CN 43-1258/TP
ISSN 1007-130X

计算机工程与科学
Computer Engineering & Science

第 42 卷第 1 期 2020 年 1 月
Vol. 42, No. 1, Jan. 2020

文章编号:1007-130X(2020)01-0089-09

图数据压缩技术综述*

李凤英,杨恩乙,董荣胜
(桂林电子科技大学可信软件重点实验室,广西 桂林 541004)

摘　要:应用合适的压缩技术对包含上亿个节点和边的图数据进行紧凑准确的表示和存储是对大规模图数据进行分析和操作的前提。紧凑的图数据表示不仅可以降低图数据的存储空间,而且还可以支持在图数据上的高效操作。从图数据的存储角度出发对图数据管理中关于图数据压缩技术的研究进展进行综述,将重点介绍以下 3 种压缩技术:基于邻接矩阵的图数据压缩技术、基于邻接表的图数据压缩技术和基于形式化方法的图数据压缩技术,以及相关的代表性算法、适用范围和优缺点。最后对图数据压缩技术的现状和面临的问题进行了总结,并给出了未来图数据压缩技术的发展趋势。

关键词:邻接矩阵;邻接表;形式化方法;图压缩
中图分类号:TP311　　**文献标志码**:A
doi:10.3969/j.issn.1007-130X.2020.01.011

Summary of graph data compression technologies

LI Feng-ying, YANG En-yi, DONG Rong-sheng
(Guangxi Key Laboratory of Trusted Software, Guilin University of Electronic Technology, Guilin 541004, China)

Abstract: Using appropriate compression techniques to compactly and accurately represent and store graph data with hundreds of millions of nodes and edges is a prerequisite for the analysis and operation of large-scale graph data. Compact graph data representation not only reduces the storage space of graph data, but also supports efficient operation on graph data. This paper summarizes the research progress of graph data compression technologies in graph data management from the storage point of graph data, and focuses on the following three compression technologies: compression technology based on adjacency matrix, compression technology based on adjacency list, and compression technology based on formal method. Their related representative algorithms, application scopes, advantages and disadvantages are discussed. Finally, the current situation and problems of graph data compression technologies are summarized, and the development trend of future graph data compression technologies is given.

Key words: adjacency matrix; adjacency list; formal method; graph compression

1　引言

随着大数据时代的到来,数据规模以前所未有的方式不断增长,数据结构也呈现出复杂性和多样性。如何对其有效地描述、存储和分析已成为当前的研究热点和难点。图通常被看成是一系列节点和节点关系的集合[1],节点对应于实体,边对应于实体关系,非常适合描述关联性数据及内部有一定结构的数据,如社交网络[2]、知识图谱[3]等。

研究表明,大多数领域的问题都可以通过图的相关理论来解决。在 Web 网络分析[4]中,实体和它们的关系可以表示成有向无权图,节点表示网页,边表示网页之间的链接,节点标识表示不同的

* 收稿日期:2019-04-29;修回日期:2019-08-16
基金项目:国家自然科学基金(61762024);广西自然科学基金(2017GXNSFDA198050,2016GXNSFAA380054);桂林电子科技大学研究生创新创业项目(2019YCXS053)
通信地址:541004 广西桂林市桂林电子科技大学可信软件重点实验室
Address: Guangxi Key Laboratory of Trusted Software, Guilin University of Electronic Technology, Guilin 541004, Guangxi, P. R. China

域名。在这种表示形式下,查询给定的实体关系和侦测特定的团体,可以转化为图的邻居查询和子图匹配[5-7]问题。在化学数据分析中,从给定的化合物集中挖掘常见的原子团,可以转化为频繁子图查询[8]问题。在蛋白质交互网络分析中,衡量给定的某2个蛋白质发生作用的概率时,可以用不确定图[9,10]建模得以解决。可见,图在众多领域都有着重要的研究价值。

随着图在各个领域的广泛应用,传统的图存储结构已经不能支持超大规模图数据的管理和分析。比如具有一百万个节点的社交网络,邻接矩阵的大小(2个节点之间的1条边存储空间为1 bit)大约为116 GB,大多数计算机不会有如此大的主存储器来加载图并执行社交网络分析。中国互联网络中心2018年发布的《第41次中国互联网络发展状况报告》[11]中提到,网页的数量约为2 600亿。若采用图模型存储上述网页节点及节点关系,至少需要42 TB的存储空间。并且随着因特网的不断发展,需要的存储开销也越来越大。如何存储和操作上亿万个节点的图数据,国内外研究人员主要从以下3个方向做了大量的研究:

(1)外部存储技术[12,13]:针对大规模图数据无法一次装入内存问题,研究人员一方面将图数据存储到价格低廉、容量大的外部存储器(硬盘或软盘),另一方面设计更加高效的I/O算法来避免更多的I/O开销。

(2)分布式存储技术[14]:将图数据分割为多个部分,存储到不同的分布式计算机中,但这会带来更多的通信开销和CPU资源消耗。

(3)图数据压缩技术[15-17]:主要思想是消除图数据中的冗余信息,将图数据以压缩的形式存储到内存中。

相对前2种技术,第(3)种技术时间开销相对降低,而且可以适用于任何类型的图数据。

本文主要从3个方面讨论图数据压缩技术:

(1)基于邻接矩阵的压缩技术,主要思想是尽可能地压缩邻接矩阵中的"0"元素。

(2)基于邻接表的压缩技术,主要思想是利用节点的邻居节点集的相似性和局部引用性来进行压缩。

(3)基于形式化方法的压缩技术,主要思想是对所给的图进行编码,使其转化为布尔代数,再利用决策图对布尔代数进行表示和化简。

本文的结构如下:首先介绍3类压缩技术,分别为基于邻接矩阵的压缩技术、基于形式化方法的压缩技术和基于邻接表的压缩技术。在此基础上,为了充分说明形式化方法对于图数据压缩的优势,我们给出了相关的实验数据对比,并预测了未来图数据压缩发展方向。

2 基于邻接矩阵的压缩技术

Broder等人[18]和Raghavan等人[19]的研究表明,大量的图特征函数服从幂律分布,其邻接矩阵往往具有一定的稀疏性和聚类性。Brisaboa等人[20]利用邻接矩阵的特征提出了k^2-tree,取得了较好的时间/空间均衡。k^2-tree的构造过程主要包括以下2个步骤:

步骤1 对于1个给定的$n\times n$邻接矩阵,首先判断n是否为k的幂。若满足条件,转到步骤2;若n不是k的幂,增加矩阵中的行和列使得$n=k^s$(s为正整数),其中增加的行和列的元素用"0"填充,然后再转到步骤2进行递归划分。

步骤2 进行递归划分:根据MXQuntree规则[21]把矩阵划分为k^2个大小一致的子矩阵。如果子矩阵中的元素至少有1个为1,那么把这种矩阵标记为1,否则标记为0,自上而下,自左而右排列这些值,它们将作为根节点的4个儿子节点,树的第1层节点构造完毕。将标记为1的矩阵再进行递归处理,它们的值将作为树的第2层节点,如此重复直到划分后的矩阵全部为0或者已经划分到原始矩阵中的某个元素,递归停止。

如图1所示是1个具有4个节点的网页图所对应的邻接矩阵以及k^2-tree(n是k的幂)。图2所示是1个具有11个节点的网页图所对应的邻接矩阵以及k^2-tree(n不是k的幂),矩阵中的深色部分是为满足条件所增加的行和列。

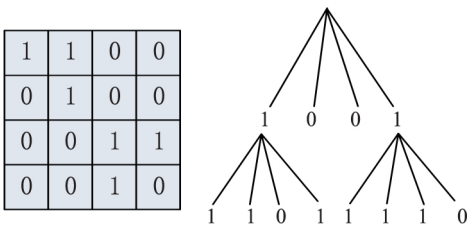

Figure 1 k^2-tree corresponding to $4*4$ adjacency matrix($k=2$)

图1 4×4 邻接矩阵所对应的k^2-tree($k=2$)

如图2所示,若用邻接矩阵来存储节点数为11的网页图,需要的存储空间为121 bit。若采用k^2-tree存储,存储空间为72 bit。可见k^2-tree比邻接矩阵具有更好的空间利用率,并且随着图节点

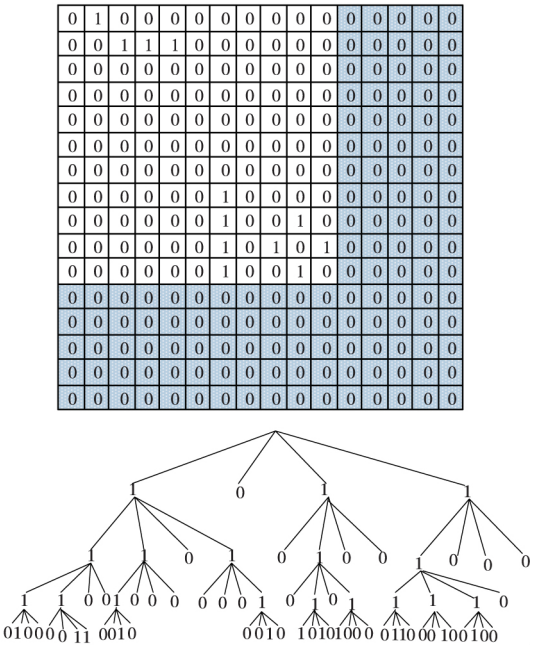

Figure 2 Extension of matrix and construction of k^2-tree when $k=2$

图 2 $k=2$ 时矩阵的拓展和 k^2-tree 构建

数目的不断增加,k^2-tree 节省的存储空间会越来越多。不仅如此,k^2-tree 还支持在常数时间范围内查询节点的直接邻居和反向邻居。

k^2-tree 能够紧凑地表示图数据,但传统的 k^2-tree 对于图的邻接矩阵的压缩还存在不足:(1)网页图中的节点和链接分布严重依赖于 URL 排序。若采用传统的 k^2-tree 思想机械地划分网页图,会导致稠密区域和稀疏区域划分到同 1 个子矩阵中,这会使得空间利用率降低。(2)k^2-tree 的邻居查询时间与 k^2-tree 的高度成正比,面对上亿个节点的图数据,若不对其进行处理,邻居查询时间会大大增加。

为了更加有效、紧凑地压缩表示网页图,Claude 等人[22]利用网页图中域特有的规律,提出了 k^2-partition。他们沿着对角线把图划分成不同的域,对不同的域用传统的 k^2-tree 表示,实现了较好的时间/空间均衡。以下用实例来说明如何对图数据进行 k^2-partition 表示。

采用传统的 k^2-tree 表示图 2 中的矩阵需要 72 bit 的存储空间,而采用 k^2-partition 表示图 3 和图 4 中的矩阵,需要的存储空间为 64 bit,存储空间需求有所改善,并且随着图节点的增加,效果会越来越明显。在传统的 k^2-tree 中查询给定节点的邻居信息的时间与树的高度成正比,在 k^2-partition 中由于降低了树的高度,因此能获得更好的查询时间。

出于对多维数据如时序图(节点关系随时间改变的动态图)、社交网络图和引文网络图紧凑表示

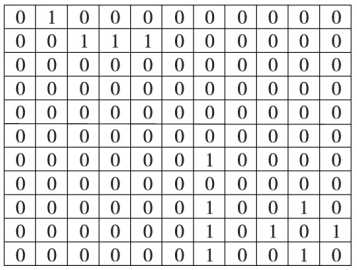

Figure 3 Dividing the graph into domains

图 3 对图进行分域处理

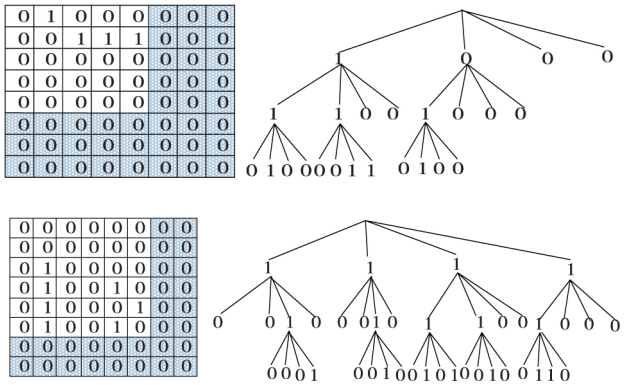

Figure 4 k^2-tree representation of different domains

图 4 不同域的 k^2-tree 表示

的需要,Caro 等人[23]利用 k^2-tree 的构造思想提出了 k^d-tree。k^d-tree 是 k^2-tree 的一般化形式,通常用来表示 d 维的二元矩阵,对于 1 个规模为 $n_1\times n_2\times\cdots\times n_d$ 的 d 维矩阵,将其递归地划分为 k^d 个子矩阵,为了简化分析,假设对于每一个 $i(d\geqslant i\geqslant 1)$都有 $n_i=n$,其中每 1 个树中的节点都拥有 k^d 个儿子节点来表示对应的子矩阵。对于每 1 个子矩阵,按照从高维到低维的次序来确定 k^d-tree 中的第 1 层至最后 1 层的值,如果子矩阵存在“1”值的元素,使用“1”来表示,如果为“0”,使用“0”来表示。图 5 表示了不同维度下 k^d-tree($k=2$)表示方法。

3 基于形式化方法的压缩技术

二叉决策图 BDD(Binary Decision Diagram)作为一种新型的数据结构,不仅是解决布尔代数表示与计算的最为有效的工具[24-26],还是符号模型检验技术[27,28]的核心。2007 年,杨志飞等人[29]将符号计算运用到大规模图数据的压缩表示中,核心思想是将图中的节点进行二进制编码,把图中的链接关系转化为布尔代数,进一步布尔代数可以通过有序二叉决策图 OBDD(Ordered Binary Decision Diagram)求解。利用布尔代数中节点取值的冗余,衍生出 OBDD 的化简规则和删除规则,以此来

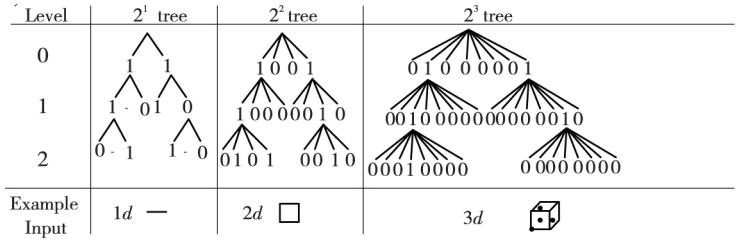

Figure 5 Construction of k^d-tree in different dimensions(d is the dimension of data,$k=2$)

图 5 不同维度的 k^d-tree 的构建(d 为数据的维度,$k=2$)

共享子图,提高存储效率。在这种表示形式下,查询图中节点的度数、判断 2 个节点是否存在链接关系、向图中增加或者删除边和图的同构问题分别可以转化为 OBDD 的可满足性问题、OBDD 的求值操作、OBDD 的 apply 操作和 OBDD 的等价性判定[30]。将图转化为 OBDD 的具体思想为:对于 1 个具有 n 个节点的有向图,利用布尔变量对节点和边进行编码,将其转化为布尔表达式。如图 6a 中具有 4 个节点,故需要 2 个布尔变量。对于图 6a 中的有向边,则需要 2 组布尔变量来分别表示有向边的起点和终点,设有向边的起点用 $x=x_1x_2$ 表示,终点用 $y=y_1y_2$ 表示,其中 $x_1,x_2,y_1,y_2\in\{0,1\}$,由于节点 0 和节点 1 存在有向边,设节点 0 的编码为 $x_1'x_2'$(表示 x_1 取 1,x_2 取 1),节点 1 的编码为 $y_1'y_2$(表示 y_1 取 1,y_2 取 1),则有向边可表示为 $x_1'x_2'y_1'y_2$。基于上述方法,编码每 1 条边,便可得到该图对应的 OBDD,如图 6b 所示是图 6a 有向图所对应的 OBDD。由于 OBDD 的终节点只能为 0 或 1,所以它只能表示无权图。Bahar 等人[31]提出了代数决策图 ADD(Algebraic Decision Diagram),进一步把布尔代数拓展到伪布尔代数,将无权图拓展到带权图,进一步丰富了图的布尔代数表示方法。将图转化为 ADD 的思想和 OBDD 类似,其中的区别在于 ADD 的终节点不再是 0 或 1,而是图中存在的每 1 条边的权重值。如图 7 所示是 4 个节点的带权有向图所对应的邻接矩阵 $\boldsymbol{M}_G$ 以及代数决策图。

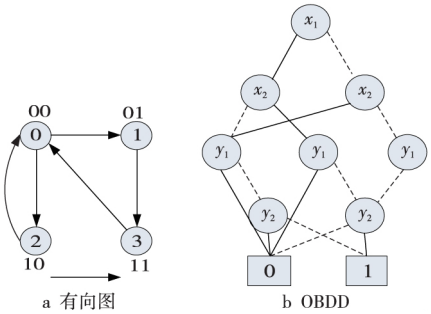

Figure 6 An OBDD representation of a digraph

图 6 有向图的 OBDD 表示

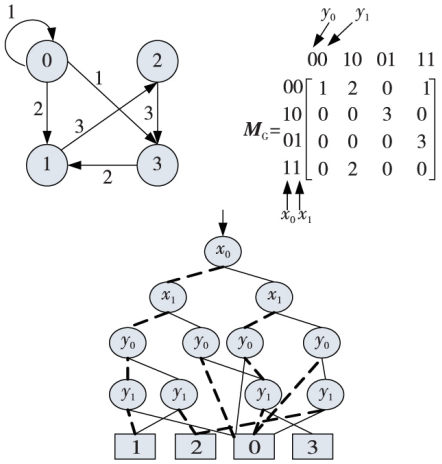

Figure 7 An ADD representation of a weighted graph

图 7 带权图的 ADD 表示

随着图数据规模的不断增长,如何更加有效地紧凑表示图数据,成为当前的一个研究重点。文献[20]提到 k^2-tree 能很好地压缩邻接矩阵,实现较好的时间/空间均衡,但 k^2-tree 还面临以下问题:(1)k^2-tree 中还存在大量的同构子树。(2)k^2-tree 只能对稀疏图进行压缩。(3)k^2-tree 只能表示静态图,不能向其中增加或者删除边。

针对上述问题,董荣胜等人[32]把多值决策图 MDD(Multi-value Decision Diagram)[33]和 k^2-tree 进行结合,提出了 k^2-MDD,利用 MDD 的删除规则和化简规则以及多变量取值性质进行相同子图的合并,构造过程主要包括 3 步:

步骤 1 将所给的邻接矩阵用传统的 k^2-tree 表示。

步骤 2 删除 k^2-tree 中所有的 0 节点,合并 k^2-tree 叶子节点中为 1 的节点。

步骤 3 对 k^2-tree 每个分支进行二进制编码($k=2$),节点值相等且儿子节点相同的节点合并(共享子图)。

图 8 所示是 11 个顶点的邻接矩阵的 k^2-tree 表示。图 9 所示是传统的 k^2-tree 中的同构子树的分布,相同的子树用大小相等的方框注明。图 10 表示 k^2-tree 的 MDD。

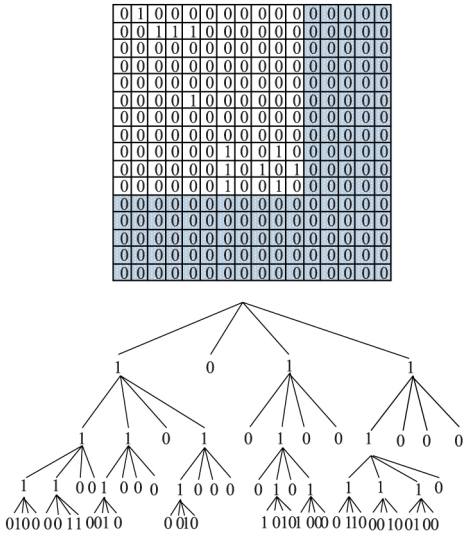

Figure 8 k^2-tree representation of an adjacency matrix

图 8 邻接矩阵的 k^2-tree 表示

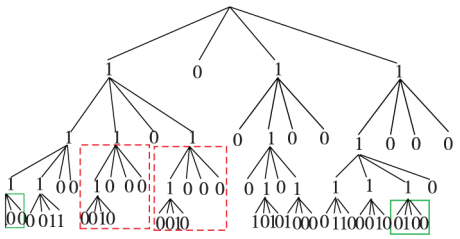

Figure 9 Isomorphic subtree distribution of k^2-tree

图 9 k^2-tree 的同构子树分布

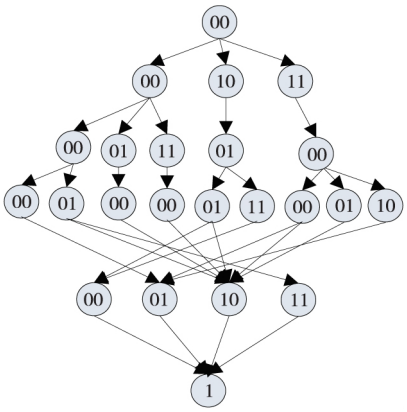

Figure 10 MDD representation of a k^2-tree

图 10 k^2-tree 的 MDD 表示

4 基于邻接表的压缩技术

根据研究表明,如果所有网页图的 URLs 按照字典序排序,大多数的网页图具有以下 2 种特性[34,35]:

(1)局部性:对于某个页面来说,它的直接邻居集合彼此之间挨得很近。

(2)相似性:位置上靠得很近的一些网页集,它们的后继集很相似。

利用网页的局部性,Boldi 等人[36]提出了空隙编码,其思想为用 2 个连续的节点标签值来替代原始节点标签值,设 $A(x)=(a_1,a_2,a_3,\cdots,a_n)$,$x$ 为第 x 个节点的标签值,$a_1,a_2,a_3,\cdots,a_n$ 为 x 的直接邻居,则 x 对应的空隙编码为 $B(x)=(c(a_1-x),a_2-a_1-1,a_3-a_2-1,\cdots,a_n-a_{n-1}-1)$。式(1)为计算 $A(x)$中 a_1 的空隙编码标签值。

$$c(x)=\begin{cases}2x, x\geqslant 0\\ 2|x|-1, x<0\end{cases} \tag{1}$$

表 1 是网页图中截取的小部分邻接表,表 2 是邻接表对应的空隙编码。

Table 1 Traditional adjacency table representation

表 1 传统的邻接表表示

节点	邻接节点
…	…
15	13,15,16,17,18,19,23,24,203,315,1034
16	15,16,17,22,23,24,315,316,317,3041
17	…
18	13,15,16,17,50
…	…

Table 2 Void coding representation

表 2 空隙编码表示

节点	邻接节点
…	…
15	3,1,0,0,0,0,3,0,178,111,718
16	1,0,0,4,0,0,290,0,0,2723
17	…
18	9,1,0,0,32
…	…

图数据规模的不断增长使得节点标签值的位数不断增加,空隙编码的本质为压缩节点的标签值,减少所需要的存储空间。

利用网页的相似性,Suel 等人[37]提出了参考压缩,其思想是用 1 个节点的邻接表来表示其余的邻接表,设 $s(x)$,$s(y)$分别为 2 个节点的出度表,$s(x)$称为参考表,$s(y)$称为复制表,$y-x$ 称为参考系数,用 r 表示。若 $s(x)$的后继在 $s(y)$中也存在,那么复制表中对应位置为 1,否则为 0。进一步若 $s(y)$的后继在 $s(x)$中不存在,记这些节点为额外节点。表 3 是邻接表的参考压缩。

上述 2 种技术都要求节点的直接邻居集合是有序的,如果网页图需要保存最原始的链接顺序,上面的方法并不适用。

Adler 和 Faust 等人[38,39]2010 年发现,

Table 3 Reference compressed of adjacency table

表 3 邻接表的参考压缩

节点	参考编号	参考列表	额外节点列表
…	…	…	
15	0		13,15,16,17,18,19,23,24,203,315,1034
16	1	01110011010	22,316,317,3041
17	…	…	…
18	3	11110000000	50

邻接表的节点之间的后继有许多相似的信息,这意味着数据还存在一定的冗余[38,39],Repair 算法[40]应运而生。其算法思想是把所有节点的后继看成 1 个序列 T,每 1 次在序列中用 s(s 为从来没有在 T 中出现的符号)替换 T 中最频繁的符号对,直到序列 T 不再出现频繁模式。假设图 $G=(V,E)$,$T(G)=v_1v_{1,1}v_{1,2}v_{1,3}\cdots v_{1,n}v_2v_{2,1}v_{2,2}v_{2,3}\cdots v_{2,n}\cdots v_nv_{n,1}v_{n,2}v_{n,3}\cdots v_{n,n}$,其中 v_1 为第 1 个节点的标签值,$v_{1,n}$为第 1 个节点的后继。$Ptrs[m]$为 1 个指针数组,记录每 1 个节点在序列 T 中的起始位置。图 11 是给定图的邻接表的 Repair 压缩。

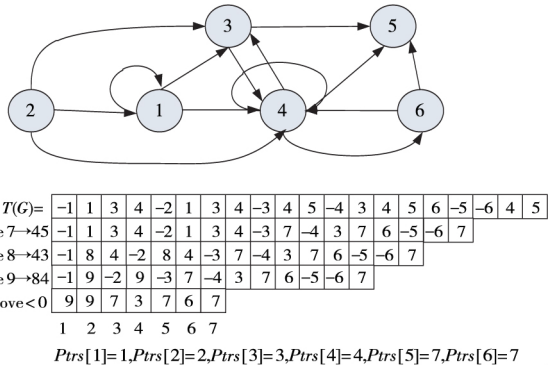

Figure 11 Repair compression of an adjacent table

图 11 邻接表的 Repair 压缩

Repair 算法将 1 个邻接表压缩成 1 个字典规则 R 的集合,1 个指针数组 $ptrs$,1 个序列 T。每次查询节点的信息时,仅仅需要找到该节点的起始位置和终止位置,然后进行部分解压缩即可。但是,对于大规模图而言,由于每次都要添加新的规则,字典规则 R 的集合越来越大,Bille 等人[41]对该算法进行了改进,取得了很好的时间/空间均衡。

另 1 种基于邻接表的压缩称为 LZ78 算法[42],由 Ziv 和 Lempel 提出,其思想为建立 1 个字典表,每读入 1 个字符,判断其是否在字典表中,若不存在,则保存字符并建立索引。若存在,则保存索引并加上新的字符作为这个字符串的表示。

具体的算法流程如下[43]:

步骤 1 建立字典表,并将字典表设置为空。

步骤 2 依次读取文本中的 1 个新的字符,设新字符为 C。

步骤 3 在词典中查找当前的前缀和新的字符的组合,也就是 $P+C$:

(1)如果在字典表中找到了这个新组合,那么就把前缀 P 重新进行改写,需要加上新读取的字符 C。

(2)如果字典表中没有这个新组合,就要执行保存新组合的操作:

①输出当前前缀的索引以及字符 C。

②把前缀和新读取的字符串保存在字典表中。

③重新改写前缀 P,将其设置为空。

(3)重复步骤 2 和步骤 3,直到所有的字符串都完成编码。

LZ78 算法和 Repair 算法在对文本压缩时有 1 个区别:$T(G)=v_{1,1}v_{1,2}v_{1,3}\cdots v_{1,n}v_{2,1}v_{2,2}v_{2,3}\cdots v_{2,n}\cdots v_{n,1}v_{n,2}v_{n,3}\cdots v_{n,n}$,表 4 所示是图 11 所示的邻接表的 LZ78 压缩。

Table 4 LZ78 compression

表 4 LZ78 压缩结果

输出	索引	字符
(0,1)	1	1
(0,3)	2	3
(0,4)	3	4
(1,3)	4	1,3
(3,4)	5	4,4
(0,5)	6	5
(2,4)	7	3,4
(6,6)	8	5,6
(3,5)	9	4,5

LZ78 算法和 Repair 最大的区别在于在压缩时不用存储和维护字典表,因为字典表以结果形式输出了,因此 LZ78 算法查询节点信息的速度比 Repair 算法快,但由于每次解压过程,从结果的最开始开始构造字典 R,所以只能对邻接表的边表进行压缩,限制了 LZ78 算法的性能。

5 图数据压缩技术比较

文献[17]采用真实的网页图数据集和社交网络数据集对其中提到的所有压缩技术进行了对比,其数据集来自于米兰大学 LAW[44],表 5 给出了网页图数据集的相关属性,表 6 给出了社交网络数据集的相关属性。通过一系列的实验对比,他们得出 k^2-tree 的压缩率明显低于其他算法(Repair,LZ78

等算法)的,能实现较好的时间/空间均衡。为了体现出符号计算对于大规模图数据压缩处理的优势,本文将 k^2-tree 分别与 OBDD,k^2-MDD 2 种压缩技术做了实验对比,实验结果如表 7 和表 8 所示。

Table 5 Web graph data set

表 5 网页图数据集

数据集	节点数	边数	边数/节点数
cnr	325 557	3 216 152	9.88
in	1 382 908	16 917 053	12.23
uk	4 769 354	50 829 923	10.66
indochina	7 414 866	194 103 311	26.18

Table 6 Social network data set

表 6 社交网络数据集

数据集	节点数	边数	边数/节点数
enron	69 244	276 143	3.99
dblp-1	326 186	1 615 400	4.95
Dblp-2	986 324	6 107 236	6.80
dewiki	1 532 254	36 722 696	23.96

Table 7 Experimental results of web graph

表 7 网页图实验结果 bit

数据集	k^2-tree	OBDD/bit	k^2-MDD/bit
cnr	11 246 165	2 435 522	342 893
in	49 442 029	10 652 901	1 205 089
uk	464 588 901	59 830 211	14 793 248
indochina	467 001 541	63 727 936	7 974 153

Table 8 Experimental results of social network

表 8 社交网络实验结果 bit

数据集	k^2-tree	OBDD/bit	k^2-MDD/bit
enron	2 490 345	242 821	125 481
dblp-1	12 018 925	3 260 150	621 021
dblp-2	69 557 893	12 269 732	3 310 652
dewiki	615 969 633	50 403 738	18 895 096

实验结论:在 k^2-tree,k^2-MDD,OBDD 中,用 1 个位串 T 来记录除最后 1 层节点的 0 值和 1 值,1 个位串 L 来记录最后 1 层节点的 0 值和 1 值,T 和 L 的总和为需要的存储空间。如表 7 所示,若采用 OBDD 来压缩网页图,T 和 L 的总和约为 k^2-tree 的 21%,若采用 k^2-MDD 来压缩网页图,T 和 L 的总和约为 k^2-tree 的 3%,如表 8 所示,我们用社交网络数据集来进行实验对比,分别在 enron,dblp-1,dblp-2,dewiki 数据集上进行实验,得到的 T 和 L 的总和分别为 k^2-tree 的 4.8%,5.1%,4.7%,3%,对存储空间的需求得到有效的改善。

6 结束语

随着图在各个领域的广泛应用,传统的图存储结构已经不能支持大规模图数据的管理和操作,如何有效地紧凑表示图数据并且支持快速的访问已经成为一项重要的研究任务。本文首先介绍图数据应用概况和相关的图数据压缩表示技术,然后详细阐述了 3 种图数据压缩技术,并给出了不同压缩技术的优缺点。通过分析发现,基于形式化方法的图压缩技术在处理大规模图数据时,具有很好的压缩效果,这也是我们下一步研究的重点。对于图数据的压缩,未来可能会结合机器学习的聚类算法和形式化方法来更好地解决数据的冗余问题。由于篇幅有限,本文不可能涵盖该领域所有的研究内容,希望这篇综述能对图数据压缩技术的研究起到一定的参考作用。

参考文献:

[1] Khan A, Wu Y, Yan X. Emerging graph queries in linked data [C]// Proc of International Conference on Data Engineering, 2012: 1218-1221.

[2] Belov Y A, Vovchok S I. Generation of a social network graph by using Apache Spark[J]. Automatic Control & Computer Sciences, 2017, 51(7): 678-681.

[3] Bordes A, Usunier N, Garcia-Duran A, et al. Translating embeddings for modeling multi-relational data[C]// Proc of International Conference on Neural Information Processing Systems, 2013: 2787-2795.

[4] Blazewicz J, Pesch E, Sterna M. A novel representation of graph structures in web mining and data analysis[J]. The international Journal of Management Science, 2005, 33(1): 65-71.

[5] Khan A, Li N, Yan X, et al. Neighborhood based fast graph search in large networks[C]// Proc of SIGMOD International Conference on Management of Data, 2011: 901-912.

[6] Atre M, Chaoji V, Zaki M J, et al. Matrix " Bit" loaded: A scalable light weight join query processor for RDF data[C]// Proc of the 19th International Conference on World Wide Web, 2010: 26-30.

[7] Zou L, Mo J, Chen L, et al. gStore: Answering SPARQL queries via subgraph matching[J]. Proceedings of the VLDB Endowment, 2011, 4(8): 482-493.

[8] Chen Xiao, Liu Feng-chun, Li Jian-jing, et al. A new algorithm for top-down mining of the most frequent subgraphs [J]. Computer Engineering & Science, 2013, 35(4): 157-162. (in Chinese)

[9] Zhao B, Wang J, Li M, et al. Detecting protein complexes

based on uncertain graph model[J]. Transactions on Computational Biology & Bioinformatics,2014,11(3):486-497.

[10] Zhang Y,Lin H,Yang Z,et al. An uncertain model-based approach for identifying dynamic protein complexes in uncertain protein-protein interaction networks[J]. BMC Genomics,2017,18(7):743-754.

[11] China internet development report[EB/OL]. [2018-03-05]. http://cnnic.cn/hlwfzyj/hlwxzbg/hlwtjbg/201803/t20180305_70249.html. (in Chinese)

[12] Vitter J S. External memory algorithms and data structures [J]. ACM Computing Surveys,2001,33(2):209-271.

[13] Kumar V, Schwabe E J. Improved algorithms and data structures for solving graph problems in external memory [C]// Proc of IEEE Symposium Parallel and Distributed Processing,1996:169-176.

[14] Badue C,Ribeiro-Neto B,Baeza-Yates R,et al. Distributed query processing using partitioned inverted files[C]// Proc of International Symposium on String Processing & Information Retrieval,2005:10-20.

[15] Chang L,Zeng X. Gu Tian-long. Optimal representation of large-scale graph data based on K^2-Tree[J]. Wireless Personal Communications,2017,95(1):1-14.

[16] Boldi P,Vigna S. The WebGraph framework I:Compression techniques[C]// Proc of International Conference on World Wide Web,2004:595-601.

[17] Zhang Yu,Liu Yan-bing,Xiong Gang,et al. Survey on succinct representation of graph data[J]. Journal of Software, 2014,25(9):1937-1952. (in Chinese)

[18] Broder A,Kumar R. Graph structure in the Web[J]. Computer Networks,2000,33(1):309-320.

[19] Raghavan S, Garcia-Molina H. Representing web graphs [C]// Proc of the 19th International Conference on Data Engineering,2003:405-420.

[20] Brisaboa N R,Ladra S,Navarro G. K^2-Trees for compact web graph representation[C]// Proc of International Symposium on String Processing and Information Retrieval, 2009:18-30.

[21] Samet H. Foundations of multidimensional and metric data structures [M]. 2nd ed. San Francisco:Morgan Kaufmann Publisher,2006.

[22] Claude F,Ladra S. Practical representations for web and social graphs[C]// Proc of the 20th ACM Conference on Information and Knowledge Management,2011:1185-1190.

[23] Caro D,Rodr,Guez M A,et al. Compressed k^d-tree for temporal graphs [J]. Knowledge & Information Systems,2016, 49(2):1-43.

[24] Gu Tian-long,Xu Zhou-bo. Ordered binary decision diagram and application[M]. 2nd ed. Beijing:Science Press,2009. (in Chinese)

[25] Fujita M,Mcgeer P C,Yang C Y. Multi-terminal binary decision diagrams:An efficient data structure for matrix representation[J]. Formal Methods in System Design,1997,10 (3):149-169.

[26] Bryant R E. Symbolic boolean manipulation with ordered binary-decision diagrams[J]. ACM Computing Surveys,1992, 24(3):293-318.

[27] Li Feng-ying,Gu Tian-long, Xu Zhou-bo. Symbolic ZBDD reachable tree analysis technique for Petri nets[J]. Chinese Journal of Computers, 2009, 32(12): 2420-2428. (in Chinese)

[28] Qian Jun-yan,Gu Tian-long,Zhao Ling-zhong. Model checking state-charts based on EHA[J]. Computer Engineering, 2006,32(3):19-21. (in Chinese)

[29] Yang Zhi-fei,Gu Tian-long. Research on storage and operation of directed graph based on OBDD[J]. Computer Science,2007,34(8):283-285. (in Chinese)

[30] Bryant R E. Graph-based algorithms for boolean function manipulation[J]. IEEE Transactions on Computers,1986, 35(8):677-691.

[31] Bahar R I,Frohm E A,Gaona C M,et al. Algebric decision diagrams and their applications[J]. Formal Methods in System Design,1997,10(2):171-206.

[32] Dong Rong-sheng, Zhang Xin-kai,Gu Tian-long,et al. K^2-MDD representation method and operation of large-scale graph data[J]. Journal of Computer Research and Development,2016,52(12):2783-2792. (in Chinese)

[33] Srinivasan A,Ham T,Malik S,et al. Algorithms for discrete function manipulation[C]// Proc of IEEE International Conference on Computer-Aided Design,1990:92-95.

[34] Bharat K, Broder A,Henzinger M, et al. The connectivity server:Fast access to linkage information on the web[C]// Proc of the 7th International World Wide Web Conference, 1998:469-477.

[35] Randall K, Stata R, Wickremesinghe R, et al. The LINK database:Fast access to graphs of the web[C]// Proc of the Data Compression Conference,2002:122-131.

[36] Boldi P,Vigna S. The webgraph framework II:Codes for the world-wide web[C]// Proc of Conference on Data Compression,2004:528-537.

[37] Suel T,Yuan J. Compressing the graph structure of the web [C]// Proc of the Data Compression Conference,2001:213-222.

[38] Adler M, Mitzenmacher M. Towards compressing web graphs[C]// Proc of the Data Compression Conference, 2001:423-435.

[39] Faust F C, Navarro G. A fast and compact web graph representation [J]. ACM Transactions on the Web,2010,4 (4):1-31.

[40] Larsson N J,Moffat A. Off-line dictionary-based compression[J]. Proceedings of the IEEE,2000,88(11):1722-1732.

[41] Bille P, Gørtz I L, Prezza N. Space-efficient Repair compression[C]// Proc of 2017 Data Compression Conference (DCC), 2017: 237-247.

[42] Ziv J, Lempel A. A universal algorithm for sequential data compression[J]. IEEE Transactions on Information Theory, 1977, 23(3): 337-343.

[43] Man Tian-xing. Improved search algorithm on LZ series compressed text [D]. Jilin: Jilin University, 2017. (in Chinese)

[44] Laboratory for web algorithmics[EB/OL]. [2016-10-13]. http://law.di.unimi.it/datasets.php.

附中文参考文献:

[8] 陈晓,刘凤春,李建晶,等. 一种新的自顶向下挖掘最大频繁子图的算法[J]. 计算机工程与科学, 2013, 35(4): 157-162.

[11] 中国互联网络发展报告[EB/OL]. [2018-03-05]. http://cnnic.cn/hlwfzyj/hlwxzbg/hlwtjbg/201803/t20180305_70249.html.

[17] 张宇,刘燕兵,熊刚,等. 图数据表示与压缩技术综述[J]. 软件学报, 2014, 25(9): 1937-1952.

[24] 古天龙,徐周波. 有序二叉决策图及应用[M]. 修订2版. 北京:科学出版社, 2009.

[27] 李凤英,古天龙,徐周波. Petri网的符号ZBDD可达树分析技术[J]. 计算机学报, 2009, 32(12): 2420-2428.

[28] 钱俊彦,古天龙,赵岭忠. 基于EHA模型检验Statecharts[J]. 计算机工程, 2006, 32(3): 19-21.

[29] 杨志飞,古天龙. 基于OBDD的有向图的存储与操作研究[J]. 计算机科学, 2007, 34(8): 283-285.

[32] 董荣胜,张新凯,古天龙,等. 大规模图数据的K^2-MDD表示方法与操作研究[J]. 计算机研究与发展, 2016, 52(12): 2783-2792.

[43] 满天星. 改进的LZ系列压缩文本上的搜索算法[D]. 吉林:吉林大学, 2017.

作者简介:

李凤英(1974-),女,辽宁朝阳人,博士,副教授,CCF会员(37785M),研究方向为图数据、符号计算和形式化方法。**E-mail**:lfy@guet.edu.cn

LI Feng-ying, born in 1974, PhD, associate professor, CCF member(37785M), her research interests include graph data, symbolic computing, and formal method.

杨恩乙(1994-),男,四川达州人,硕士生,研究方向为图数据压缩和机器学习。**E-mail**:1037239419@qq.com

YANG En-yi, born in 1994, MS candidate, his research interests include graph data compression, and machine learning.

董荣胜(1965-),男,湖北红安人,教授,CCF会员(090425),研究方向为大规模图数据的管理和计算思维的结构。**E-mail**:ccrsdong@guet.edu.cn

DONG Rong-sheng, born in 1965, professor, CCF member(090425), his research interests include large-scale graph data management, and structure of computational thinking.