(19)中华人民共和国国家知识产权局

(12)发明专利申请

(10)申请公布号 CN 106709006 A
(43)申请公布日 2017.05.24

(21)申请号 201611209081.1

(22)申请日 2016.12.23

(71)申请人 武汉科技大学
地址 430081 湖北省武汉市青山区和平大道947号
申请人 武汉楚天云科技股份有限公司

(72)发明人 顾进广 彭燊 黄智生 符海东 梅琨

(74)专利代理机构 武汉科皓知识产权代理事务所(特殊普通合伙) 42222
代理人 鲁力

(51)Int.Cl.
*G06F 17/30*(2006.01)

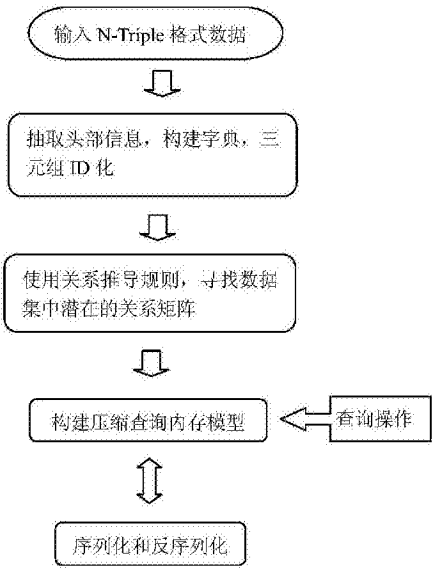
权利要求书2页 说明书4页 附图3页

(54)发明名称

一种对查询友好的关联数据压缩方法

(57)摘要

本发明涉及一种对查询友好的关联数据压缩方法，该方法包括：定义关系挖掘规则，挖掘三元组中潜在的关联关系；定义压缩查询内存模型，由主语向量、谓语向量和宾语矩阵组成；定义压缩查询内存模型的序列化方式，使用三个辅助符号实现序列化和反序列化；定义在压缩查询内存模型上执行SPARQL的查询方式，主语和谓语查询使用二分查找方法，宾语查询使用线性遍历方法；定义当宾语矩阵过大导致查询缓慢的解决方案，将大的数据块拆分为多个小的数据块。使用本发明方法处理的关联数据集，相对于大部分现有的压缩方案，提高了压缩率，并且在压缩状态下，可以直接进行SPARQL查询操作。




CN 106709006 A

1.一种对查询友好的关联数据压缩方法,其特征在于,

一个构建结构模型的步骤,具体包括:

步骤1,将三元组内存模型基于N-Triple格式关联数据并解析,得到三元组集合,然后构建字典,并将三元组ID化,其中,解析的过程包括:

步骤1.1,过滤掉以#开始的行或者空行;

步骤1.2,读取每一行数据按空格切分字符串;

步骤1.3,将切分后数据映射到三元组的主语、谓语和宾语,构建成一个三元组;

步骤2,基于关系挖掘约束,挖掘三元组中潜在的关联;

步骤3,定义压缩查询内存模型,由头部信息、字典和数据块集合组成,每个数据块由主语向量、谓语向量和宾语矩阵组成;所述压缩查询内存模型使用主语向量、谓语向量和宾语矩阵的方式表示三元组关系:定义主语向量为一个长度为m的列向量,谓语向量为一个长度为n的行向量,宾语矩阵为一个m*n的矩阵,主语向量和谓语向量做向量乘法,得到一个和宾语矩阵大小相同的由主谓语组成的矩阵,再与宾语矩阵的数据项一一映射,映射后的每一项为一个三元组关系;

一个基于结构模型进行高压缩率数据存储的步骤,具体包括:

步骤4,定义每个ID的存储长度相同,基于压缩查询内存模型的序列化方式:使用辅助标示进行序列化和反序列化操作;

步骤4.1,序列化:对于每个数据块,将宾语矩阵扁平化,用介词辅助标识将主语向量、谓语向量和扁平化的宾语矩阵数据连接成线性数据结构,再将处理后的线性数据结构用数据块辅助标识连接在一起;

步骤4.2,反序列化:根据数据块辅助标识符将序列化的数据划分成每个一个个数据块,对于每个数据块,根据介词辅助标识符划分为主语向量、谓语向量和扁平化的宾语矩阵,再根据主语向量的长度将扁平化的宾语矩阵还原成真实矩阵;

一个基于结构模型进行数据查询的步骤,具体包括:

步骤5,基于压缩查询内存模型的SPARQL的查询:对主语和谓语的查询使用二分查找,对于宾语的查询使用线性遍历查找,具体包括:

步骤5.1,主语查询方法具体包括:遍历所有数据块的主语向量,因为主语向量内部已排序,使用二分查找方法,因此主语查询时间复杂度为O(log2n);

步骤5.2,谓语查询方法具体包括:遍历所有数据块的谓语向量,因为谓语向量内部已排序,使用二分查找方法,因此谓语查询时间复杂度为O(log2n);

步骤5.3,宾语查询方法具体包括:遍历所有数据块的宾语矩阵,因为宾语矩阵内部未排序,只能顺序查找,时间复杂度为O(n)。

2.根据权利要求1所述的一种对查询友好的关联数据压缩方法,其特征在于,基于N-Triple格式关联数据并解析,得到三元组集合具体包括:

步骤2.1,过滤掉以#开始的行或者空行;

步骤2.2,读取每一行数据按空格切分字符串;

步骤2.3,将切分后数据映射到三元组的主语、谓语和宾语,构建成一个三元组。

3.根据权利要求1所述的一种对查询友好的关联数据压缩方法,其特征在于,构建字典,并将三元组ID化具体包括:

步骤3.1,将上一步得到的三元组进行扁平化操作,去掉重复数据项;

步骤3.2,为每一项数据分配一个唯一ID,得到Dictionary;

步骤3.3,抽取Dictionary中每项数据相同的头部信息,得到Header;将原始的三元组数据用ID替换,得到ID化的三元组集合。

4.根据权利要求1所述的一种对查询友好的关联数据压缩方法,其特征在于,关系挖掘约束包括:

约束条件一:合并具有相同主语和谓语的三元组;

约束条件二:将所有三元组按照主语进行分类,合并相同主语的所有谓语和宾语,形成谓语向量和宾语向量,提取每个主语的谓语向量;

约束条件三:合并具有相同谓语向量和宾语的三元组;

约束条件四:将所有三元组按照谓语向量分类,合并相同谓语向量的主语和宾语,形成主语向量和宾语矩阵。

5.根据权利要求1所述的一种对查询友好的关联数据压缩方法,其特征在于,压缩查询内存模型,使用主语向量、谓语向量和宾语矩阵的方式表示三元组关系:假定主语向量为一个长度为m的列向量,谓语向量为一个长度为n的行向量,宾语矩阵为一个m*n的矩阵,主语向量和谓语向量做向量乘法,得到一个和宾语矩阵大小相同的由主谓语组成的矩阵,再与宾语矩阵的数据项一一映射,映射后的每一项为一个三元组关系。

6.根据权利要求1所述的一种对查询友好的关联数据压缩方法,其特征在于,SPARQL的查询还包括:

步骤5.4,复杂查询,所有的复杂查询都可以分解为成前面三种简单查询,再合并简单查询的结果;所述步骤5.1至步骤5.4能够并发执行。

7.根据权利要求1所述的一种对查询友好的关联数据压缩方法,其特征在于,还包括一个宾语矩阵拆分的步骤,对于宾语矩阵过大的数据块,拆分为多个数据块,具体是当宾语矩阵过大导致宾语查询缓慢的解决方案,保持谓语向量不变,将主语向量和宾语矩阵对应拆分,得到多个小的数据块,此拆分方法能维护压缩查询内存模型结构,保证拆分后的压缩查询内存模型仍然能进行并发查询操作。

# 一种对查询友好的关联数据压缩方法

## 技术领域

[0001] 本发明涉及大数据领域，用于海量RDF、LOD及知识图谱相关数据的存储、传输和查询。尤其涉及一种对查询友好的关联数据压缩方法

## 背景技术

[0002] 现有的关联数据压缩方案有很多种，但大部分对于查询并不友好。普遍认同的压缩方案有HDT，这种压缩方案压缩率较高，但是查询时需要先解压缩，对查询并不友好。受HDT方案的启发，很多基于HDT方案的压缩技术也被提出，如HDT FoQ、WaterFowl、HDT++，这些压缩技术都有一个共同的特点：高压缩率，但对查询并不友好。

[0003] 也有一些对查询友好的方案，譬如BitMat方法，这种压缩方案采用三维矩阵的方式表述三元组关系，为很多不存在的三元组关系也预留存储空间。当关联数据集大到一定规模时，这个三维矩阵就变成了一个超级稀疏矩阵，由于存储了很多冗余信息，压缩率并不理想。为了减少存储的冗余信息，$K^2$-triple方案被提出来，它按照谓语将三维矩阵划分为多个二维矩阵，用K2树的结构存储二维矩阵。这种方法在一定程度上提高了压缩率，但也破坏了原有直观的矩阵结构，所以在进行查询的时候需要先还原矩阵，而这个操作会降低RDF的查询效率。

[0004] 越来越多的关联数据充斥着整个数据网络，当需要管理和查询这些数据时，查询性能和数据可扩展便成了焦点问题。虽然可以使用足够多的存储介质来存储越来越庞大的关联数据集，但是庞大的数据集不仅会导致查询效率降低，还会加剧其它常见的过程(如RDF发布和交换)的性能问题。随着通过网络传输执行结果的远程SPARQL端点查询方式越来越受欢迎，RDF的发布和交换在关联数据的查询中使用得越来越频繁。因此寻找一种查询友好的关联数据压缩方法具有重大意义。

## 发明内容

[0005] 针对上述问题，本发明的目的是找到一种对查询友好的压缩方案。在没有对关联数据压缩数据解压缩的情况下，能直接进行SPARQL查询，同时尽可能提高压缩率。

[0006] 本发明目标由挖掘关联数据集中潜在的关系矩阵实现。该方法包括：

[0007] 一种对查询友好的关联数据压缩方法，其特征在于，

[0008] 一个构建结构模型的步骤，具体包括：

[0009] 步骤1，将三元组内存模型基于N-Triple格式关联数据并解析，得到三元组集合，然后构建字典，并将三元组ID化，其中，解析的过程包括：

[0010] 步骤1.1，过滤掉以“#”开始的行或者空行；

[0011] 步骤1.2，读取每一行数据按空格切分字符串；

[0012] 步骤1.3，将切分后数据映射到三元组的主语、谓语和宾语，构建成一个三元组；

[0013] 步骤2，基于关系挖掘约束，挖掘三元组中潜在的关联；

[0014] 步骤3，定义压缩查询内存模型，由头部信息、字典和数据块集合组成，每个数据块

由主语向量、谓语向量和宾语矩阵组成;所述压缩查询内存模型使用主语向量、谓语向量和宾语矩阵的方式表示三元组关系:定义主语向量为一个长度为m的列向量,谓语向量为一个长度为n的行向量,宾语矩阵为一个m*n的矩阵,主语向量和谓语向量做向量乘法,得到一个和宾语矩阵大小相同的由主谓语组成的矩阵,再与宾语矩阵的数据项一一映射,映射后的每一项为一个三元组关系;

[0015] 一个基于结构模型进行高压缩率数据存储的步骤,具体包括:

[0016] 步骤4,定义每个ID的存储长度相同,基于压缩查询内存模型的序列化方式:使用辅助标示进行序列化和反序列化操作;

[0017] 步骤4.1,序列化:对于每个数据块,将宾语矩阵扁平化,用介词辅助标识将主语向量、谓语向量和扁平化的宾语矩阵数据连接成线性数据结构,再将处理后的线性数据结构用数据块辅助标识连接在一起;

[0018] 步骤4.2,反序列化:根据数据块辅助标识符将序列化的数据划分成每个一个个数据块,对于每个数据块,根据介词辅助标识符划分为主语向量、谓语向量和扁平化的宾语矩阵,再根据主语向量的长度将扁平化的宾语矩阵还原成真实矩阵;

[0019] 一个基于结构模型进行数据查询的步骤,具体包括:

[0020] 步骤5,基于压缩查询内存模型的SPARQL的查询:对主语和谓语的查询使用二分查找,对于宾语的查询使用线性遍历查找,具体包括:

[0021] 步骤5.1,主语查询方法具体包括:遍历所有数据块的主语向量,因为主语向量内部已排序,使用二分查找方法,因此主语查询时间复杂度为O(log2n);

[0022] 步骤5.2,谓语查询方法具体包括:遍历所有数据块的谓语向量,因为谓语向量内部已排序,使用二分查找方法,因此谓语查询时间复杂度为O(log2n);

[0023] 步骤5.3,宾语查询方法具体包括:遍历所有数据块的宾语矩阵,因为宾语矩阵内部未排序,只能顺序查找,时间复杂度为O(n)。

[0024] 在上述的一种对查询友好的关联数据压缩方法,基于N-Triple格式关联数据并解析,得到三元组集合具体包括:

[0025] 步骤2.1,过滤掉以#开始的行或者空行;

[0026] 步骤2.2,读取每一行数据按空格切分字符串;

[0027] 步骤2.3,将切分后数据映射到三元组的主语、谓语和宾语,构建成一个三元组。

[0028] 在上述的一种对查询友好的关联数据压缩方法,构建字典,并将三元组ID化具体包括:

[0029] 步骤3.1,将上一步得到的三元组进行扁平化操作,去掉重复数据项;

[0030] 步骤3.2,为每一项数据分配一个唯一ID,得到Dictionary;

[0031] 步骤3.3,抽取Dictionary中每项数据相同的头部信息,得到Header;将原始的三元组数据用ID替换,得到ID化的三元组集合。

[0032] 在上述的一种对查询友好的关联数据压缩方法,关系挖掘约束包括:

[0033] 约束条件一:合并具有相同主语和谓语的三元组;

[0034] 约束条件二:将所有三元组按照主语进行分类,合并相同主语的所有谓语和宾语,形成谓语向量和宾语向量,提取每个主语的谓语向量;

[0035] 约束条件三:合并具有相同谓语(谓语向量)和宾语的三元组;

[0036] 约束条件四:将所有三元组按照谓语向量分类,合并相同谓语向量的主语和宾语,形成主语向量和宾语矩阵。

[0037] 在上述的一种对查询友好的关联数据压缩方法,压缩查询内存模型,使用主语向量、谓语向量和宾语矩阵的方式表示三元组关系:假定主语向量为一个长度为m的列向量,谓语向量为一个长度为n的行向量,宾语矩阵为一个m*n的矩阵,主语向量和谓语向量做向量乘法,得到一个和宾语矩阵大小相同的由主谓语组成的矩阵,再与宾语矩阵的数据项一一映射,映射后的每一项为一个三元组关系。

[0038] 在上述的一种对查询友好的关联数据压缩方法,SPARQL的查询还包括:

[0039] 步骤5.4,复杂查询,所有的复杂查询都可以分解为成前面三种简单查询,再合并简单查询的结果;所述步骤5.1至步骤5.4能够并发执行。

[0040] 在上述的一种对查询友好的关联数据压缩方法,还包括一个宾语矩阵拆分的步骤,对于宾语矩阵过大的数据块,拆分为多个数据块,具体是当宾语矩阵过大导致宾语查询缓慢的解决方案,保持谓语向量不变,将主语向量和宾语矩阵对应拆分,得到多个小的数据块,此拆分方法能维护压缩查询内存模型结构,保证拆分后的压缩查询内存模型仍然能进行并发查询操作。

[0041] 因此,本发明具有如下优点:使用本发明处理的关联数据集,相对于大部分现有压缩方案,提高了压缩率,并且在压缩状态下,可以直接进行SPARQL查询操作。

## 附图说明

[0042] 图1为本发明实施例的压缩原理图。

[0043] 图2为本发明实施例的压缩查询内存模型图。

[0044] 图3为本发明实施例的四条关系挖掘规则描述图。

[0045] 图4为本发明实施例的大数据块拆分规则描述图。

[0046] 图5是本发明的方法流程示意图。

## 具体实施方式

[0047] 以下结合附图和实施例详细说明本发明技术方案。

[0048] 本发明提供的技术方案是基于关系矩阵的关联数据集压缩算法,具体包括以下步骤:

[0049] 1.定义三元组内存模型,包含主语S、谓语P和宾语O三个数据段;

[0050] 2.输入N-Triple格式关联数据并解析,得到三元组集合;

[0051] 详细过程如下:

[0052] 2.1.过滤掉以“#”开始的行或者空行;

[0053] 2.2.读取每一行数据按空格切分字符串;

[0054] 2.3.将切分后数据映射到三元组的主语、谓语和宾语,构建成一个三元组;

[0055] 3.构建字典,并把三元组ID化;

[0056] 详细过程如下:

[0057] 3.1.将上一步得到的三元组进行扁平化操作,去掉重复数据项;

[0058] 3.2.为每一项数据分配一个唯一ID,得到Dictionary;

[0059] 3.3.抽取Dictionary中每项数据相同的头部信息,得到Header;

[0060] 3.4.将原始的三元组数据用ID替换,得到ID化的三元组集合;

[0061] 4.关系挖掘第一步,合并具有相同主语和谓语的三元组,参照附图1中Step1,推导公式参照附图3中Rule1;

[0062] 5.关系挖掘第二步,将所有三元组按照主语进行分类,合并相同主语的所有谓语和宾语,形成谓语向量和宾语向量,提取每个主语的谓语向量,并对谓语向量内部进行排序,参照附图1中Step2,推导公式参照附图3中Rule2;

[0063] 6.关系挖掘第三步,合并具有相同谓语(谓语向量)和宾语的三元组,参照附图1中Step3,推导公式参照附图3中Rule3;

[0064] 7.关系挖掘第四步,将所有三元组按照谓语(谓语向量)分类,合并相同谓语(谓语向量)的主语和宾语,形成内部已排序的主语向量和宾语矩阵,将这样一个由主语向量、谓语向量和宾语矩阵组成的结构称为一个数据块,参照附图1中Step4,推导公式参照附图3中Rule4;

[0065] 8.提取每个数据块的主语向量、谓语向量和宾语矩阵并建立压缩查询内存模型,压缩查询内存模型参照附图2;

[0066] 9.压缩状态下的SPARQL查询方式,在所有的数据块上可进行并发查询操作;

[0067] 详细过程如下:

[0068] 9.1.主语查询,遍历所有数据块的主语向量,因为主语向量内部已排序,使用二分查找方法,因此主语查询时间复杂度为O(log2n);

[0069] 9.2.谓语查询,遍历所有数据块的谓语向量,因为谓语向量内部已排序,使用二分查找方法,因此谓语查询时间复杂度为O(log2n);

[0070] 9.3.宾语查询,遍历所有数据块的宾语矩阵,因为宾语矩阵内部未排序,只能顺序查找,时间复杂度为O(n),针对宾语矩阵特别大的数据块,可以进行数据分块,以减少线性遍历查找带来的时间开销,参考附图4;

[0071] 9.4.复杂查询,所有的复杂查询都可以分解为成前面三种简单查询,再合并简单查询的结果。

[0072] 10.序列化写入文件,设定每个ID的存储长度相同,使用辅助符号“|”(或标识符)、“,”(主谓宾语分割符)和“/”(数据块分割符)实现序列化和反序列化。

[0073] 本文中所描述的具体实施例仅仅是对本发明精神作举例说明。本发明所属技术领域的技术人员可以对所描述的具体实施例做各种各样的修改或补充或采用类似的方式替代,但并不会偏离本发明的精神或者超越所附权利要求书所定义的范围。

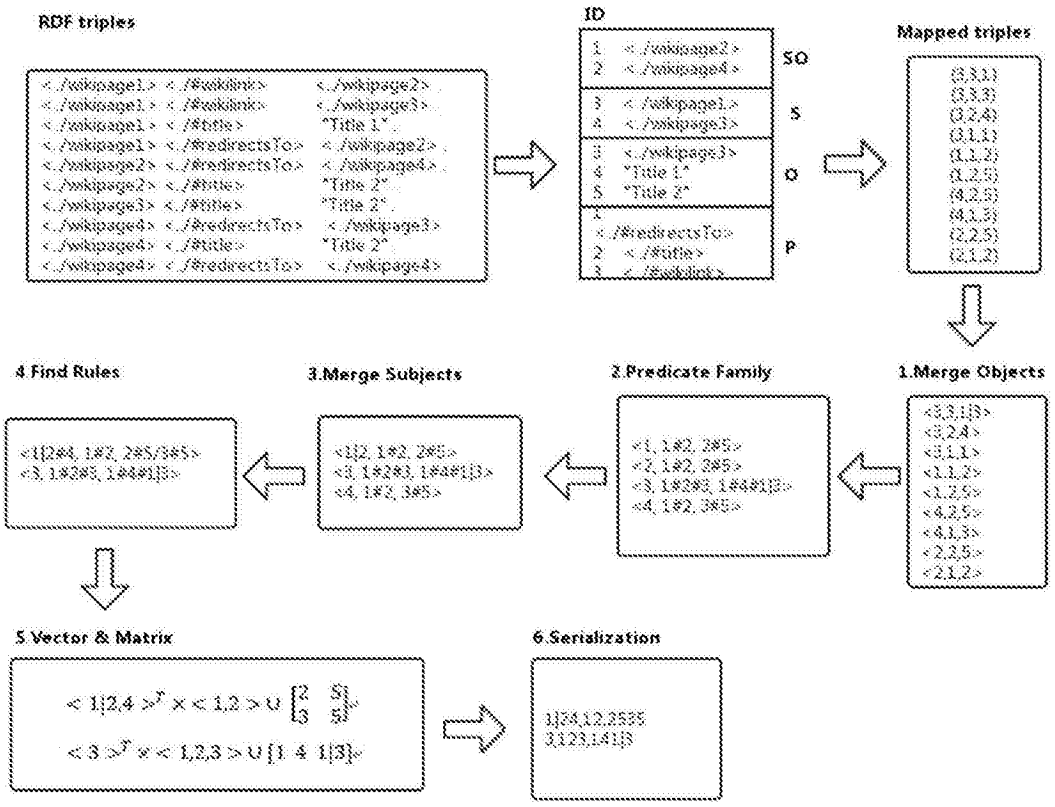

RDF triples
ID
Mapped triples
SO
S
O
P
1.Merge Objects
2.Predicate Family
3.Merge Subjects
4 Find Rules
5 Vector & Matrix
6.Serialization

图1

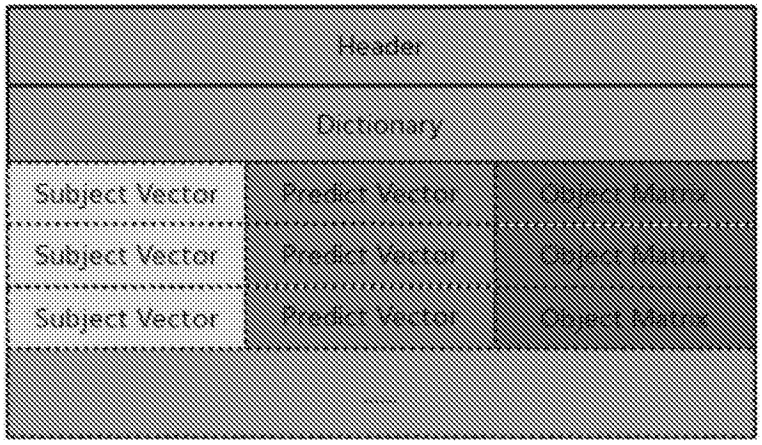

Header
Dictionary
Subject Vector
Predict Vector
Object Matrix
Subject Vector
Predict Vector
Object Matrix
Subject Vector
Predict Vector
Object Matrix

图2

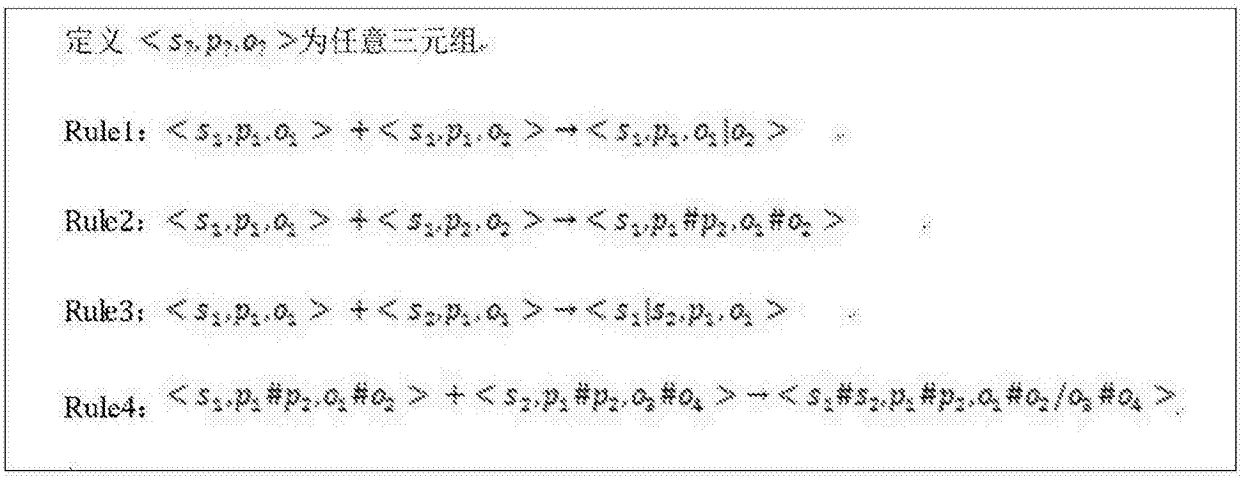



图3

$$< s_1, s_2, s_3, s_4 >^T \times < p_1, p_2 > \cup \begin{bmatrix} o_{11} & o_{12} & o_{13} & o_{14} \\ o_{21} & o_{22} & o_{23} & o_{24} \end{bmatrix}$$

$$\Downarrow$$

$$< s_1, s_2 >^T \times < p_1, p_2 > \cup \begin{bmatrix} o_{11} & o_{12} \\ o_{21} & o_{22} \end{bmatrix}$$

$$\Uparrow$$

$$< s_3, s_4 >^T \times < p_1, p_2 > \cup \begin{bmatrix} o_{13} & o_{14} \\ o_{23} & o_{24} \end{bmatrix}$$

图4

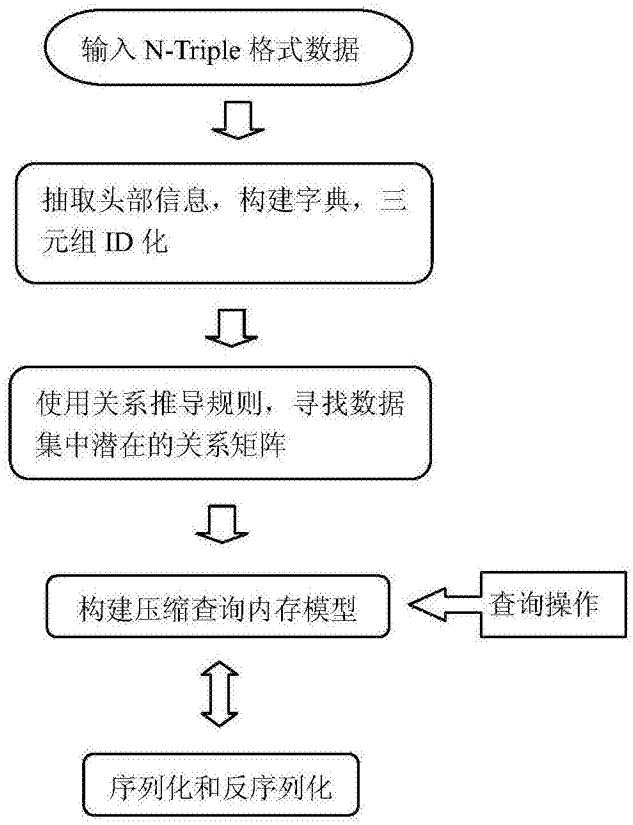

输入 N-Triple 格式数据
抽取头部信息，构建字典，三元组 ID 化
使用关系推导规则，寻找数据集中潜在的关系矩阵
构建压缩查询内存模型
查询操作
序列化和反序列化

图5