ISSN 1004-9037,CODEN SCYCE4
Journal of Data Acquisition and Processing Vol. 33,No. 4,Jul. 2018,pp. 758－768
DOI:10. 16337/j. 1004-9037. 2018. 04. 020

http://sjcj. nuaa. edu. cn
E-mail:sjcj@nuaa. edu. cn
Tel/Fax: ＋86-025-84892742

支持多模态医学数据融合的并行加载算法

翟　霄　潘海为　谢晓芹　张志强　韩启龙

(哈尔滨工程大学计算机科学与技术学院,哈尔滨,150001)

摘　要:由于多模态数据中的数据分属多种模态且相互之间存在互补关系,所以利用传统单模态数据的分析和处理方法无法有效地融合不同模态的数据并表示和处理不同模态数据之间的相互关系。为了解决多模态数据的建模、表示和存储问题,使得更好地融合不同模态的数据及更有效地表示数据之间的相互关系,本文提出了一种新的数据模型,即模态结构图,用于多模态医学数据的建模。该模型利用图结构对多模态数据中的模态及模态间的关系进行建模和表示。基于此模型,本文提出了一种并行的数据加载技术,用于抽取出多模态医学数据中分属不同模态和模态间关系的数据并存储到图数据库中。通过使用批量医学数据文件进行实验,验证了该提取加载技术能够获得较好的数据处理效率。
关键词:并行加载;医学数据;多模态融合;图数据库;模态结构图
中图分类号:TP391　　**文献标志码**:A

Parallel Loading Algorithm for Multimode Medical Data Fusion

Zhai Xiao, Pan Haiwei, Xie Xiaoqin, Zhang Zhiqiang, Han Qilong

(College of Computer Science and Technology, Harbin Engineering University, Harbin, 150001, China)

Abstract: Since the different parts of multimode data belong to multiple modes and there are complementary relationships between them, the traditional analysis and processing techniques cannot fuse the data from different modes and their relationships. In order to solve the problems of modeling, representation and storage of multimode data, a new data model of modal structure graph is proposed to model the multimode medical data. The model utilizes the graph structure to model and represent the modes and their relationships in the multimode data. And based on the model, the paper proposes a parallel loading algorithm, which can extract the data of different modes and relationships, and store them in a graph database. Experimental results by using batch medical data files show that the proposed algorithm can obtain good data processing efficiency.
Key words: parallel loading; medical data; multimode fusion; graph database; mode graph

引　言

近年来,由于医院信息系统(Hospital information system,HIS)的普及,医疗机构掌握了海量的医

基金项目:国家自然科学基金(61672181, 61370084, 61272184, 61202090)资助项目;黑龙江省自然科学基金(F2016005)资助项目。
收稿日期:2016-09-10;**修订日期**:2016-11-18

学数据,如何利用这些数据进行医学信息检索、数据挖掘引起了越来越多的关注。因此,如何更有效地表示和存储医学数据使其更适合进行信息检索和数据挖掘,以及更快速地加载医学数据成为迫切需要解决的问题。

医学数据包括多种类型:(1)检查手段产生的测量数值,如体温、血压、血氧饱和度和化验值等;(2)仪器记录的信号,如心电图、脑电图等;(3)医学影像设备生成的图像,如X线图像、CT图像和MRI图像等;(4)文本形式呈现的报告结果,如医生结合自身医学知识给出的针对测量数值、信号、图像的解释和医生做出的病理诊断等;(5)叙述性的数据,如医生记录的主诉(患者口述的病情);(6)元数据文本,如关于器官、药物、疾病以及治疗方法的知识、医疗设备的参数等;(7)社会特征,如医院的机构信息、医生和患者的个人信息等。上述的部分类型本身还含有不同结构的数据,例如医学影像数据,除了图像内容,还包含成像参数等文本数据。这些不同类型的医学数据来自不同的渠道,拥有相对完整且独立的语义,虽然互为差异,但是能够相互印证相互补充[1],都从特定的角度表达了医学信息的内容和特点,构成了多样且互补的数据集合。文献[2]定义模态为描述同一事物的不同方法或角度,基于该定义,本文将这些来自多个渠道且具有相对独立语义的医学数据称为多模态医学数据。

多模态融合通过分析多模态数据中各个模态的数据在表达能力或者信息倾向上的相关性和互补性,达到对数据高层语义的深层次理解,从而可以更好地实现信息检索或数据挖掘。文本领域的研究表明,多模态融合的检索系统通常可取得优于单模态检索系统的表现[3]。当前,在一些领域已经开展了多模态数据分析的探索性研究[4-7],这些研究的思路主要考虑从不同模态的数据中分别提取特征,构成多模态特征空间,发展具有多模态特征的模式发现理论与方法[1]。对当前研究进行分类,主要集中在3个方面:基于多核学习的多模态数据分类[8],基于多字典协同表达的多模态数据建模[9]和基于深度学习的多模态数据融合[10]。但是,这些研究都还没有涉及如何对不同模态的数据本身进行融合后的表示和存储,而这需要在存储医学数据时能够对数据中的模态统一建模并充分体现模态与模态之间的关系。

现有医学数据的处理技术主要面向特定类型医学数据的存储和查询,例如医学影像存档与通信系统(Picture archiving and communication systems, PACS),它是HIS系统的重要组成部分,也是现有医学影像数据处理的典型技术。它是一个由相关设备和技术组成的综合系统,可以进行医学影像数据的获取、存储、检索和展示[11]。PACS通常使用关系型数据库对医学影像数据中的病人信息、成像参数等数据进行存储,并使用文件系统存储影像文件。关系型数据库在系统建立之初就一次性完成全部实体和实体间关系的定义,而这样只能适应系统建立之初对于存储和查询的需求。由于特定类型医学数据中的数据种类和数据之间的关系是有限且显然的,所以在系统建立之初一次性完成实体和关系的定义是合理的,但是多模态医学数据的模态融合并不是静态的定义,而是一个动态描述模态及模态间关系的过程,这就需要数据库对于数据间的拓扑结构具有较强的表达能力和足够的灵活性,能够描述复杂的数据间关系并动态适应新加载的模态和新发现的模态间关系。此外,在进行多模态或跨模态查询时,关系数据库会产生大量的表连接操作,而这是降低关系数据库查询效率的主要因素。

随着关系数据在社交网络等领域中的应用越来越广泛,图数据库逐渐成为关系数据处理中的研究热点。图数据库应用于数据之间的关联或者拓扑关系相比数据本身更加重要或者同等重要的领域。首先,在图数据库中,数据和数据间的关系通常位于同一层次[12],所有的数据都存储在一张图中,图中的点与边可以进行增添或修改,所以数据的拓扑结构具有很强的灵活性,能够适应各种拓扑变化。此外,由于图数据库引入了一定程度的抽象,使图类型数据的建模更加自然、直观,查询语言和操作直接基于图结构[13],因此也避免了关系数据库易产生大量表连接操作的缺陷。为了达到模态融合的目的,本文采用图数据库存储多模态医学数据。由于面对的是海量数据,所以数据加载过程中有必要使用并行策略。本文提出了一种并行加载医学数据的算法,该算法利用单机环境下的并行策略将多模态医学数据加载至图数据库中,具有较好的数据处理效率。

1 多模态医学数据模型

通过使用图数据库,来自不同模态的数据以及它们之间的关系被以点和边的形式存储在同一个图结构中,这从存储层面实现了不同模态数据的融合。但是,也正由于不同模态的数据被无差别地存储为同一个图结构中的点和边,导致在访问这些点或边时,无法明确它们所属的模态或模态间的关系,也就无法根据不同模态的特点读取其对应的数据。在关系型数据库中,由于数据以表的形式分别存储,每张表都具有已知的表结构,所以在查询关系型数据库中存储的数据时,数据所属的实体或者关系是显然的。通过对比,可以看出,由于图数据库侧重了数据之间关联的表达,而数据之间的关联因其复杂而灵活的特点无法借助实体关系模型进行定义,进而也就不具备类似关系型数据中的表结构,这样便导致图数据库实际上缺少对所存储数据的类型进行描述的元数据。

现有利用图数据库处理图数据的技术主要有在线查询类和离线挖掘类两类。在线查询类的如 TAO[14] 和 FlockDB[15],都应用于社交网络中。由于社交网络中的数据全部来源于社交网站,数据的采集渠道相对单一,所以数据种类有限且相对固定。如图 1 是 Facebook 实体图的一个示意片段[16],实体的种类包括用户、页面和位置等,不同实体之间的关系是固定的,使用这些数据的方式也是明确的,所以此类图数据库并没用对数据所属的模态及模态间关系进行分析和描述。离线挖掘类的如 Pregel[17],更强调底层计算模型的设计,而数据的类型、采集渠道和对应的语义则是由使用这些技术的用户负责维护和解释的,同样没有提供对数据所属模态以及模态之间关系进行描述的方法。相比社交网络中的数据,医学数据存在大量不同的类型,不同类型的数据之间也存在着复杂的关系,所以对医学数据所属模态以及模态间关系进行抽象和表示是必要的,这也是对医学数据进行模态融合从而提高信息检索和数据挖掘效果的本质原因。

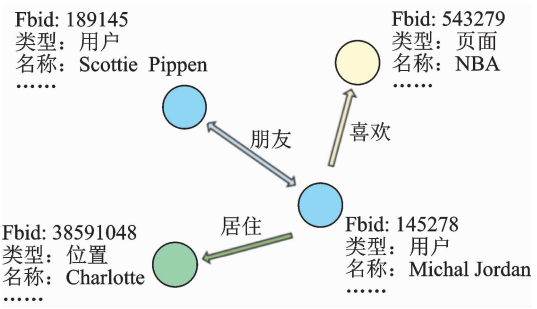

图 1 Facebook 实体图

Fig. 1 Entity graph of Facebook

为了解决这个问题,本文提出了一种数据模型,即模态结构图。该模型以图的形式对医学数据所属的不同模态和模态间关系进行描述。具体来说,对图数据库中存储着具体数据的“大图”按照它们所属的模态或模态间关系进行投影,可以得到数据所属的模态与模态间关系的“小图”,这个“小图”就是模态结构图。在模态结构图中,点代表一个模态,点的属性描述该模态具有的属性,点与点之间的边代表模态与模态之间的关系,边的属性描述该关系所具有的属性,图数据库中的点和边都是模态结构图中点和边的实例。以下给出关于模态、模态间关系和模态结构图的具体定义。

定义 1 模态 mode 可以用五元组{Name, Dep, Pro, Uni, UniT}表示。其中,Name 为模态名称;Dep=$\{\langle \text{Name}_i, \text{DepT}_i\rangle \mid i\in\{1,\cdots,n\}\}$表示该模态所依赖的其他模态,其中$\text{Name}_i$是被依赖模态的名称,$\text{DepT}_i\in${唯一依赖,非唯一依赖},表示依赖类型,唯一依赖表示被依赖模态对应的点只能拥有一个当前模态的点,而非唯一依赖则可以有多个;Pro 是该模态的属性集合;Uni$\subseteq$Pro 是具有唯一性的属性集合,集合中属性的值在规定范围内具有唯一性;UniT$\in${全局唯一,依赖唯一},表示唯一性属性的适用范围,其中全局唯一指 Uni 中的属性值在整个图数据库中唯一存在,依赖唯一则指在依赖同一个的点的范围内唯一存在。

例 1 由于医学数据中多种模态的数据往往都和患者的个人信息等社会特征相关联,所以首先建立一个社会特征模态 M_1 用来表示医学数据中关于患者的个人信息。$M_1=\langle$Patient, $\varnothing$, Pro, Uni,

UniT〉,其中 Dep 为空集表示该模态不依赖于其他模态;Pro={"ID", "Name", "Sex", "Age"},Pro 中属性的值都可以从 HIS 中按照对应的格式分别采集到;Uni={"ID"},ID 属性作为该模态的唯一性属性且 UniT 选择全局唯一,确保病人的 ID 属性值在数据库范围内不会重复。然后建立一个叙述性的文本模态 M_2 用来表示由医生记录的病人主诉,M_2=〈"Complaint",Dep,Pro,∅,Null〉,其中 Dep={〈"Patient",DepT〉},DepT 选择非唯一依赖,因为一个病人每进行 1 次诊断,就会生成 1 份主诉;Pro={"Content"},该属性只能采集到不含格式的文本数据;由于只根据文本无法判断数据是否唯一,所以该模态的 Uni 为空集,UniT 无效。由于医学数据中很多信号数据、图像数据都是用来描述某一器官的,所以还可以再建立一个元数据模态 M_3 表示脑部器官,M_3=〈"Head",Dep,Pro,∅,Nul〉, 其中 Dep={〈"Patient",DepT〉},DepT 选择唯一依赖,因为一个病人只能拥有一颗头颅;Pro 中的属性可以从相关的医学知识库中进行采集,获得能够描述脑部器官的相关参数;Uni 为空集,因为从医学元数据中获取的数据无法区分出不同的脑部器官,UniT 无效。然后再建立一个医学图像模态 M_4=〈"Image",Dep,Pro,Uni,UniT〉,其中 Dep={〈"Head",DepT〉},DepT 选择非唯一依赖,因为对同一器官的的影像检查会生成多张图像;Pro 中除了图像的内容,还包括图像的宽度、高度、采样数等图像的格式数据,这些数据都可以从 DICOM 等类型文件中采集到,Uni 选择图像像素作为唯一性属性且 UniT 选择全局唯一,因为图像内容本身是独一无二的。最后,再建立一个文本报告形式的模态 M_5 用来表示病人的诊断结果,M_5=〈"Diagnosis",Dep,Pro,Uni,UniT〉,其中 Dep={〈"Patient",$DepT_1$〉,〈"Complaint",$DepT_2$〉,〈"Image",$DepT_3$〉},$DepT_1$,$DepT_2$和$DepT_3$都选择非唯一依赖,因为病人可以拥有不止一份诊断结果;Pro={"Content"},该属性也只能采集到不含格式的文本数据;其 Uni 也为空集,UniT 也无效。

定义 2　模态间关系 Relation 可以用六元组〈Name,BegM,EndM,Type,Pro,Uni〉表示,其中,Name 为关系名称;BegM 为该关系所关联的起点模态;EndM 为该关系所关联的终点模态;Type∈{单边,重边}为关系类型,其中单边表示在起点模态和终点模态对应的点之间只能存在一条当前关系对应的边,重边则可以存在多条;Pro,Uni 和 Mode 中的意义相同,且当 Type 选择为单边时,Uni 无效。

例 2　建立例 1 中 M_1 和 M_2 的模态间关系〈"Describe",M_1,M_2,Type,Pro,Null〉,该模态间关系的名称为"Describe";起点模态为 M_1,终点模态为 M_2,表示主诉数据是由患者叙述的;Type 选择为单边,即一个表示患者的社会特征模态对应的点和一个表示患者主诉的叙述性模态对应的点之间只能存在一条叫做"Describe"的边;Pro={"Date"},表示病人叙述主诉的日期;Uni 无效。

定义 3　模态结构图可以用二元组〈{Mode},{Relation}〉表示,其中{Mode}是全部的模态集合,{Relation}是全部的模态间关系的集合。

例 3　建立例 1 所述模态的模态关系图,如图 2 所示。其中 M_1 和 M_2 之间存在"描述"关系;M_1 和 M_3 之间存在"所属"关系;M_3 和 M_4 之间存在"成像"关系;M_2 和 M_5,M_4 和 M_5 之间存在"参考"关系;M_1 和 M_5 之间存在"诊断"关系。

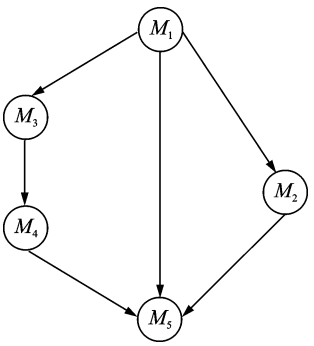

图 2　模态结构图示例

Fig. 2　Example of modal structure graph

从底层数据存储的角度来看,由于模态结构图提供了图数据库所缺少的元数据,所以借助模态结构图,图数据库除了本身就具备的对复杂拓扑关系强大的表达能力,还拥有了关系型数据库对数据良好的定义能力。因此,基于模态结构图和图数据库的数据查询与检索就同时具备了图数据库和关系型数据库两者的优势:(1)如引言中所述,相比关系型数据库,图数据库在执行多模态、跨模态的查询与检索时,具有非常显著的效率优势,而且查询语言也十分直观;(2)由于真实多模态医学数据中的拓扑关系非常复杂,例如存在大量的孤立数据点或互相不连通的数据点集,所以在查询与检索时,仅凭数据间的相互关系无法完整地对数据库中的数据进行查询与检索,但是

图数据库中的数据按照其所属的模态或模态间关系都映射到了模态结构图，所以仍然可以通过数据所属的模态访问到所有该模态的数据，进而完成对数据的查询与检索。从上层数据语义的角度来看，由于模态结构图同时也是模态语义独立性和相互关联性的一种抽象和表达，所以其本身就是一种以图结构表示的多模态融合策略。借助模态结构图，可以有效地描述基于数据融合的查询与检索需求，进而在此基础上设计相应的检索与查询算法。

综上所述，通过图数据库存储数据并借助模态结构图对数据所属模态或模态间关系进行描述，多模态医学数据可以实现从存储层面到语义层面的模态融合。

2 多模态并行加载算法

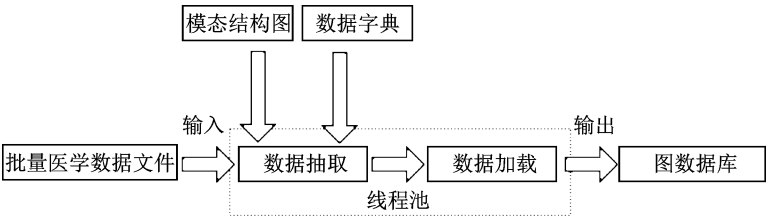

图 3 多模态并行加载算法整体流程图

Fig. 3 Flowchart of multimode parallel loading algorithm

图 3 是多模态并行加载算法的整体流程图。并行加载算法在图中虚线所示的线程池中运行，分为数据抽取和数据加载两个阶段。当批量的医学数据文件输入后，线程池中的空闲线程会随机取得一个尚未加载的文件进入数据抽取阶段。在数据抽取阶段，线程首先根据模态结构图并借助指定的数据字典将属于不同模态或者模态间关系的数据从文件中抽取出来，并整合成对应点或边的数据集合，然后把从一个文件中抽取出的全部数据集合组成一个图更新事务序列，最后，将这个序列发送给一个空闲线程进入数据加载阶段。在数据加载阶段，线程在图数据库中建立各个数据集合对应的点或边，完成整个加载过程。

2.1 预处理

由于并行加载算法需要根据具体的模态结构图和数据字典才能对批量医学数据进行建模和提取，进而将数据加载进图数据库中，所以在执行加载算法之前，首先需要给出待加载数据的模态结构图和数据字典。在预处理过程中，由人工对待加载医学数据的模态组成和数据融合的需求进行综合分析，并按照第 1 节中模态结构图的定义给出具体的模态结构图。通过查询各种模态数据对应的行业标准或者数据本身的特点，人工可以很容易地生成解析不同模态数据的数据字典。

2.2 加载顺序构建算法

由于模态之间存在依赖关系，所以需要根据模态结构图中这些关系遍历模态结构图生成模态的加载顺序。线程的数据抽取和加载都将按照这个顺序依次处理模态结构图中的全部模态。算法的伪代码见算法 1。

算法 1 加载顺序构建算法

输入：模态网络 N

输出：加载顺序表 L

```
(1) while (L. size() < N. Mode. size())
(2)   current = L. size();
(3)   for each mode in N. Mode
(4)     if (mode 未加入 L)
(5)       初始化可添加标识 flag 为 true
(6)       list = mode. Dep;
(7)       for each m in list
(8)         if (m 未加入 L)
```

```
(9)              flag= false;
(10)         if (flag)
(11)           向 L 中加入 mode;
(12)   if (L. size() == current && L. size() < N. Mode. size())
(13)      模态结构图存在依赖环构建失败并退出
(14) return L
```

算法 1 不断遍历模态网络 N 中的全部模态,找出其中还未加入 L 的模态,判断它的依赖模态是否已全部加入 L,若是则将该模态也加入 L,否则继续遍历,直到 L 中所含模态数量等于 N 中的模态数量时停止,这时 L 即为按照模态间的依赖关系进行排序所得的列表。其中步骤(2)中 current 在每次遍历开始时都赋值为当前 L 中模态的数量,在每次遍历结束时,比较当前 L 中所含模态数量和 current,若两者相等,则可知本次遍历没有任何一个模态被加入 L,若此时仍有模态未加入 L,则可知人工给出的模态结构图存在依赖环,而存在依赖环的模态结构图是无法被用来加载数据的,算法退出。

2.3　多模态数据抽取

模态结构图只是对多模态医学数据中所包含的模态及其关系进行建模后形成的一个数据模型,在进行具体数据的加载时,需要根据该模型得到每个文件中各个模态及其关系对应的具体数据。本文给出了一个根据模态结构图定义从数据文件中抽取具体数据的算法,该算法在数据文件中寻找并抽取出每一个模态和模态间关系的属性的值,拼装成待创建的点和边对应的属性集合,最后这些属性集合组成了一个图更新事务序列作为该算法的输出。该抽取算法的伪代码见算法 2。

算法 2　DataExtract

```
输入:模态网络 N,加载顺序表 L,文件 F,数据字典 D
输出:图更新事务序列 UpdateSeq
(1) for each m in L
(2)   初始化属性值列表 valueList
(3)   for each pro in m. Pro
(4)     根据 D 在 F 中找到 pro 的值 value
(5)     valueList [pro] = value
(6)   生成图更新事务 T
          T=<m. Name, m. Dep, proValue, m. Uni, m. UniT>
(7)   将 T 加入 UpdateSeq
(8) for each r in N. Relations
(9)    初始化属性值列表 valueList
(10)   for each pro in r. Pro
(11)    根据 D 在 F 中找到 pro 的值 value
(12)    value [pro] = value
(13)   生成图更新事务 T
           T=<r. Name, r. BegM, r. EndM, r. Type, valueList, r. Uni>
(14)   将 T 加入 UpdateSeq
(15) return UpdateSeq
```

算法 2 中按照模态加载顺序表中的顺序,循环执行步骤(2—7),抽取出每个模态的属性值,生成包含模态名称、依赖模态、属性值、唯一性属性和唯一性适用范围的新建点事务 T,并加入图更新事务序列 UpdateSeq。同样地,循环执行步骤(9—14),抽取出每个模态间关系的属性值,生成包含关系名称、起点模态、终点模态、关系类型、关系属性值和唯一性属性的新建边事务,并加入 UpdateSeq。

2.4 并行加载

通过数据提取,一个数据文件被转换成了一个图更新事务序列,该序列将继续被分发给线程池中的一个空闲线程,该线程将按照序列中的顺序依次执行其中的图更新事务完成数据的加载。

由于线程池中的若干线程同时执行图更新事务,待新建点的依赖点及其唯一性属性和待新建边的唯一性属性就成为了线程间的临界资源,线程需要以互斥的方式访问这些数据,所以并行算法需要选择一种协调线程运行的机制保护这些临界资源不被破坏。常用的机制包括互斥量等线程同步机制和线程通信机制。在并行加载过程中,临界资源的使用有如下特点:(1)同时访问临界资源的线程在加载过程中是随机出现的,事先无法判断哪些线程会同时访问某个临界资源;(2)线程在临界区内只对数据库中某些数据的状态进行检查,并不需要在线程间传递数据或者其他复杂操作,所以线程进入临界区内的时间非常短;(3)所有临界资源的数量都是 1,并且线程对于临界资源的访问是有序的。根据这些特点可知,临界资源被同时访问的随机性将增大线程间通信的复杂程度,而资源使用的单一性和短暂性将使线程通信的作用得不到真正的发挥;同时,临界资源的有序访问和数量的单一性避免了线程同步可能出现的死锁风险。因此,文中采用线程同步中的互斥量实现加载算法的并行化。

并行加载算法在执行接收到的图更新事务序列中的每一个事务时,若事务为新建点事务,则首先调用图 4 中所示算法建立该点;若事务为新建边事务,则调用算法 3 建立该边。其中,并行加载算法在执行每一个图更新事务序列时,创建两个互斥量列表 MList1 和 MList2,其中 MList1 中的互斥量用来同步多个线程以互斥的方式访问数据库中的点,MList2 中的互斥量用来同步多个线程对某个唯一性属性的互斥访问。同时,由于 MList1 和 MList2 本身也是临界资源,所以并行加载算法还需要创建两个互斥量 M_1 和 M_2 来确保对 MList1 和 MList2 的修改是互斥的。除此之外,并行加载算法还创建 nodeList 记录当前已成功建立的点,从而在之后执行新建边事务时仍然可以找到这些点。

图 4 所示算法首先检查待建立点的依赖关系及唯一性是否符合模态结构图的定义,如果满足则在图数据库中建立该点。首先检查待创建点依赖关系是否正确,若全部依赖点的依赖关系和“依赖唯一”的属性都满足模态结构的定义,则开始在图数据库中建立该点。其中,为了检查和确保待建立点的依赖关系正确,需要对待建立点的依赖点对应的互斥量进行加锁,图 4 中自 t.Dep 赋值开始的全部流程均处于或可能处于依赖点对应互斥量加锁的范围内;为了确保点的全局唯一属性的全局唯一性,同样需要对全局唯一性属性对应的互斥量进行加锁,图中 name 为空分支的全部流程除了可能处于依赖点对应互斥量加锁的范围外,还有可能处于全局唯一性属性对应互斥量的加锁范围内。name 为空的流程中的“依赖边”不存储具体数据,只用来表示依赖关系,其他流程正是利用“依赖边”完成了对依赖关系的检查。

算法 3 CreateEdge

输入:新建边事务 T,已建立点列表 nodeList,点锁列表 MList1

(1) beginID = nodeList[T. BegM. Name]
(2) endID = nodeList[T. EndM. Name]
(3) if (beginID 和 endID 均存在)
(4) 对 MList1[beginID]加锁
(5) 对 MList1[endID]加锁
(6) if (T. Type 为单边)
(7) if (ID 为 beginID 和 ID 为 endID 的点之间不存在名称为 T. Name 的边)
(8) 建立 T. Name 的边
(9) else
(10) if (ID 为 beginID 和 ID 为 endID 的点之间不存在 Uni 为 T. Uni 的边)
(11) 建立 T. Name 的边
(12) 对 MList1[beginID]解锁

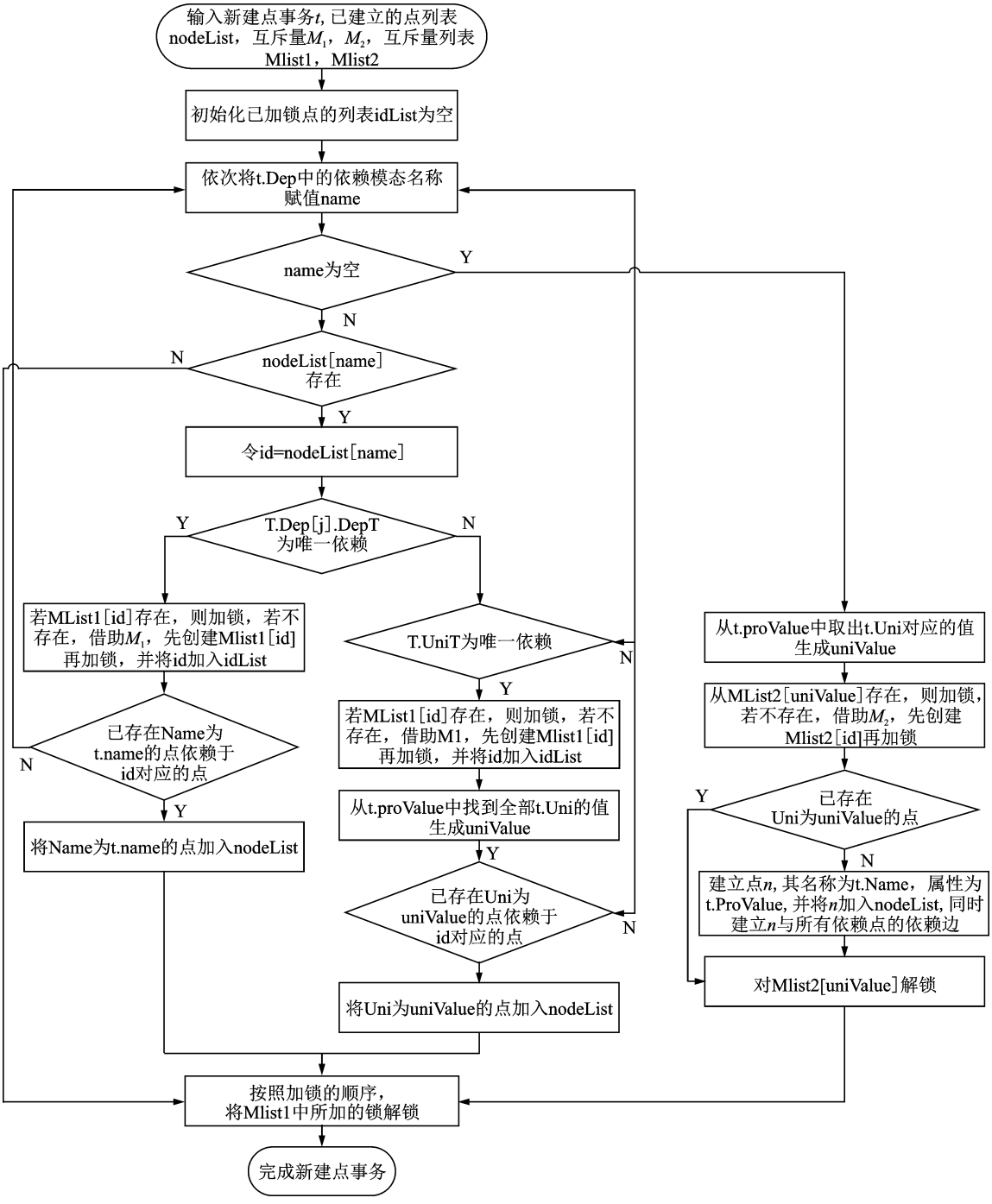

图 4　新建点算法流程图

Fig. 4　Flowchart of new node creation algorithm

(13)　对 MList1[endID]解锁

算法 3 根据边的类型采取不同策略完成两点间边的建立。其中,步骤(1—2)从 nodeList 中获取当前边需要连接的起点和终点的 ID;步骤(3)判断起点和终点是否存在,若存在,步骤(4—8)建立当前边,

其中，步骤(4—5)首先锁定这两个点对应的互斥量，然后步骤(6)判断边的类型，若为单边，则(7)检查两点间是否已存在当前类型的边，已存在则不再建立当前边，否则步骤(8)建立当前边；若为重边，则步骤(10)检查两点间是否已存在相同唯一性属性的边，若已存在，同样不再建立当前边。

2.5 算法的进一步优化

为了确保模态间的依赖性不被破坏、唯一性属性的唯一性得到保证，并行加载算法引入了大量互斥量用于协调和同步不同线程的算法执行过程，所以若同时加载的数据在依赖性和属性唯一性方面存在较强的相关性时，将使多个线程在同一时间段频繁试图对相同的互斥量执行加锁操作，这会花费较多时间使线程间互相等待其他线程释放互斥量，降低了并行算法的并行加速比。根据对现有医学数据特点的分析，在批量医学数据文件中，连续的数据文件往往描述同一对象。例如在按序加载病人的脑部 CT 图像序列 $D=\{d_1,d_2,d_3,\cdots,d_i,\cdots,d_j,\cdots,d_n\}$ 时，其中连续的文件序列片段组成了不同病人的脑部切片序列，即 $\{\underbrace{d_1,d_2,d_3,\cdots}_{p_1},\cdots,\underbrace{d_i,\cdots}_{p_i},\cdots,\underbrace{d_j,\cdots,d_n}_{p_n}\}$，其中 d_i 为存储有 CT 图像的 DICOM 文件，p_i 对应某个病人。在某个切片序列对应的若干 DICOM 文件 $\{d_i \mid d_i \in p_j\}$ 中，每一个 d_i 都存储了 p_i 的信息，所以若该序列中的 d_i 被分给了不同的线程进行加载，将导致这些线程在检查 p_i 的 ID 等唯一性属性和检查 p_i 与对应医院的依赖关系时共享相同的互斥量，从而造成并行加速比的下降。通过实验发现，若数据序列中数据文件的个数和线程数相当时，这种线程的竞争现象十分显著。

为了降低不同线程加载数据的相关性，当输入原始的批量医学数据文件序列后，对其中的数据文件执行算法 2 之前，使用一个随机函数 $f(D)=D'$ 生成最终的数据文件序列 $D'=\{d'_1,d'_2,d'_3,\cdots,d'_i,\cdots,d'_j,\cdots,d'_n\}$，通过该函数，$D$ 中连续的数据文件被随机分布在 D' 中，如图 5 所示。如此，不同线程同时加载的文件的相关性被尽可能降低。反映到图数据库中，这样做可以使每个线程尽可能在图数据库中的不同区域工作，使新建的点和边在图中尽可能均匀分布，降低了不同新建点共同拥有同一个依赖点的概率和新建边需要连接相同点的概率。

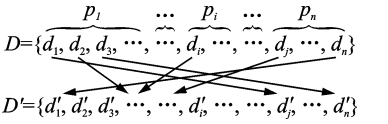

图 5 生成用于加载的数据文件序列

Fig. 5 Generation of data files sequence for loading

3 实验结果与分析

为了评估模态结构图在具体建模过程中的合理性、并行加载算法的正确性和执行效率，本文使用真实脑部 CT 的 DICOM 文件进行建模和加载实验。DICOM 文件除了包括 CT 图像、成像参数、医院和患者信息的社会特征，部分文件还包含有诊断数据。DICOM 文件涵盖了医学数据中较有代表性的多个模态，所以使用其进行实验具有一定的说服力。实验环境：操作系统版本 OS X EI Capitan 10.11.4；内存 16 GB；处理器 2.2 GHz Intel Core i7，物理四核，虚拟八核。

在 5 个数据集和 5 个线程池上分别执行并行加载算法。图 6 给出了图数据库中存储的部分数据的可视化显示结果。由图 6 可知，加载算法按照模态结构图正确地完成了数据的加载，主要体现为图数据库中的点和边上的属性均按照模态结构图中的定义被正确地从数据文件中提取，不同模态对应的点也被对应的边正确连接，不同模态间的依赖关系也符合模态结构图中的规定。

表 1 给出了算法的执行时间。图 7 给出了算法运行时间随着数据集大小和线程个数变化的变化趋势。从表 1 可以看出，当线程数量增大，算法的运行时间不断下降，但随着线程数量越来越大，运行时间的下降幅度变得越来越不明显，甚至当线程数量从 8 线程变为 16 线程时，运行时间反而有所增加。这是由于硬件环境能够提供的同时运行的线程数是一定的，当线程数增大时，并行带来的系统开销也在不断增加，所以算法运行时间的减少幅度随着线程数的增加会趋于不显著，当线程数超过了硬件环境能够提供的最大并行线程数时，系统开销的增长幅度超过了算法运行时间的减少幅度，导致了运行时间不降

反升。从图 7 可以看出,当线程数较少时,随着数据集逐渐增大,算法运行时间的增长速度在加快。这是由于数据集越大,需要维护的互斥量就越多,遍历寻找某个互斥量所需要的时间就越长,由此增加的运行时间比较显著;当线程数较多时,随着数据集的增大,虽然每个线程寻找指定互斥量的时间依然在增加,但多线程的并行执行在一定程度上抵消了由此增加的运行时间。

图 8 给出了对批量 DICOM 文件进行顺序加载和随机加载的运行时间差值的变化趋势。从实验结果可以看到,随着数据集的增大,随机加载相比顺序加载减少的运行时间呈增长趋势,但是增长逐渐趋于平缓且在数据集大小为 520 个文件后,不再继续增长。原因是当数据集较小时,其中的大部分 DICOM 文件都来自同一个患者的脑部切片序列,不论是顺序加载还是随机加载,线程间对互斥量的竞争仍然非常频繁;当数据集增大后,由于数据集中的 DICOM 文件属于多个患者,这时随机加载就降低了不同线程在同一时刻需要锁定同一个互斥量的概率,但当数据集继续增大,由于数据集中属于每个患者的 DICOM 文件个数基本相同,所以这个概率将逐渐趋于稳定。

由实验结果可知,当选择合适的线程数后,随着数据集逐渐增大,算法运行时间的增长速度近似线性,同时结合算法的优化策略和算法加载不同大小数据集所消耗的绝对时长,可知该算法能够在较短时间内完成批量多模态医学数据的建模和加载。

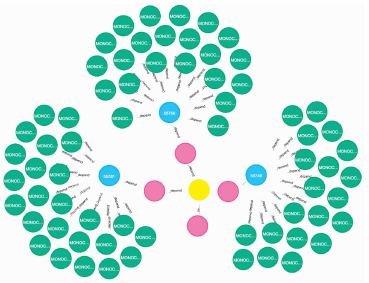

图 6　图数据的可视化

Fig. 6　Visualization of graph data

表 1　算法的执行时间

Tab. 1　Execution time of the algorithm　s

文件个数	18	51	120	291	520	1 026
1 线程	0.66	1.55	3.63	8.11	18.51	44.74
2 线程	0.43	1.40	2.73	7.38	15.20	25.16
4 线程	0.32	0.61	2.16	4.89	7.91	17.10
8 线程	0.25	0.52	1.38	3.07	4.67	11.31
16 线程	0.34	0.91	1.73	3.79	6.80	13.81

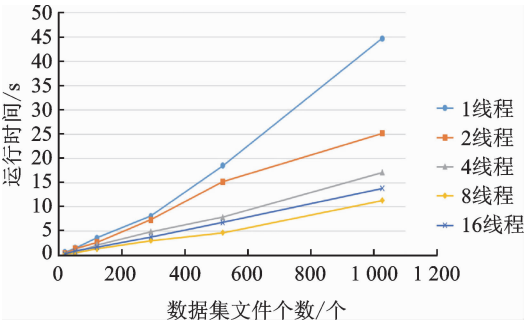

图 7　算法执行时间随数据集大小和线程个数的变化趋势

Fig. 7　Trend of execution time with data set size and the number of threads

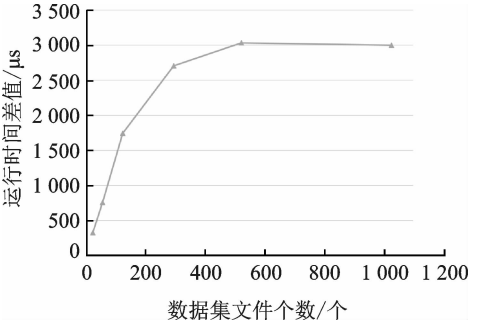

图 8　算法优化前后运行时间差值的变化趋势

Fig. 8　Trend of running time difference before and after optimization

4　结束语

本文基于医学数据中数据类型的多样性,分析了医学数据的多模态特性,同时针对该特性,分析了模态融合对于数据的表示和存储的重要意义。为了实现数据表示和存储阶段的模态融合,通过对比现有技术,提出了使用图数据库表示和存储医学数据,同时为了解决图数据库缺乏元数据的缺陷,又提出

了一种新的数据模型，即模态结构图，用于多模态医学数据的建模和表示。该模型以点和边表示数据中的不同模态及其相互关系，为图数据库提供了关于所存储数据的元数据，同时也提供了一种图结构的模态融合方案。基于此模型，提出了一种并行加载算法，该算法以并行的方式抽取出分属不同模态和模态间关系的数据并存储至图数据库中。实验结果表明，并行加载算法能够保证数据按照模态结构图的定义正确建模并具有较好的数据处理效率。下一步将主要研究如何利用图数据库中的多模态数据和模态结构图构建索引结构进而实现多模态、跨模态的信息检索以及如何在模态融合后的数据中进行数据挖掘。

参考文献：

[1] 梁吉业，钱宇华，李德玉，等. 大数据挖掘的粒计算理论与方法[J]. 中国科学：信息科学，2015，45(11)：1355-1369.
Liang Jiye, Qian Yuhua, Li Deyu, et al. Granular computing theory and method for large data mining[J]. SCIENTIA SINICA Informationis, 2015, 45(11): 1355-1369.

[2] 钱宇华，成红红，梁新彦，等. 大数据关联关系度量研究综述[J]. 数据采集与处理，2015，30(6)：1147-1159.
Qian Yuhua, Cheng Honghong, Liang Xinyan, et al. Review for variable association measures in big data[J]. Journal of Data Acquisition and Processing, 2015, 30(6): 1147-1159.

[3] Lee J H. Analyses of multiple evidence combination[J]. Acm Sigir Forum, 1996, 31(SI): 267-276.

[4] Friston K J. Modalities, modes, and models in functional neuroimaging[J]. Science, 2009, 326(5951): 399-403.

[5] Zhang D, Shen D. Multi-modal multi-task learning for joint prediction of multiple regression and classification variables in Alzheimer's disease[J]. Neuroimage, 2012, 59(2): 895-907.

[6] Zeng Z, Pantic M, Roisman G I, et al. A survey of affect recognition methods: Audio, visual, and spontaneous expressions [J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2009, 31(1): 39-58.

[7] Guo Z, Zhang Z, Xing E, et al. Enhanced max margin learning on multimodal data mining in a multimedia database[C]// Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. New York, USA: ACM, 2007: 340-349.

[8] Bucak S S, Jin R, Jain A K. Multiple kernel learning for visual object recognition: A review[J]. IEEE Transactions on Pattern Analysis & Machine Intelligence, 2014, 36(7): 1354-1369.

[9] Wang S, Zhang D, Zhang L, et al. Relaxed collaborative representation for pattern classification[C]// Proceedings of the 2012 IEEE Conference on Computer Vision and Pattern Recognition(CVPR). Providence, Rhode Island, USA: IEEE, 2012: 2224-2231.

[10] Wu P, Hoi S C H, Xia H, et al. Online multimodal deep similarity learning with application to image retrieval[C]// Proceedings of the 21st ACM International Conference on Multimedia(MM2013). New York, USA: ACM, 2013: 153-162.

[11] 胡伟标. 医学影像存储和通讯系统[J]. CT 理论与应用研究，2001，10(1)：18-20.
Hu Weibiao. Picture archiving and communication system[J]. CT Theory and Applications, 2001, 10(1): 18-20.

[12] Angles R, Gutierrez C. Survey of graph database models[J]. ACM Computing Surveys, 2008, 40(1): 178-187.

[13] Angles R. A comparison of current graph database models [C]//2012 IEEE 28th International Conference on. Data Engineering Workshops (ICDEW). Arlington, Virginia, USA: IEEE, 2012: 171-177.

[14] Bronson N, Amsden Z, Cabrera G, et al. TAO: Facebook's distributed data store for the social graph[C]//2013 USENIX Annual Technical Conference. San Jose, California, USA: USENIX, 2013: 49-60.

[15] Freels, boazavital, rcohen, et al. Twitter FlockDb [EB/OL]. https://github.com/twitter/flockdb, 2012-04-29/2016-08-01.

[16] 张俊华. 大数据日知录：架构与算法[M]. 北京：电子工业出版社，2014：274.
Zhang Junhua. Big data: Architecture and algorithm[M]. Beijing: Publishing House of Electronics Industry, 2014: 274.

[17] Malewicz G, Austern M H, Bik A J C, et al. Pregel: A system for large-scale graph processing[C]//Proceedings of the 2010 ACM SIGMOD International Conference on Management of Dada. New York, USA: ACM, 2010: 135-146.

作者简介：

翟霄(1990-)，男，硕士研究生，研究方向：数据挖掘、图像处理、海量信息处理等，E-mail：zhaixiao@hrbeu.edu.cn。

潘海为(1974-)，男，副教授，研究方向：并行数据库、数据挖掘、海量信息处理等。

谢晓芹(1973-)，女，副教授，研究方向：面向服务的计算、网络模式下知识处理、软件复用、智能信息处理等。

张志强(1973-)，男，教授，研究方向：数据库、Web 信息检索、软件工程等。

韩启龙(1974-)，男，教授，研究方向：隐私保护、大数据处理、移动物联网、社会网络。

（编辑：王静）