# 一种基于 IDEF1x 模型的层次多关系聚类算法

黄少滨 [1] 程 媛 [1] 万庆生 [1] 刘国峰 [1] 申林山 [1]

**摘 要** 多关系聚类仍存在利用统计方法提取一对多联系对应的信息时会忽略数据的原始特征、不同关系表间的联系出现的回路可能导致信息重复利用等问题, 且尚未见有效的解决方法. 本文认为利用 IDEF1x 模型中不同联系的特点, 可重构有助于解决上述问题的模型. 因此基于 IDEF1x 模型构建多关系数据集中表间关联关系层次模型的框架, 然后定义框架中不同种类的联系对聚类结果传递的影响, 以及整合多个子节点聚类结果的方法, 并以此为基础提出新的多关系聚类算法. 在真实的以及人工数据集上的实验效果表明, 相较于单关系聚类算法以及对比的多关系聚类算法, 所提算法可获得较准确的聚类结果.

**关键词** 多关系聚类, IDEF1x 模型, 最短路径, 结果传递

**引用格式** 黄少滨, 程媛, 万庆生, 刘国峰, 申林山. 一种基于 IDEF1x 模型的层次多关系聚类算法. 自动化学报, 2014, **40**(8): 1740−1753

**DOI** 10.3724/SP.J.1004.2014.01740

## A Hierarchical Multi-relational Clustering Algorithm Based on IDEF1x

HUANG Shao-Bin[1] CHENG Yuan[1] WAN Qing-Sheng[1] LIU Guo-Feng[1] SHEN Lin-Shan[1]

**Abstract** There still exist some problems in multi-relational clustering, such as the primitive characters of data may be ignored when getting information reflected by one-to-many relationship through statistical methods, and information reutilization may result from the loops in the relationship between tables. We feel that a model can be rebuilt to solve these problems based on the characters of relationships in the IDEF1x model. Therefore, a hierarchical model of relationships between tables in multi-relational dataset is built based on the IDEF1x model, the effect brought by different relationships in the model as well as the integration of multiple clustering results are defined, and then a new multi-relational clustering algorithm is proposed based on these definitions. Finally, the experiments on real-world and synthetic dataset indicate the accuracy of our algorithm.

**Key words** Multi-relational clustering, IDEF1x model, shortest path, transfer results

**Citation** Huang Shao-Bin, Cheng Yuan, Wan Qing-Sheng, Liu Guo-Feng, Shen Lin-Shan. A hierarchical multi-relational clustering algorithm based on IDEF1x. *Acta Automatica Sinica*, 2014, **40**(8): 1740−1753

传统聚类算法用于发现一张关系表中数据的类别, 并已有效应用于多个领域中[1−3]. 然而在很多应用中为了产生有意义的结果, 所使用的特征 (属性) 可能来自于多个关系表. 虽然可通过连接或聚合操作将多张关系表合并为一张单表, 但这种处理方式不仅会产生高维数据, 而且并非所有对象在每张关系表上均存在对应的数据信息, 因此整合后可能会存在数据稀疏的情况, 即数据点可能会分布在不同维的子空间中, 进而导致位于不同维的数据对象可能是距离相等的, 失去距离度量的意义; 还很难体现不同表间的联系对聚类造成的影响[4−6]. 因此出现了多关系聚类算法来解决此类问题. 在多关系数据集中, 包含若干相互联系的关系表, 其中存在一个目标关系表, 表中对象为聚类的目标对象; 其他关系表作为非目标关系提供辅助聚类的信息, 因此可称其为属性关系表; 多关系聚类即是利用多个关系表的信息, 基于对象的相似性, 将目标关系表中的数据对象划分成聚簇的过程[7].

目前多关系聚类算法的研究多从数据集的表示、对象间距离定义以及聚类过程三个主要步骤着手, 或基于一种常见的聚类算法, 并结合各自定义的多关系数据集中表间联系的特点, 提出多关系聚类算法[8−12], 如基于分层聚类的思想[8]、基于谱聚类的思想[9], 以及基于 $K$-means 算法[10−11] 等; 或基于矩阵分解的思想进行多关系聚类[13−15]. 然而仍存在一些尚待解决的问题, 如不同关系表间可能存在一对多联系; 若将各关系表看作点, 并将表间联系看作边, 那么各关系表及其联系构成了一个连通图, 从图

收稿日期 2013-08-29 录用日期 2013-12-23
Manuscript received August 29, 2013; accepted December 23, 2013
国家科技支撑计划 (2012BAH08B02), 哈尔滨工程大学中央高校基本科研业务专项资金项目 (HEUCF100603, HEUCFZ1212) 资助
Supported by National Key Project of Scientific and Technical Supporting Programs (2012BAH08B02), Fundamental Research Funds for the Central Universities (HEUCF100603, HEUCFZ1212)
本文责任编委 刘成林
Recommended by Associate Editor LIU Cheng-Lin
1. 哈尔滨工程大学计算机科学与技术学院 哈尔滨 150001
1. College of Computer Science and Technology, Harbin Engineering University, Harbin 150001

论的角度分析, 其中还可能存在回路. 如上问题均可能对多关系聚类结果产生一定的影响. 如图 1 所示为一个基本养老保险领域的数据集, 描述了参与基本养老保险的单位基本信息及其参保信息, 以及在职人员的缴费信息和退休人员的享受待遇信息. 下面以该数据集为例, 解释上述问题为何会影响聚类效果.

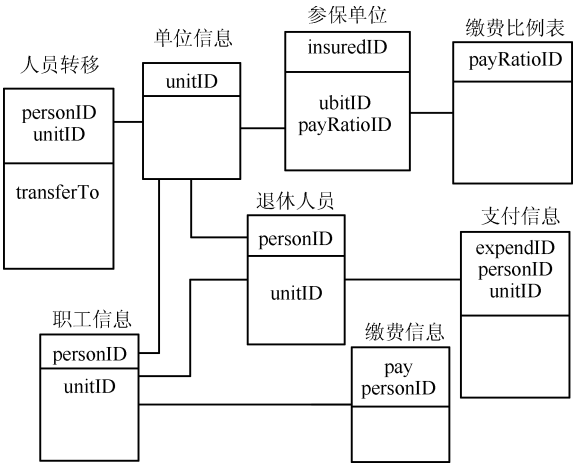



图 1　基本养老保险领域数据集

Fig. 1　Dataset of basic endowment insurance

若从数据库理论的角度考虑, 多关系数据集中各表之间会存在一对多联系, 如图 2 所示人员 1 和人员 2 分别对应多条缴费记录以及支付记录 (对应的实际业务含义为每个月均至少存在一条缴费记录以及支付记录). 若比较这两个人员的相似度必然涉及两者关于缴费和支付信息的区别, 因此需要获取两者关于缴费信息以及支付信息的特点, 现有算法的处理方式是取得其对应的多条缴费记录以及支付记录的统计信息 (如均值、最大值等). 然而应用统计方法易忽略原始数据的特征, 进而可能会影响目标对象的聚类效果.

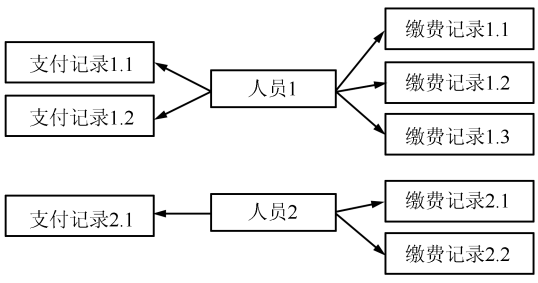



图 2　不同表间的一对多联系

Fig. 2　One-to-many relationships between tables

多关系数据集中各表及其之间的联系可构成一个连通图, 而连通图中可能存在回路. 如图 1 中的单位、个人以及退休人员信息三个表构成了一个回路. 若以传统方法进行多关系聚类, 对其中一种类型的对象聚类, 或计算其对象间相似度时, 均需另两种类型对象的信息进行辅助, 而回路的存在会导致信息反复迭代的利用. 例如计算两个单位的相似度, 不仅依赖于描述单位信息的属性, 还需借助与其关联的职工和退休人员信息; 相应的进一步计算职工 (或退休人员) 间相似度时, 同样需基于其自身的属性, 以及与其关联的单位、退休人员 (或职工) 信息而进行度量. 如图 3 中的任意一条 $(e_1, e_{1.1}, e_{1.2})$ 路径均描述了信息反复利用的过程. 因此面对该问题时, 需尽量不重复利用数据信息, 同时又保证聚类结果的准确率. 然而现有多关系聚类算法并未分析并处理该问题.

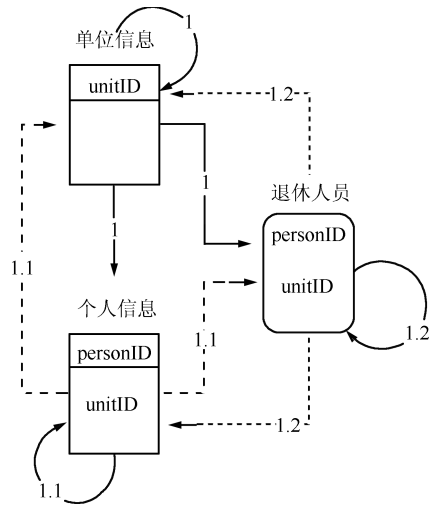



图 3　回路可能引起的信息重复利用

Fig. 3　Information reutilization resulting from loop

由第 2 节可知不同联系的特点是产生上述问题的主要原因, 那么如果更细致地划分表间联系, 并围绕对目标关系表聚类的目标, 可重新构建有助于解决上述问题的表间联系图. 数据库的 IDEF1x (Integrated DEFinition 1 extended) 概念模型[16−17] 丰富了表间的联系类型以及对象类型, 这种划分有助于细化表间联系图. 因此本文首先基于 IDEF1x 模型构建表间的关联关系层次模型, 构成层次多关系聚类过程分析的基础, 并将其作为解决上述问题的雏形; 另外多关系数据集中表间存在联系, 那么除了描述对象的属性外, 还需考虑联系对聚类的影响; 因此本文接着定义模型中不同类型的联系对聚类结果传递的影响, 以及整合多个子节点聚类结果的方法; 最后以上述方法为基础提出一种层次多关系聚类算法.

本文将从数据集的表示到多关系聚类过程, 完整地提出一种新的层次多关系聚类算法. 首先在第 1 节讨论 IDEF1x 模型及其特点; 然后在第 2 节介

绍描述多关系数据集的关联关系层次模型; 并在第 3 节讨论不同联系对聚类的影响及其相关问题, 进而提出多关系聚类算法; 在第 4 节通过实验验证该算法的有效性; 最后在第 5 节得出本文结论.

## 1 IDEF1x 模型

IDEF1x 模型是美国空军集成计算机辅助制造 (Integrated computer aided manufacturing, ICAM) 工程在结构化分析和设计方法基础上发展的一套系统分析和实际方法. 并基于 ER (Entity-relationship) 模型中的实体、属性与联系等概念, 构建数据库的概念模型, 具有更丰富的语义和表达能力.

相较于 ER 模型, IDEF1x 模型更细致地划分了实体类型, 进而通过不同类型的实体以及属性信息, 还丰富了实体间联系的类型: 首先它将实体分为独立实体、从属实体和相交实体; 其中独立实体不依赖于其他实体而单独存在; 而从属实体的存在依赖于其他实体, 可理解为对独立实体细节的描述; 相交实体用于描述两个具有多对多联系的独立实体间的关系, 将其抽象为一个实体不仅避免信息冗余, 还可以描述联系特征的属性信息.

通过描述实体以及属性信息, 可进一步将实体间的联系划分为标定联系、非标定联系和不确定联系. 其中,

1) 标定联系 (Identifying relationship): 表示从属实体和独立实体之间的联系, 由于从属实体依赖于独立实体而存在, 因此继承自独立实体的属性需作为该从属实体的主键;

2) 非标定联系 (Non-identifying relationship): 表示不存在依赖关系的实体之间所产生的联系, 并将继承属性作为继承实体的一般属性;

3) 不确定联系 (Non-specific relationship): 将多对多的联系转换成两个一对多的联系, 产生的关联关系表为相交实体, 且一对多的联系通过在多端继承一端的属性来实现.

下面将同一表内的对象视为同种类型, 而不同表中的对象则属于不同类型. 图 4 描述了图 1 中数据集对应的 IDEF1x 模型, 其中退休人员、缴费信息和支付信息表是从属实体, 人员转移属于相交实体, 其余表为独立实体. 首先单位信息、职工信息等表均不依赖于其他实体而存在; 而缴费明细信息需依赖于职工信息而存在, 因此缴费信息表属于从属实体, 那么这两个表间的联系属于标定联系, 因为缴费信息描述了职工的缴费行为, 缴费明细即从属于职工而存在, 同时缴费信息表继承了职工信息表中的主键 personID 作为关键字, 且一个参保职工对应多条缴费记录明细; 而参保单位与缴费比例表两个独立实体之间构成了非标定联系, 参保单位表将缴费比例表的主键 payRatioID 作为外键, 两个参保单位对象是否具有相同的缴费比例可通过该联系而体现, 即该联系可表示同种类型对象间的关系; 另外单位与职工信息表两个独立实体之间由于存在人员转移而构成了多对多的不确定联系, 即一个职工可能会多次转移单位, 而一个单位也可能会具有多个转移职工, 相交实体人员转移表即描述了该关联关系, 并分别继承了单位和职工信息表中的主键作为该表的主键, 因此单位与职工这两种不同类型对象间的转移关系可由该联系所表示.

可见 IDEF1x 模型相较于 ER 模型具有更为丰富的表达能力, 其描述的几种联系不仅可表示作为多关系聚类基础的同种类型以及不同类型对象间的关系, 如后文所述, 这几种联系的特点还是引起多关系聚类中存在问题的主要原因. 因此本文将基于 IDEF1x 模型的特点, 提出一种新的层次多关系聚类算法以解决上述问题.

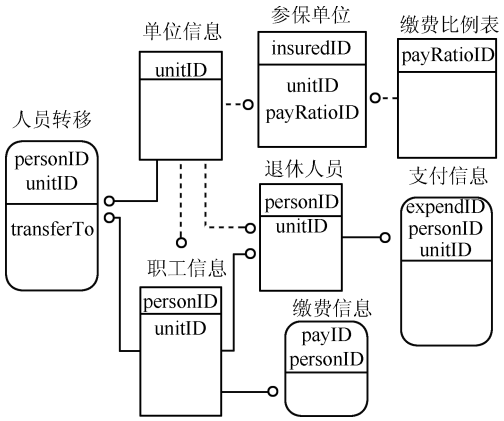



图 4 图 1 中数据集对应的 IDEF1x 模型

Fig. 4 IDEF1x model corresponding to dataset in Fig. 1

## 2 多关系数据集的关联关系层次模型

已有研究构建的多关系数据集表示, 多集中于关系模型和图这两种表示形式, 且存在如前文所述的回路, 进而造成聚类时的信息重复利用. 因此需构建一种新的多关系数据集的表示方法, 为拆分回路提供解决途径. 用于描述多关系数据集中实体联系的 IDEF1x 模型, 刻画了不同类型实体间的联系特点, 可用于区分不同联系对聚类的影响. 因此本文基于其构建一种表示多关系数据集的层次关联模型, 初步拆分原关联图中的回路, 进而作为多关系聚类的基础.

首先构建框架 (Clustering graph framework, CGK), 表示多关系数据集通过表间的相互联系所形成的连通图:

**定义 1.** 框架 CGK=$\langle T, \Sigma, \delta, t\rangle$ 是一个四元组,

其中,

1) $T$ 为数据集中的关系表集合;

2) $\Sigma$ 为 IDEF1x 模型中多种联系的集合, $\Sigma = \{n, i, s, n', i', s'\}$, 其中 $n'$ 为非标定联系, $i'$ 为标定联系, $s'$ 为非确定联系; 并且 $n$、$i$ 和 $s$ 表示由被继承实体指向继承实体的联系, 而 $n'$、$i'$ 和 $s'$ 则表示由继承实体指向被继承实体的联系;

3) $\delta$ 为联系函数, 即 $\delta : T \times \Sigma \to P(T)$, 其中 $P(T)$ 为 $T$ 的幂集, 即与某个表具有某种联系的表的集合为 $T$ 的一个子集;

4) $t$ 为目标关系表.

框架扩展了 IDEF1x 中联系的类型, 因为构建用于表示连通图的框架, 需从多关系聚类的目标关系表出发, 并沿原 IDEF1x 模型中的联系将各表逐步加入该框架中; 若考虑联系中继承实体与被继承实体的关系, 那么这种联系则是有向的. 为了有效区分这些有向联系, 以及进一步的逐层构建层次框架, 将联系类型扩展为如上的 $\{n, i, s, n', i', s'\}$.

下面以图 4 所示的多关系数据集 IDEF1x 模型为例, 构建其对应的框架 CGK=$\langle T, \Sigma, \delta, t\rangle$, 其中:

1) $T = \{T_i | 0 \le i \le 7\}$, 其中 $T_0 \sim T_7$ 分别表示单位信息表、职工信息表、参保单位表、退休人员表、缴费信息表、缴费比例表、支付信息表和人员转移表;

2) $\Sigma = \{n, i, s, n', i', s'\}$;

3) $t = T_0$, 即以单位为多关系的聚类目标, 其他关系表中的信息均用于辅助其聚类;

4) 联系函数 $\delta$ 如表 1 所示进行定义, 如其中的 $T_0 \xrightarrow{n} \{T_1, T_2, T_3\}$, 即与表 $T_0$ 具有非标定联系 $n$ 的表为 $\{T_1, T_2, T_3\}$.

表 1　定义 CGK 的联系函数 $\delta$

Table 1　Definition of relationship function $\delta$ in CGK

| | $n$ | $i$ | $s$ | $n'$ | $i'$ | $s'$ |
|---|---|---|---|---|---|---|
| $T_0$ | $\{T_1, T_2, T_3\}$ | {} | $\{T_7\}$ | {} | {} | {} |
| $T_1$ | {} | $\{T_3, T_4\}$ | $\{T_7\}$ | $\{T_0\}$ | {} | {} |
| $T_2$ | {} | {} | {} | $\{T_0, T_5\}$ | {} | {} |
| $T_3$ | {} | $\{T_6\}$ | {} | $\{T_0\}$ | {} | {} |
| $T_4$ | {} | {} | {} | {} | $\{T_1\}$ | {} |
| $T_5$ | $\{T_2\}$ | {} | {} | {} | {} | {} |
| $T_6$ | {} | {} | {} | {} | $\{T_3\}$ | {} |
| $T_7$ | {} | {} | {} | {} | {} | $\{T_0, T_1\}$ |

尽管基于该框架构建的各关系表形成的连通图中引入了不同类型的有向边, 由表间联系构成的回路却可能依然存在. 若再引入层次信息, 即可构建能进一步体现表间联系的层次模型.

以目标关系表为根节点, 与其直接联系的表作为它的子节点, 并结合继承实体与被继承实体的关系, 确定指向子节点的联系类型. 若所有与其直接相关的表均加入层次模型, 则依次处理其下层节点: 对于与当前处理的表直接相联系的关系表, 若已加入层次模型且位于该表的上层, 则不做处理直接处理下一关联的表; 若与该表位于同一层次或位于其后层, 且两表间尚未构建关联边, 则同样结合继承实体与被继承实体的关系, 确定指向关联表的联系类型; 否则作为该表的子节点加入层次模型; 如此逐层构建每一层次直至所有关系表均被处理, 即可形成体现多关系数据集中表间联系的层次模型.

显然下层节点不具有指向上层节点的关联边, 因此该层次模型可初步拆分原图中的回路. 另外其与树结构的区别在于同层节点间可能存在联系, 且位于下层的节点可能存在多个父节点. 那么由此而可能构成的回路将在第 3 部分中给出相应的解决方案.

如图 5 所示为以上述框架 CGK 为基础构建的体现表间关联关系的层次模型 CGKH, 共三个层次. 若仍以框架形式描述该层次模型, 可将 CGK 中的对象表示为 $(T_i, l)$, 其中 $l$ 为表 $T_i$ 在模型中的层次. 关系表间的联系保留层次模型中各节点间的有向边所表示的联系即可.

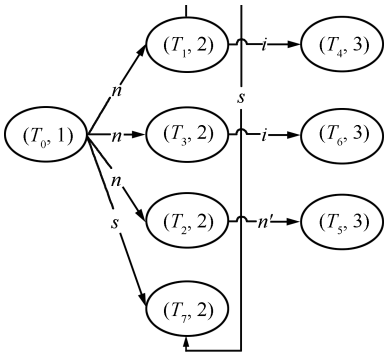



图 5　多关系数据集关联关系层次模型

Fig. 5　Relationship hierarchical model of multi-relational dataset

该层次模型体现了目标表以及与其相联系的其他表间的关系. 同时该模型为有向图, 多关系聚类中存在的问题, 是在将图中作为每条边起点的关系表作为目标表处理时所面临的, 而模型中几种联系的特点是产生问题的主要原因, 图 6 即描述了不同类型的联系与其可能导致的问题之间的对应关系. 首先由各自的定义可知, 三种联系均可能引起目标对象与非目标对象间存在一对多的联系. 另外非标定联系和不确定联系可能使得模型中表间的联系存在回路, 因为回路是在已有联系的表之间, 仍存在另一条由联系所连接的路径而形成的. 在两个表已关联的基础上, 能够再次将两个表联系起来的只有非标

定联系或不确定联系, 因为不可能有两个独立实体与同一从属实体具有标定联系.

可见丰富的表间联系类型有助于层次模型清晰地刻画多关系数据集中各表间相互关联的状态; 若针对其各自的特点, 分析不同联系对聚类的影响, 同时尽量避免因回路造成的信息重复利用问题, 则能有效引导多关系聚类过程的进行. 该模型即可作为多关系聚类的基础. 下面将进一步分析基于表间关联关系层次模型的多关系聚类算法, 使得该算法能够解决前文所述问题, 同时有效地完成多关系聚类任务.

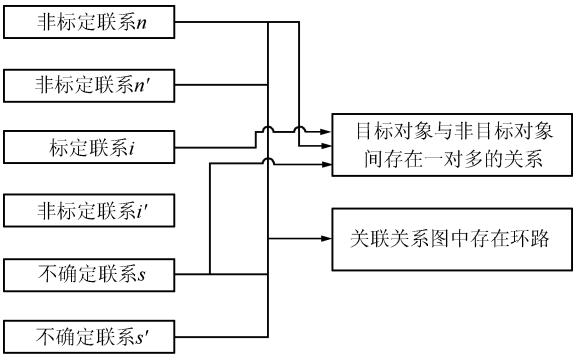



图 6 不同联系与可能导致的问题之间的对应关系

Fig. 6 Correspondence between relationship and problems in multi-relational clustering

## 3 多关系聚类算法

对于基于 IDEF1x 模型构建的表间关联关系层次模型而言, 由于划分了层次可初步拆分原关联图中存在的回路. 而为了结合其他关系表中的信息对位于根节点的目标对象聚类, 可从位于层次模型最底层的表开始聚类, 并将其聚类结果基于关联路径传递给与其联系的表, 以辅助相关表中对象的聚类; 与直接获取一对多联系所对应的统计信息相比, 能充分考虑数据的原始特征. 因此下面首先分析不同类型的联系对子节点聚类结果传递的影响; 同时由于非叶节点可能包含多个子节点, 考虑子节点聚类结果的整合方法; 特别的由于层次模型中关系表可能与多个同层表相关联, 需要一种同层表间的关联路径选择算法, 以获取一个无回路的且尽量保留表间联系的关联图; 然后基于路径选择后的关联图以及子节点结果整合方法, 确定根据关联路径传递子节点聚类结果的方法; 最后以上述过程为基础, 提出新的层次多关系聚类算法.

### 3.1 不同联系对聚类结果传递的影响

本文所提多关系聚类算法以 CGKH 框架为基础, 为了充分利用数据原始特征, 将作为辅助信息的表中对象的聚类结果逐层沿关联路径 (即表间联系) 向上传递直至作为根节点的目标表. 下面首先针对 CGKH 框架下不同类型联系的特点, 讨论其如何影响聚类结果的传递. 对于框架中的六种联系 $n$、$n'$、$i$、$i'$、$s$、$s'$:

非标定联系 $n$: 若 $T_x \xrightarrow{n} T_y$, 则表 $T_y$ 继承表 $T_x$ 的主键 $pk_x$ 作为一般属性, 且一个 $T_x$ 对象可能对应多个 $T_y$ 对象, 即存在一对多的情况. 此时 $T_x$ 作为目标表, 而将 $T_y$ 作为属性关系表, 基于 $T_y$ 所含信息对 $T_x$ 中的对象进行聚类. 为避免因直接获取多个 $T_y$ 对象的统计信息而破坏其原始数据特征, 本文根据 $T_y$ 自身特征获取其聚类结果, 进而辅助 $T_x$ 中对象聚类, 即利用多个 $T_y$ 对象的聚簇分布信息代替对应的如均值、最大值等初始统计信息.

设 $T_x$ 的对象集合 $D_x$, 基于自身属性聚类生成 $k$ 个聚簇 $C_x = \{C_{x1}, \cdots, C_{xk}\}$; 而 $T_y$ 中对象集合 $D_y$, 聚类生成 $k'$ 个聚簇 $C_y = \{C_{y1}, \cdots, C_{yk'}\}$; 对于 $\forall d_{x1}, d_{x2} \in D_x$, 在继承属性 $pk_x$ 上的值分别为 $pk_{x1}$ 和 $pk_{x2}$, 且 $d_{x1}$ 和 $d_{x2}$ 对应于表 $T_y$ 中的对象集合分别为 $S_{y1} = \{d_{y11}, \cdots, d_{y1m}\}$ 和 $S_{y2} = \{d_{y21}, \cdots, d_{y2n}\}$, 即表 $T_y$ 中在属性 $pk_x$ 上的值为 $pk_{x1}$ 和 $pk_{x2}$ 的对象集合分别为 $S_{y1}$ 和 $S_{y2}$;

利用 $S_{y1}$ 和 $S_{y2}$ 中对象的聚簇分布情况度量 $d_{x1}$ 和 $d_{x2}$ 关于 $T_y$ 的相似性, 而非基于 $S_{y1}$ 和 $S_{y2}$ 中对象关于各属性的统计值. 设 $S_{y1}$, $S_{y2}$ 关于 $C_y$ 的聚簇分布分别由 $dc_{y1}$ 和 $dc_{y2}$ 表示, 两者均包含 $k'$ 个分量, 分别对应于表 $T_y$ 的 $k'$ 个聚簇:

$dc_{y1} = (p_{y11}, \cdots, p_{y1k'})$, 其中 $p_{y1i} = \frac{|\{d_{y1a} | d_{y1a} \in C_{yi},\ d_{y1a} \in S_{y1}\}|}{m}$, $1 \le i \le k'$, 且 $m = |\ S_{y1}\ |$, 即第 $i$ 个分量为 $S_{y1}$ 中属于第 $i$ 个聚簇的对象在 $S_{y1}$ 中所占比例;

而 $dc_{y2} = (p_{y21}, \ldots, p_{y2k'})$, 其中 $p_{y2j} = \frac{|\{d_{y2b} | d_{y2b} \in C_{yj},\ d_{y2b} \in S_{y2}\}|}{n}$, $1 \le j \le k'$, 且 $n = |\ S_{y2}\ |$;

那么向量 $dc_{y1}$, $dc_{y2}$ 间的余弦相似度可用于计算 $d_{x1}$ 和 $d_{x2}$ 关于 $T_y$ 的相似性:

$$\mathrm{sim}_{py}(d_{x1}, d_{x2}) = \frac{\sum_{i=1}^{k'} p_{y1i} \cdot p_{y2i}}{\sqrt{\sum_{i=1}^{k'} (p_{y1i})^2} \times \sqrt{\sum_{i=1}^{k'} (p_{y2i})^2}} \tag{1}$$

以式 (1) 为相似性度量方式, 计算 $T_x$ 中任意两个对象关于 $T_y$ 的相似性, 即可得到相似性矩阵 $S_{n_x \times n_x}$, 其分量如式 (2) 所示:

$$s_{ij} = \begin{cases} 1, & \mathrm{sim}_{py}(d_{xi}, d_{xj}) \ge th \\ 0, & \mathrm{sim}_{py}(d_{xi}, d_{xj}) < th \\ -1, & d_{xi}\, or\, d_{xj} \notin T_y \end{cases} \tag{2}$$

其中 $1 \le i, j \le n_x$, 而 $n_x$ 为表 $T_x$ 中的对象个数, $th$ 为相似度阈值; 且若有对象在 $T_y$ 中不存在对应

信息, 其对应的行和列中的值均为 $-1$. 其中阈值 $th$ 的选择目标为保证尽量聚集相似对象, 值不宜过大或过小, 因为较大的阈值可能产生离群点, 而较小的阈值不利于有效的划分对象. 因此本文将任意两个 $T_x$ 对象间相似性规范化至 $(0,1)$ 区间且保留至十分位, 并以上三角矩阵记录该相似性, 取每行中的最大值组成集合 $Th$, 取该集合的众数作为阈值 $th$; 即选取可将多数对象划分为同一聚簇的相似度作为阈值.

然后根据如下步骤生成表 $T_x$ 关于 $T_y$ 中信息的 $k_{xy}$ 个聚簇: $C_{xy}=\{C_{xy1},\cdots,C_{xyk_{xy}}\}$, 体现了 $T_x$ 中对象基于 $T_y$ 中对应信息的不同分布.

1) 设 $k_{xy}=1, i=1$;

2) 对于 $T_x$ 中第 $i$ 个对象, 若未被处理过, 则将其加入聚簇 $C_{xyk_{xy}}$, 令 $j=i+1$;

a) 对于第 $j$ 个对象, 若未被处理过, 且 $\mathrm{sim}(d_i,d_j)\geq th$,

i) 若存在被处理的对象 $d_k$, 满足 $\mathrm{sim}(d_k,d_j)\geq \mathrm{sim}(d_i,d_j)$ 且 $\mathrm{sim}(d_i,d_k)\geq th$, 则将 $d_k,d_j$ 加入聚簇 $C_{xyk_{xy}}$, 并标记 $d_k,d_j$ 为已处理;

ii) 若不存在对象 $d_k$, 则将 $d_j$ 加入聚簇 $C_{xyk_{xy}}$, 并标记 $d_j$ 为已处理;

b) 若第 $j$ 个对象被处理过, 转到步骤 c);

c) $j++$, 转到步骤 a);

3) $k_{xy}++, i++$, 转到步骤 2).

此后, 再结合 $T_x$ 基于自身属性生成的聚类结果 $C_x$, 获取 $T_x$ 中对象基于两种信息的聚簇, 从而体现联系 $n$ 对聚类结果传递的影响. 由维增长子空间算法 CLIQUE (Clustering in QUEst)[18] 可知, 如果一个 $k$ 维单元是稠密的, 该单元在 $k-1$ 维子空间的投影也是稠密的. 因此本文认为若合并后仍在一个聚簇的对象, 无论是基于其自身属性, 还是基于 $T_y$ 中的相关对象, 也均属于同一个聚簇. 若以矩阵 $MC_{xy}$ 标记任意两个 $T_x$ 对象关于 $T_y$ 的所属聚簇是否一致, 其分量 $mc_{ij}$ 如式 (3) 所示; 并以相同方式针对 $T_x$ 对象构建关于自身属性的所属聚簇是否一致的矩阵 $MC_x$, 那么利用 Hadamard 矩阵乘积合并两个结果, 可获得 $T_x$ 中对象关于其自身属性以及 $T_y$ 信息的聚类结果, $MC_{xy}\circ MC_x$ 如式 (4) 所示, 每个分量的含义与前述相同. 其中 $1\leq i,j\leq|\ D_x\ |$, 即若对象 $d_{xi},d_{xj}$ 属于同一聚簇, 对应的 $mc_{ij}$ 为 1; 否则为 0; 若 $d_{xi}$ 或 $d_{xj}$ 不属于任何聚簇, 对应的 $mc_{ij}$ 为 $-1$.

$$mc_{ij}=\begin{cases}1, & d_{xi},d_{xj}\in C_{xyp}\\ 0, & d_{xi}\in C_{xyp}, d_{xj}\in C_{xyq}\\ -1, & d_{xi}\ or\ d_{xj}\notin C_{xy}\end{cases}\tag{3}$$

非标定联系 $n'$, 若 $T_x\xrightarrow{n'}T_y$, 则表 $T_x$ 继承表 $T_y$ 的主键 $pk_y$ 作为一般属性. 一个 $T_y$ 对象可能对应多个 $T_x$ 对象, 然而一个 $T_x$ 目标对象仅对应用于辅助聚类的 $T_y$ 中的一个对象. 那么基于 $T_y$ 的信息对 $T_x$ 中对象聚类, 无需获取 $T_y$ 的统计信息, 直接计算其关于 $T_y$ 的相似性, 并基于传统聚类算法进行即可.

下面分析 $T_x$ 中对象基于 $T_y$ 的相似性度量方法. 设 $T_x$ 中对象集合 $D_x$, 而 $T_y$ 中对象集合 $D_y$, 聚类生成 $k'$ 个聚簇 $C_y=(C_{y1},\cdots,C_{yk'})$.

对于 $\forall d_{x1},d_{x2}\in D_x$, 在继承属性 $pk_y$ 上的值分别为 $pk_{y1}$ 和 $pk_{y2}$, 且分别对应于 $D_y$ 中的 $d_{y1}$ 和 $d_{y2}$; 另外 $d_{y1}\in C_{y1}$, 而 $d_{y2}\in C_{y2}$, 那么 $d_{x1}$ 和 $d_{x2}$ 关于属性 $pk_y$ 的相似性度量如式 (5) 所示. 即若 $d_{x1}$ 和 $d_{x2}$ 对应于同一 $T_y$ 对象, 则两者关于 $T_y$ 的相似性为 1; 若对应属于不同聚簇的 $d_{y1}$ 和 $d_{y2}$, 说明 $d_{x1}$ 和 $d_{x2}$ 关于 $T_y$ 的差异较大. 因此将 $pk_{y1}$ 和 $pk_{y2}$ 间以及 $C_{y1}$ 和 $C_{y2}$ 间相似度中的较小值作为求解值; 另外如果对应属于同一聚簇的 $T_y$ 对象 $d_{y1}$ 和 $d_{y2}$, 则认为 $d_{x1}$ 和 $d_{x2}$ 关于 $T_y$ 虽具有一定差别, 却无需扩大该差异, 因此直接将 $pk_{y1}$ 和 $pk_{y2}$ 间的相似度

$$MC_{xy}\circ MC_{xy}=\begin{bmatrix}1 & m_{12}\cdot m'_{12} & \cdots & m_{1n_x}\cdot m'_{1n_x}\\ m_{21}\cdot m'_{21} & 1 & \cdots & m_{2n_x}\cdot m'_{2n_x}\\ \vdots & \vdots & \ddots & \vdots\\ m_{n_x1}\cdot m'_{n_x1} & m_{n_x2}\cdot m'_{n_x2} & \cdots & 1\end{bmatrix}\tag{4}$$

$$\mathrm{sim}_{pky}(d_{x1},d_{x2})=\begin{cases}1, & pk_{y1}=pk_{y2}\\ \min(\mathrm{sim}(C_{y1},C_{y2}),\mathrm{sim}(pk_{y1},pk_{y2})), & pk_{y1}\neq pk_{y2}, C_{y1}\neq C_{y2}\\ \mathrm{sim}(pk_{y1},pk_{y2}), & pk_{y1}\neq pk_{y2}, C_{y1}=C_{y2}\end{cases}\tag{5}$$

作为 $d_{x1}$ 和 $d_{x2}$ 关于 $T_y$ 的相似度. 再结合其自身属性信息, 基于传统聚类算法即可对 $T_x$ 中的对象进行聚类, 同时体现了非标定联系 $n'$ 对 $T_y$ 的聚类结果传递的影响.

联系 $i$, 若 $T_x \xrightarrow{i} T_y$, 则表 $T_y$ 继承表 $T_x$ 的主键 $pk_x$ 作为主键, 且 $T_x$ 与 $T_y$ 中的对象可能存在一对多的情况, 即一个 $T_x$ 目标对象可能对应多个 $T_y$ 对象. 因此与联系 $n$ 类似, 基于 $T_y$ 包含的信息对 $T_x$ 中的对象进行聚类时, 分析方法与联系 $n$ 相同.

联系 $i'$, 若 $T_x \xrightarrow{i'} T_y$, 则表 $T_x$ 继承表 $T_y$ 的主键 $pk_y$ 作为主键. 与联系 $n'$ 类似, 一个 $T_x$ 目标对象仅对应于一个 $T_y$ 对象. 因此基于 $T_y$ 的信息对 $T_x$ 中对象聚类时, 分析方法与 $n'$ 相同.

联系 $s$, 若 $T_x \xrightarrow{s} T_y$, 则表 $T_y$ 继承表 $T_x$ 的主键 $pk_y$ 作为主键, 且 $T_x$ 与 $T_y$ 中的对象可能存在一对多的情况. 同理, 其分析方法与联系 $n, i$ 相同.

联系 $s'$, 若 $T_x \xrightarrow{s'} T_y$, 则表 $T_x$ 继承表 $T_y$ 的主键 $pk_y$ 作为主键. 同理, 其分析方法与 $n'$ 相同.

### 3.2 整合子节点聚类结果

下面为叙述简明, 将直接用节点表示多关系数据集中的表, 即子节点表示作为子节点的表, 父节点表示作为父节点的表.

尽管确定了一个子节点的聚类结果沿不同联系向父节点传递聚类结果的方法, 然而关联层次模型中每个非叶节点均可能存在多个子节点, 且每个子节点的聚类结果均会影响其父节点的聚类, 同时子节点间也可能存在联系, 此时即需整合多个子节点的聚类结果. 子节点间相互关联的形式主要分为子节点间存在联系、子节点间相互独立以及前两种的混合等几种情况, 下面分别针对这三种情况具体分析整合子节点聚类结果的方法.

1) 子节点间具有联系

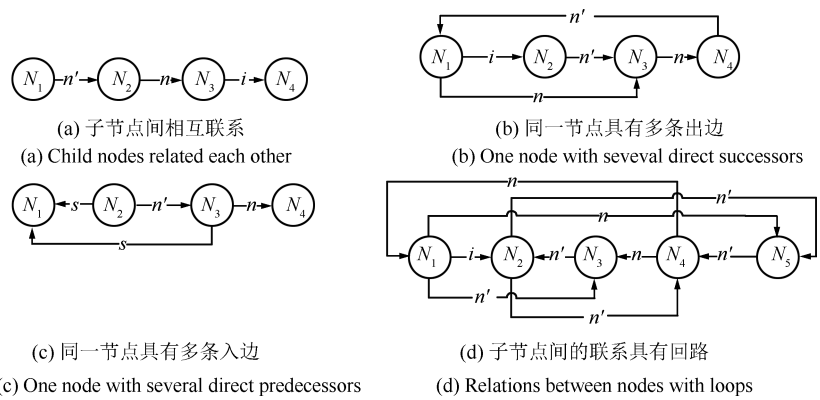
子节点间具有联系, 即任意两个子节点间均具有至少一条关联路径, 可能存在的情况如图 7 所示. 7 (a) 为子节点间的联系形成了一条路径, 此时为了体现联系对各节点聚类结果传递的影响, 可将路径上的其他节点看作该路径的始点的描述信息, 仅保留由父节点至路径始点间的联系即可. 那么无论是子节点间还是作为路径的始点向父节点传递聚类结果, 均可利用第 3.1 节中的方法, 根据联系的不同种类而进行. 若任意两个直接关联的节点间还存在通过其它节点关联的路径, 如 7 (b) 中的节点 $N_1$ 与 $N_3$, 此时为了避免反复利用 $N_3$ 的聚类结果, 可直接通过路径 $(N_1, N_2, N_3)$ 传递聚类结果; 同样的, 若父节点与子节点间存在此种情况也可如此处理. 如 7 (d) 所示更为复杂需要路径选择的情况的处理方法将在下一节详细讨论.

2) 子节点间相互独立

子节点间相互独立, 即它们之间不存在关联边, 可按照如下方法向父节点传递结果: 对于 $n'$、$i'$、$s'$ 类型的联系, 由式 (5) 可知, 能够直接计算任意两个父节点对象基于每个作为辅助信息的子节点的相似度, 那么按照第 3.1 节中的对应方法可分别向父节点传递聚类结果. 而对于 $n$、$i$、$s$ 类型的联系, 由第 3.1 节中相应方法可知, 需首先通过矩阵的 Hadamard 乘积整合多个子节点间的聚类结果, 然后基于相应联系的方法向其父节点传递该整合结果.

3) 上述两种情况的混合

采用的方法也需结合前两种处理方法. 对于关联的表可按照 1) 中方法加以处理; 相互独立的表若具有同一起始节点, 则整合其聚类结果, 否则无需整合.




(a) 子节点间相互联系
(a) Child nodes related each other

(b) 同一节点具有多条出边
(b) One node with seveval direct successors

(c) 同一节点具有多条入边
(c) One node with several direct predecessors

(d) 子节点间的联系具有回路
(d) Relations between nodes with loops

图 7 子节点间存在联系的几种情况

Fig. 7 Situations of different relationships between child nodes

### 3.3　子节点关联路径选择

如前所述关联关系层次模型可初步拆分原关联图中的回路, 图 5 构建的关联层次模型已不存在有向回路; 然而该模型允许同层节点间存在联系, 使得同层的表之间仍可能存在回路, 如图 7 (d) 中节点 $N_1$ 与 $N_4$ 之间; 此时需从中确定传递聚类结果且尽量不重复利用信息的关联路径.

虽然最小生成树算法可用于选择路径, 但其得到的是一个具有保持图连通的最少边的极小连通图, 可能会忽略某些体现节点间关联关系的边. 另外由图论可知, 最短路径的子路径也为最短路径, 即如果两点间的最短路径上仍包含其他点, 那么该路径上的任意两点间的最短路径为该路径的子路径. 因此本文以图论中求顶点间最短路径的方法以及最小生成树方法为基础, 解决用于传递聚类结果的路径选择问题.

首先几种联系的权值与其各自的优先级相关, 即具有高优先级的联系的权值更低, 那么基于顶点间最短路径确定传递路径时, 可优先被选择. 由于联系 $i$ 表示了一种附属关系, 具有较高的优先级, 而第 3.1 节中指出了联系 $i'$、$n'$ 和 $s'$ 传递聚类结果的方式相同, 因此本文定义几种联系的优先级如下: $i > n > s > i' = n' = s'$.

然后由最小生成树的 Prim 算法每步添加到树中的边都是使树的权尽可能小的边可知, 如果两点间存在回路, 则保留权值较小的路径, 并可忽略与其构成回路的路径. 接下来以矩阵 $L$ 构造每对节点间的最短路径, 如果两点间存在最短路径, 其对应的分量 $l_{ij} \neq \infty$. 那么基于矩阵所体现的每对顶点间的最短路径, 根据算法 1 可进行传递聚类结果路径的选择.

**算法 1. ChooseTransferPath($L$)**

**输入.** 子节点间最短路径权值矩阵 $L_{n\times n}$

**输出.** 保留的路径集合 $remainP$

**步骤 1.** 设 $candidate := \{\}$;

**步骤 2.** 对于任意第 $i, j(i \neq j)$ 个节点, 若 $N_i, N_j$ 间的路径 $p_{ij} \notin remainP$.

1) 若两点间无回路, 则 $remainP += p_{ij}$;

2) 若两点间存在回路, 选择较小权值 $l_{ij}, l_{ji}$ 对应的路径加入 $remainP$, 并将其逆向路径加入 $candidate$;

**步骤 3.** 对于 $candidate$ 中的每条路径 $canpath$, 若 $canpath$ 中存在未处理节点, 则将 $canpath$ 的逆向路径加入 $remainP$;

**步骤 4.** 若 $remainP$ 中仍存在回路, 对于任意 $(N_s, N_t), (N_t, N_s)$, 删除包含较少节点的路径, 并将其逆向路径加入 $remainP$;

下面以图 7 (d) 为例, 解释利用算法 1 选择传递聚类结果路径的过程. 首先根据边的优先级确定其权值, 如图 8 (a) 所示为其加权图, 同时利用自底向上求节点间最短路径的方法获得如图 8 (b) 所示的权值矩阵. 具体选择的过程如图 9 所示, 第一次循环

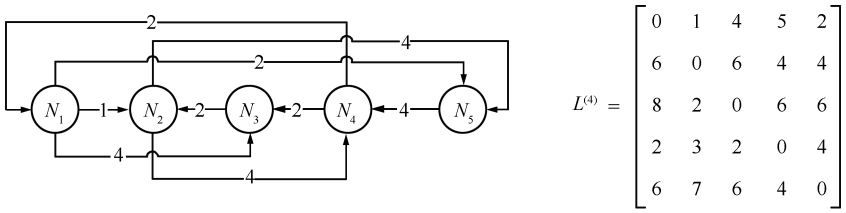



图 8　图 7 (d) 对应的加权图及顶点间最短路径的权值矩阵

Fig. 8　Weighted graph and weight matrix about shorted path between nodes corresponding to Fig. 7 (d)

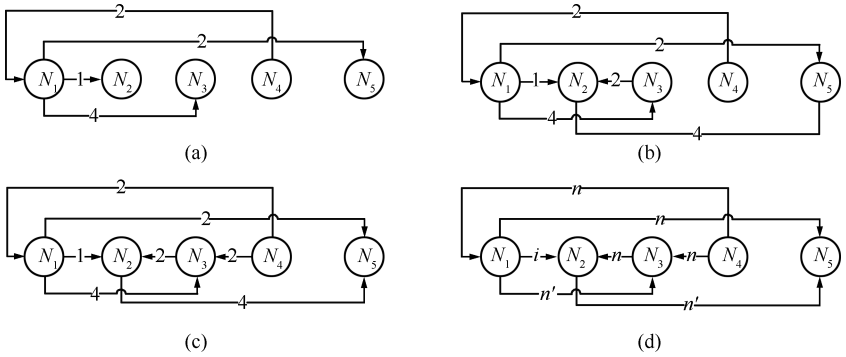



图 9　路径选择过程

Fig. 9　Processes of route selection

后, 通过第一行与第一列对应值的比较, 确定保留的 $N_1$ 与其他顶点间的路径如 (a) 所示; 第二次与第三次循环后的结果分别如 (b) ∼ (c) 所示; (d) 为最终保留的路径, 可见 $N_1$ 与 $N_4$ 之间已不存在有向回路. 因为根据算法 1, 若两节点间存在回路, 则保留权值较小的路径; 另外若待保留的路径中存在未处理的节点, 则保留其逆向路径 (如 $n$ 转换为 $n'$). 因此由算法 1 能够得到一个尽量保留节点间联系的无回路的关联图.

### 3.4 子节点聚类结果传递

接下来需基于路径选择后的关联图, 以及子节点聚类结果的整合方法, 确定根据关联路径向父节点传递子节点聚类结果的方法. 考虑到前述子节点间联系包含的 3 种情况, 算法 2 将分别对其进行处理.

**算法 2. TransferCluster**$(C_{child}, L_{n_c\times n_c})$

**输入.** 子节点聚类结果 $C_{child}=\{C_{c1},\cdots,C_{cn_c}\}$;
子节点间最短路径权值矩阵 $L_{n_c\times n_c}$;

**输出.** 父节点聚类结果 $C_f$;

**步骤 1.** 若子节点间相互独立, 根据 3.1 节中的方法, 首先处理联系 $n'$、$i'$、$s'$, 然后处理联系 $n$、$i$、$s$ 向父节点传递子节点的聚类结果, 得到 $C_f$;

**步骤 2.** 若子节点间形成一条关联路径, 则由该路径的终点依次沿路径传递聚类结果, 并将该路径起点的聚类结果根据其与父节点间的联系类型传递聚类结果, 得 $C_f$;

**步骤 3.** 若子节点存在回路或同时存在相互关联及独立的子节点, 由 chooseTransferPath$(L)$ 获取子节点间构成的关联图, 并将入度为 0 的节点集合记为 $sSet$;

1) 对于每个 $s_i \in sSet$, 接受其关联路径上各节点的聚类结果传递:

$$Cs_i = \text{TransferCluster}(s^i_{child}, L^i_s);$$

2) $sSet$ 中各节点相互独立, 将其聚类结果传递给共同的父节点: TransferCluster$(Cs, L_s)$, 得 $C_f$.

该算法考虑了 3.2 节中子节点间关联的几种情况. 特别的对于较复杂的情况, 首先由算法 1 获取相关联节点构成的关联图; 由于该图中入度为 0 的节点是相互独立的, 那么通过递归调用完成其各自聚类结果的整合; 再将这几个独立节点的聚类结果, 按照子节点与父节点间联系的类型, 根据 3.1 节提出的方法向其共有的父节点传递聚类结果. 算法中子节点间相互独立时, $n'$、$i'$、$s'$ 类型关系传递结果的时间复杂度为 O(1); $n$、$i$、$s$ 类型关系基于矩阵的 Hadamard 乘积传递聚类结果的时间复杂度为 $\mathrm{O}(n^2)$, 其中 $n$ 为目标对象的个数; 而子节点相互关联时, 结果传递的时间复杂度最多为 $\mathrm{O}(n_c \cdot n^2)$, 其中 $n_c$ 为子节点的个数. 因此其时间复杂度为 $\mathrm{O}(n^2)$.

### 3.5 多关系聚类算法

综合前述方法, 同时结合算法 3 不仅可完成基于 IDEF1x 模型的层次多关系聚类任务, 还可避免上述多关系聚类中存在的问题. 即基于 IDEF1x 模型构建的多关系数据集的关联层次模型 (IDEF1x multi-relational clustering, IDEFMC), 从位于模型底层的表开始, 逐层完成每张表中对象的聚类. 对于作为叶节点的关系表, 根据构成该表的属性完成聚类; 而对于非叶节点的表而言, 由于与其子节点构成了一个多关系数据子集, 因此可通过 TransferCluster() 方法将子节点的聚类结果传递给当前处理的节点. 最终位于根节点的表的聚类结果即为多关系聚类的结果.

**算法 3. IDEFMC(CGKH)**

**输入.** 多关系数据集构成的关联层次模型 CGKH

**输出.** 目标对象聚类结果 $C=\{C_1,\cdots,C_k\}$

**步骤 1.** 设 CGKH 共 $l$ 个层次, 并令 $i=l$;

**步骤 2.** 对于每个叶子节点, 即 $i=l$, 基于其自身属性聚类;

**步骤 3.** 令 $i--$;

**步骤 4.** 对于第 $i$ 个层次中的每个节点 $t^i_j$, 令其子节点聚类结果集合为 $C_{c^i_j}$, 子节点间最短路径权值矩阵 $L_{c^i_j}$;

1) 获取其聚类结果:

$$C^i_j = \text{TransferCluster}(C_{c^i_j}, L_{c^i_j});$$

2) 若 $i\neq 1$, 则 $i--$, 转入步骤 4; 否则返回 $C^1_j$.

由算法 2 可知, 每层节点向父节点传递结果的复杂度为 $\mathrm{O}(n^2)$, 其中 $n$ 为目标对象的个数. 若基于 IDEF1x 模型构建的表间关联层次模型 CGKH 共有 $l$ 层, 那么该算法整合聚类结果的时间复杂度为 $\mathrm{O}(l\cdot n^2)$.

## 4 实验

本文在真实以及人工数据集上进行实验, 通过与单表上的聚类算法 $K$-means, 以及多关系聚类算法 CrossClus[10]、DIVA[8]、NMTF[14] 和 ONRC[15] 的对比, 说明本文所提算法能够有效地获得较准确合理的聚类结果.

对于预先标记类别并可用于验证聚类效果的数据集, 本文采用两种常用的度量方法评价聚类结果: 纯度 $Purity$[19] 和信息熵 $Entropy$[19]. 设聚类结果为 $C = C_1, C_2, \cdots, C_k$, 而已知的分类 $P = P_1, P_2, \cdots, P_l$, 其中 $k$ 为生成的聚簇数目, $l$ 为原始分类数.

1) 纯度: $Purity = \sum_{i=1}^{k} \frac{1}{N} \max_{j}(n_i^j)$;

2) 信息熵: $Entropy = \sum_{i=1}^{k} \frac{n_i}{N} (-\frac{1}{\log l} \sum_{j=1}^{i} \frac{n_i^j}{n_i} \log \frac{n_i^j}{n_i})$;

其中 $N$ 为数据集中的对象个数, $n_i^j$ 为原属于第 $j$ 类的数据在求得的第 $i$ 个聚簇中的个数, $n_i$ 为第 $i$ 个聚簇中的对象个数. 纯度 $Purity$ 度量了生成聚簇的纯净度, 值越大说明每个聚簇由属于同一原始类别的对象构成的程度越高. 而信息熵 $Entropy$ 基于熵的定义, 度量了生成聚簇的混杂程度, 值越大说明每个聚簇由不同原始类别中对象组成的程度越高. 因此纯度越高、信息熵越低表明聚类效果越好; 且理想情况下, $Purity = 1.0$, 而信息熵 $Entropy = 0.0$.

### 4.1　真实数据集

本文选取的真实数据集为基本养老保险领域的数据集, 其对应的 IDEF1x 模型与图 4 类似, 仅不考虑其中的人员转移表, 数据集中包含的记录条数以及目标对象的原始分类数如表 2 所示.

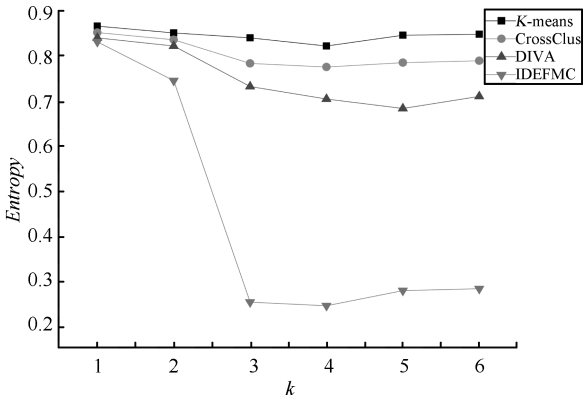
表 2　养老保险数据集中各关系表信息

Table 2　Information in every table of basic endowment insurance dataset

| 单位 | 参保情况 | 职工 | 缴费 | 退休人员 | 支付 | 原始分类数 |
|---|---|---|---|---|---|---|
| 1 300 | 1 300 | 23 818 | 32 560 | 2 203 | 3 738 | 2 |

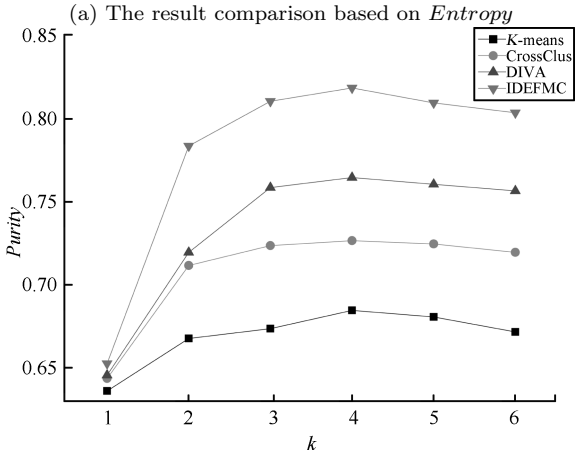
实验中的聚类过程采用 $K$-means 的思想, 以单位为目标对象进行多关系聚类. 目标为根据单位的特点及其参保情况、职工信息及其缴费信息, 以及退休人员及其支付信息, 将单位分为事业单位与企业单位两个类别, 以便于分析两种类型的单位在基本养老保险业务中各具有何种特点.

如图 10 所示为基于不同的输入参数, 本文所提算法 IDEFMC 与对比算法 $K$-means、CrossClus 和 DIVA 的聚类结果对比. 图 10 (a) 为面向不同的 $k$ 值输入时, 基于信息熵 $Entropy$ 的聚类结果对比. 可见除了 $k$ 为 1 时, IDEFMC 得到的信息熵值均明显小于其他算法. 这是因为实际的单位类型具有两种, 若 $k$ 值为 1, 显然聚类结果中所有类型的单位均同属一个聚簇, 信息熵值必然为最大, 因为信息熵体现了聚簇的混杂程度; 而此时由于有一种类型的对象被分到同一聚簇, 因此对应的纯度值不会过低. 而当 $k$ 为 3 或 4, 即接近于实际分类数时, 信息熵值小于 $k$ 取其它输入时的值; 这是因为当 $k$ 值与实际类别个数相差较大时, 如 $k$ 取 5 或 6, 较多聚簇中心的选择使得每个聚簇中心可能吸收不同类型的对象, 进而影响聚类效果, 使得信息熵值增加. 另外 $k$ 为 2 时, 可能由于其中一个聚簇同时吸收了属性差异不够明显的两种类别的对象, 而获得了较大的信息熵值. 图 10 (b) 则为同等条件下基于纯度值 $Purity$ 的聚类结果对比. 相对于 $Entropy$, 纯度值越大说明聚类效果越好. 可见同样的, 除了 $k$ 为 1 时, IDEFMC 相较于几个对比算法, 均具有较高的纯度值. 而且当 $k$ 接近于实际分类数时, 纯度值也相对较高, 达到了 0.8 以上.




(a)　基于信息熵的结果对比

(a) The result comparison based on $Entropy$




(b)　基于纯度的结果对比

(b) The result comparison based on $Purity$

图 10　真实数据集上基于不同输入的多关系聚类结果对比

Fig. 10　Comparisons of clustering results of real dataset based on different parameters

能够产生上述结果, 是由于 $K$-means 算法仅针对单位信息表进行聚类, 并未利用其他关系表作为辅助信息; 而 CrossClus 和 DIVA 则在表间存在一对多联系时, 利用传统的统计方法获取对应信息, 相较于 IDEFMC, 并未很好地保留了数据的原始特征; 另外如 CrossClus 由于算法特点, 并非选取了与业务相关的多关系特征参与聚类, 因此聚类效果一般; 而 IDEFMC 基于层次关联模型的聚类方法, 不仅保留了数据的原始特征, 所提的传递聚类结果的路径选择算法, 也尽量避免了信息的重复利用, 因而能够获得较好的聚类效果.

而算法 NMTF、ONRC 的主要思想为将同一

类型以及不同类型对象间的联系均抽象为矩阵形式, 并基于矩阵分解的方法进行多关系聚类, 可同时获得不同表中对象的聚类结果. 因此本文将 IDEFMC 在 $k$ 的输入分别为 3 和 4 时的结果与这两种算法在该数据集上的聚类结果进行比较, 对比结果如图 11 所示.

可见在该数据集上, IDEFMC 所获得的信息熵值均低于两种对比算法, 同时纯度值也相对高于对比算法. 显然当输入 $k$ 值接近于目标对象的实际分类数目时, 相较于这两种对比算法, IDEFMC 也能够获得较准确的聚类结果. 虽然对比算法能够通过矩阵分解同时获得多种类型对象的聚类结果, 但主要基于对象间联系进行聚类的特点, 使其在聚类时忽略了数据的属性信息. 另外 ONRC 仅适用于星型结构, 而图 4 中的养老保险数据集结构则较为复杂, 因此 ONRC 的结果具有相对较高的信息熵以及较低的纯度值. 而 IDEFMC 虽然采用不同的方式组织聚类过程, 但其基于 IDEF1x 模型构建多关系数据集的模型, 不仅可表示同种类型以及不同类型对象间的联系. 聚类算法也同时充分考虑了数据对象的属性以及不同表间的联系对聚类的影响, 因此 IDEFMC 能够获得较好的聚类效果.

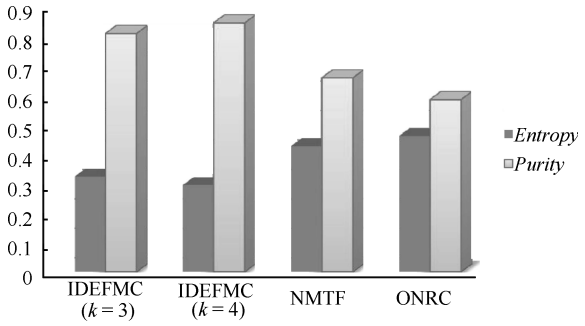



图 11 真实数据集上的多关系聚类结果对比

Fig. 11 Comparisons of clustering results of real dataset

## 4.2 人工数据集

为了进一步验证所提算法的有效性, 本文还选取两个人工数据集作为实验数据集.

1) Movie 数据集[1]

首先选取人工数据集为 UCI 的多关系数据集 Movie, 主要包括 5 个关系表, 描述了电影、导演、演员间的关系. 该数据集的 IDEF1x 模型以及基于 IDEF1x 模型构建的关联关系层次模型如图 12 所示.

基于 Movie、Actor、Casts 以及 Studio 等表的信息作为辅助信息, 以电影导演为目标对象, 进行多关系聚类. 数据集包含的对象信息以及目标对象的原始分类数如表 3 所示.

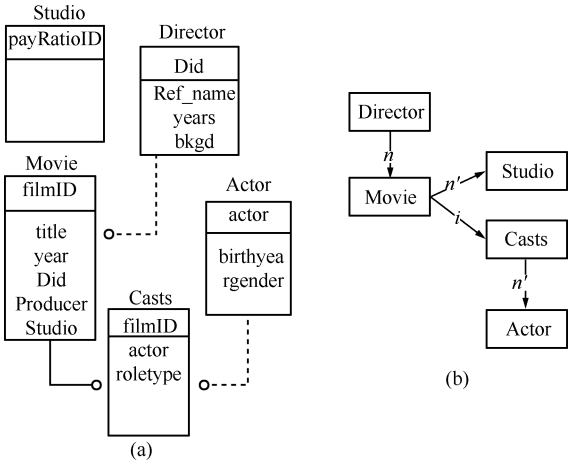
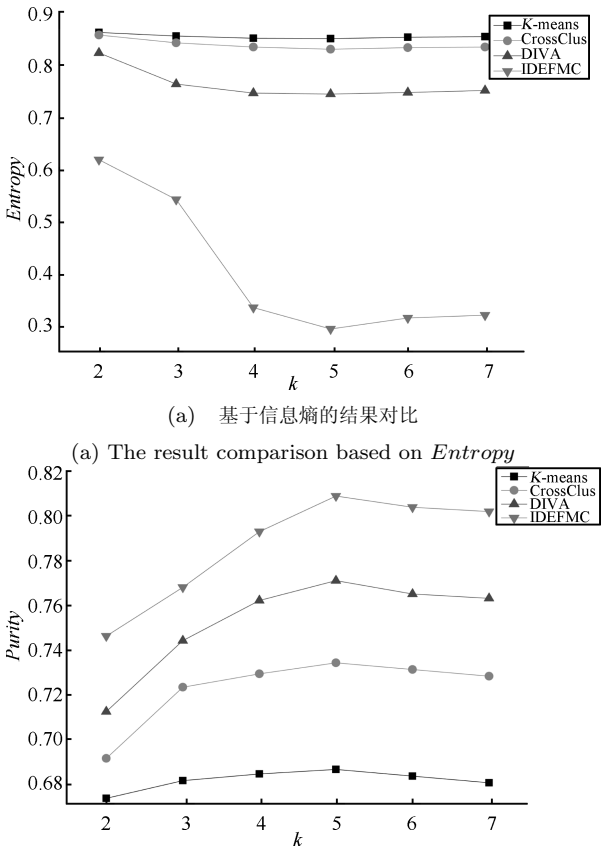



图 12 人工数据集 Movie 对应的 IDEF1x 模型与关联层次模型

Fig. 12 IDEF1x model and CGKH model of synthetic dataset movie




(a) 基于信息熵的结果对比

(a) The result comparison based on *Entropy*

(b) 基于纯度的结果对比

(b) The result comparison based on *Purity*

图 13 数据集 Movie 上基于不同输入的多关系聚类结果对比

Fig. 13 Comparisons of clustering results of dataset Movie based on different parameters

[1] http://achive.ics.uci.edu/ml/datasets.html

图 13 所示为 IDEFMC、$K$-means、CrossClus 和 DIVA 几个算法以电影导演为目标对象的聚类结果比较. 同样的, 13 (a) 为面向不同的输入 $k$ 值时, 基于信息熵 *Entropy* 的聚类结果对比, 而 13 (b) 则为同等条件下基于纯度值 *Purity* 的聚类结果对比. 可见相较于单表聚类算法 $K$-means, 以及 CrossClus、DIVA 等多关系聚类算法, 应用 IDEFMC 能够均取得了相对较小的信息熵值, 以及较高的纯度值. 特别的, 当 $k$ 在接近目标对象原始分类数时, 由相应的信息熵以及纯度值可知, 所得聚类效果更为准确. 因为 IDEFMC 基于 IDEF1x 模型构建多关系数据集的模型进而提出多关系聚类算法, 充分利用了数据的原始特征以及各关系表间的联系对聚类的影响.

表 3　Movie 数据集包含的对象信息

Table 3　Object information of Movie dataset

| Movie | Actor | Director | Casts | Studio | Partitions |
|---|---|---|---|---|---|
| 11 284 | 6 773 | 3 112 | 45 884 | 194 | 4 |

而图 14 则描述了 IDEFMC 在 $k$ 的输入值为 4 和 5 时与 NMTF、ONRC 在该数据集上的聚类结果对比. 可以看出相较于这两种算法, IDEFMC 也依然能获得相对较小的信息熵值和较高的纯度值. 说明在人工数据集上 IDEFMC 仍能够取得较准确的聚类结果.

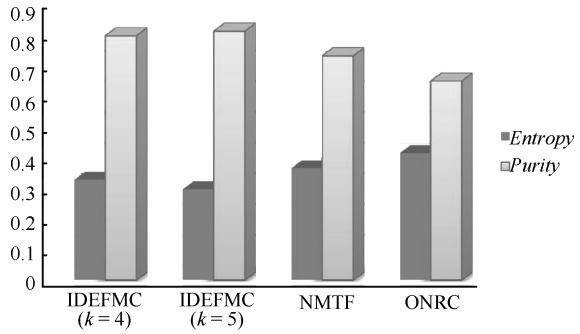



图 14　数据集 Movie 上的多关系聚类结果对比

Fig. 14　Comparisons of clustering results of dataset Movie

2) DBLP 数据集[2]

DBLP 数据集由 DBLP 的 XML 数据中提取, 包含计算机领域文献的相关信息, 如论文、作者、发表日期等. 本文以其他表中的信息作为辅助信息, 并以作者为目标对象, 按照其研究领域进行多关系聚类. 数据集包含的对象信息以及目标对象的原始分类数如表 4 所示, 共包含 4 张关系表, 其中 Author-Paper 表示作者与论文间的对应关系.

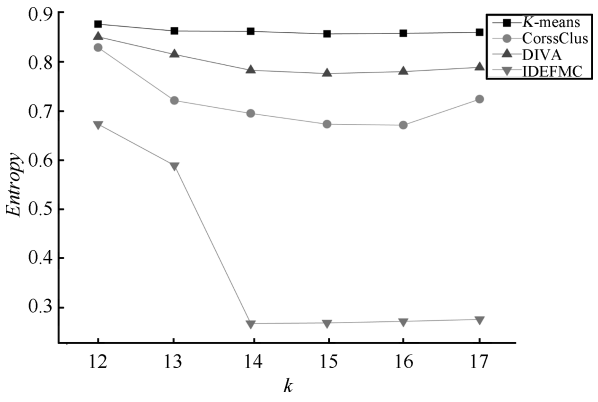
[2]http://dblp.uni-trier.de/xml

表 4　DBLP 数据集包含的对象信息

Table 4　Object information of DBLP dataset

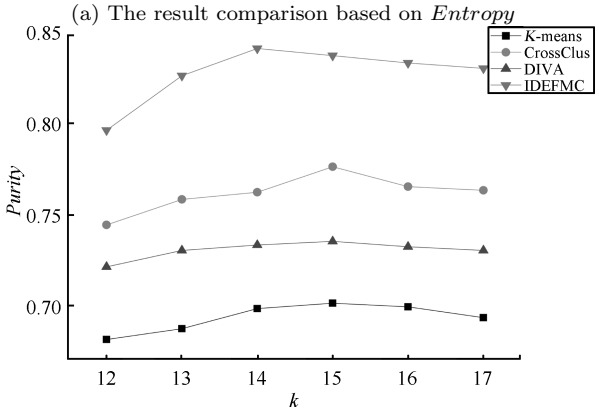
| Author | Paper | Author-Paper | Book | Partitions |
|---|---|---|---|---|
| 15 120 | 15 102 | 21 791 | 307 | 14 |

如图 15 所示为 IDEFMC 与对比算法 K-means、CrossClus 和 DIVA 基于不同输入聚类结果的比较, 可见相较于对比算法, IDEFMC 仍取得了相对较低的信息熵值, 以及较高的纯度值. 并且当 $k$ 为 14, 即与目标对象原始分类数一致时, 聚类效果相对较好; 同样的, 当 $k$ 值与实际类别数不一致时, 或由于 $k$ 小于实际分类数而导致不同类别的对象被划分至一个聚簇, 或如前所述较大的 $k$ 值导致每个聚簇中心可能吸收不同类型的对象, 影响了聚类效果, 进而使得信息熵值相对较大, 同时纯度值较小.




(a)　基于信息熵的结果对比

(a) The result comparison based on *Entropy*




(b)　基于纯度的结果对比

(b) The result comparison based on *Purity*

图 15　DBLP 数据集上基于不同输入的多关系聚类结果对比

Fig. 15　Comparisons of clustering results of dataset DBLP based on different parameters

另外 IDEFMC 与算法 NMTF、ONRC 在 DBLP 数据集上的聚类结果对比如图 16 所示.

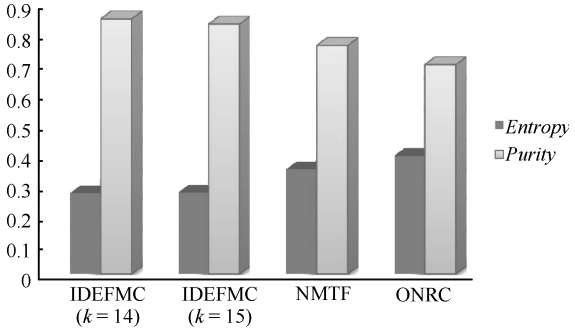



图 16 DBLP 数据集上的多关系聚类结果对比

Fig. 16 Comparisons of clustering results of dataset DBLP

可见同样的, 当 IDEFMC 的输入 $k$ 值接近于目标对象的原始分类数时, 依然能够获得相较于两种对比算法而言更小的信息熵值, 以及更大的纯度值; 而因为该数据集接近于星型结构, 所以相较于前两个数据集, ONRC 获得了相对较好的聚类效果. 可见 IDEFMC 基于 IDEF1x 模型对实体和实体间联系丰富的表达能力, 以及兼顾数据原始特征以及表间联系对聚类影响的特点, 使其在该实验数据集上依旧能获得较准确的聚类结果, 进一步验证了本文所提算法的有效性.

## 5 结论

由于 IDEF1x 模型中的联系类型是引起多关系聚类尚存问题的主要原因, 因此本文提出一种基于 IDEF1x 模型的层次多关系聚类算法, 该算法可充分利用每张表体现的原始信息. 同时由于同一层次的两表之间可能存在多条关联路径, 另外表间联系可能构成有向回路, 还提出了传递聚类结果的路径选择算法, 该算法能够获得一个无回路的且尽量保留表间联系的子节点间关联图. 再结合聚类结果整合方法以及不同联系对结果传递的影响, 即构成了层次多关系聚类算法的基础. 在真实以及人工数据集上的实验表明了所提算法的有效性与适用性. 然而由于业务执行等原因多关系数据集中仍然存在目标对象可能由不同阶数信息描述的问题, 进一步研究将围绕该问题而展开.

## References


1 Shen Xiao-Wei, Wang Fei-Yue, Cheng Chang-Jian, Liu Xi-Wei. Application of clustering analysisto team management. *Acta Automatica Sinica*, 2012, **38**(4): 563−569
(沈小伟, 王飞跃, 程长建, 刘希未. 聚类分析方法在企业班组管理中的应用. 自动化学报, 2012, **38**(4): 563−569)

2 Zhou Lin, Ping Xi-Jian, Xu Sen, Zhang Tao. Cluster ensemble based on spectral clustering. *Acta Automatica Sinica*, 2012, **38**(8): 1335−1342
(周林, 平西建, 徐森, 张涛. 基于谱聚类的聚类集成算法. 自动化学报, 2012, **38**(8): 1335−1342)

3 Peng Yu, Luo Qing-Hua, Wang Dan, Peng Xi-Yuan. WSN location method using interval data clustering. *Acta Automatica Sinica*, 2012, **38**(7): 1190−1199
(彭宇, 罗清华, 王丹, 彭喜元. 基于区间数聚类的无线传感器网络定位方法. 自动化学报, 2012, **38**(7): 1190−1199)

4 Taskar B, Segal E, Koller D. Probabilistic classification and clustering in relational data. In: Proceedings of the 17th International Joint Conference on Artificial Intelligence. San Francisco, USA: Morgan Kaufmann Publishers, 2001. 870−878

5 Wang J D, Zeng H J, Chen Z, Lu H J, Li T, Ma W Y. ReCoM: reinforcement clustering of multi-type interrelated data objects. In: Proceeding of the 26th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. Toronto, Canada: ACM, 2003. 274−281

6 Long B, Zhang Z F, Yu P S. A probabilistic framework for relational clustering. In: Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. New York, USA: ACM, 2007. 470−479

7 Sun Y Z, Han J W, Zhao P X, Yin Z J, Cheng H, Wu T Y. Rankclus: integrating clustering with ranking for heterogeneous information network analysis. In: Proceedings of the 12th International Conference on Extending Database Technology: Advances in Database Technology. New York: ACM, 2009. 565−576

8 Li T, Anand S S. DIVA: a variance-based clustering approach for multi-type relational data. In: Proceedings of the 16th ACM Conference on Conference on Information and Knowledge Management. Lisboa, Portugal: ACM, 2007. 147−156

9 Long B, Zhang Z F, Wu X Y, Yu P S. Spectral clustering for multi-type relational data. In: Proceedings of the 23rd International Conference on Marchine Learning. Pittsburgh, USA: ACM, 2006. 585−592

10 Yin X X, Han J W, Yu P S. CrossClus: user-guided multi-relational clustering. *Data Mining and Knowledge Discovery*, 2007, **15**(3): 321−348

11 Gao Ying, Liu Da-You, Qi Hong, Liu He. Semi-supervised $K$-means clustering algorithm for multi-type relational data. *Journal of Software*, 2008, **19**(11): 2814−2821
(高滢, 刘大有, 齐红, 刘赫. 一种半监督 K 均值多关系数据聚类算法. 软件学报, 2008, **19**(11): 2814−2821)

12 Lin Y R, Sun J M, Cao N, Liu S X. Contextour: contextual contour visual analysis on dynamic multi-relational clustering. In: Proceedings of the SIAM Conference on Data Mining. Columbus, USA: ASA, 2010. 418−429

13 Wang H, Huang H, Ding C. Simultaneous clustering of multi-type relational data via symmetric nonnegative matrix tri-factorization. In: Proceedings of the 20th ACM International Conference on Information and Knowledge Management. Glasgow, UK: ACM, 2011. 279−284

14 Wang H, Nie F P, Huang H, Ding C. Nonnegative matrix tri-factorization based high-order co-clustering and its fast implementation. In: Proceedings of the 11th IEEE International Conference on Data Mining. Vancouver, Canada: IEEE, 2011. 774−783

15 Liu Y, Shen C. Orthogonal nonnegative matrix factorization for multi-type relational clustering. *International Journal of Computer and Information Technology*, 2013, **2**(2): 215−221

16 Kusiak A, Letsche T, Zakarian A. Data modelling with IDEF1x. *International Journal of Computer Integrated Manufacturing*, 1997, **10**(6): 470−486

17 Ma Z M, Zhang W J, Ma W Y. Extending IDEF1X to model fuzzy data. *Journal of Intelligent Manufacturing*, 2002, **13**(4): 295−307

18 Agrawal R, Gehrke J, Gunopulos D, Raghavan P. Automatic subspace clustering of high dimensional data for data mining applications. In: Proceeding of the 1998 ACM SIGMOD International Conference on Management of Data. New York, USA: ACM, 1998. 94−105

19 Zhao Y, Karypis G. Criterion Functions for Document Clustering: Experiment and Analysis, Technical Report TR 01-40, Department of Computer Science, University of Minnesota, USA, 2001



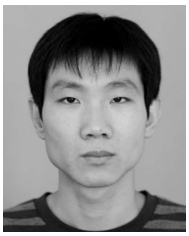
**黄少滨** 哈尔滨工程大学计算机科学与技术学院教授. 主要研究方向为分布式计算与仿真, 模型检测和数据集成.
E-mail: huangshaobin@hrbeu.edu.cn
(**HUANG Shao-Bin** Professor at the College of Computer Science and Technology, Harbin Engineering University. His research interest covers distributed computing and simulation, model checking, and data integration.)



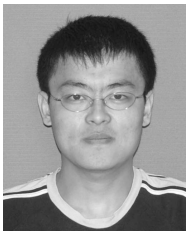
**程 媛** 哈尔滨工程大学计算机科学与技术学院博士研究生. 主要研究方向为数据挖掘, 机器学习. 本文通信作者.
E-mail: changuang7@sina.com
(**CHENG Yuan** Ph. D. candidate at the College of Computer Science and Technology, Harbin Engineering University. Her research interest covers data mining, and machine learning. Corresponding author of this paper.)



**万庆生** 哈尔滨工程大学计算机科学与技术学院博士研究生. 主要研究方向为专家系统, 问答系统.
E-mail: wanqingsheng@hrbeu.edu.cn
(**WAN Qing-Sheng** Ph. D. candidate at the College of Computer Science and Technology, Harbin Engineering University. His research interest covers expert system and answering system.)



**刘国峰** 哈尔滨工程大学计算机科学与技术学院博士研究生. 主要研究方向为数据集成, 模式匹配, 数据挖掘.
E-mail: liuguofeng22@163.com
(**LIU Guo-Feng** Ph. D. candidate at the College of Computer Science and Technology, Harbin Engineering University. His research interest covers data integration, schema matching, and data mining. )



**申林山** 哈尔滨工程大学计算机科学与技术学院讲师. 主要研究方向为分布式计算, 自主计算.
E-mail: shenlinshan@hrbeu.edu.cn
(**SHEN Lin-Shan** Lecturer at the College of Computer Science and Technology, Harbin Engineering University. His research interest covers distributed computing and autonomic computing.)