对网络进行了分析,并在真实、大规模的数据集上验证了相关算法的性能.实验结果表明,我们提出的分析框架具有一定的有效性和高效性.本文的工作旨在基于 OLAP 与数据立方体技术的基础上,设计一种高效的、有效的多维网络分析框架,并在丰富的应用场景中对网络数据进行分析.未来的工作中,我们将继续探索图立方物化策略以提高物化效率.同时,我们也将结合图挖掘技术,提出更为丰富的 GraphOLAP 操作,提高对多维网络的分析挖掘能力.

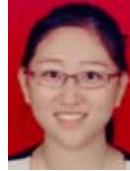
References:

- [1] Li X, Ng MK, Ye Y. MultiComm: Finding community structure in multi-dimensional networks. *IEEE Trans. on Knowledge & Data Engineering*, 2014,26(4):929–941. [doi: 10.1109/TKDE.2013.48]
- [2] Tang L, Wang X, Liu H. Community detection in multi-dimensional networks. 2010. 352–359. https://www.researchgate.net/publication/228778236_Community_Detection_in_Multi-Dimensional_Networks
- [3] Berlingerio M, Coscia M, Giannotti F, Monreale A, Pedreschi D. Foundations of multidimensional network analysis. In: *Proc. of the Int'l Conf. on Advances in Social Networks Analysis and Mining*. IEEE Computer Society, 2011. 485–489. [doi: 10.1109/ASONAM.2011.103]
- [4] Berlingerio M, Coscia M, Giannotti F, Monreale A, Pedreschi D. Multidimensional networks: Foundations of structural analysis. *World Wide Web-Internet & Web Information Systems*, 2013,16(5-6):567–593. [doi: 10.1007/s11280-012-0190-4]
- [5] Rossetti G, Berlingerio M, Giannotti F. Scalable link prediction on multidimensional networks. In: *Proc. of the IEEE Int'l Conf. on Data Mining Workshops*. IEEE Computer Society, 2011. 979–986. [doi: 10.1109/ICDMW.2011.150]
- [6] Gray J, Chaudhuri S, Bosworth A, Layman A, Reichart D, Venkatrao M. Data cube: A relational operator generalizing group-by, cross-tab, and roll-up. In: *Proc. of the Int'l Conf. on Data Engineering*. 1996.
- [7] Gupta A, Mumick IS. Implementing data cubes efficiently. *ACM Sigmod Record*, 1996,25(2):343–360. [doi: 10.1145/233269.233333]
- [8] Gupta A, Mumick IS. *Materialized Views: Techniques, Implementations, and Applications*. MIT Press, 1999.
- [9] Zhao P, Li X, Xin D, Han J. Graph cube: On warehousing and OLAP multidimensional networks. In: *Proc. of the ACM SIGMOD Int'l Conf. on Management of Data (SIGMOD 2011)*. Athens: DBLP, 2011. 853–864. [doi: 10.1145/1989323.1989413]
- [10] Wang Z, Fan Q, Wang H, Tan KL, Agrawal D, Abbadi AE. Pagrol: Parallel graph olap over large-scale attributed graphs. In: *Proc. of the IEEE Int'l Conf. on Data Engineering*. IEEE, 2014. 496–507. [doi: 10.1109/ICDE.2014.6816676]
- [11] Wang P, Wu B, Wang B. TSMH graph cube: A novel framework for large scale multi-dimensional network analysis. In: *Proc. of the IEEE Int'l Conf. on Data Science and Advanced Analytics*. IEEE, 2015. 1–10. [doi: 10.1109/DSAA.2015.7344826]
- [12] Malewicz G, Austern MH, Bik AJC, Dehnert JC, Horn I, Leiser N, Czajkowski G. Pregel: A system for large-scale graph processing. In: *Proc. of the ACM SIGMOD Int'l Conf. on Management of Data*. ACM Press, 2010. 135–146. [doi: 10.1145/1582716.1582723]
- [13] Gonzalez JE, Low Y, Gu H, Bickson D, Guestrin C. PowerGraph: Distributed graph-parallel computation on natural graphs. In: *Proc. of the Usenix Conf. on Operating Systems Design and Implementation*. USENIX Association, 2012. 17–30.
- [14] Xin RS, Gonzalez JE, Franklin MJ, Stoica I. GraphX: A resilient distributed graph system on Spark. In: *Proc. of the Int'l Workshop on Graph Data Management Experiences and Systems*. ACM Press, 2013. [doi: 10.1145/2484425.2484427]
- [15] Apache Software Foundation. The apache Hama project. 2017. <http://hama.apache.org/>
- [16] Kang U, Tsourakakis CE, Faloutsos C. PEGASUS: A peta-scale graph mining system implementation and observations. In: *Proc. of the 9th IEEE Int'l Conf. on Data Mining*. IEEE Computer Society, 2009. 229–238. [doi: 10.1109/ICDM.2009.14]
- [17] Chen C, Yan X, Zhu F, Han J, Yu P. Graph OLAP: Towards online analytical processing on graphs. In: *Proc. of the 8th IEEE Int'l Conf. on Data Mining*. IEEE, 2008. 103–112. [doi: 10.1109/ICDM.2008.30]
- [18] Chuan LI, Lei Z, Tang CJ, Chen Y, Li L, Zhao XM, Liu XL. Modeling, design and implementation of graph OLAPing. *Ruan Jian Xue Bao/Journal of Software*, 2011,22(2): 258–268. [doi: 10.3724/SP.J.1001.2011.03771]
- [19] Yin M, Wu B, Zeng Z. HMGraph OLAP: A novel framework for multi-dimensional heterogeneous network analysis. In: *Proc. of the 15th Int'l Workshop on Data Warehousing and Olap*. 2012. 137–144. [doi: 10.1145/2390045.2390067]
- [20] Shi C, Kong X, Yu PS, Xie S, Wu B. Relevance search in heterogeneous networks. In: *Proc. of the Int'l Conf. on Extending Database Technology*. ACM Press, 2012. 180–191. [doi: 10.1145/2247596.2247618]
- [21] Sun Y, Han J. *Mining Heterogeneous Information Networks: A Structural Analysis Approach*. ACM Press, 2013. [doi: 10.1145/2481244.2481248]
- [22] Sun Y, Han J, Yan X, Yu PS. Mining knowledge from interconnected data: A heterogeneous information network analysis approach. *Proc. of the VLDB Endowment*, 2012,5(12):2022–2023. [doi: 10.14778/2367502.2367566]
- [23] Fang M, Shivakumar N, Garcia-Molina H, Motwani R, Ullman JD. Computing iceberg queries efficiently. In: *Proc. of the 24rd Int'l Conf. on Very Large Data Bases*. Morgan Kaufmann Publishers Inc., 1998. 299–310.
- [24] Milo T, Altschuler E. An efficient MapReduce cube algorithm for varied data distributions. In: *Proc. of the Int'l Conf. on Management of Data*. ACM Press, 2016. 1151–1165. [doi: 10.1145/2882903.2882922]
- [25] Beheshti SMR, Benatallah B, Motahari-Nezhad HR, Allahbakhsh M. A framework and a language for on-line analytical processing on graphs. In: *Proc. of the Int'l Conf. on Web Information Systems Engineering*. 2012. 213–227. [doi: 10.1007/978-3-642-35063-4_16]
- [26] Zhou L, Lv J, Wu B. Social network construction of the role relation in unstructured data based on multi-view. In: *Proc. of the 2017 IEEE 2nd Int'l Conf. on Data Science in Cyberspace (DSC)*. 2017. 382–388. [doi: 10.1109/DSC.2017.78]

- [27] Wang Z, Zhang J, Feng J, Chen Z. Knowledge graph and text jointly embedding. In: Proc. of the Conf. on Empirical Methods in Natural Language Processing. 2014. 1591–1601. [doi: 10.3115/v1/D14-1167]
- [28] Blanco R, Cambazoglu BB, Mike P, Torzec N. Entity recommendation in Web search. In: Proc. of the 12th Int'l Semantic Web Conf. (ISWC). Berlin: Springer-Verlag, 2013. 33–48. [doi: 10.1007/978-3-642-41338-4_3]
- [29] Cao Q, Zhao YM. The realization process and related applications of knowledge graph. Information Studies: Theory & Application (ITA), 2015,12(38):127–132.
- [30] Mtimes. Retrieved January 13, 2018, from Mimes: <http://www.mtime.com/>
- [31] Sinha A, Shen Z, Song Y, Ma H, Eide D, Hsu B, Wang K. An overview of Microsoft academic service (MAS) and applications. 2015. 243–246. [doi: 10.1145/2740908.2742839]
- [32] Nandi A, Yu C, Bohannon P, Ramakrishnan R. Distributed cube materialization on holistic measures. In: Proc. of the IEEE Int'l Conf. on Data Engineering. IEEE Computer Society, 2011. 183–194. [doi: 10.1109/ICDE.2011.5767884]
- [33] Sergey K, Yury K. Applying Map-Reduce paradigm for parallel closed cube computation. In: Proc. of the Int'l Conf. on Advances in Databases, Knowledge, and Data Applications. IEEE, 2009. 62–67. [doi: 10.1109/DBKDA.2009.32]
- [34] Hannachi L, Benlidia N, Bentayeb F, Boussaid O. Social microblogging cube. In: Proc. of the 16th Int'l Workshop on Data Warehousing and Olap. 2013. 19–26. [doi: 10.1145/2513190.2513200]
- [35] Rehman NU, Weiler A, Scholl MH. OLAPing social media: The case of Twitter. In: Proc. of the IEEE/ACM Int'l Conf. on Advances in Social Networks Analysis and Mining. IEEE, 2013. 1139–1146. [doi: 10.1145/2492517.2500273]
- [36] Wang Z, Chu Y, Tan KL, Agrawal D, Abbadi AE, Xu X. Scalable data cube analysis over big data. Computer Science, 2013.
- [37] Li X, Han J, Gonzalez H. High-Dimensional OLAP: A minimal cubing approach. In: Proc. of the 30th Int'l Conf. on Very Large Data Bases. VLDB Endowment, 2004. 528–539.
- [38] Beheshti SMR. Scalable graph-based OLAP analytics over process execution data. Distributed & AMP; Parallel Databases, 2016, 34(3):379–423. [doi: 10.1007/s10619-014-7171-9]
- [39] Valiant LG. A bridging model for parallel computation. Communications of the ACM, 1990,33(8):103–111. [doi: 10.1145/79173.79181]
- [40] Apache Software Foundation. The apache giraph project. 2017. <http://giraph.apache.org/>
- [41] Low Y, Gonzalez JE, Kyrola A, Bickson D, Guestrin C, Hellerstein JM. GraphLab: A new framework for parallel machine learning. Computer Science, 2014.



张子兴(1994—),男,河北唐山人,硕士,主要研究领域为 Graph mining,OLAP.



孙思瑞(1995—),女,主要研究领域为数据挖掘,复杂网络.



吴斌(1969—),男,博士,教授,博士生导师,CCF 高级会员,主要研究领域为数据挖掘,复杂网络.



彭程(1996—),女,主要研究领域为数据挖掘,社会网络分析.



吴心宇(1993—),男,硕士,主要研究领域为 Graph mining,OLAP.



刘昱彤(1996—),女,主要研究领域为数据挖掘,社会网络分析.



张有杰(1996—),男,主要研究领域为数据挖掘,社会网络分析.