一种基于超节点理论的本体关系消冗算法

于洪涛 丁悦航* 刘树新 黄瑞阳 谷允捷
(国家数字交换系统工程技术研究中心 郑州 450002)

摘 要：本体作为指导知识图谱数据构建的上层结构，在知识图谱技术中具有重要意义。本体在发展的过程中会形成结构上的冗余。现有的本体消冗方法无法处理含有等价关系的本体结构，只能针对单一类属关系进行冗余的检测与消除。该文针对含有等价关系的本体提出一种基于超节点理论的消冗算法，首先将相互等价的节点看作超节点，消除单一类属关系之间的的冗余；然后还原等价节点，消除等价关系与类属关系之间的冗余。在计算机生成网络和真实网络上的实验和分析表明，该算法能够准确识别关系冗余，具有较高的稳定性和综合性能。

关键词：本体；等价关系；超节点；关系冗余；类属关系
中图分类号：TP311.1 **文献标识码**：A **文章编号**：1009-5896(2019)07-1633-08
DOI: 10.11999/JEIT180793

Eliminating Structural Redundancy Based on Super-node Theory

YU Hongtao DING Yuehang LIU Shuxin HUANG Ruiyang GU Yunjie
(*National Digital Switching System Engineering & Technological R & D Center*, *Zhengzhou* 450002, *China*)

Abstract: Ontology, as the superstructure of knowledge graph, has great significance in knowledge graph domain. In general, structural redundancy may arise in ontology evolution. Most of existing redundancy elimination algorithms focus on transitive redundancies while ignore equivalent relations. Focusing on this problem, a redundancy elimination algorithm based on super-node theory is proposed. Firstly, the nodes equivalent to each other are considered as a super-node to transfer the ontology into a directed acyclic graph. Thus the redundancies relating to transitive relations can be eliminated by existing methods. Then equivalent relations are restored, and the redundancies between equivalent and transitive relations are eliminated. Experiments on both synthetic dynamic networks and real networks indicate that the proposed algorithm can detect redundant relations precisely, with better performance and stability compared with the benchmarks.

Key words: Ontology; Equivalent relation; Super node; Relation redundancy; Transitive relationship

1 引言

近年来，知识图谱作为可推理的结构化数据集，已经越来越多地应用于智能语义搜索、移动个人助理和深度问答系统[1]。本体作为知识图谱的上层结构，在知识推理和数据层的构建中都发挥着重要作用：结构清晰、逻辑正确的本体不仅能指导构建多个符合标准的知识图谱，实现本体重用；而且能通过推理得到图谱中没有显式表达的信息，实现结构的自我完善。然而，在本体发展和维护的过程中[2,3]，可能会产生这样一类关系：两个概念之间显式地存在某种关系，与此同时，这个关系也可以通过推理的方式间接得到。称这种关系为“冗余关系”，称蕴含这种关系的本体为“冗余本体”。

冗余本体违反了本体建模的简约原则[4]，增加了本体发展和维护的负担。若本体中已经存在某种关系的隐式表达，向本体中重复添加其显式表达会造成本体内容有所更新的假象[5]。当冗余本体中的某个显式关系发生变化时，如果其隐式表达未能及时更新，就会造成本体的不一致，从而导致本体使用时逻辑混乱、效率低下[6]。此外，无冗余本体的结构更加清晰，便于本体重用[7]、本体抽取[8]和本体合并[9]。本体消冗技术可以与模块抽取技术相结合[10,11]，直接抽取出无冗余的子本体，便于后续的本体拼接、合并操作。因此，在网络信息海量化、复杂化的背景下，消除本体中存在的冗余具有重要的应用意义。

收稿日期：2018-08-09；改回日期：2019-02-25；网络出版：2019-03-04
*通信作者： 丁悦航 data_rabbit@163.com
基金项目：国家自然科学基金(61521003, 61803384)
Foundation Items: The National Natural Science Foundation of China (61521003, 61803384)

目前主要的本体关系消冗算法将本体中的概念看作节点，本体中连接概念的关系看作边，构造本体网络，进而在本体网络中检测冗余关系。文献[12]抽取相似概念描述中的特殊词构成词对，然后通过词对间的比较判断可能的缺失关系和冗余关系。文献[4]通过比较某节点的直接邻居和间接邻居集合检测类属关系间的冗余。文献[6]通过将本体转化为哈斯图实现冗余的消除。上述方法通过抽取本体中的单一类属关系，构造有向无环图，用基于图论的方法消除类属关系中的冗余。然而，等价关系的存在使得本体网络不再是有向无环图，从而导致上述方法陷入检测盲区，无法有效检测冗余。无环性的破坏使得关系消冗算法的性能受网络中等价关系数量影响较大，稳定性和准确性难以得到保障。

基于此，本文提出一种基于超节点理论的本体关系消冗算法。主要贡献如下：(1)针对本体关系消冗算法，提出了等价关系的存在会使得现有方法失效的问题；(2)将相互等价的节点看作一个“超节点”，进而将本体网络转化为有向无环图，通过类属关系消冗算法实现网络中类属冗余的检测与消除；(3)通过设置等价关系向量，运用向量扫描的方式在网络中检测和消除等价-类属冗余。在计算机生成网络和真实网络上的实验表明，本文算法能够基于超节点理论对本体网络进行转化，从而发现网络中的真实冗余，具有相对较高的稳定性和准确率。

2 问题描述

现有算法将本体中的概念看作节点，类属关系看作有向边，将本体文件转化为有向无环图$G(V,E)$，将一步可达且多步可达的节点对视为冗余边。然而，等价关系的存在破坏了网络的无环性，使得相互等价的节点对既一步可达又多步可达，从而使得现有算法错误地将等价关系判断为冗余关系。

下面首先给出本文中本体关系冗余的严格定义。

定义1 本体关系冗余：在本体网络$G(V,E)$中，若$\exists(i,j)\in E$，满足$\exists k_1,k_2,\cdots,(n\geq 1)\,k_n\in V$使得$(i,k_1),(k_1,k_2),\cdots,(k_n,j)\in E$，则称网络中存在关系冗余，称从点$i$到点$j$的边为冗余边。

现有的本体关系消冗方法只考虑了类属关系之间的冗余，等价关系的加入使得本体关系冗余相较于单一类属关系冗余的分析更为复杂。为了更清楚地描述关系冗余的结构，本文定义了如图1所示的4种基本冗余结构。

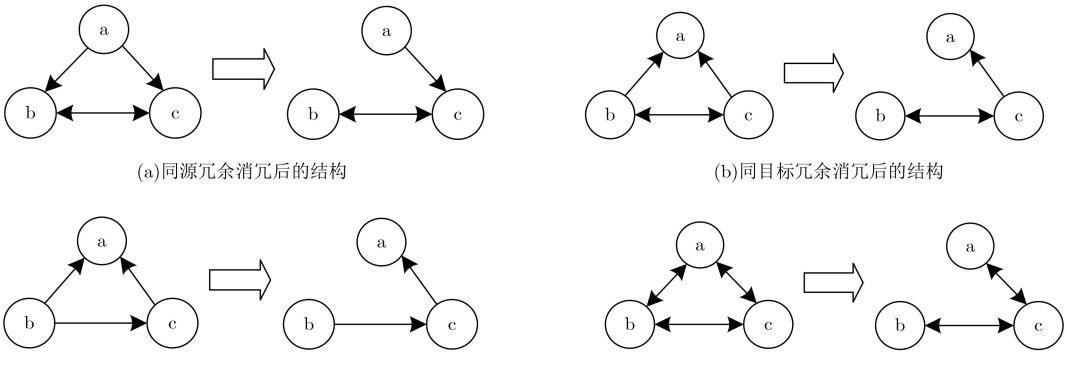
在4种基本冗余结构中，同源冗余和同目标冗余是等价关系和类属关系共同构成的冗余，本文称之为等价-类属冗余。传递冗余是类属关系构成的冗余，目前已有的研究都基于这种冗余结构。等价冗余是等价关系构成的冗余，本文不考虑这种冗余的消除。原因在于，等价关系通过本体合并被引入当前本体，去掉看似多余的等价关系会减弱两个子本体间的联系。需要注意的一点是，等价-类属冗余的消除方式不是唯一的。以图1(a)所示的同源冗余为例，去除边(a,b)或边(a,c)都能达到消冗的目的。在实际消冗过程中，去除其中任意一条边即可完成消冗任务。称这种现象为冗余边的不唯一性。

通过上述转化，本文将本体关系消冗问题转化成了等价-类属冗余和传递冗余的消除问题。

3 基于超节点理论的关系消冗

本文基于超节点化思想，首先将相互等价的节点看作一个超节点，将本体网络转化为有向无环图，进而通过层次冗余检测算法(Fast, Exhaustive Detection of Redundant hierarchical Relations, FEDRR)[4]实现有向无环图上传递冗余的消除。随后通过向量扫描检测消除网络中的等价-类属冗余，恢复等价关系，从而在本体网络上实现基于超节点理论的关系消冗。

为了消除网络中的传递冗余，将相互等价的节点转化为超节点。首先计算网络中的等价节点集



(a)同源冗余消冗后的结构　(b)同目标冗余消冗后的结构

(c)传递冗余消冗后的结构　(d)等价冗余消冗后的结构

图 1 本体网络中的4种冗余

合。其中，每个集合代表一种等价关系，集合中所有节点相互等价。将每个集合中的点看作一个超节点，在网络中构造所有可能的同源冗余和同目标冗余，并去除等价关系。经过上述步骤，网络在连通性不变的情况下转化为一个有向无环图，因此，可利用现有算法消除网络中的传递冗余。随后利用等价关系向量进行扫描，消除同源冗余和同目标冗余，恢复等价关系，最终得到无冗余的网络。

图2展示的是一个示例本体消冗的过程。图2中单向箭头表示类属关系，双向箭头表示等价关系。

3.1 超节点的转化

等价关系的存在破坏了本体网络的无环性，因此，基于有向无环图的消冗算法不再适用于含有等价关系的本体网络。如果在不破坏网络连通性的条件下将网络中的等价关系转化为类属关系，就能将网络转化为有向无环图，进而利用现有算法消除传递冗余。

基于上述思想，本文通过将本体网络中相互等价的节点看作超节点[13]的方法，将本体网络转化为有向无环图。具体地，去除等价关系前，在网络中添加相应的同源冗余和同目标冗余以保证网络的连通性不变。

记本体网络的邻接矩阵为$\boldsymbol{M}$。首先求出本体网络中蕴含的等价关系。称表示等价关系的矩阵为等价关系矩阵，记为$\mathbf{Me}$。

$$\begin{aligned}\mathbf{Me}&=t\left(\boldsymbol{M}\otimes\boldsymbol{M}^{\mathrm{T}}\right)\\&=\left(\boldsymbol{M}\otimes\boldsymbol{M}^{\mathrm{T}}\right)\oplus\left(\boldsymbol{M}\otimes\boldsymbol{M}^{\mathrm{T}}\right)^{2}\cdots\end{aligned}\tag{1}$$

其中，n为网络规模，$\boldsymbol{A}\otimes\boldsymbol{B}$表示矩阵$\boldsymbol{A}$和$\boldsymbol{B}$的数量积，$\boldsymbol{A}\oplus\boldsymbol{B}$表示矩阵$\boldsymbol{A}$和$\boldsymbol{B}$的布尔加，$t(\boldsymbol{R})$表示求矩阵$\boldsymbol{R}$的传递闭包。

得到网络中的所有等价关系后，将相互等价的节点看作一个超节点，去除超节点内部的等价关系，添加与当前超节点相关的等价-类属冗余，从而将网络转化为有向无环图。

记初始矩阵$\boldsymbol{M}$第i行j列的元素为M_{ij}，等价关系矩阵$\mathbf{Me}$第i行j列的元素为Me_{ij}。接下来分别构造同源冗余和同目标冗余。构造过程分别对应图2(c)和图2(d)。

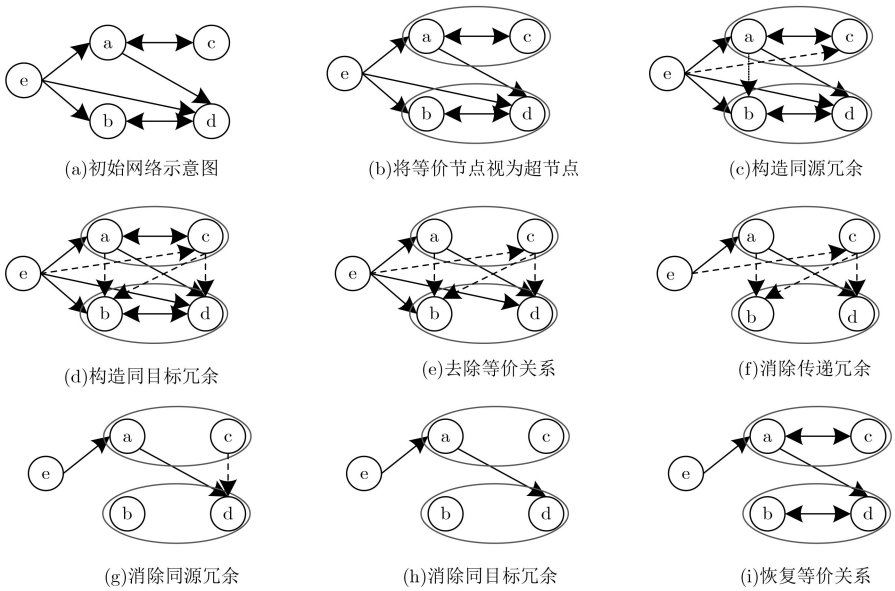
为构造同源冗余，在原网络中添加源节点到等价节点的连边。设添加同源冗余后的网络为$\mathbf{Ms}$，网络中第i行j列的元素记为Ms_{ij}。$\mathbf{Ms}$需保留原网络$\boldsymbol{M}$中的连边，并添加所有可能的同源冗余。即$\mathrm{Ms}_{ij}=1\Leftrightarrow M_{ij}=1\cup(\exists k\in V,\ \mathrm{s.t.}M_{ik}=1\cap\mathrm{Me}_{kj}=1)$。则有

$$\mathrm{Ms}_{ij}=\mathrm{bool}\left(\sum_{k}M_{ik}\times\mathrm{Me}_{kj}\right)\tag{2}$$

其中，$\mathrm{bool}(\cdot)$表示取布尔值，即若当前值大于1，置当前值为1；若当前值小于0，置当前值为0。

为构造同目标冗余，在原网络中添加等价节点到目标节点的连边。设添加同目标冗余后的网络为$\mathbf{Mt}$，网络中第i行j列的元素记为Mt_{ij}。$\mathbf{Mt}$需保留原网络$\boldsymbol{M}$中的连边，并添加所有可能的同目标冗余。即$\mathrm{Mt}_{ij}=1\Leftrightarrow M_{ij}=1\cup(\exists k\in V,\ \mathrm{s.t.}M_{kj}=1\cap\mathrm{Me}_{ik}=1)$。则有

$$\mathrm{Mt}_{ij}=\mathrm{bool}\left(\sum_{k}\mathrm{Me}_{ik}\cdot M_{kj}\right)\tag{3}$$



(a)初始网络示意图　(b)将等价节点视为超节点　(c)构造同源冗余

(d)构造同目标冗余　(e)去除等价关系　(f)消除传递冗余

(g)消除同源冗余　(h)消除同目标冗余　(i)恢复等价关系

图 2 本体消冗过程示意图

构造等价-类属冗余后，本体网络中所有至少经过一个等价关系相连的两个节点，都能通过类属关系直接相连，这意味着，等价关系的“桥梁”作用可完全由类属关系取代，可去除本体中的等价关系。至此，网络在连通性不变的情况下实现了到有向无环图的转化。

等价-类属冗余的构造使得等价关系的去除对本体中蕴含的结构信息没有损坏。设进行超节点转化后的网络为**Mr**，根据以上分析可以得到超节点转化公式为

$$\begin{aligned}\mathrm{Mr}_{ij}&=\mathrm{bool}(\mathrm{Ms}_{ij}+\mathrm{Mt}_{ij})-\mathrm{Me}_{ij}\\&=\mathrm{bool}\left(\sum_k(M_{ik}\cdot\mathrm{Me}_{kj}+\mathrm{Me}_{ik}\cdot M_{kj})\right)-\mathrm{Me}_{ij}\end{aligned}\tag{4}$$

3.2 传递冗余的消除

本体网络$\boldsymbol{M}$通过超节点转化的处理，被转化成了只含有单一类属关系的有向无环图**Mr**。可以分别通过求传递闭包的Warshall算法[14]、文献[4]中的FEDRR算法和文献[6]中的基于Hasse图的算法消除冗余边。下面通过实验，从中选出一个最高效的消冗算法用于本文。

随机生成100个含100个节点的有向无环图，100个含1000个节点的有向无环图和10个含10000个节点的有向无环图。3种消冗算法在不同规模网络上的运行时间如图3所示。

从图3中可以看出，在网络规模较大时，FEDRR算法的运行时间最短。相较于另外两种方法，FEDRR的运行时间取决于网络中节点和边的数量，抖动较大。但是，在3种规模的网络中，FEDRR的平均运行时间都是最短的。因此，本文选用FEDRR算法消除传递冗余，消冗过程对应图2(f)。记消除传递冗余后的网络为$\boldsymbol{R}$。

3.3 等价-类属冗余的消除

经过传递冗余的消除，有向无环图**Mr**被转化成了无冗余网络$\boldsymbol{R}$。然而，原网络$\boldsymbol{M}$中仍然存在等价-类属冗余。因此，本节首先在网络$\boldsymbol{R}$上通过等价关系向量检测出当前网络中的等价-类属冗余并进行消冗，然后恢复原网络中的等价关系。网络中的等价-类属冗余有两个来源：网络中原本存在的冗余和超节点化时人为添加的冗余。

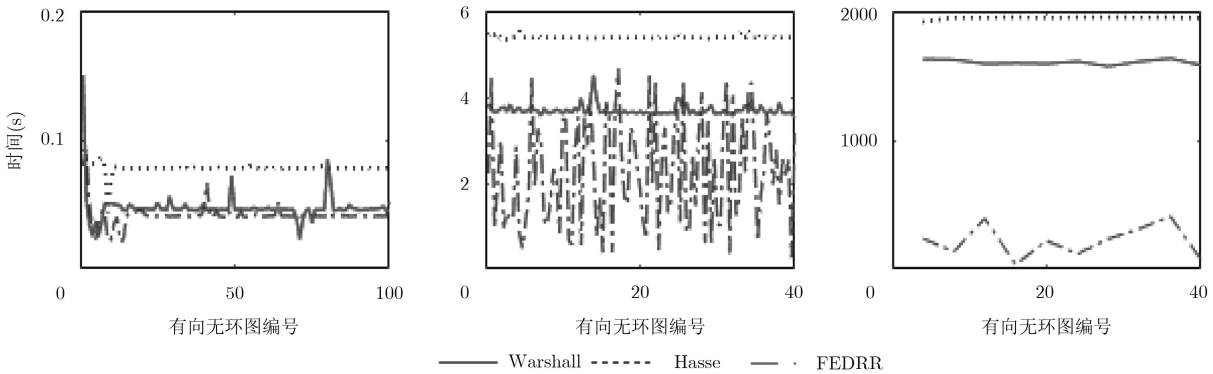
等价关系矩阵**Me**中的每一行代表一个等价节点集合：一行中对应位置值为1的节点相互等价。为方便计算，将等价关系矩阵**Me**中不相等的行存入新的矩阵$\boldsymbol{E}$, $\boldsymbol{E}$的每一行都是一个等价关系向量。接下来检测网络中的同源冗余和同目标冗余。

首先检测同源冗余，检测过程对应图2(g)。计算每个等价关系向量与矩阵$\boldsymbol{R}$每一行的数量积，得到冗余向量。若向量中有多个值为1的元素，说明当前等价关系与冗余向量对应位置值为1的元素之间存在同源冗余。上述判别方法可转化为，若冗余向量中各元素的和大于1，则当前等价关系存在同源冗余。因此，同源冗余的检测可转化为等价关系向量与$\boldsymbol{R}^{\mathrm{T}}$的向量积

$$\begin{aligned}S_{ij}&=\mathrm{bool}\left(\boldsymbol{E}.\mathrm{row}(i)\otimes\boldsymbol{R}.\mathrm{row}^{\mathrm{T}}(j)-1\right)\\&=\mathrm{bool}\left(\sum_k E_{ik}\cdot R_{jk}-1\right)\end{aligned}\tag{5}$$

其中，$\boldsymbol{R}$为消除传递冗余后的超节点化网络，$\boldsymbol{S}$为同源冗余检测矩阵，S_{ij}=1表示网络中第i种等价关系与第j个节点间存在同源冗余，即存在$\sum_k E_{ik}\cdot R_{jk}$条边从节点$j$指向等价关系$i$(超节点$i$)。

接下来检测同目标冗余，检测过程对应图2(h)。计算每个等价关系向量与矩阵$\boldsymbol{R}$每一列的数量积，得到冗余向量。若向量中有多个值为1的元素，说明当前等价关系与冗余向量对应位置值为1的元素



(a)网络规模为100 (b)网络规模为1000 (c)网络规模为10000

图 3 3种算法在不同规模网络下的运行时间

之间存在同目标冗余。上述判别方法可转化为，若冗余向量中各元素的和大于1，则当前等价关系存在同目标冗余。因此，同目标冗余的检测可转化为等价关系向量与$\boldsymbol{R}$的向量积

$$\begin{aligned} T_{ij} &= \text{bool}(\boldsymbol{E}.\text{row}(i) \otimes \boldsymbol{R}.\text{column}(j) - 1) \\ &= \text{bool}\left(\sum_k E_{ik} \cdot R_{kj} - 1\right) \end{aligned} \tag{6}$$

其中，$\boldsymbol{T}$为同目标冗余检测矩阵。同理，若T_{ij}=1，说明存在$\sum_k E_{ik} \cdot R_{kj}$条边从等价关系$i$(超节点$i$)指向节点$j$。

由于冗余边的不唯一性，保留超节点与普通节点之间若干条边中的一条，即可消除网络中的等价-类属冗余。最后恢复等价关系，就得到了消冗后的网络。恢复等价关系的计算方法如式(7)所示

$$R_{ij}=\text{bool}(R_{ij}+\text{Me}_{ij}) \tag{7}$$

3.4 算法实现

算法的实现分为以下5个步骤：

步骤 1　抽取本体网络$G(V,E)$中的等价关系；

步骤 2　将网络中的等价关系转化为类属关系；

步骤 3　消除传递冗余；

步骤 4　消除等价-类属冗余；

步骤 5　恢复等价关系。

算法伪代码如表1所示，算法的总体复杂度量级为$O(n^3)$。其中，超节点转化阶段的算法复杂度为$O(n^3)$：求传递闭包算法的复杂度为$O(n^3)$，将等价关系转化为类属关系的复杂度为$O(n^3)$。冗余消除阶段的算法复杂度为$O(n^3)$：消除传递冗余，算法复杂度为$O(c \cdot n+|E|)=O(n^2)$，向量检测算法复杂度为$O(n^3)$，恢复等价关系的操作只涉及加法运算，复杂度为$O(1)$。

表 1　消冗算法伪代码

输入：本体网络$\boldsymbol{M}$

输出：消冗后的本体网络$\boldsymbol{R}$

(1) /*超节点的转化*/

(2) **Me** $\leftarrow t(\boldsymbol{M} \otimes \boldsymbol{M}^{\text{T}})$/*抽取本体网络中的等价关系，存入**Me***/

(3) $\text{Mr}_{ij} \leftarrow \text{bool}\left(\sum_k (M_{ik} \cdot \text{Me}_{kj}+\text{Me}_{ik} \cdot M_{kj})\right) - \text{Me}_{ij}$/*将等价关系转化为类属关系*/

(4) /*传递冗余的消除*/

(5) $\boldsymbol{R} \leftarrow$ **Mr**

(6) $D[V], I[V]$=FEDRR(G(V, E))/*用FEDRR算法求出网络中每个节点的孩子集合$D[V]$与后代集合$I[V]$*/

(7) **for all** $v \in V$ **do**

(8)　**for all** $s \in I[v] \cap D[v]$ **do**

(9)　　delete (s, v, R) /*置R_{sv}=0*/

(10)　**end for**

(11) **end for**

(12) /*等价-类属冗余的消除*/

(13) $\boldsymbol{E} \leftarrow$Equiv(**Me**)/*$E$每行表示一种等价关系*/

(14) S_{ij}=bool$\left(\sum_k E_{ik} \cdot R_{jk} - 1\right)$/*构造同源冗余*/

(15) T_{ij}=bool$\left(\sum_k E_{ik} \cdot R_{kj} - 1\right)$/*构造同目标冗余*/

(16) **for** i **from** 1 to n **do**/*去除矩阵中等价关系与偏序关系之间的冗余，n是矩阵大小*/

(17)　**for** j **from** 1 to n **do**

(18)　　**if** $S_{ij} > 1$ **then**/*去除同源冗余*/

(19)　　Elim_Line($\boldsymbol{E}$.row(i), $\boldsymbol{R}$.row(j))/*只保留节点j到超节点i的一条边*/

(20)　　**end if**

(21)　　**if** $T_{ij} > 1$ **then**/*去除同目标冗余*/

(22)　　Elim_Line($\boldsymbol{E}$.row(i), $\boldsymbol{R}$.column(j))/*只保留超节点i到节点j的一条边*/

(23)　**end if**

(24)　**end for**

(25) **end for**

(26) /*恢复等价关系*/

(27) $R_{ij} \leftarrow \text{bool}(R_{ij}+\text{Me}_{ij})$

4　实验验证

为了验证本文算法的准确性，分别在生成网络和真实本体网络上进行实验。通过自回避随机游走[15]的方式在两种网络上随机添加冗余，然后用已有的Warshall算法、基于Hasse图的算法和FEDRR算法3种算法，以及本文中的算法进行消冗，通过查准率p、查全率r和调和指标F进行性能对比，评价本文消冗算法的性能。实验表明，本文算法较之于其他算法更适用于真实情况，有更高的综合性能。

4.1 评价准则与方法

若已知网络的冗余边集合为E，将算法检测出的冗余边集合$E1$映射到E上，则p表示检测出的冗余边中真正冗余边的比例，r表示正确检测出的冗余边占所有冗余边的比例，F是前两种指标的调和指标。3种参数的计算公式为

$$p=|E \cap E1|/|\text{E1}| \tag{8}$$

$$r=|E \cap E1|/|E| \tag{9}$$

$$F=2 \times (p \times r) / (p+r) \tag{10}$$

4.2 计算机生成网络上的实验

研究表明，绝大多数本体层次网络是有向无环图[16]。因此，本文基于生成有向无环图的算法生成网络数据集。具体地，首先生成有向无环图，然后基于当前网络添加等价节点对，最后随机添加冗

余。生成网络的节点数、等价关系数、冗余边数等相关参数都可灵活设置，能够生成较高质量的网络测试数据。本文生成了一组规模为1000，冗余边数量间隔为100的测试数据集和一组规模间隔为100，冗余边数量为随机数的测试数据集。具体配置如表2所示。

表 2 随机生成网络配置参数

	网络规模N	网络个数n	最大等价节点对数	冗余边数
配置1	1000	100	500	(100,1000)
配置2	(100,1000)	100	$0.5N$	$(1,N)$

分别用Warshall算法、基于Hasse图的算法、FEDRR算法和本文算法对上述每种配置的网络检测冗余，将检测出的冗余与真实冗余进行对比，计算3种度量指标的均值。其中，p值越大，说明算法检测出的边越可能是冗余边；r值越大，说明有越多的冗余边被算法检测出来。F 值是p值和r值的调和平均值，反映了算法的综合性能。好的算法必须同时有较高的p值和r值，使得综合指标F 值也处于较高水平。

图4展示了4种算法在网络规模相同，冗余边数量不同时的性能指标。如图4(a)所示，冗余边数量对冗余检测的p值有一定影响。随着冗余边数量的增加，4种算法的p值都呈上升趋势。当冗余边数量大于500时，4种算法的p值都趋于稳定。这是因为当随机添加的冗余边数量较少时，等价-类属冗余占比较大，Hasse算法和Warshall算法无法有效识别此种冗余；此外，由于冗余边的不唯一性，FEDRR算法和本文算法识别的冗余边未必是原定的冗余边。因此，4种算法的p值与等价-类属冗余的占比负相关，当节点数量增加时，等价-类属冗余占比减小，p值增大。

如图4(b)所示，本文算法在r值上震荡较严重。这是因为，由于冗余边的不唯一性，算法在等价-类属冗余的检测中，检测出的可能不是原定的冗余边，但同样达到了消除冗余的效果。而当前实验设定等价-类属冗余的数量是随机的。因此，本文算法的r值具有一定的随机性，所以出现了震荡的现象。

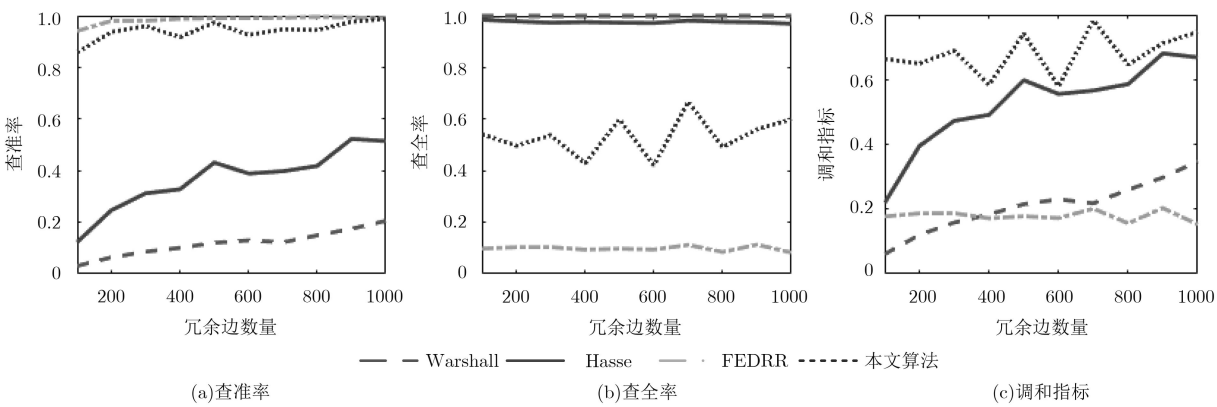
分析冗余边检测的整体性能发现，FEDRR算法的p值较高而r值较低，Hasse算法和Warshall算法的r值较高而p值较低。FEDRR算法r值较低的原因是，该算法将没有入边的节点作为初始节点，按先广搜索的方式计算每个节点的子集合与后代集合。然而，在含有等价关系的网络中，有的子图不含有无入边的节点，形成了算法的盲区，从而造成了部分冗余边的遗漏，使得r值较低。例如，FEDRR算法无法识别图1(b)所示子图中的同目标冗余。Hasse算法和Warshall算法p值较低的原因是，两种算法只保留网络中一步可达且其余步不可达的节点对，当网络中存在等价关系时，算法会将等价关系也误判作冗余关系，因此降低了p值。

图4(c)展示的是p值和r值的调和指标。从图中可以看出，本文算法在网络规模固定，冗余边数量变化的情况下F 值较其他算法更高，说明在冗余边不同的网络中，本文算法的综合性能最好。

图5展示了4种算法在网络规模不同时的性能指标。从图中可以看出，算法性能在网络规模变化时波动较小，说明其受网络规模影响较小。与图4呈现的规律相同，图5中FEDRR算法的p值较高而r值较低，Hasse算法和Warshall算法的r值较高而p值较低。原因同上，在此不再赘述。受等价-类属冗余数量随机性的影响，本文算法的r值抖动仍然较大。从图5(c)中可以看出，本文算法的调和指标F在网络规模变化的情况下较其他算法更高，说明在规模不同的网络中，本文算法的综合性能最好。

4.3 真实网络上的实验

真实网络相比计算机生成网络更不规则，因而



(a)查准率 (b)查全率 (c)调和指标

图 4 网络规模为1000时4种算法的查准率、查全率、调和指标的性能对比

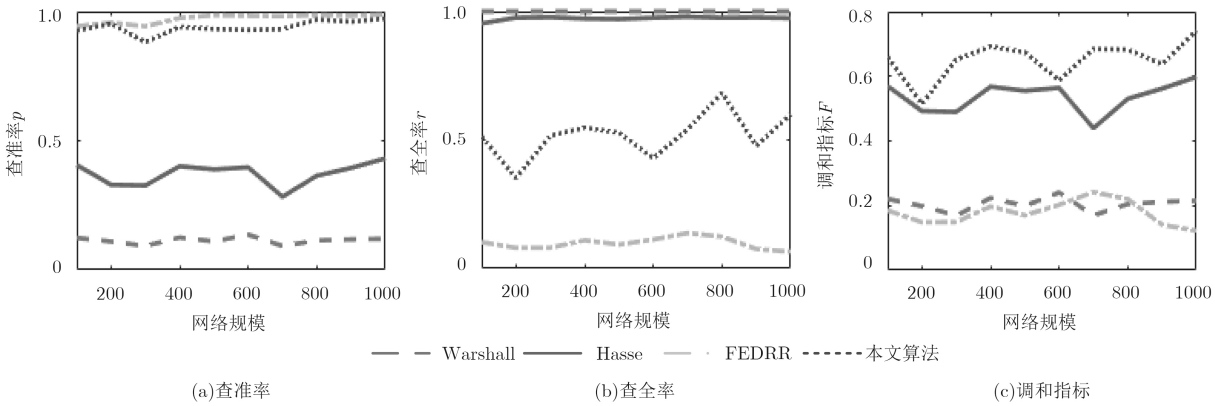


图 5 网络规模间隔为100时4种算法的查准率，查全率，调和指标的性能对比

其不确定性往往也更大。目前，经典的公开本体数据集大多由专家构建，不存在冗余。然而，本体在演化的过程中可能会产生冗余，新构建的本体中也可能存在冗余。基于此，本文采用地球与环境术语语义网(Semantic Web for Earth and Environmental Terminology, SWEET)[17]数据集来进一步测试算法性能。SWEET是由美国宇航局地球办公室支持建设的规模最大的地球科学数据与术语研究项目[18]。数据集中所有本体由专家构建并广泛使用，因此，可认为实验所用的原始本体皆为无冗余本体。

通过自回避随机游走的方式在选定的测试数据中随机添加冗余，然后用Warshall算法、基于Hasse图的算法、FEDRR算法和本文算法对测试数据进行消冗处理，分别得到其查准率、查全率和调和指标的均值。图6给出了各算法的各项性能表现，综合比较发现：Hasse算法和Warshall算法同时在p值上较低，而r值较高，使得两项指标相差悬殊；FEDRR算法p值较高，r值较低，使得两项指标相差悬殊。本文算法的p值和r值相差较小，使得F值维持在较高水平，证明了其检测冗余的有效性和准确性。此外，本文算法在真实网络上的p值相较于计算机生成网络明显偏低。这是因为，SWEET数据集中本体蕴含的等价关系数量多于生成网络，从而可能生成更多的等价-类属冗余。等价-类属冗余中冗余边的不唯一性造成了本文算法p值的下降。

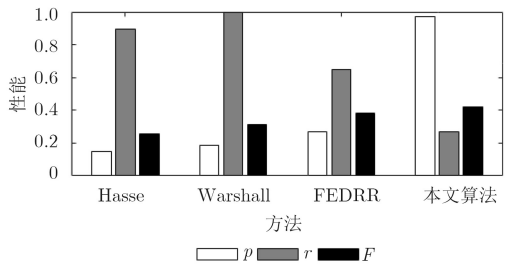


图 6 4种算法基于真实网络的查准率、查全率、调和指标性能对比

综合计算机生成网络和真实网络上的实验可以得出结论：在不同网络规模、不同冗余边数的条件下，基于超节点理论的本体关系消冗算法在综合性能方面相比其他算法有所改善。此外，由于冗余边的不唯一性，算法可能通过消除非冗余边达到消冗目的。因此，性能指标反映的是算法性能的下限，算法的真实性能理论上高于指标值。

5　结束语

本文提出了一种基于超节点理论的本体关系消冗算法，采用将网络中的等价节点转化为超节点的方式将本体网络转化为有向无环图，然后利用类属关系消冗算法进行传递冗余的消除，最后利用向量检测算法消除网络中存在的等价-类属冗余，从而解决已有方法仅针对单一类属关系，难以处理等价关系的问题。实验和分析表明：基于超节点理论的本体关系消冗算法能够有效转化本体网络中的等价关系，在计算机生成网络和真实本体网络的消冗中具有较高的准确率、稳定性和广泛的适用性。在保证稳定性的前提下，如何将冗余检测算法进一步应用于大规模本体网络，以及准确评估本体网络中冗余的数量等问题，是下一步研究工作的重点。

参 考 文 献

[1] 刘峤, 李杨, 段宏, 等. 知识图谱构建技术综述[J]. 计算机研究与发展, 2016, 53(3): 582–600. doi: 10.7544/issn1000-1239.2016.20148228.
LIU Qiao, LI Yang, DUAN Hong, *et al*. Knowledge graph construction techniques[J]. *Journal of Computer Research and Development*, 2016, 53(3): 582–600. doi: 10.7544/issn1000-1239.2016.20148228.

[2] BORO D and BHUYAN Z. A game for shared ontology evolution[C]. Proceedings of the International Conference on Computing and Communication Systems, Shillong, India, 2018: 95–101. doi: 10.1007/978-981-10-6890-4_9.

[3] 刘树新, 季新生, 刘彩霞, 等. 局部拓扑信息耦合促进网络演

化[J]. 电子与信息学报, 2016, 38(9): 2180–2187. doi: 10.11999/JEIT151338.

LIU Shuxin, JI Xinsheng, LIU Caixia, *et al*. Information coupling of local topology promoting the network evolution[J]. *Journal of Electronics & Information Technology*, 2016, 38(9): 2180–2187. doi: 10.11999/JEIT151338.

[4] XING Guangming, ZHANG Guoqiang, and CUI Licong. FEDRR: Fast, exhaustive detection of redundant hierarchical relations for quality improvement of large biomedical ontologies[J]. *BioData Mining*, 2016, 9: 31–42. doi: 10.1186/s13040-016-0110-8.

[5] DENTLER K and CORNET R. Intra-axiom redundancies in SNOMED CT[J]. *Artificial Intelligence in Medicine*, 2015, 65(1): 29–34. doi: 10.1016/j.artmed.2014.10.003.

[6] 于娟, 熊振辉, 欧忠辉. 基于哈斯图的本体偏序关系消冗方法研究[J]. 情报学报, 2015, 34(3): 279–285. doi: 10.3772/j.issn.1000-0135.2015.003.006.

YU Juan, XIONG Zhenhui, and OU Zhonghui. Eliminating redundant ontology relations based on Hasse Diagram[J]. *Journal of the China Society for Scientific and Technical Information*, 2015, 34(3): 279–285. doi: 10.3772/j.issn.1000-0135.2015.003.006.

[7] OCHS C, PERL Y, GELLER J, *et al*. An empirical analysis of ontology reuse in BioPortal[J]. *Journal of Biomedical Informatics*, 2017, 71: 165–177. doi: 10.1016/j.jbi.2017.05.021.

[8] SHARP M E. Toward a comprehensive drug ontology: Extraction of drug-indication relations from diverse information sources[J]. *Journal of Biomedical Semantics*, 2017, 8: 2–12. doi: 10.1186/s13326-016-0110-0.

[9] CHATTERJEE N, KAUSHIK N, GUPTA D, *et al*. Ontology merging: A practical perspective[J]. *Information and Communication Technology for Intelligent Systems*, 2018: 136–145. doi: 10.1007/978-3-319-63645-0_15.

[10] JHA M, VELTRI P, GUZZI P H, *et al*. Network based algorithms for module extraction from RNASeq data: A quantitative assessment[C]. Proceedings of 2017 IEEE International Conference on Bioinformatics and Biomedicine, Kansas City, USA, 2017: 1312–1315. doi: 10.1109/BIBM.2017.8217852.

[11] DORAN P, TAMMA V, and IANNONE L. Ontology module extraction for ontology reuse: An ontology engineering perspective[C]. Proceedings of the Sixteenth ACM Conference on Conference on Information and Knowledge Management, Lisbon, Portugal, 2007: 61–70. doi: 10.1145/1321440.1321451.

[12] ABEYSINGHE R, HINDERER E W, MOSELEY H N B, *et al*. Auditing subtype inconsistencies among gene ontology concepts[C]. Proceedings of 2017 IEEE International Conference on Bioinformatics and Biomedicine, Kansas City, USA, 2017: 1242–1245. doi: 10.1109/BIBM.2017.8217835.

[13] DIETZE F, VALDEZ A C, KAROFF J, *et al*. That's so meta! Usability of a hypergraph-based discussion model[C]. Proceedings of the 8th International Conference on Digital Human Modeling. Applications in Health, Safety, Ergonomics, and Risk Management: Health and Safety, Vancouver, Canada, 2017: 248–258.

[14] WARSHALL S. A theorem on boolean matrices[J]. *Journal of the ACM (JACM)*, 1962, 9(1): 11–12. doi: 10.1145/321105.321107.

[15] 刘树新, 季新生, 刘彩霞, 等. 一种信息传播促进网络增长的网络演化模型[J]. 物理学报, 2014, 63(15): 158902. doi: 10.7498/aps.63.158902.

LIU Shuxin, JI Xinsheng, LIU Caixia, *et al*. A complex network evolution model for network growth promoted by information transmission[J]. *Acta Physica Sinica*, 2014, 63(15): 158902. doi: 10.7498/aps.63.158902.

[16] 徐雷. 本体网络结构及其演化研究[D]. [博士论文], 武汉大学, 2014.

XU Lei. A research on the structure of ontology networks and its evolution[D]. [Ph.D. dissertation], Wuhan University, 2014.

[17] RASKIN R G and PAN M J. Knowledge representation in the semantic web for Earth and environmental terminology (SWEET)[J]. *Computers & Geosciences*, 2005, 31(9): 1119–1125. doi: 10.1016/j.cageo.2004.12.004.

[18] SAVIĆ M, IVANOVIĆ M, and JAIN L C. Complex Networks in Software, Knowledge, and Social Systems[M]. Cham: Springer, 2019: 143–175.

于洪涛：男，1970年生，研究员，研究方向为网络安全、网络大数据分析.

丁悦航：女，1995年生，硕士生，研究方向为数据挖掘、知识图谱.

刘树新：男，1987年生，博士生，研究方向为复杂网络、链路预测、移动网络安全.

黄瑞阳：男，1986年生，副研究员，研究方向为网络大数据分析，大图挖掘.

谷允捷：男，1994年生，硕士生，研究方向为新型网络体系结构，数据挖掘与网络优化.