一种基于聚类模式的RDF数据聚类方法

袁 柳[1] 张龙波[2]
(陕西师范大学计算机科学学院 西安 710062)[1] (山东理工大学计算机科学与技术学院 淄博 255049)[2]

摘 要 如何有效管理并利用日益庞大的RDF数据是当今Web数据管理领域面临的挑战之一。对大规模的RDF数据集进行聚类操作从而得到数据集的有效划分是RDF数据存储和应用时通常采取的策略。针对现有RDF聚类过程中忽略RDF三元组自身模式特征的问题,在对RDF聚类结果的形式深入分析的基础上,定义了3种不同类型的聚类模式,从而提出基于模式的聚类方法。通过对RDF数据集的重新描述,自动生成适用于RDF数据集特征的聚类模式,在此基础上实现数据聚类的任务。在不同测试集上的实验结果验证了所提方法的正确性和有效性。

关键词 RDF,聚类,开放关联数据,聚类模式

中图法分类号 TP311.1 **文献标识码** A **DOI** 10.11896/j.issn.1002-137X.2015.10.054

Cluster Pattern Based RDF Data Clustering Method

YUAN Liu[1] ZHANG Long-bo[2]
(School of Computer Science, Shaanxi Normal University, Xi'an 710062, China)[1]
(School of Computer Science and Technology, Shandong University of Technology, Zibo 255049, China)[2]

Abstract How to manage and exploit the large mount of RDF dataset availably has become a vital issue in Web data management field. In order to partition the large scale RDF dataset for efficient data processing, clustering is usually adopted. The related researches tend to use classical clustering methods, and neglect the structure features of RDF triples. This paper analyzed the RDF clustering results intensively, and defined three types of cluster patterns. Based on the cluster patterns, a novel RDF data clustering strategy was proposed. By redescribing the RDF dataset, the cluster patterns can be generated automatically. The experiments on different test benches prove the accuracy and efficiency of the new method.

Keywords RDF, Clustering, Linked open data, Clustering pattern

1 引言

Web发展至今已经从文档的Web(Web of Document)发展演化成为数据的Web(Web of Data)。尤其是随着开放关联数据(Linked Open Data,LOD)[1]项目规模的不断壮大,越来越多的数据以遵循关联数据规范的形式在Web上发布。关联数据最典型的特征之一就是数据以RDF(Resource Description Framewok)的形式呈现。统计数据显示,截至2014年9月,已发布的LOD数据集的总量已达1048个,包含了约700亿的RDF三元组(http://stats.lod2.eu)。面对着海量的RDF数据,越来越多的研究致力于如何有效地对这些数据进行存储、索引和查询[2,3];同时利用数据挖掘技术发现RDF数据集中有价值的信息和知识。由于数据规模巨大,通常需要对数据集进行分块处理,因此聚类的思想通常被用于RDF数据集以提升应用的性能。例如,可利用层次聚类的结果构建层次化的路径索引以提高RDF数据查询处理的效率[4,5];分布式环境下,可根据聚类结果对RDF数据集进行有效的分割并将其存储在不同的机器上,同时尽可能地降低分布式查询处理过程中节点间数据交换的代价[3];可利用聚类结果发现同构的RDF资源集合,进而实现基于本体的知识发现[6,7]。由此可见,对RDF数据聚类是一种利用RDF资源的有效途径,研究适合于RDF数据特征的聚类技术是大数据环境下必须面对的问题。然而针对这一问题的研究成果目前还较为少见,已有的研究方法和技术在聚类过程中很少考虑到RDF三元组的结构特征和RDF数据之间的链接关系。针对这一问题,本文以实现有效的RDF数据聚类为目标,提出一种基于RDF数据特征的聚类方法。

2 相关工作

现有的RDF聚类技术几乎都基于经典的聚类方法[8],将这些经典的算法用于RDF数据集主要面临着两个困难:首先,传统的聚类方法作用的对象为数据实例,而不是大规模地将资源相互链接的RDF图。因此使用传统方法实现RDF数据聚类的前提条件是从RDF数据集中抽取实例。众所周知,RDF图中的实例抽取是一个非常耗时且复杂的过程[9],而且实例抽取可能丢失RDF链接所表达的重要的语义信息。其

到稿日期:2014-10-16 返修日期:2014-12-05 本文受国家自然科学基金项目:云计算环境下旅游信息个性化服务模型研究(41271387)资助。
袁 柳(1979-),女,博士,讲师,主要研究方向为Web数据管理、语义信息检索,E-mail:yuanliu@snnu.edu.cn;张龙波(1968-),男,博士,教授,主要研究方向为数据流与数据挖掘。

次，聚类结果类簇之间距离的度量也是一个困难的过程，无论是利用所抽取实例的特征计算实例间的相似性，还是利用RDF图的拓扑结构以子图间的重叠程度来表示实例间的相似性，都忽略了RDF数据自身的语义信息。可以看出，以实例抽取为前提的经典聚类思想并不适合于RDF这一特殊的数据模型。一般地，基于经典聚类方法的RDF聚类都将RDF聚类转化为图聚类，借鉴图挖掘方法，根据某些标准将图中节点分割为若干个组件，同时提出识别RDF图内相似结构子图的方法，但使用这种方式所发现的图节点集合不能通过RDF图中路径所对应的属性序列来区别，即它只关注了路径的结构而忽视了路径所表达的属性-对象模式。因此目前基于图挖掘技术的RDF数据聚类研究也没有在聚类过程中充分利用RDF模型蕴含的语义信息。

一般地，聚类算法的结果都是将数据集分割为若干个互不相交的类簇。而在现实情况下，一个个体(或RDF资源)可能同时属于多个不同的概念，因此RDF数据集上的聚类结果应该允许一个资源同时属于多个类簇，即允许类簇的重叠。但是基于个体间相似性计算的RDF数据集聚类过程很难实现重叠类簇的目标，将不同的资源归属为同一个类簇的标准也难以定义。在与数据语义相关的研究领域中，个体间的相似性计算可以基于数据层级和知识层级，每一级别又分别可实现基于值匹配、个体匹配和数据集匹配的相似性计算。对于RDF图中的资源应该采用什么样的相似性匹配策略，目前还没有令人满意的研究成果[10]。有研究通过计算连接图中节点的链接(边)间的相似性来实现可重叠聚类的目标[11]，该思路可用于RDF数据聚类过程，但在计算链接(边)的相似性的过程中如何利用RDF图中链接的语义信息，而不仅仅根据拓扑结构计算边的相似性，仍需要更深入的研究。

目前国内RDF相关研究主要集中在RDF数据查询处理优化方面[15,16]，关于RDF聚类方法比较成熟的研究成果仍非常少见。

3 基于RDF图结构的聚类特征分析

3.1 RDF数据聚类的定义

为了方便下文的论述，首先对文中出现的主要概念进行形式化的描述和说明。

RDF三元组(RDF triple)。给定一个URI集合R、空结点集合B、文字描述集合L，一个RDF三元组t是形如(s,p,o)的三元组，其中$s\in R\cup B$，$p\in R$。这里的s通常称为主语(subject)、资源(resource)或主体，p称为谓词(predicate)或属性(property)，o称为宾语(object)、属性值(value)或客体。后续章节使用s,p,o分别表示组成三元组的主语、谓词和宾语。

RDF数据图(RDF data graph)。RDF数据图G是一个三元组(V,E,L)，其中V为顶点集合，E为边的集合，L为标签集合且$L=L_v\cup L_p$，L_v为顶点的标签集合，L_p为边的标签集合。这里的V对应三元组中的s和o，边对应p。即RDF数据图是以s和o为顶点、p为边并且顶点和边上都带有标签的图。

三元组模式(triple pattern)。三元组模式$tp=(s,p,o)$是一个三元组，VAR代表变量集合，则有$s\in VAR\cup R\cup B$，$p\in VAR\cup R$，$o\in VAR\cup R\cup B\cup L$。

通常认为，图G上的聚类分析的目标是将其顶点集合V分割为若干个子集，$V=\{v_1,v_2,\cdots,v_n\}$，$v_i\cap v_j=\emptyset$，$i,j\in\{1,2,\cdots,n\}$。对于RDF数据图G，虽然从拓扑结构上看与一般的图没有本质差别，但对于RDF数据图G所对应的数据集合来说，其最直观的表现是RDF三元组的集合。因此对RDF数据聚类分析结果最直观的呈现方式应该是一系列规模较小的RDF三元组的集合。由于目前存储RDF三元组的主要方法仍然是基于关系数据库的垂直存储方案和三元组表方案[12]，这种聚类结果的形式可更方便地指导大规模RDF数据分布式存储与查询。因此对RDF数据的聚类定义如下。

RDF聚类(RDF Data Clustering)：给定一个RDF数据图G，其上的聚类分析是发现G的子图$G_1,G_2,\cdots,G_m$，$G_i=\{(s,p,o)\mid s,o\in V(G_i),p\in E(G_i)\}$，$V(G_i)\cap V(G_j)\neq\emptyset$，$E(G_i)\cap E(G_j)=\emptyset$。其中$V(G_i)\cap V(G_j)\neq\emptyset$表示同一资源可归属于不同的类簇。

3.2 RDF聚类结果特征分析

上文对RDF聚类结果的定义最大程度上保留了RDF三元组(s,p,o)所包含的原始语义信息，且可以容易地转化为对图节点(即基于实例)的聚类结果。该聚类过程的实现可以借鉴基于边相似性的聚类方法。这类方法的具体思路是：对于RDF图中具有相同顶点k的边，计算两边之间的相似性。边e_{ij}与边e_{jk}之间的相似性定义为$S(e_{ij},e_{jk})=|n_+(i)\cap n_+(j)|/|n_+(i)\cup n_+(j)|$，其中$n_+(i)$表示结点$i$的邻居结点的集合[11]。然后采用单链(single-linkage)层次聚类的方法构建边的树状图。最后依据设定的阈值对树状图进行修剪，得到最终的边聚类结果。

为了验证这种方法作用在RDF数据集上的效果，本文首先利用LUBM测试集生成了一个较小规模的RDF数据集，并采用文献[11]的方法实现对该数据集的聚类。测试集生成的详细过程可参考文献[13]。所生成数据集包含的类、属性及对应的实例信息如表1所列，RDF三元组总计约4500个。执行如上所述算法，最终产生的聚类树状图在叶子节点包含类簇1271个，其中有979个类簇中只包含一个三元组，属于trivial类簇。仔细分析利用上述方法所产生的非trivial的聚类结果，可发现类簇呈现出3种不同的类型。

表1 LUBM测试集数据特性

LUBM类	属性个数	实例个数
FullProfessor	9	10
AssociateProfessor	9	14
Lecturer	8	7
UndergraduateStudent	4	532
GraduateStudent	7	146
ResearchGroup	0	10
Publication	2	460
Course	1	61
GraduateCourse	1	67
TeachingAssistant	1	29
ResearchAssistant	0	39

类型Ⅰ 类簇中的三元组具有相同的主语s：即一个聚类就是描述同一个资源s的节点的集合，这种形式的聚类结果等价于基于实例抽取的聚类结果。设三元组集合$CT1_i$表示满足该类型的某个类簇，则对于$\forall(s_j,p_j,o_j),(s_k,p_k,o_k)\in CT1_i$，有$s_j=s_k$。

在RDF数据图G上，这种聚类表现为G的子图，该子图仅包含资源节点s、以s为起点的对应s各个属性的边，以及表示相应属性值的节点o。

类型II　类簇中的三元组具有相同的属性-对象模式(p,o)：一个聚类中聚集了t个不同的资源，但是这些资源都具有相同的属性p，并取值为同一个值o。设三元组集合$CT2_i$表示满足该类型的某个类簇，则对于$\forall(s_j,p_j,o_j),(s_k,p_k,o_k)\in CT2_i, j\neq k, s_i\in S^t, s_j\in S^t$，有$p_j=p_k, o_j=o_k$。$S^t$表示$t$个不同资源的集合。

在RDF数据图G中，这种聚类也可表现为G的子图，该子图仅包含表示属性取值的对象节点o、以o为终点的表示属性p的边以及相关的t个资源节点。

类型III　类簇中的三元组具有相同的属性-对象模式$\{(p_1,o_1),\cdots,(p_f,o_f)\}$集合：一个聚类中包含了$t$个不同的资源，描述这些资源的(属性-对象)二元组同属于一个有f个不同的(属性-对象)模式的集合。设三元组集合$CT3_i$表示满足该类型的某个类簇，则对于$\forall(s_j,p_j,o_j)\in CT3_i$满足$(p_k,o_k)\in\{(p_1,o_1),\cdots,(p_f,o_f)\}, s_j\in S^t$。$S^t$表示$t$个不同资源的集合。

在RDF数据图G中，这种聚类所对应的子图包含有f个能够表示聚类特征的对象节点、f条表示属性的边以及t个资源节点，其中f条边的起点取自t个资源节点，终点取自f个对象节点。可以看出，该种类型的聚类可理解为t个类型I的聚类集合，或者f个类型II的聚类集合。

显然，通过文献[11]在RDF图G上获得的聚类结果所呈现出的子图结构与RDF三元组模式(s,p,o)有着密切的对应关系。聚类结果同时显示，绝大多数的类簇属于类型I，约占非trivial的类簇总量的87%，每个属于该类型的类簇都是具有相同主语的RDF三元组集合。这个现象也可以直观地理解为一个属于类型I的类簇就是对一个资源所属的各种类型和具有的各种属性进行描述的集合。符合类型II和类型III的类簇占整个聚类结果的比例相对较少，其中属于类型II的类簇约占非trivial的类簇总量的8%左右，类型III类簇所占比例约为5%。其原因在于类型II与类型III的类簇考虑的是资源属性的具体取值，而非属性的值域，无论是数值型属性还是对象型属性，属性值的数量都是非常可观的，因此RDF三元组中对象取相同值的概率是相对较低的。根据以上分析，结合RDF三元组的语义和RDF数据聚类结果的不同类型，提出了一种新的RDF数据聚类方法。

4　基于聚类模式的RDF聚类的实现

4.1　RDF资源特征的重新描述

实现基于模式的RDF数据聚类，首先需要建立RDF数据集的"属性-值"模型。一个RDF数据集中的资源可以通过一个特征集合rfs(RDF features set)表示，rfs中的每一个特征都关联着一个取值集合。rfs可定义如下：

$rfs_k=\{f_1,\cdots,f_n\}$

对于$\forall f_i\in rfs_k$，同时定义其取值集合$Vf_i=\{vf_1,\cdots,vf_k\}$。

例如，资源1和资源2分别具有属性profession、type、degree、study field，每个资源在一个属性上可能有多个取值，如表2所列。

表2　资源1和资源2的属性及其取值

属性	资源1	资源2
profession	entrepreneur	—
profession	scientist	—
profession	engineer	engineer
profession	Programmer	—
profession	Computer_scientist	Computer_scientist
profession	businessman	—
profession	Inventor	—
type	Computer_designer	academic
degree	Bachelor_of_science	Master_of_science
degree	—	doctorate
study_field	SE	SE
study_field	—	DBMS

对于资源1，其rfs形式的描述如下：

$rfs_1=\{profession,degree,type,study\_field\}$

$Vf_1^1=\{entrepreneur, scientist, engineer, programmer, compuer\ scientist, businessman, inventor\}$

$Vf_1^2=\{bachelor\ of\ science\}$

$Vf_1^3=\{computer\ designer\}$

$Vf_1^4=\{SE\}$

对于资源2，其rfs形式的描述如下：

$rfs_2=\{profession,degree,type,study\_field\}$

$Vf_2^1=\{engineer,computer\ scientist\}$

$Vf_2^2=\{master\ of\ science, doctorate\}$

$Vf_2^3=\{academic\}$

$Vf_2^4=\{SE,DBMS\}$

在此形式上，可定义资源属性及其取值的相似性。属性值的相似性sim_v可直观地定义为：

$$sim\_v(vf_1,vf_2)=\begin{cases}1, & \text{if } vf_1=vf_2\\ 0, & \text{否则}\end{cases}$$

在属性值相似性计算基础上，属性间的相似性计算可定义为：

$$sim\_f(f_1,f_2)=\frac{\sum_{s=1}^{k}\max(sim\_v(v_s,v_t))\ \forall t\in[1,h]}{k}$$

其中，$v_s\in Vf_1, v_t\in Vf_2, k=|Vf_1|, h=|Vf_2|$。

值得注意的是，sim_f的计算是非对称的，即当$k\neq h$时，$sim\_f(f_1,f_2)\neq sim\_f(f_2,f_1)$。在本文研究中为了消除这种非对称性对属性相似性计算的影响，将同时计算$sim_f(f_1,f_2)$和$sim_f(f_2,f_1)$，只要两者其中之一满足所设定的相似性阈值即可对属性间的相似性做出判断。

给定一个rfs的集合R、特征f_i和该特征的一个取值$vf_s\in Vf_i$，本文同时定义值vf_s的价值系数$sig(vf_s)$来反映该属性值在整个数据集中的重要性。$sig(vf_s)=\frac{T(vf_s)}{T(f_i)}$，其中$T(vf_s)$为集合$R$中属性$f_i$取值为$vf_s$的次数，$T(f_i)$为属性$f_i$在$R$中出现的总次数。$sig(vf_s)$从出现频率的角度反映了属性值的重要程度。

4.2　聚类模式的生成及聚类的实现

基于4.1节的RDF资源特征描述，根据RDF数据聚类结果所呈现出的类型特征，本文提出了一种新的RDF聚类策略：首先，对于数据集中的RDF资源，建立其rfs资源描述方式；然后根据3.2节的聚类类型，分别找出满足类型I、类型II、类型III的RDF三元组遵循的聚类模式集合CMI、CMII和CMIII；最后根据聚类模式从RDF数据集中分别找到满足

相应模式的 RDF 三元组集合。

4.2.1 RDF 数据集上聚类模式的生成

可以看出，从资源的 rfs 描述定义聚类模式是该聚类方法的核心。对于满足类型 I 的聚类结果，只要 RDF 资源是同名的就属于同一个聚类，不涉及属性和属性值的问题，因此通过遍历整个三元组集合就可直接获得。对于满足类型 II 的聚类结果，需要定义属性模式 (p,o)，根据资源的 rfs 描述，n 个属性、每个属性取值有 k 种情况的 RDF 资源理论上可以生成 nk 个候选 (p,o) 模式，但这些模式并非全部有意义，可根据属性取值的价值系数 sig_v 设定使得 (p,o) 有意义的阈值，最终得到满足阈值的 (p,o) 模式集合 CMII。对于满足类型 III 的聚类结果，需要定义模式集合 $\{(p_1,o_1),\cdots,(p_f,o_f)\}$，其生成过程可以在 CMII 的基础上进行。要使模式 $\{(p_1,o_1),\cdots,(p_f,o_f)\}$ 有意义，则其中的任何一个二元组 (p_i,o_i) 必须是有意义的，因此可以利用 CMII 的结果生成 CMIII，具体过程如算法 1 所述。

算法 1 RDF 聚类模式生成算法

输入：RDF 资源集合 R 的 rfs 描述集合，属性值价值系数阈值 s，属性相似性阈值 sf

输出：聚类模式 CMIII

1. CMII=∅;
2. CMIII=∅;

//对于数据集 R 中的每一个属性 f_i，计算其所有可能的取值 vf_i，如果取值满足一定的价值系数，则将模式 (f_i,vf_i) 放入聚类模式集合 CMII 中。

3. for each $f_i \in R$
4. for each $vf_i \in Vf_i$
5. if $sig(vf_i) \geqslant s$
   CMII=CMII∪{(f_i,vf_i)};
6. else if $\exists f_j \in$ CMII 且满足 $sim\_f(f_i,f_j) \geqslant sf$ 或 $sim\_f(f_j,f_i) \geqslant sf$
7. CMII=CMII∪{(f_i,vf_i)};

//找到聚类模式集合 CMII 的所有子集，如果子集中的各个元素之间不存在语义上的矛盾，则将该子集放入模式集合 CMIII 中。

8. Sub=SubSetof(CMII);
9. for each se in Sub
10. if compatible(element(se))
11. CMIII=CMIII∪se;

可以看出，sim_v、sim_f 的定义方式直接影响着聚类模式的最终结果。rfs、CMII 和 CMIII 的计算可看作是对 RDF 数据集的预处理，作为 RDF 数据聚类的准备工作，这个过程可在聚类算法执行之前完成，不会影响聚类过程的效率。

4.2.2 基于聚类模式的聚类实现

在获取了不同类型的聚类模式之后，聚类的实现可以有两种方式：第一种方式适用于使用经典的关系数据库系统存储 RDF 三元组的情况，遍历数据库中 RDF 三元组表，将三元组与聚类模式进行匹配，如果符合模式的要求则将该三元组分配到以该模式为解释的类簇中。第二种方式针对提供了 SPARQL 查询端口的 RDF 数据集，可以利用 SPARQL 查询实现不同类型的聚类。例如，对于 CMII 中的模式 (p_i,o_i)，可建立查询：SELECT ?s FROM http://example.org/dataset1 WHERE {?s p_i o_i}，该查询结果即为满足模式 (p_i,o_i) 的资源集合。本文将在实验部分分别对两种方式的效果进行测试。

5 实验

本文分别使用 LUBM 和 SP2Bench 两个测试集对所提出的方法进行验证，采用 Sesame 作为 RDF 数据存储系统。Sesame 以三元组表方式存储 RDF 数据，同时也提供了 SPARQL 查询接口，因此便于检测不同的 RDF 数据聚类实现策略的效果。实验中算法的实现基于 Java 1.6，程序运行在 Windows 7 环境下，硬件平台为 Intel Xeon 处理器，4GB RAM。

5.1 传统关系系统上聚类的实现

对于 LUBM 测试集，首先使用经典聚类方法作用在表 1 的数据集上，以发现 RDF 聚类结果在结构上的特征，从而定义聚类类型。其次将算法 1 作用在同一数据集上，采用第一种聚类实现方式，并将聚类结果与使用文献[11]的方法获取的结果进行比较，由于文献[11]的研究成果在社区发现等与聚类任务高度相关的应用中已得到广泛应用，因此以该方法为基准验证所提出算法对 RDF 数据进行聚类的效果。表 3 是在该数据集上使用算法 1 生成的聚类模式的结果，表 4 为与使用文献[11]的方法所产生的聚类结果的比较。

表 3 LUBM 测试集上聚类模式生成的结果

参数	生成数量		消耗时间 (s)
	CMII	CMIII	
s=0.1,sf=0.8	12	3	7.7
s=0.1,sf=0.6	12	5	8.1
s=0.05,sf=0.8	14	3	7.5

表 4 LUBM 测试集上的聚类结果特征比较

	基于边相似性	基于聚类模式
聚类依据	边相似性	聚类类型的模式匹配
类簇数量	1271	1123
支持多类簇	支持	支持
数据的存储方式	图	三元组表
聚类结果的解释	非直观	可从模式获得
时间复杂度	$O(n^2)$ (n 表示图中节点数据)	$O(n^2)$ (n 表示三元组表的行数)

表 3 展示了在属性值价值系数阈值 s、属性相似性阈值 sf 分别取不同值的情况下，所产生第二种和第三种类型的聚类模式数量和所耗费的时间。结果同时反映了符合类型 II 和类型 III 的聚类结果数量相对较少的现象。由于 CMII 和 CMIII 的数量较少，阈值 s、sf 对其产生的影响并不显著。从生成聚类模式所耗费的时间上看，这一过程在实际应用中是可接受的。

从表 4 可以看到，本文方法的聚类结果与使用文献[11]得到的结果基本一致，并且都支持将同一资源归属于多个类簇。从数量上看，基于聚类模式的类簇数量较少，其主要原因在于聚类模式可能会遗漏少量的类簇特征，这说明聚类模式的准确性需要进一步提高。这两种方式将个体资源聚集在一起的依据大相径庭，文献[11]是基于图结构，通过边的相似性计算实现节点的聚类：相似性较高的边，其端点也应该具有较高相似性，可以归属为同一类；本文方法是基于聚类类型的模式匹配，不需要直接计算资源或者资源间链接的相似性，同时充分利用了 RDF 三元组结构的信息。三元组表存储上的各种优化技术将有助于提高聚类的效率。从对聚类结果的解释上看，本文所提出的聚类类型本质上就是对聚类特征的一种描述，因此可以较为容易地描述聚类结果并对其进行解释。从算法执行的时间复杂度上看，对于文献[11]，最坏情况下处理的对象是对一个完全有向图，即需要计算 $n(n-1)$ 条边之间的相似性。由于 RDF 图的稀疏性，这种极端情况几乎不会

出现。对于本文方法而言，由于算法是直接基于三元组的匹配，对于类型 I 和类型 II 的聚类，只需要对三元组表进行一次遍历；对于类型 III，最坏情况下对三元组表进行两次遍历即可完成。因此三元组上的存储优化对提高算法效率是十分有意义的。

5.2 基于 SPARQL 的聚类方法

使用 SP2Bench 数据集的主要目的是验证使用 SPARQL 查询完成聚类的效率，并将其与使用基本的遍历三元组表方式获得的聚类结果进行对比。SP2Bench 生成的详细方法见文献[14]。表 5 是文中使用的 SP2Bench 数据集的主要特征。

表 5 使用 SP2Bench 生成的三个数据集特征

三元组个数		10k	50k	100k
文件大小		1.0MB	5.2MB	11.2MB
实例数量	# Authors	1.5k	6.9k	14.7k
	# Journals	25	104	236
	# Articles	920	3.9k	8.1k
	# Proc.	6	35	74
	# InProc.	175	1.3k	2.2k
	# InColl	20	60	92
	# Books	0	0	30

针对该数据集，分别采用实现聚类的两种途径发现三种不同类型的聚类，表 6 是当 $s=0.1, sf=0.8$ 时，在不同规模的数据集上所生成的 CMII 和 CMIII 的数量以及消耗的时间，该结果与表 3 描述的结果一致。表 7 是聚类过程的执行时间，从中可以看出，两种聚类实现方法都可以在可接受的时间范围内完成聚类操作。基于 SPARQL 查询的实现过程所花费的时间明显多于基于关系表的时间，随着数据集规模的增大，两者时间上的差别越发显著。其主要原因在于作用在经典的关系数据库系统上的数据访问操作更容易实现性能上的优化。当然，也可以通过引入一些 SPARQL 查询优化策略来提高方法 2 的效率。方法 2 的最大优点在于其描述聚类过程的简洁性，可避免直接处理大量的三元组。

表 6 SP2Bench 测试集上聚类模式生成的结果($s=0.1, sf=0.8$)

三元组个数	生成数量		消耗时间
	CMII	CMIII	(s)
10k	17	9	12.4
50k	24	11	27.7
100k	36	17	61.8

表 7 不同聚类实现方式所耗费的时间

三元组个数		10k	50k	100k
CMI	方法 1	0.8s	1.9s	5.2s
	方法 2	1.2s	3.2s	6.6s
CMII	方法 1	1.1s	3.7s	7.4s
	方法 2	1.6s	4.6s	9.3s
CMIII	方法 1	4.9s	6.2s	10.4s
	方法 2	6.3s	7.1s	13.7s

结束语 对大规模 RDF 数据集进行聚类是有效利用 RDF 数据资源的一种重要方式。本文针对 RDF 数据自身结构的特征以及聚类结果中 RDF 聚类类簇所呈现出的聚类模式，提出了基于模式的 RDF 数据聚类方法。该方法的主要优势在于可以充分利用 RDF 数据集和 RDF Schema 所提供的信息，使得聚类过程更适合于 RDF 数据，并且聚类的结果更容易理解。聚类模式的生成是本文方法的核心，为了构造聚类模式，定义了 RDF 资源的 rfs(RDF features set)描述。生成聚类模式的过程可看作对 RDF 数据进行预处理的过程。

进一步的研究工作主要集中在如何提高聚类模式的质量上，具体包括如何提高聚类模式的准确性，减少被遗漏的类簇的数量，以及如何减少模式生成的时间。与此同时，如何利用 RDF Schema 的语义信息对聚类结果进行更有效的组织以提高结果的可理解性和可用性也是需要关注的问题。

参 考 文 献

[1] Bizer C, Heath T, Berners-Lee T, et al. Linked data on the Web [C]// Proceedings of the 17th International Conference on World Wide Web. 2008: 1265-1266

[2] Tran T, Wang H, Haase P. Hermes: Data web search on a pay-as-you-go integration infrastructure [J]. Web Semantics: Science, Services and Agents on the World Wide Web, 2009, 7(3): 189-203

[3] Zeng K, Yang J, et al, A distributed graph engine for web scale rdf data[C]// Proceedings of the 39th International Conference on Very Large Data Bases. 2013: 265-276

[4] Wu A Y, Garland M, Han J. Mining scale-free networks using geodesic clustering[C]// Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. 2004: 719-724

[5] Kaushik R, Shenoy P, Bohannon P, et al. Exploiting local similarity for indexing paths in graph-structured data[C]// Proceedings of the 18th International Conference on Data Engineering. 2002: 129-140

[6] Konrath M, Gottron T, Staab S, et al. Schemex efficient construction of a data catalogue by stream-based indexing of linked data[J]. Web Semantics: Science, Services and Agents on the World Wide Web, 2012, 16: 52-58

[7] Böohm C, Lorey J, Naumann F. Creating void descriptions for Web-scale data[J]. Web Semantics: Science, Services and Agents on the World Wide Web, 2011, 9(3): 339-345

[8] Fanizzi N, d ' Amato C. A hierarchical clustering method for semantic knowledge bases[C]// Proceedings of KES 2007. 2007: 653-660

[9] Grimnes G A, Edwards P, Preece A D. Instance based clustering of semantic web resources[C]// Proceedings of ESWC 2008. 2008: 303-317

[10] Alzogbi A, Lausen G. Similar structures inside rdf-graphs[C]// Proceedings of Proceedings of the WWW 2013 Workshop on Linked Data on the Web. 2013

[11] Ahn Y Y, Bagrow J P, Lehmann S. Link communities reveal multiscale complexity in networks [J]. Nature, 2010, 466 (7307): 761-764

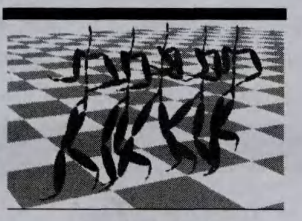
[12] 杜小勇，王琰，吕彬. 语义 Web 数据管理研究进展[J]. 软件学报，2009，20(11)：2050-2964
Du Xiao-yong, Wang Yan, Lv Bin. Research and Develompment on Semantic Web Data Management[J]. Journal of Software, 2009, 20(11): 2050-2964

[13] Guo Yuan-bo, Pan Zheng-xiang, Jeff H. LUBM: A Benchmark for OWL Knowledge Base Systems[J]. Web Semantics, 2005, 3 (2): 158-182

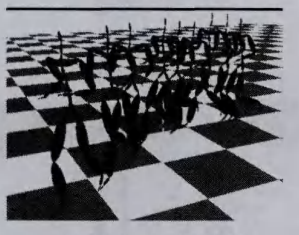
[14] Schmidt M, Hornung T, Lausen G, et al. SP²Bench: a SPARQL performance benchmark[M]. Semantic Web Information Management. 2010: 371-393

(下转第 296 页)

到达点 A"、"在 B 点捡起物体"这样高层次的控制命令，来准确地、交互式地控制角色。语义层次的角色动画控制更多关注高层次的任务行为，而不是低层次的运动细节。图 7(a)展示了一个随机生成的跑步运动序列，图 7(b)展示了一个向左偏转的跑步运动序列，通过调节角色的运动轨迹和朝向的低维语义参数来实现。



(a)随机跑步序列



(b)改变轨迹的跑步序列

图 7　随机运动合成

结束语　本文主要介绍了多角色可变形运动模型的概念和构建方法，实现了多角色相似运动的分解与合成。通过对运动模型时空关系的处理，采用高效的降维方法，经过统计分析提取出运动数据的内在特征，构建低维的语义空间，最终实现通过调节参数来灵活地控制虚拟人群运动。实验结果表明，可变形运动模型能够合成丰富、逼真的多角色运动环境。

参 考 文 献

[1] Reynolds C W. Flocks, herds and schools: A distributed behavioral model[J]. ACM SIGGRAPH Computer Graphics, 1987, 21(4): 25-34

[2] Reynolds C W. Steering behaviors for autonomous characters [C]// Game Developers Conference. 1999: 763-782

[3] Duives D C, Daamen W, Hoogendoorn S P. State-of-the-art crowd motion simulation models[J]. Transportation Research Part C: Emerging Technologies, 2013, 37: 193-209

[4] Brand M, Hertzmann A. Style machines[M]. New York: ACM Press, 2000: 183-192

[5] Urtasun R, Glardon P, Boulic R, et al. Style-Based Motion Synthesis[J]. Computer Graphics Forum, 2004, 23(4): 799-812

[6] Li Y, Wang TS, Shum HY. Motion texture: A two-level statistical model for character motion synthesis[J]. ACM Trans. on Graphics, 2002, 21(3): 465-472

[7] Glardon P, Boulic R, Thalmann D. PCA-based walking engine using motion capture data [C] // Proceedings of Computer Graphics International, 2004. IEEE, 2004: 292-298

[8] Glardon P, Boulic R, Thalmann D. A coherent locomotion engine extrapolating beyond experimental data [C] // Proceedings of CASA, 2004: 73-84

[9] Hsu E, Pulli K, Popović J. Style translation for human motion [J]. ACM Transactions on Graphics, 2005, 24(3): 1082-1089

[10] Niwase N, Yamagishi J, Kobayashi T. Human walking motion synthesis with desired pace and stride length based on HSMM [J]. IEICE Transactions on Information and Systems, 2005, 88 (11): 2492-2499

[11] Tanco L M, Hilton A. Realistic synthesis of novel human movements from a database of motion capture examples[C] // Proceedings of Workshop on Human Motion, 2000. IEEE, 2000: 137-142

[12] Chai J, Hodgins J K. Constraint-based motion optimization using a statistical dynamic model[J]. ACM Transactions on Graphics, 2007, 26(3): 8

[13] Lau M, Bar-Joseph Z, Kuffner J. Modeling spatial and temporal variation in motion data[J]. ACM Transactions on Graphics, 2009, 28(5): 171

[14] Wei X, Min J, Chai J. Physically valid statistical models for human motion generation[J]. ACM Transactions on Graphics, 2011, 30(3): 19

[15] Lerner A, Chrysanthou Y, Lischinski D. Crowds by example[J]. Computer Graphics Forum, 2007, 26(3): 655-664

[16] Courty N, Corpetti T. Crowd motion capture[J]. Computer Animation and Virtual Worlds, 2007, 18(4/5): 361-370

[17] Lee K H, Choi M G, Hong Q, et al. Group behavior from video: a data-driven approach to crowd simulation[C] // Proceedings of the 2007 ACM SIGGRAPH/Eurographics Symposium on Computer animation. Eurographics Association, 2007: 109-118

[18] Ju E, Choi M G, Park M, et al. Morphable crowds[J]. ACM Transactions on Graphics, 2010, 29(6): 140

[19] Blanz V, Vetter T. A morphable model for the synthesis of 3D faces[C] // Proceedings of the 26th annual conference on Computer graphics and interactive techniques. ACM Press/Addison-Wesley Publishing Co., 1999: 187-194

[20] Min J, Chen Y L, Chai J. Interactive generation of human animation with deformable motion models[J]. ACM Transactions on Graphics, 2009, 29(1): 9

[21] Teknomo K, Takeyama Y, Inamura H. Tracking algorithm for microscopic flow data collection [C] // Proceedings of JSCE Student Conference. Sendai, Japan, 2000

[22] 王鑫，孙守迁，邵明. 运动路径驱动的角色动画合成方法[J]. 计算机辅助设计与图形学学报，2009，21(3)：319-324
Wang Xin, Sun Shou-qian, Shao Ming. A Path-Driven Character Animation Synthesis Mechod[J]. Journal of Computer-Aided Design & Computer Graphics, 2009, 21(3): 319-324

[23] Lee J, Chai J, Reitsma P S A, et al. Interactive control of avatars animated with human motion data[J]. ACM Transactions on Graphics, 2002, 21(3): 491-500

(上接第 270 页)

[15] 杜芳，陈跃国，杜小勇. RDF 数据查询处理技术综述[J]. 软件学报，2013，24(6)：1222-1242
Du Fang, Chen Yue-guo, Du Xiao-yong. Survey of RDF Query Processing Techniques[J]. Journal of Software, 2013, 24(6): 1222-1242

[16] 李慧颖，瞿裕忠. KREAG：基于实体三元组关联图的 RDF 数据关键词查询方法[J]. 计算机学报，2011，34(5)：825-836
Li Hui-ying, Qu Yu-zhong. KREAG: Keyword Query Approach over RDF Data based on Entity-Triple Association Graph[J]. Chinese Journal of Computers, 2011, 34(5): 825-836