计算机科学
COMPUTER SCIENCE

基于超图的知识提取算法

刘川, 杜宝苍, 毛华

引用本文

刘川, 杜宝苍, 毛华. 基于超图的知识提取算法[J]. 计算机科学, 2025, 52(4): 147-160.

LIU Chuan, DU Baocang, MAO Hua. Knowledge Extraction Algorithm Based on Hypergraphs[J]. Computer Science, 2025, 52(4): 147-160.

计算机科学 COMPUTER SCIENCE
http://www.jsjkx.com
DOI:10.11896/jsjkx.240100084

基于超图的知识提取算法

刘 川[1] 杜宝苍[2] 毛 华[1]
1 河北大学数学与信息科学学院 河北 保定 071002
2 河北金融学院管理学院 河北 保定 071051
(18811527086@163.com)

摘 要 知识提取一直是计算机领域研究的主题之一,然而现有的一些知识提取方法还不能满足可视化以及潜在知识的提取两方面的实际需求。众所周知,知识是由可定义知识和潜在知识组成,并且可定义知识可以在潜在知识的提取过程中同时得到,反之则不然。有关可定义知识的提取目前已有许多成果,但针对潜在知识的提取的研究相对较少,特别是如何通过可视化方法提取潜在知识是一个急需解决的问题。为此,文中利用超图的可视化特点,在信息系统的背景下,探究了信息系统与超图之间的对应关系,并且给出了两者之间相互转化的方法。利用此方法,结合超图理论与粗糙集理论,定义了基于超图的一对上下近似算子,进一步地,提出近似超图的概念,探究近似超图的相关性质,完成近似超图的构建,并在此基础上创建了一种有效方法以实现超图框架下的知识提取。将所提方法与经典的和新近提出的近似理论以及知识提取方法进行了对比,结果表明所提方法在近似方案和知识提取等方面具有多种优势。通过实际案例验证了所提方法的正确性,从而说明了其可应用性。所提方法是现有的知识提取理论的发展和补充。

关键词:知识提取;信息系统;超图;近似超图;可视化方法
中图分类号 TP391;TP182

Knowledge Extraction Algorithm Based on Hypergraphs

LIU Chuan[1],DU Baocang[2] and MAO Hua[1]
1 College of Mathematics and Information Science,Hebei University,Baoding,Hebei 071002,China
2 College of Management,Hebei Finance University,Baoding,Hebei 071051,China

Abstract Knowledge extraction has always been one of the topics in computer science research. However,some existing knowledge extraction methods are not sufficient to meet the practical needs in terms of visualization and latent knowledge extraction. It is well known that knowledge consists of definable knowledge and latent knowledge,and definable knowledge can be obtained while the latent knowledge is extracted,but not vice versa. Regarding the extraction of definable knowledge,many achievements have been made,but relatively less attention has been paid to the extraction of latent knowledge,especially how to extract latent knowledge through visualization methods,which is an urgent problem to be solved. Therefore,utilizing the visualization characteristics of hypergraphs in the context of information systems,this paper explores the correspondence between information systems and hypergraphs,and proposes methods for their mutual conversion. Using this method,combined with hypergraph theory and rough set theory,a pair of hypergraph-based upper and lower approximation operator is defined. Furthermore,the concept of approximate hypergraphs is proposed,and its properties are explored. The construction of approximate hypergraphs is completed,and an effective method for knowledge extraction under the hypergraph framework is implemented. By comparing with classical and recently proposed approximation theories and knowledge extraction methods,the advantages of the proposed method in terms of approximation and knowledge extraction are demonstrated. For the proposed method,its correctness is verified through practical examples,so that its applicability is indicated. The proposed method is a development and supplement to existing knowledge extraction theories.

Keywords Knowledge extraction,Information system,Hypergraph,Approximate hypergraph,Visualization methods

到稿日期:2024-01-08 返修日期:2024-07-12
基金项目:国家自然科学基金(61572011)
This work was supported by the National Natural Science Foundation of China(61572011).
通信作者:毛华(mh@hbu.edu.cn)

1 引言

知识提取一直是计算机领域研究的主题之一。随着科技的发展，如今的网络中蕴藏着大量质量参差不齐的资源，需要大量的人力资源去寻找有价值的知识。如何在这些资源中挖掘出有价值的知识是知识提取领域的一个主要研究方向。

由于知识提取中有些知识不能精确地进行表达，因此Pawlak提出了粗糙集理论[1]。作为处理模糊数据的数学工具，粗糙集理论不仅能有效地分析不精确、不一致等不完备信息，还可以对数据进行分析和推理，从中发现隐含知识，揭示潜在的规律。知识在不同领域的解释不同，其中具有代表性的是Pawlak[2]对知识的描述，即：将一个论域U的任一子集称为知识，也就是对整个论域进行分类的能力，一般由特征属性进行分类。此外，Pawlak还根据粗糙集中的近似算子将知识细分为可定义知识与不可定义知识(又称潜在知识)。

目前，在知识提取上的研究已取得许多成果，例如Mao等[3]利用超图在一类信息系统——形式背景下进行知识提取并进行了属性约简，做到了知识提取的可视化，但是在知识提取的范围上却不够广，未能涉及潜在知识的提取；Yao等[4]通过粗糙集及三支决策信息系统对知识进行提取，拓宽了传统知识提取的范围，提供了三支近似算子这一新的知识提取工具，但是在可视化方面却并未涉及；Jaradeh等[5]对知识图谱这一特殊的信息系统进行可视化知识提取，并且将其推广至大数据模型之中，但是对于潜在知识的提取却未涉足。

根据Pawlak[2]的研究可知，可定义知识既可以单独提取，也可以在潜在知识提取的过程中表现出来，反之则不然。因此有必要对潜在知识进行深入研究。

总的来说，现有的研究大多是从多种不同类型的信息系统的角度对知识提取的可视化和潜在知识提取这两方面进行探索，但是如何在一般信息系统而非特殊类型的信息系统中实现知识提取的可视化和潜在知识的挖掘，目前尚未有可行的解决方案。事实上，可视化方法在知识提取领域中有着巨大的优势：1)可以使得数据的趋势、关联等特征更加直观并易于理解；2)可以将知识提取结果直接呈现给包括专业人员在内的相关人员，进而更好地传达和解释提取的知识；3)更易于帮助研究者对知识提取的效果进行评估和优化；4)可视化方法能够提供用户交互界面和工具，使用户能够主动参与到知识提取的过程当中，从而有利于改进提取结果的准确性和完整性。至于潜在知识的可视化提取，一方面可以帮助挖掘和发现以前未知的知识；另一方面可以将其用于知识推理和应用。因此，现有的知识提取方法在可视性以及对潜在知识的提取能力上存在的弱势限制了其发展和应用范围。

超图(Hypergraph)[6]是图论的一个分支，同时也是经典图论的扩展，是有限集的子集系统。它与经典图(简称图)之间的区别在于每条边所能连接的最多节点数。在图中，每条边连接的节点数至多为2，而超图中每条超边至多可连接的节点数可以大于2，这使得超图更适合用来描述具有复杂多元关系的数据。在物理[7]、数学中的一些其他分支[8]以及计算和生物学[9]等自然科学领域，超图都有着广泛的应用。20世纪60年代，Berge[10]提出了超图理论，随后法国和匈牙利的数学家对其进行了完善，提出了有向超图、超图着色和超图设计等理论[11]。1970年，Berge首次系统地建立了超图理论，并应用矩阵结构研究了超图理论在运筹学中的应用。其后又有许多研究人员深入探索了正规超图[12]、完全图及三色超图等理论[13]。然而，早期的超图理论主要用于求解组合数学问题。20世纪80年代，随着数据库理论[14]研究的不断深入，信息科学家们发现了超图与数据库之间的联系，并引入了非循环超图等概念以解决实际问题[15-16]。事实上，超图在信息检索[17]、推荐系统[18-19]、自然语言处理[20]等场景中均发挥了重要作用。不仅如此，20世纪90年代，随着信息技术的不断发展和普及，人类积累的各种复杂网络系统的数据量呈指数级增长，但与规模日益增大的数据量相比，人们分析数据的能力以及从中获取知识的能力都存在着相当大的差距，因此形成了“数据过剩”而又“信息匮乏”的被动局面。在这种情况下，超图基于其描述多变量、高阶、复杂关系的能力，被广泛应用到基于复杂网络发展的图像分割[21]、社交网络[22]、疾病诊断[23]和数据融合[24]等现代生活实践领域之中。

总的来说，使用超图理论具有以下优势：

1)超图对于复杂多元关系的处理能力更强，这使得超图理论在大数据时代相较于其他理论在知识提取领域中有着更为广阔的应用前景；

2)超图可以做到知识提取的可视化，使得知识提取的过程可以通过图像更为直观地展现出来，降低了知识提取的相关操作的难度，使知识提取的过程更容易理解。

事实上，每个超图都可以用一个可视的示图来表达，这使得超图具有良好的可视性，而这一优势与知识提取相结合的成果与实际需求存在差距。因此，若将超图理论应用到粗糙集理论的知识提取的研究上，相信超图可以有效地利用自身良好的可视性，获得很好的成果，进而扩大知识提取的应用范围，推动知识提取研究的发展。

根据以上的分析可知目前在知识提取的研究中存在的急需解决的问题之一是：对任何一个信息系统，如何通过可视化的方法提取潜在的知识？

针对该问题，本文在信息系统中研究了基于超图的知识提取问题，其主要创新包括两点：

1)提出了一种将超图与信息系统相互转换的方法；

2)基于所提出的E-距离构建了近似超图，结合1)，实现超图框架下的潜在知识的提取。

上述两个创新点的具体实现过程如下：

1)从信息系统、超图的定义及性质出发，探究了信息系统与超图的各个组成部分之间的对应关系，进而提出了信息系统与超图之间相互转化的方法，并用这种方法进行知识提取。

2)类似于粗糙集中关于上下近似的研究，本文在超图上定义了E-距离，并据此给出了基于超图的上下近似算子，进一步完成构建近似超图，并通过近似超图实现了潜在知识的提取。

3)在文献[25]中的真实生物数据的基础上，用所提出的方法实现了知识提取，并与原文结果进行了对比。

由于超图可视化，而近似超图是基于超图理论进行工作的，因此具有良好的可视性。这使得本文的主要贡献不仅限

于提取了知识（特别是潜在知识提取方面），而且整个提取过程是在可视化框架下完成的，并用实际案例验证了成果，从而不仅解决了存在的问题，而且在解决问题的同时说明了解决该问题的实用性。

2　相关工作

下面将给出本文后续工作中所需的一些有关超图、序理论、粗糙集的基本定义及定理。更详细的内容，超图见文献[6]、序理论见文献[26]、粗糙集见文献[1,2,27]。

2.1　超图

定义 1（超图）[6]　超图 $H=(V;E=\{e_i\mid i=1,\cdots,m\})$，其中集合 V 为点集合，集合 $E=\{e_i\mid i=1,\cdots,m\}$ 为超边集合，e_i 为超边且满足 $e_i\subseteq V(i=1,\cdots,m)$，$0<|V|,|E|<+\infty$。特别地，当 $|e_i|=0$ 时，超边 e_i 被称为空超边。

为了方便描述，后续工作中的一些简记符号如下：

1）记集合 V 的幂集合为 2^V；

2）记构成超边的点集 $V(e)$ 为 e；

3）对于 $V_1\subseteq V,e\subseteq E$，记 $V_1\bigcap V(e)$ 为 $V_1\bigcap e$；

4）对于 $\forall e_1,e_2\in E$，记 $V(e_1)\bigcap V(e_2)$ 为 $e_1\bigcap e_2$。

另外，根据定义 1，后续工作中有关超图使用的符号如表 1 所列。

表 1　基本符号表示

Table 1　Notations of basic symbols

符号	定义
e^*	$e^*\in 2^V$
T	$T=\max\{\lvert e^*\rvert\mid e^*\in 2^V\}$
A_{e^*}	$A_{e^*}=\begin{cases}1, & e^*\in E\\ 0, & e^*\notin E\end{cases}$
$\boldsymbol{W}^{[t]}$	t 阶 $n\times n$ 邻接矩阵
$W_{ij}^{[t]}$	$W_{ij}^{[t]}=\sum_{e^*\in 2^V,\lvert e^*\rvert=t}A_{e^*}$
$D_{ii}^{[t]}$	$D_{ii}^{[t]}=\sum_{j\in V}W_{ij}^{[t]}$
$L^{[t]}$	$L^{[t]}=D^{[t]}-W^{[t]}$

Gong 等[28]通过超图拉普拉斯矩阵的二次形式定义了一种超图上的线性不相干关系。

引理 1[28]　对于任意 $\boldsymbol{x}\in\mathbb{R}^n$，有 $\boldsymbol{x}'L\boldsymbol{x}=\frac{1}{2}\eta_{\text{lin}}(G,\boldsymbol{x})$。其中，$L=\sum_{t=2}^{T}c_tL^{[t]}$，$\eta_{\text{lin}}(G,\boldsymbol{x})$ 是一种超图上的完全线性不相干关系。

Mao 等[3]探究了形式背景与超图之间的同构关系。

引理 2[3]　对于超图 H，$H(\mathbb{K}(H))$ 同构于 H。

2.2　序理论

定义 2（偏序关系）[26]　对于集合 R 上的关系 $\leqslant$，若其具有自反性、反对称性和传递性，则称 $(R,\leqslant)$ 为偏序集。

自反性：对于任意 $r\in R$，有 $r\leqslant r$；

反对称性：对于任意 $r,s\in R$，若 $r\leqslant s$，且 $s\leqslant r$，则 $r=s$；

传递性：对于任意 $r,s,t\in R$，若 $r\leqslant s$，且 $s\leqslant t$，则 $r\leqslant t$。

2.3　粗糙集

2.3.1　知识及信息系统

定义 3（知识空间）[1]　记 U 为论域，即一个有限的非空集合，$\boldsymbol{R}$ 为论域 U 上的等价关系的集合，知识空间 $K=(U,R=\{R_1,\cdots,R_m\})$ 表示等价关系集 R 中所有可能的关系对论域 U 的划分。

定义 4（信息系统）[27]　信息系统是一个四元组 $S=(U,A,Z,f)$，其中 U 表示研究对象的论域；A 表示属性的非空有限集合；$Z=\bigcup Z_a$，Z_a 是属性 a 的值域；f 表示 $U\times A\to Z$ 是一个信息函数，它为每个属性赋予一个信息值，即对 $\forall a\in A$，$\forall x\in U$，有 $f(x,a)\in Z$。

2.3.2　粗糙集的上下近似

定义 5（上下近似）[2]　对于一个论域 U，$\overline{apr}(X):2^U\to 2^U$ 及 $\underline{apr}(X):2^U\to 2^U$ 分别定义为：

$$\begin{cases}\overline{apr}(X)=\{x\in U\mid [x]\bigcap X\neq\varnothing\}\\ \underline{apr}(X)=\{x\in U\mid [x]\subseteq X\}\end{cases}$$

并分别称 $\overline{apr}$，$\underline{apr}$ 为上、下近似算子。

据此，Pawlak 将知识划分为可定义知识以及不可定义知识，并通过上下近似算子进行描述。可定义知识与不可定义知识统称为知识。

定义 6（可定义知识）[2]　X 为可定义知识，当且仅当 $\underline{apr}(X)=\overline{apr}(X)$ 成立。

（不可定义知识（或称潜在知识））X 为不可定义知识，当且仅当 $\underline{apr}(X)\neq\overline{apr}(X)$ 成立。

2.3.3　Mean-Shift 聚类算法

聚类算法作为知识提取中常用的一类方法，在统计分析、生物信息学、数据压缩、计算机图像识别、医学影像分析等方面有着大量应用。本文选取了 Mean-Shift 聚类算法[29]，研究了聚类算法在基于超图的知识提取中的应用。算法过程伪代码如算法 1 所示。

算法 1　Mean-shift 聚类算法

输入：带宽 r，收敛阈值 t，数据集 X=｛x_i|i=1,…,n｝

输出：每个数据点所属的聚类簇

1. 选择一个数据点作为初始种子点 sd.
2. for i=1 to n do
3.　对于每个数据点 x_i，计算其在半径 r 内的邻域 $N(x_i)$；
4.　对于种子点 sd，计算其邻域内数据点的均值向量 **m**；
5.　如果 m 与 sd 的距离大于收敛阈值 t，则更新 sd 为 m，并重复上述步骤；
6. end for
7. Return 每个数据点 x_i 所属的聚类簇

3　信息系统与超图的关系

用超图进行知识提取，首先需要探究超图与知识系统之间的关系，即如何由一个信息系统构建出与之相对应的超图；进一步地，如何由一个超图构建出与之相对应的信息系统。

3.1　信息系统的超图表示

超图作为一种解决问题的数学工具，在实际生活中需要将对应的实际背景以信息系统的形式进行记录，并通过信息系统来生成与实际问题相对应的超图，这样才可以借助超图的可视化优势对所研究的问题进行讨论。为此，下面讨论信息系统与超图之间的对应关系。

首先，讨论如何由一个信息系统产生一个超图。

定理 1（对于信息系统）　$IS=(U=\{u_1,u_2,\cdots,u_n\},A=$

$\{a_1,a_2,\cdots,a_m\}$, $Z=\{0,1\}$, f), 一定存在一个超图 $H(IS)=(V,E)$, 且满足下列条件:

1) $V=\{v_i \mid i=1,\cdots,n\}$, $E=\{e_j \mid j=1,\cdots,m\}$;

2) $v_i=u_i$, $e_j=\{v_i \mid f(u_i,a_j)=1\}$。

证明:分步完成证明

1)点集 V 由信息系统 IS 的论域 U 中的点 $u_1,u_2,\cdots,u_n$ 一一对应生成,因此满足条件 $0<|V|<+\infty$。

2) e_j 为根据信息系统 IS 中的对象 u_i 和属性 a_j 之间的关系 $f(u_i,a_j)=1$ 生成,即

$$e_j=\{v_i \mid f(u_i,a_j)=1, i=1,\cdots,n; j=1,\cdots,m\}$$

又因为 $v_i \in V$, 所以有 $e_j \subseteq V(j=1,\cdots,m)$, $0<|E|=m<+\infty$。

3)根据定义 1, $H(IS)$ 是超图,再由上述两步可知 $H(IS)$ 满足条件 1)和 2)。

事实上,定理 1 的表述中给出了一种由信息系统 IS 构建超图 $H(IS)$ 的方法。不同于 Mao 等[3]的超图构建方法,定理 1 给出了由属性值为 $\{0,1\}$ 的信息系统生成超图的方法,而 Mao 等给出的是由属性值为 $\{0,1\}$ 的形式背景生成超图的方法。因为属性值为 $\{0,1\}$ 的形式背景是一种特殊的信息系统,所以本文提出的生成超图的方法不同于文献[3]中生成超图的方法,其适用范围更广。事实上,定理 1 中的超图生成方法可以进一步推广至属性值非 $\{0,1\}$ 的信息系统(见推论 4)。

不止于此,由定理 1 可以得到如下推论。

推论 1 定理 1 中的信息系统 IS 和超图 $H(IS)$ 满足对应关系 $v_i \in e_j \Leftrightarrow f(u_i,a_j)=1$, $(i=1,\cdots,n; j=1,\cdots,m)$。

推论 1 由定理 1 的证明过程直接可得。

推论 2 对于属性值域非 $\{0,1\}$ 的信息系统 $IS=(U,A,Z,f)$ 而言,同样存在一个超图 $H(IS)=(V,E)$ 与之对应。

推论 2 与定理 1 的区别在于信息系统中属性值 Z 的取值不同,因此只需说明可以通过一些过程将属性值取值不为 $\{0,1\}$ 的信息系统转化为属性值取值为 $\{0,1\}$ 的信息系统,即可根据定理 1 得到推论 2。事实上,聚类算法完全可以完成由 Z 到 $\{0,1\}$ 的转化。

其次,讨论如何由一个超图构建一个信息系统。

定理 2 设 $H=(V;E=\{e_i \mid i=1,\cdots,m\})$ 为一个超图,令 $U=V$, $A=\{a_j \mid j=1,\cdots,m\}$,映射 f 满足 $f(u_i,a_j)=1 \Leftrightarrow v_i \in e_j$, $i=1,\cdots,n; j=1,\cdots,m$,则 $IS(H)=(U,A,\{0,1\},f)$ 是一个信息系统。

证明:分步完成证明

1)取超图 H 中的点 $v_1,v_2,\cdots,v_n$ 作为结构 $IS(H)$ 中的集合 U,即满足 $v_i=u_i$, $(i=1,\cdots,n)$。由超图的定义可知, U 和 E 是非空有限集合,从而有 $|U|,|A|<+\infty$。

2)考虑映射 f,由 f 的定义可知其满足 $f:U\times A\to\{0,1\}$。因此由定义 4 可知, f 是一个信息函数。

3)由定义 4 及上述两步可知, $IS(H)=(U,A,\{0,1\},f)$ 是一个信息系统。□

定理 2 给出了一种由超图 H 构建信息系统 $IS(H)$ 的方法。事实上,由定理 2 还可得到推论 3。

推论 3 定理 2 中的信息系统 $IS(H)$ 和超图 H 满足对应关系 $v_i \in e_j \Leftrightarrow f(u_i,a_j)=1, i=1,\cdots,n; j=1,\cdots,m$。

推论 3 可由定理 2 的证明过程直接得到。

事实上,由超图同样可以构建属性取值非 $\{0,1\}$ 的信息系统(见推论 2)。

推论 4 对于超图 $H=(V;E=\{e_i \mid i=1,\cdots,m\})$, $IS=(U,A,Z,f)$ 是一个信息系统,其中 $U=\{u_i \mid i=1,\cdots,n\}$, $A=\{a_1,\cdots,a_m\}$,属性值域 $Z_1=[0,1]$。对于信息函数,有 $f(u_j,a_i)=|v_j-v_{i*}|/\max\{|v_t-v_{i*}| \mid v_t \in e_i\}$,其中 v_{i*} 是超边 e_i 的中心点。

推论 4 的证明过程与定理 2 类似,只需将第二步中映射 f 的定义变为 $f(u_j,a_i)=|v_j-v_{i*}|/\max\{|v_t-v_{i*}| \mid v_t \in e_i\}$ 即可完成证明。

结合定理 1、定理 2 以及引理 2,可以得到推论 5。

推论 5 对于一个超图 $H=(V;E)$,有 $H(IS(H))=H$;而对于一个信息系统 $IS=(U,A,Z_2=\{0,1\},f)$,有 $IS(H(IS))=IS$。

1)由推论 5 可知,对于超图 H(属性值取值为 $\{0,1\}$ 的信息系统 IS,其所生成的信息系统对应的超图(信息系统)是唯一的。

2) 由定理 2 可得,对于任意一个超图 H,都存在一个信息系统 $IS(H)$ 与之对应。

3)由定理 1 可得,对于任意一个属性值取值为 $\{0,1\}$ 信息系统 IS,都存在一个超图 $H(IS)$ 与之对应。

依据 1)-3)可得超图与信息系统中术语的对应关系,如表 2 所列。

表 2 超图与信息系统的对应关系

Table 2 Correspondence between hypergraph and information system

Information System	Hypergraph
(U,A,Z,f)	(V,E)
A	E
U	V
$f:U\to Z$	$g:V(H)\to E(H)$

此外,Zhao[30]给出了信息系统和知识空间之间的对应关系(见表 3)。

表 3 知识空间与信息系统的对应关系

Table 3 Correspondence between knowledge space and information system

Information System	Knowledge Space
(U,A,Z,f)	$(U,\boldsymbol{R}\subseteq 2^U)$
A	R
U	U
$f:U\to Z$	$R\in\boldsymbol{R}$

由表 2 和表 3 可以得到表 4。

表 4 超图、知识空间与信息系统的对应关系

Table 4 Correspondence among hypergraph, knowledge space and information system

Information System	Knowledge Space	Hypergraph
(U,A,Z,f)	$(U,\boldsymbol{R}\subseteq 2^U)$	(V,E)
A	$\boldsymbol{R}$	E
U	U	V
$f:U\to Z$	$R\in\boldsymbol{R}$	$g:V(H)\to E(H)$

由表 4 可知,信息系统、知识空间和超图之间的术语存在一一对应关系,因此可以利用这种对应关系,讨论在信息系统

下和知识空间中知识提取的超图框架和超图方法。

3.2　信息系统的超图生成算法

由 3.1 节可知，每一个已知的信息系统 IS 都存在唯一一个与之对应的超图 $H(IS)$。因此有必要提出一种通过信息系统构建超图的算法。具体的算法生成过程如算法 2 所示。

3.2.1　算法生成过程

算法 2　{0,1}-IS 超图生成算法

输入：信息系统($U=\{u_j|j=1,\cdots,n\}$，$A=\{a_i|i=1,\cdots,m\}$，$Z=\{0,1\}$，f)

输出：超图 $H=(V=\{v_j|j=1,\cdots,n\};E=\{e_i|i=1,\cdots,m\})$

1. 初始化：V，E
2. for j=1 to n do
3. 　　$v_j=u_j$；
4. end for
5. for i=1 to m do
6. 　　for j=1 to n do
7. 　　$e_i=e_i\cup\{v_j|f(u_j,a_i)=1\}$；
8. 　　end for
9. end for
10. Return H=(V;E)

算法 2 是根据定理 1 直接得到的超图生成算法，其正确性由定理 1 保证。下面对算法的复杂度进行分析。

1)第 1 步为初始化由信息系统构建超图的点集合 V 与超边集合 E。

2)第 2－4 步为基于信息系统的对象集构造超图的点集 V，这一步需要对每一个对象进行赋值运算，共需进行 $|U|$ 次操作，其时间复杂度为 $O(|U|)=O(n)$。

3)第 5－9 步为根据信息系统中属性的个数来进行相应次数的循环。在每一次循环中，根据对象对应的属性值进行判定，共需进行 $|U||A|$ 次操作，其时间复杂度为 $O(|U||A|)=O(mn)$。

由 2)的结果可知，这一循环过程共需进行 $|m|$ 次，因此算法第 5－9 步共需 $|A|$ 次循环，其时间复杂度为 $O(|A||U||A|)=O(m^2n)$。

4)由 1)，2)，3)可知，算法 2 的时间复杂度为 2)和 3)的求和 $O(n+m^2n)$。

对于信息系统中属性值只有 0 和 1 的情况，使用算法 2 即可，但当信息系统中的属性值为其余离散取值乃至于区间值时，算法 2 不再有效。事实上，算法 2 的超边生成过程核心为判定对象的属性值是否为 1，而当属性值为其余取值时，这一过程可以通过聚类算法实现。这里利用 Mean-Shift 聚类算法[29]对算法 2 进行修改，得到算法 3。

算法 3　IS 超图生成算法

输入：信息系统 IS=($U=\{u_j|j=1,\cdots,n\}$，$A=\{a_i|i=1,\cdots,m\}$，Z，f)

输出：超图 $H=(V=\{v_j|j=1,\cdots,n\};E=\{e_i|i=1,\cdots,m\})$

1. 初始化：V，E
2. for j=1 to n do
3. 　　$v_j=u_j$；
4. end for
5. for i=1 to m do
6. 　　for j=1 to n do
7. 　　e_i=Meanshift(U，a_i，f)；
8. 　　end for
9. end for
10. Return H=(V;E)

算法 3 与算法 2 的主要区别在于超边的生成方式，针对属性值取值不为 0，1 的信息系统，这里采用 Mean-Shift 聚类算法替代算法 2 中的超边生成过程，即算法 2 的第 5－9 步。算法 3 的正确性同样可以由定理 1 保证。

下面对算法 3 的复杂度进行分析：

由于算法 3 是对算法 2 的修改，这里只对更改的地方进行分析。

1)算法 3 的第 5－9 步为根据属性的个数来进行相应次数的循环。在每一次循环中，根据对象对应的属性值进行聚类，这一步的算法复杂度基于所使用的聚类算法的复杂度，而 Mean-Shift 聚类算法的复杂度为 $O(T|U||A|\log(|U||A|))$，其中 T 为在迭代过程中选取的中心点数，在这里的超图构建过程中取为 1。因此每一次循环的复杂度为 $O(|U||A|\log(|U||A|))=O(mn\log(mn))$。

算法 2 的循环过程为根据属性集中的每一条属性进行聚类运算，因此算法的第 5－9 步共需 $|A|$ 次循环。

2)由算法 2 的结果可得算法 3 的时间复杂度为 $O(|U|)+O(|A|)O(|U||A|\log(|U||A|))=O(m^2n\log n+n)$。

3.2.2　实验

下面用一个实际案例来实现算法 2 和算法 3。

案例 1　从文献[25]的表 3 中任意选取部分数据(见表 5)。

对所选取的几类特征进行重新标号，令 a_1＝“贮液囊呈卵形”，a_2＝“囊体着生于第Ⅴ可见腹板近基部 2/3 之前”，a_3＝“贮液囊末端超过第Ⅳ可见腹板基部”，a_4＝“贮液囊壁厚”，a_5＝“贮液囊外表具有稀疏花纹”，具体如表 5 所列。

表 5　琵甲族 7 属昆虫防御腺特征值

Table 5　Characteristics codes of defensive glands of 7 genera of Blaptini

	a_1	a_2	a_3	a_4	a_5
琵甲属	1	0	1	1	0
异琵甲属	1	0	1	0	1
贞琵甲属	0	0	0	0	0
小琵甲属	0	0	1	1	0
乾琵甲属	0	0	1	1	1
贝琵甲属	0	0	1	1	0
格琵甲属	0	1	1	1	0

注：0＝“该属生物无此特征”；1＝“该属生物有此特征”。

令 u_1＝“琵甲属”，u_2＝“异琵甲属”，u_3＝“贞琵甲属”，u_4＝“小琵甲属”，u_5＝“乾琵甲属”，u_6＝“贝琵甲属”，u_7＝“格琵甲属”，具体如表 6 所列。

表 6　表 5 的数学表示

Table 6　Mathematical representation of Table 5

	a_1	a_2	a_3	a_4	a_5
u_1	1	0	1	1	0
u_2	1	0	1	0	1
u_3	0	0	0	0	0
u_4	0	0	1	1	0
u_5	0	0	1	1	1
u_6	0	0	1	1	0
u_7	0	1	1	1	0

注：0＝“该属生物无此特征”；1＝“该属生物有此特征”。

根据算法2,基于表6所列的信息系统生成超图。

根据算法2的第1步,为将要生成的超边分配存储空间,可得超图边集为 $E(IS)=\{e_1,\cdots,e_5\}$,其中 $e_i=\varnothing, i=1,\cdots,5$。

由算法2的第2-4步,超图点集为 $U(IS)=\{u_1,u_2,u_3,u_4,u_5,u_6,u_7\}$。

之后,首先对属性 a_1 进行算法2的第5-9步的实现。

$f(u_1,a_1)=1$, $e_1=\varnothing\cup\{v_1\}=\{v_1\}$; $f(u_2,a_1)=1$, $e_1=e_1\cup\{v_2\}=\{v_1,v_2\}$;而 $f(u_3,a_1)=f(u_4,a_1)=\cdots=f(u_7,a_1)=0$。对于点 $\{v_3,v_4,v_5,v_6,v_7\}$,超边 e_1 不再进行更新。因此对属性 a_1 进行算法2的第6-7步可得 $e_1=\{v_1,v_2\}$。

类似地,对属性 a_2 进行算法的第5-9步,可得 $e_2=\{u_7\}$;对属性 a_3 进行算法2的第5-9步,可得 $e_3=\{u_1,u_2,u_4,u_5,u_6,u_7\}$;对属性 a_4 进行算法2的第5-9步,可得 $e_4=\{u_1,u_4,u_5,u_6,u_7\}$;对属性 a_5 进行算法2的第5-9步,可得 $e_5=\{u_2,u_5\}$。

由算法2第10步,有 $H(IS)=(U=\{u_1,u_2,u_3,u_4,u_5,u_6,u_7\},E=\{e_1,e_2,e_3,e_4,e_5\})$。据此可得 $H(IS)$ 的示图表示,如图1所示。此处可知算法1在此案例中的复杂度为 $O(n+m^2n)=O(7+5^2\times7)=O(182)$。

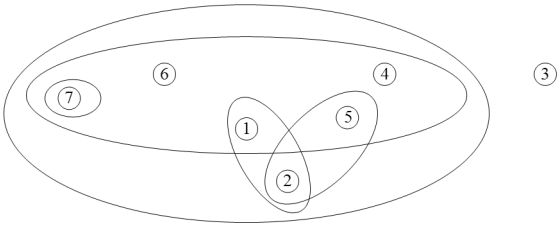

图1 $H(IS)$的示图表示

Fig. 1 Graphic representation of $H(IS)$

下面根据算法3,基于表6所列的信息生成超图。

根据算法2的第1步,为将要生成的超边分配存储空间,可得超图边集为 $E(IS)=\{e_1,\cdots,e_5\}$,其中 $e_i=\varnothing, i=1,\cdots,5$;由算法2的第2-4步,超图点集为 $U(IS)=\{u_1,u_2,u_3,u_4,u_5,u_6,u_7\}$。

之后,首先对属性 a_1 进行算法3的第5-9步的实现。对属性 a_1 进行 Mean-Shift 聚类,这里取 $U=\{u_1,u_2,u_3,u_4,u_5,u_6,u_7\}$, $a_i=a_1$, $f(u_1,a_1)=1$, $f(u_2,a_1)=1$, $f(u_3,a_1)=f(u_4,a_1)=\cdots=f(u_7,a_1)=0$。据此所得的结果为 $e_1=\{v_1,v_2\}$。

类似地,对属性 a_2 进行算法3的第5-9步,可得 $e_2=\{u_7\}$;对属性 a_3 进行算法3的第5-9步,可得 $e_3=\{u_1,u_2,u_4,u_5,u_6,u_7\}$;对属性 a_4 进行算法3的第5-9步,可得,$e_4=\{u_1,u_4,u_5,u_6,u_7\}$;对属性 a_5 进行算法3的第5-9步,可得,$e_5=\{u_2,u_5\}$。

由算法3第10步,有 $H(IS)=(U=\{u_1,u_2,u_3,u_4,u_5,u_6,u_7\},E=\{e_1,e_2,e_3,e_4,e_5\})$。

可以发现,由算法3基于表6生成的超图与由算法2生成的超图相同,即为图1。

由此可知算法2在此案例中的复杂度为 $O(m^2n\log n+n)=O(7+5^2\times7\times\log7)=O(347.5)$。

综合算法2及算法3的运行结果,如图1所示,包含表6中信息的超图 $H(IS)$,可以由5条超边构成,记为 $E'=\{\{u_1,u_2\},\{u_7\},\{u_1,u_2,u_4,u_5,u_6,u_7\},\{u_1,u_4,u_5,u_6,u_7\},\{u_2,u_5\}\}$。可以看到,当多个节点同时具有某种性质或满足某个条件时,只需用1条超边进行表示即可。

3.2.3 算法对比

选取几种现有的优秀的超图生成算法与本文提出的超图生成算法(即算法3)进行对比,结果如表7所列。记信息系统 $IS=(U=\{u_1,\cdots,u_n\},A=\{a_1,\cdots,a_m\},Z,f)$。

表7 超图构建算法对比

Table 7 Comparison of hypergraph generation algorithms

算法	时间复杂度	存储方式	输入数据格式
本文	$O(m^2n\log mn+n)$	邻接矩阵	任意信息系统
文献[31]	$O(m^2n^2)$	邻接矩阵	图
文献[32]	$O(mn)$	邻接矩阵	超图
文献[33]	$O(m^3n^2)$	多层邻接矩阵	元数据

由表7可知:

1)在算法复杂度以及输入数据格式方面,文献[31]中提出的超图构建算法的时间复杂度为 $O(m^2n^2)$。相比之下,当 $n>\log m$ 时,有 $O(m^2n^2)$ 远大于 $O(m^2n\log mn+n)$,也就是说本文算法在时间复杂度方面有着一定的优势。在输入数据格式方面,文献[31]的方案对于原始数据是否具有图结构,给出了不同的超边生成方法,但最终均为通过邻接矩阵或关联矩阵的方式进行存储和超边生成,实际上仍要求输入数据具有图结构,在这方面本文算法的要求更低,适用范围更广。

文献[32]的超图构建算法只提到了通过超图的对角矩阵进行超边的生成及存储,其时间复杂度为 $O(mn)$。尽管文献[33]中算法的时间复杂度略低于本文算法的 $O(m^2n\log mn+n)$,但是该算法对输入数据格式有着严格的要求,其适用范围远不如本文算法。

综合来看,虽然本文提出的算法在时间复杂度方面略有弱势,但具有很好的普适性。在输入数据格式方面,文献[32]的算法要求严苛,输入的数据必须为超图结构,适用性方面不如本文算法。

文献[33]中提到的超图构建方案针对同一节点多次调用其节点信息以生成超边进行学习。多次调用造成这一模块的时间复杂度为 $O(m^3n^2)$,尽管提升了算法学习效果,但在时间复杂度方面,因为 $O(m^2n\log mn+n)$ 远小于 $O(m^3n^2)$,所以本文算法相比文献[33]具有较大的优势。在输入数据格式的方面,文献[33]中的算法要求格式为元数据,而对于其余的数据格式处理能力仍有不足。相比之下,本文算法对于输入数据的格式没有任何要求,适用于各类格式的数据。因此在适用性方面,本文算法强于文献[33]中的算法。

2)在存储方式方面,本文与文献[31]和文献[32]均为邻接矩阵方式,而文献[33]采取了多层邻接矩阵方式。多层邻接矩阵是邻接矩阵中的一种特殊方法,主要针对某些关系较为复杂的信息系统,但对于一般的信息系统而言,其复杂度较单层邻接矩阵而言高了很多,这说明邻接矩阵在一般情况下是优于多层邻接矩阵方式的。在文献[31-33]这3种新方法中,有两种在存储方式方面采用了邻接矩阵的方式,这说明该方式的使用更为广泛,适用性也更强。

3.3 超图的信息系统生成算法

由3.1节可知,对于每一个已知超图 H,都存在着一个与

之对应的信息系统 $IS(H)$，因此有必要提出一个通过超图构建信息系统的算法，见算法 3 和算法 4。

3.3.1　算法实现过程

算法 3　{0，1}-信息系统生成算法

输入：超图 H=(V={v_j|j=1，…，n}；E={e_i|i=1，…，m})

输出：信息系统(U={u_j|j=1，…，n}，A={a_i|i=1，…，m}，Z={0，1}，f)

1. 初始化：U，A
2. for j=1 to n do
3.　$u_j=v_j$；
4. end for
5. for i=1 to m do
6.　for j=1 to n do
7.　if $v_j\in e_i$
8.　$a_i=a_i\cup\{u_j\}$
9.　$f(u_j,a_i)=1$；
10.　else
11.　$f(u_j,a_i)=0$；
12.　end if
13.　end for
14. end for
15. Return IS=(U={u_j|j=1，…，n}，A={a_i|i=1，…，m}，Z={0，1}，f)

算法 3 是根据定理 2 直接得到的信息系统生成算法，其正确性由定理 2 保证。下面对算法 3 的复杂度进行分析。

1)算法 3 的第 1 步为初始化由超图构建信息系统的对象集 U 和属性集 A。

2)算法 3 的第 2－4 步为基于超图的点集 V 构造信息系统的对象集 U，这一步需要将每一个对象进行赋值运算，共需进行 $|V|$ 次操作，其时间复杂度为 $O(|V|)=O(n)$。

3)算法 3 的第 5－14 步为根据超图中的超边进行循环，在每一次循环中根据点是否在超边上进行判定，并对其对应的值函数取值进行赋值，共需进行 $|V||E|$ 次操作，其时间复杂度为 $O(|V||E|)=O(mn)$。

由 2)的结果可知，这一循环过程共需进行 $|m|$ 次，因此算法的第 5－14 步共需 $|E|$ 次循环，其时间复杂度为 $O(|E||V||E|)=O(m^2n)$。

4)由 1)，2)，3)可知，算法 3 的时间复杂度为 $O(n+m^2n)$。

算法 4　[0，1]信息系统生成算法

输入：超图 H=(V={v_j|j=1，…，n}；E={e_i|i=1，…，m})

输出：信息系统(U={u_j|j=1，…，n}，A={a_i|i=1，…，m}，[0，1]，f)

1. 初始化：U，A
2. for j=1 to n do
3.　$u_j=v_j$；
4. end for
5. for i=1 to m do
6.　v_{i*}=Meanshift(e_i)；
7. for j=1 to n do
8.　if $v_j\in e_i$
9.　$a_i=a_i\cup\{u_j\}$
10.　$f(u_j,a_i)=|v_j-v_{i*}|/\max\{|v_t-v_{i*}||v_t\in e_i\}$；
11.　else
12.　$f(u_j,a_i)=0$；
13.　end if
14.　end for
15. end for
16. Return IS=(U={u_j|j=1，…，n}，A={a_i|i=1，…，m}，[0，1]，f)

算法 4 与算法 3 的主要区别在于信息函数的生成方式。针对要生成属性值取值不为{0，1}的信息系统的情况，这里采取的思想为聚类算法的逆过程，即从超边所含的位于中间的节点开始，根据其余节点与中心节点的距离来定义信息函数。算法 4 的正确性由推论 4 保证。下面对算法的复杂度进行分析。

由于算法 4 是对算法 3 的修改，这里只对更改的部分进行分析。

1)算法 4 的第 6 步为根据超图的超边判定超边的中心节点，此步的中心提取算法同样采用 Mean-shift 聚类算法以得到超边的中心点 v_{i*}。

2)算法 4 的第 7－14 步为根据超图中的超边进行循环，在每一次循环中计算超图中的点和正在循环的超边间的距离。若点在超边上，则距离为 0；若点不在超边上，则根据公式计算距离。此外，对其对应的值函数取值进行赋值，共需进行 $|V||E|$ 次操作，其时间复杂度为 $O(|V||E|)=O(mn)$。

由 2)的结果可知，这一循环过程共需进行 $|m|$ 次，因此算法的第 7－14 步共需 $|E|$ 次循环，其时间复杂度为 $O(|E||V||E|)=O(m^2n)$。

3)结合算法 3 的复杂度分析可得，算法 4 的时间复杂度为 $O(n+m^2n)$。

3.3.2　实验

下面用一个实际案例来实现算法 3。

案例 2　这里考察的案例是由案例 1 生成的图 1，即超图 $H(IS)$。

据算法 3，基于图 1 所示的信息系统生成超图。

由算法 3 的第 1 步，给定将要生成的属性集存储空间，$A(H(IS))=\{a_1,\cdots,a_5\}$，其中 $a_i=\emptyset$，$i=1,\cdots,5$；

由算法 3 的第 2－4 步，信息系统的论域为 $U(H(IS))=\{u_1,u_2,u_3,u_4,u_5,u_6,u_7\}$。

之后，首先对超边 e_1 进行算法 3 第 5－14 步的实现。属性 $a_1=\{u_1,u_2\}$，且有 $f(u_1,a_1)=1$，$f(u_2,a_1)=1$，$f(u_3,a_1)=f(u_4,a_1)=\cdots=f(u_7,a_1)=0$；类似地，对于超边 e_2 进行算法 3 的第 5－14 步可得，$a_2=\{u_7\}$，$f(u_7,a_2)=1$。

对于属性 a_3，进行算法 3 的第 5－14 步可得，$a_3=\{u_1,u_2,u_4,u_5,u_6,u_7\}$，$f(u_1,a_3)=\cdots=f(u_7,a_3)=1$，$f(u_3,a_7)=0$；对于属性 a_4 进行算法 3 的第 5－14 步得，$a_4=\{u_1,u_4,u_5,u_6,u_7\}$，$f(u_1,a_4)=f(u_4,a_4)=f(u_5,a_4)=f(u_6,a_4)=f(u_7,a_4)=1$，$f(u_2,a_4)=f(u_3,a_4)=0$；对于属性 a_5 进行算法 3 的第 5－14 步可得，$a_5=\{u_2,u_5\}$，$f(u_2,a_5)=f(u_5,a_5)=1$；

由算法 3 第 15 步，有 $IS(H(IS))=(U=\{u_1,u_2,u_3,u_4,u_5,u_6,u_7\},A=\{a_1,a_2,a_3,a_4,a_5\},\{0,1\},f)$。

据此可得 $H(IS)$ 生成的信息系统如表 8 所列。

表 8 图 1 生成的信息系统

Table 8 Information system generated from Fig. 1

	a_1	a_2	a_3	a_4	a_5
u_1	1	0	1	1	0
u_2	1	0	1	0	1
u_3	0	0	0	0	0
u_4	0	0	1	1	0
u_5	0	0	1	1	1
u_6	0	0	1	1	0
u_7	0	1	1	1	0

算法 4 的实现过程与算法 3 类似。

这里给出算法 1－算法 4 与不同信息系统直接的适用关系，如图 2 所示。

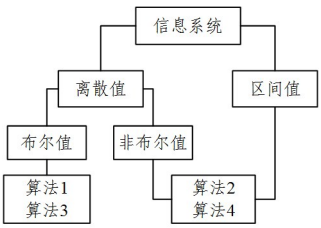

图 2 算法 1－4 与信息系统之间的对应关系

Fig. 2 Correspondence between algorithms 1－4 and information systems

4 近似超图的构建

以案例 1 为例，在生物种属分类中根据不同特征值的取值进行精细的分类。但对于非精确分类而言，例如只需寻找翅长小于定长且多足的生物时，如果按照精确值进行分类，则会带来巨大的计算资源消耗，并且会造成满足分类条件的生物种类数量不足。在实际生活中，还存在着一些类似的只需得到近似结果的问题。为了解决这类问题，借鉴粗糙集理论处理不确定性的方法，可以同时结合粗糙集对于不确定性信息的处理能力以及超图对于多元复杂系统的处理能力。

4.1 E-距离

在超图理论中引入上近似及下近似算子以得到基于超图的正、负和边界域。引入“距离”概念，用“距离”刻画超图下的上下近似。

定义 7（E-距离） 对于超图 $H(V,E)$ 中的任意一条超边 e，用 e 中所有节点对之间的欧几里得距离的平方来形式化表示在超图上这条超边所包含的信息“距离”，即：

$$D_{\mathrm{HE}}(V,e)=\sum_{v_i,v_j\in e}|v_i-v_j|^2$$

当 e 只包含一个节点时，令 $D_{\mathrm{HE}}(V,e)=0$。此处 $|v_i-v_j|=|i-j|(v_i,v_j\in e)$。

事实上，E-距离在信息系统中的本质是：信息系统中具有某一属性的所有对象间的整体差异。E-距离是欧几里得距离在超图理论下的拓广，这里使用 E-距离，主要是用于说明其对于不同的距离定义而言都有较好的适应性。

4.2 近似超图

对于一个信息系统 $IS=(U,A,v_a,f)$，利用第 3 章中的理论可以得到超图 $H(IS)$。由此，根据定义 3 可知 $(U,E(H(IS)))$ 构成了一个知识空间。根据算法 2 得到可定义知识为全体 $E(H(IS))$。因此，如何发现并提取此空间中的潜在知识是要解决的首要问题。下面给出超图上的近似算子及可定义边集的定义，并给出相关性质。

定义 8（近似算子） 设 $H=(V,E)$ 是一个超图，在 H 中定义近似算子如下。

上近似算子 $^{\vee_s}:E\to 2^V$ 定义为：

$$e\mapsto e\cup\{v\mid|D_{\mathrm{HE}}(V,e\cup\{v\})-D_{\mathrm{HE}}(V,e)|\leqslant s,v\in V\}$$

即：

$$e^{\vee_s}=e\cup\{v\mid|D_{\mathrm{HE}}(V,e\cup\{v\})-D_{\mathrm{HE}}(V,e)|\leqslant s,v\in V\}$$

下近似算子 $^{\wedge_s}:E\to 2^V$ 定义为：

$$e\mapsto e\backslash\{v\mid|D_{\mathrm{HE}}(V,e\backslash\{v\})-D_{\mathrm{HE}}(V,e)|\leqslant s,v\in V\}$$

即：

$$e^{\wedge_s}=e\{v\mid|D_{\mathrm{HE}}(V,e\backslash\{v\})-D_{\mathrm{HE}}(V,e)|\leqslant s,v\in V\}$$

由定义 5 易知，定义 8 中提出的算子是一对上下近似算子。

定义 8 中提出的上下近似算子是基于 E-距离提出的，而传统 IS 上的上下近似是基于 Pawlak 的等价关系提出的。对于集合 $X\subseteq U$，IS 之前的上下近似利用定理 1 和推论 3 转化到超图上时，可以分别表示为：

$$\overline{apr}(X)=\{x\in V\mid x(e)\bigcap X\neq\varnothing\}$$

$$\underline{apr}(X)=\{x\in V\mid x(e)\subseteq X\}$$

其中，$x(e)$ 表示超图中包含点 x 的所有超边中所包含的所有点。在引入 E-距离后，利用定义 7 可以将上述表示方式转化成为定义 8 的形式，但其中的参数 s 为一个固定值，且这一固定值会随实际背景不同而变动。

与之相对应的，对于集合 $Y\subseteq V$，超图上的上下近似在固定参数 s 的取值后利用定理 2 和推论 4 转化到 IS 上时，分别表现为：

$$Y^{\vee_s}=\{y\in U\mid[y]\bigcap Y\neq\varnothing\}$$

$$Y^{\wedge_s}=\{y\in U\mid[y]\subseteq Y\}$$

其中，$[y]$ 表示在 IS 中包含对象 y 的等价类。

事实上，由推论 5 和引理 2 可知，IS 和超图之间有着一一对应关系，并且这种关系保证了超图和 IS 上的上下近似下是一致的。但 IS 上的上下近似是基于 Pawlak 的等价关系作为前提定义的，而超图上的上下近似是基于 E-距离给出的。总地来说，超图上的上下近似是 IS 上的上下近似的一种拓广，是 IS 上的上下近似在复杂多元系统下的表示，弥补了 IS 之前的上下近似方式在复杂多元系统下的不足，并且给出了原来的约简一类问题的不同解决方案[3]。

为了便于表示，记经过上（下）近似算子 $^{\vee_s}$（$^{\wedge_s}$）运算得到的超边集合为 $E^{\vee_s}=\{e^{\vee_s}\mid e\in E\}$（$E^{\wedge_s}=\{e^{\wedge_s}\mid e\in E\}$）。由定义 1 和定义 8 可知，$H^{\vee_s}=(V,E^{\vee_s})$。（$H^{\wedge_s}=(V,E^{\wedge_s})$）是一个超图，并被称为超图 H 的 s-上（下）近似超图。

在这里引入参数 s 的目的是增加近似超图的可控性，即可以根据近似超图的用途需求，通过简单地调整参数 s 的大小来改变所得的近似超图的规模大小，并且参数 s 需要根据对应问题的现实意义进行选取。事实上，由定义 8 可知，对于任何 $e\in E$，其通过上（下）近似算子映射的像 $e^{\vee_s}$（$e^{\wedge_s}$）中参数 s 的取值通过 E-距离进行约束。由定义 7 可知，本文中参数 s 的取值均为非负整数，即 $s\in\boldsymbol{N}$。

为了实现用超图挖掘知识，下面先给出超边的上下近似算子满足的一些性质。

定理 3　在超图 $H=(V,E)$ 中，对于 $\forall e\in E$ 及 $\forall E_1, E_2\subseteq E$，超边上下近似算子有以下性质成立：

1) $e^{\wedge_s}\subseteq e\subseteq e^{\vee_s}$

2) $e^{\wedge_s}=e=e^{\vee_s}\Leftrightarrow s=0$

3) $(E_1\cup E_2)^{\wedge_s}=E_1^{\wedge_s}\cup E_2^{\wedge_s}$

4) $(E_1\cup E_2)^{\vee_s}=E_1^{\vee_s}\cup E_2^{\vee_s}$

5) $E_1\subseteq E_2\Rightarrow E_1^{\wedge_s}\subseteq E_2^{\wedge_s}$

6) $E_1\subseteq E_2\Rightarrow E_1^{\vee_s}\subseteq E_2^{\vee_s}$

这里满足 $e\in E, E_1, E_2\subseteq E$。

证明：

1)对于 $\forall e\in E$，根据定义 8，$e^{\vee_s}=e\cup\{v\mid |D_{HE}(V,e\cup\{v\})-D_{HE}(V,e)|\leqslant s, v\in V, e\in E\}$，$e^{\wedge_s}=e\backslash\{v\mid |D_{HE}(V,e\backslash\{v\})-D_{HE}(V,e)|\leqslant s, v\in V, e\in E\}$，而空集合 $\varnothing\subseteq\{v\mid |D_{HE}(V,e\cup\{v\})-D_{HE}(V,e)|\leqslant s, v\in V, e\in E\}$，且 $\varnothing\subseteq e\backslash\{v\mid |D_{HE}(V,e\backslash\{v\})-D_{HE}(V,e)|\leqslant s, v\in V, e\in E\}$。

由定义 2 易知 $(2^V,\leqslant)$ 是一个偏序集，由集合间的偏序关系可知，有 $e^{\wedge_s}\subseteq e\backslash\varnothing\subseteq e\subseteq e\cup\varnothing\subseteq e^{\vee_s}$ 成立，即有 $e^{\wedge_s}\subseteq e\subseteq e^{\vee_s}$ 成立。

2)将分“⇒”和“⇐”两方面完成证明。

“⇒”：若满足 $e^{\wedge_s}=e=e^{\vee_s}$，则有

$\{v\mid |D_{HE}(V,e\cup\{v\})-D_{HE}(V,e)|\leqslant s, v\in V, e\in E\}=\varnothing$

$\{v\mid |D_{HE}(V,e\backslash\{v\})-D_{HE}(V,e)|\leqslant s, v\in V, e\in E\}=\varnothing$

即 $|D_{HE}(V,e\cup\{v\})-D_{HE}(V,e)|\leqslant s$ 和 $|D_{HE}(V,e\backslash\{v\})-D_{HE}(V,e)|\leqslant s$ 关于点 v 的解集为空。由定义 7 及 $s\in N$ 可知，要满足上述条件即需 $s=\min\{|e|\mid e\in E\}=0$。

“⇐”：若有 $s=0$，根据定义 8 可得关于点 v 的不等式组

$|D_{HE}(V,e\cup\{v\})-D_{HE}(V,e)|\leqslant s$

$|D_{HE}(V,e\backslash\{v\})-D_{HE}(V,e)|\leqslant s$

的解集为空，从而有 $e^{\wedge_s}=e=e^{\vee_s}$ 成立。

3)由定义 8，$E^{\wedge_s}=\{e^{\wedge_s}\mid e\in E\}$ 有

$(E_1\cup E_2)^{\wedge_s}$

$=\{e^{\wedge_s}\mid e\in E_1\cup E_2\}$(定义 12)

$=\{e^{\wedge_s}\mid e\in E_1\}\cup\{e^{\wedge_s}\mid e\in E_2\}$(集合的分配律)

$=E_1^{\wedge_s}\cup E_2^{\wedge_s}$(定义 8)

4)由性质 3)的证明过程，类似可得性质 4)。

5)若 $E_1\subseteq E_2$，即对于 $\forall e\in E_1$，都有 $e\in E_2$ 成立。根据定义 8，$E_1^{\wedge_s}=\{e^{\wedge_s}\mid e\in E_1\}$，$E_2^{\wedge_s}=\{e^{\wedge_s}\mid e\in E_2\}$，对于 $\forall x\in E_1^{\wedge_s}$，有 $x\in E_2^{\wedge_s}$ 成立。因此有 $E_1^{\wedge_s}\subseteq E_2^{\wedge_s}$。

6)由性质 5)的证明过程，类似可得性质 6)。□

4.3　可定义超边及不可定义超边

与 Pawlak 的粗糙集[2]相对应，这里给出在近似超图框架下的可定义边集与不可定义边集的概念。

定义 9(可定义超边)　称超边 e 是 s-可定义超边，当且仅当 e 满足 $e^{\vee_s}=e^{\wedge_s}$。进一步地，称所有由 s-可定义超边组成的集合为 s-可定义超边集，记为：

$E_{def}=\{e\mid e\in E, e^{\vee_s}=e^{\wedge_s}\}$；

(不可定义超边) 称超边 e 是 s-不可定义超边，当且仅当 e 满足 $e^{\vee_s}\neq e^{\wedge_s}$。进一步地，称所有由 s-可定义超边组成的集合为 s-不可定义超边集，记为：

$E_{undef}=\{e\mid e\in E, e^{\vee_s}=e^{\wedge_s}\}$

5　知识提取方法

5.1　近似超图生成算法

在给出定义 8 和定义 9 之后，给出近似超图的生成算法，并对所提出的算法进行分析与对比。

5.1.1　算法生成过程

算法 5　上近似超图生成算法

输入：超图 H=(V={v_j|j=1,…,n};E={e_i|i=1,…,m})，近似参数 s

输出：s-上近似超图 $H^{\vee_s}$=(V={v_j|j=1,…,n};$E^{\vee_s}$={$e_i^{\vee_s}$|i=1,…,m})

```
1. 初始化：E^{∨s}
2. for i=1 to m do
3.     通过定义 7 计算 e_i 的 D_HE(V,e_i)；
4.     for j=1 to n do
5.       if v_j∈e_i
6.       continue;
7.       else
8.       通过定义 7 计算 e_i∪{v_j} 的 D_HE(V,e_i∪{v_j})；
9.       if |D_HE(V,e_i∪{v_j})−D_HE(V,e_i)|<s
10.      e_i=e_i∪{v_j}；
11.      e_i^{∨s}=e_i;
12.    end if
13.  end if
14.  end for
15. E^{∨s}=(E\{e_i})∪{e_i^{∨s}}；
16. end for
17. Return H^{∨s}=(V;E^{∨s})
```

算法 5 的正确性由定义 7、定义 8 及定理 3 保证。

下面对算法 5 的复杂度进行分析。

1)算法 5 的第 1 步为初始化构建 s-上近似超图 $H^{\vee_s}$ 的超边集合 $E^{\vee_s}$。

2)算法 5 的第 3 步为计算超图 H 上的 E-距离 $D_{HE}(V,e)$，超图 H 中的最大超边所含顶点个数为 $\max\{|e|\mid e\in E\}\leqslant n$。根据 E-距离的定义，最多需要 $O(mn^2)$ 次运算。

3)算法 5 的第 4－14 步为更新满足如下 E-距离约束条件的新超边 $e_i^{\vee_s}$，这一步的时间复杂度为 $O(mn^2\log_2 n)$。

4)综合 1)，2)和 3)可知，算法 5 的时间复杂度为 $O(mn^2\log_2 n+mn^2)$。

5.1.2　实验

下面用第 2 章中案例 1 所生成的超图为例实现算法 5，生成其上近似超图 $H^{\vee_1}$。

根据算法 5，基于图 1 所示超图生成其上近似超图 $H^{\vee_1}$。

由算法 5 的第 1 步，初始化 s-上近似超图 $H^{\vee_s}$ 的超边集合 $E^{\vee_s}$。

由算法第 2-4 步，首先对超边 e_1 进行 E-距离的计算，有 $D_{HE}(V,e_1)=|v_1-v_2|^2=1$。然后遍历除 v_1，v_2 以外的其余节点 v，计算 $D_{HE}(V,e\cup\{v\})$，分别为 5，13，25，41，61，均大于 s，因此不存在满足条件的点 v，即 $e_1^{\vee_1}=e_1$。由类似过程可得，$e_2^{\vee_1}=\{v_6,v_7\}$，$e_3^{\vee_1}=\{v_1,v_2,v_4,v_5,v_6,v_7\}$，$e_4^{\vee_1}=\{v_1,v_4,v_5,v_6,v_7\}$，$e_5^{\vee_1}=\{v_2,v_5\}$。

$H^{\vee_1}$ 如图 3 所示。

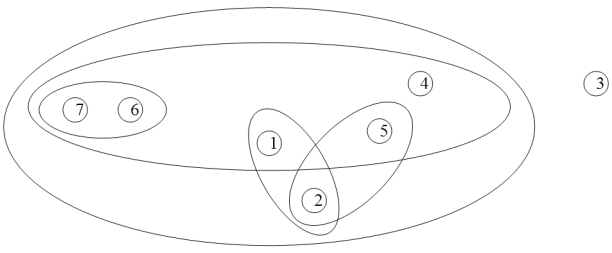

图 3 $H(IS)$的上近似超图

Fig. 3 Upper approximate hypergraph of $H(IS)$

5.1.3 算法对比

在这里选取了两种运用超图进行知识提取方面的优秀近似算法与本文提出的近似超图生成算法(即算法 5)分别从算法的时间复杂度、设计方案及输出规模 3 个方面进行对比分析,对比结果如表 9 所列。

记超图 $H=(V=\{v_1,\cdots,v_n\},E=\{e_1,\cdots,e_m\})$。

表 9 超图上超边的近似算法对比

Table 9 Comparison of approximate algorithms for hyperedges on hypergraph

算法	时间复杂度	设计方案	输出规模
本文	$O(mn^2\log_2 n+mn^2)$	近似超边	$m\times n$
文献[34]	$O(m^3)$	2-近似	$m\times n$
文献[35]	$O(m^2n^3)$	超图正则化	$m\times n$

由表 9 可以得到:

1)本文与文献[34]、文献[35]这 3 种算法都是基于超图理论进行设计的,并且在输出结果的规模上是相同的,这说明本文提出的算法 5 与文献[34]、文献[35]中的算法在超图的近似理论研究方面具有可比性,并且本文算法在输出规模上是常规的,适用范围较广。

2)在时间复杂度方面,本文算法的时间复杂度为 $O(mn^2\log_2 n+mn^2)$,文献[34]中算法的时间复杂度为 $O(m^3)$。可以发现,在超边数量 m 较大的情况下,本文方法要明显优于文献[34]方案,这一点在小群体的社交网络中尤为明显;而在节点数量与超边数量相差不大时,本文算法的时间复杂度与文献[34]基本一致。

文献[35]中算法的时间复杂度为 $O(m^2n^3)$远远高于 $O(mn^2\log_2 n+mn^2)$,故本文算法的时间复杂度明显优于文献[35]的算法。

3)在算法设计方案方面,文献[34]依据 2-近似的方案在超图上提出了一种基于超边的近似方案;而文献[35]是在超图上利用非负张量环分解,在超边上进行近似操作;本文则是利用提出的超图上超边的近似算子,对超图的超边进行近似操作。相比文献[34]和文献[35],本文方案近似操作的实现更为简便和直观。

5.1.4 下近似超图生成算法

根据上近似超图的生成算法,对偶地可得下近似超图生成算法,如算法 6 所示。

算法 6 下近似超图生成算法

输入:超图 H=(V={v_j|j=1,…,n};E={e_i|i=1,…,m}),近似参数 s

输出:s-下近似超图$H^{\wedge_s}=(V=\{v_j|j=1,\cdots,n\};E^{\wedge_s}=\{|i=1,\cdots,m\})$

1. 初始化:$E^{\wedge_s}$
2. for i=1 to m do
3. 通过定义 9 计算e_i的$D_{HE}(V,e_i)$;
4. for j=1 to n do
5. if $v_j\in e_i$
6. 通过定义 9 计算$e_i\cup\{v_j\}$的$D_{HE}(V,e_i\cup\{v_j\})$;
7. if $|D_{HE}(V,e_i\backslash\{v_j\})-D_{HE}(V,e_i)|<s$
8. $e_i=e_i\cup\{v_j\}$;
9. $e_i^{\wedge_s}=e_i$;
10. end if
11. else
12. continue;
13. end if
14. end for
15. $E^{\wedge_s}=(E\backslash\{e_i\})\cup\{e_i^{\wedge_s}\}$;
16. end for
17. Return $H^{\wedge_s}=(V;E^{\wedge_s})$

算法 6 的正确性由定义 7、定义 8 及定理 3 保证。并且由于算法 6 与算法 5 过程对偶,依据算法 5 的算法分析过程类似可得算法 6 的时间复杂度同样为 $O(mn^2\log_2 n)$。其与文献[34]、文献[35]的对比过程也与表 9 类似。

5.2 知识提取方法

在 5.1 节给出超图的上下近似超图生成算法之后,本节基于算法 5、算法 6 及定义给出可定义知识及不可定义知识的提取算法,即分别对应可定义超边提取算法和不可定义超边提取算法。

5.2.1 算法生成过程

算法 7 可定义超边提取算法

输入:超图 H=(V={v_j|j=1,…,n};E={e_i|i=1,…,m}),近似参数 s

输出:可定义超边集合$E_{def}=\{e|e\in E,e^{\vee_s}=e^{\wedge_s}\}$

1. 初始化:E_{def};
2. for i=1 to m do
3. 通过定义 8 计算e_i的$D_{HE}(V,e_i)$;
4. 通过算法 5 计算超边e_i的上近似超边
5. 通过算法 6 计算超边e_i的下近似超边
6. if $e_i^{\vee_s}=e_i^{\vee_s}$
7. $E_{def}=E_{def}\cup\{e_i\}$;
8. else
9. continue;
10. end if
11. end for
12. Return E_{def}

下面对算法 7 的复杂度进行分析。

1)算法 7 的第 1 步为初始化构建可定义超边集合E_{def}。

2)算法 7 的第 3—5 步为对超图 H 中的每一条超边进行操作,得到其上下近似超边。依据算法 5、算法 6 的复杂度分析过程可知,算法 7 的第 3—5 步的复杂度为 $O(mn^2\log_2 n)$。

3)算法 7 的第 6—10 步为判定超边是否为可定义超边的过程,其复杂度为 $O(n)$。

4)算法 7 的第 2—11 步整体为一个循环过程,需对每一条超边进行循环,综合 1)—3)可知,算法 7 的时间复杂度为 $O(m^2n^2\log_2 n+mn)$。

5.2.2　案例实现

下面用第 2 章中案例 1 所生成的超图为例实现算法 7，得到其可定义超边集E_{def}。

根据算法 7，基于图 1 所示超图生成其可定义超边集E_{def}。

令$E_{\text{def}}=\emptyset$，由算法 7 第 3－5 步，首先对超边e_1进行 E-距离的计算，有$D_{\text{HE}}(V,e_1)=|v_1-v_2|^2=1$；然后生成其上近似超边$e_1^{\vee_s}=\{v_1,v_2\}$，再生成其下近似超边$e_1^{\wedge_s}=\{v_1,v_2\}$。

根据算法 7 第 6－11 步进行判别过程，$e_1^{\vee_s}=e_1^{\wedge_s}$，因此将其并入$E_{\text{def}}$中。

遍历除e_1以外的其余超边 e，由类似过程可得$E_{\text{def}}=\{e_1,e_3,e_4,e_5\}$。

事实上，不可定义超边的提取过程可被视为可定义超边提取过程的另一方向。在这里同样给出不可定义超边的提取过程。

算法 8　不可定义超边提取算法

输入：超图 H=(V={v_j|j=1,…,n}；E={e_i|i=1,…,m})，近似参数 s

输出：不可定义超边集合$E_{\text{undef}}=\{e|e\in E,e^{\vee_s}\neq e^{\wedge_s}\}$

1. 初始化：E_{undef}；
2. for i=1 to m do
3.　通过定义 8 计算e_i的$D_{\text{HE}}(V,e_i)$；
4.　通过算法 5 计算超边e_i的上近似超边
5.　通过算法 6 计算超边e_i的下近似超边
6.　if $e_i^{\vee_s}=e_i^{\wedge_s}$
7.　continue；
8.　else
9.　$E_{\text{undef}}=E_{\text{undef}}\cup\{e_i\}$；
10.　end if
11. end for
12. Return E_{undef}

由于算法 8 与算法 7 过程对偶，依据算法 7 的算法分析过程，类似可得算法 8 的时间复杂度同样为 $O(m^2n^2\log_2 n+mn)$。

下面同样用第 2 章中案例 1 所生成的超图为例实现算法 8，得到其不可定义超边集E_{undef}。

根据算法 8，基于图 1 所示超图生成其不可定义超边集E_{undef}。

令$E_{\text{undef}}=\emptyset$，由算法 8 第 3－5 步，首先对超边$e_1$进行 E-距离的计算，有$D_{\text{HE}}(V,e_1)=|v_1-v_2|^2=1$；然后生成其上近似超边$e_1^{\vee_s}=\{v_1,v_2\}$，再生成其下近似超边$e_1^{\wedge_s}=\{v_1,v_2\}$。

根据算法 8 第 6－11 步进行判别过程，因为$e_1^{\vee_s}=e_1^{\wedge_s}$，因此$e_1\notin E_{\text{undef}}$。

遍历除e_1以外的其余超边 e，由类似过程可得$E_{\text{undef}}=\{e_2\}$。

5.2.3　结果对比

为了进一步说明本文方法的有效性和正确性，本文将从与文献[25]的原始聚类结果对比以及选取的生物特征能否体现所考虑的生物样本的共同祖先特征两个方面进行对比分析。

首先，将得到的结果与文献[25]中原始结果，即文献[25]中的图 2 进行对比分析。

1）SPSS 软件是目前生物聚类分析常用的方法，文献[25]说明了这一点。图 4 是对选取的表 1 中的数据按照文献[25]中的方法利用 SPSS 27 进行实现得到的结果，该结果与文献[25]中图 2 关于表 5 中的对象集{u_i，$i=1,\cdots,7$}的相关结果完全一致。

本文的结果在案例 1 中给出。对案例 1 中得到的超边集合 E'（见图 1）进行分析。因为节点u_3（贞琵甲属）没有超边覆盖，所以在这里先不作考虑，留待后续分析。

在有超边覆盖的节点中，u_4（小琵甲属）与u_6（贝琵甲属）为所在超边数量最少的两个节点，均为两条超边{e_1，e_3}，这说明u_4（小琵甲属）、u_6（贝琵甲属）相较于其余的点为特征最为接近的节点。这一点与图 4 中显示的对象u_4（小琵甲属）、u_6（贝琵甲属）均处于聚类结果的从左向右的第一层一致。

其次，如图 1 所示，节点u_1（琵甲属）、u_5（乾琵甲属）、u_7（格琵甲属）为除去与节点u_4（小琵甲属）、u_6（贝琵甲属）共同所在的两条超边外，所在超边数量最少的节点，均为两条包含节点u_4（小琵甲属）、u_6（贝琵甲属）的超边{e_1，e_3}，一条不含节点u_4（小琵甲属）、u_6（贝琵甲属）的超边{e_1}，{e_2}，{e_5}。这说明u_1（琵甲属）、u_5（乾琵甲属）、u_7（格琵甲属）这 3 个节点为与节点u_4（小琵甲属）、u_6（贝琵甲属）的特征最为接近的节点。事实上，因为超边$e_2=\{u_7\}$只包含一个节点，相较于其余两条超边所包含的节点数较少，因此节点u_7（格琵甲属）相较于节点u_1（琵甲属）、u_5（乾琵甲属）而言特征与节点u_4（小琵甲属）、u_6（贝琵甲属）的特征更为接近；而节点u_1（琵甲属）与u_5（乾琵甲属）则无明显差异。这里按照节点倒序进行分析得到的结果与图 4 得到的结果完全一致，即聚类结果的从左向右的第二层说明与对象u_4（小琵甲属）、u_6（贝琵甲属）最接近的对象为 u_7（格琵甲属），聚类结果的从左向右的第三层说明与对象u_4（小琵甲属）、u_6（贝琵甲属）、u_7（格琵甲属）最接近的对象为u_5（乾琵甲属），聚类结果的从左向右的第四层说明与对象u_4（小琵甲属）、u_5（乾琵甲属）、u_6（贝琵甲属）、u_7（格琵甲属）最接近的对象为u_1（琵甲属）。

在图 1 中，剩下的还有超边覆盖的节点为u_2（异琵甲属），这些超边为一条包含节点u_4（小琵甲属）、u_6（贝琵甲属）的超边以及两条不包含节点u_4（小琵甲属）、u_6（贝琵甲属）的超边。这说明节点u_2（异琵甲属）与节点u_4（小琵甲属）、u_6（贝琵甲属）相较于其余节点为与节点u_4（小琵甲属）、u_6（贝琵甲属）特征最为接近的节点，也就是说，异琵甲属与小琵甲属、乾琵甲属、贝琵甲属、格琵甲属和琵甲属具有更为接近的祖先特征。

这一点与图 4 所示的从左向右的第五层的聚类结果（即u_3（贞琵甲属）与u_4（小琵甲属）、u_6（贝琵甲属）、u_7（格琵甲属）、u_5（乾琵甲属）、u_1（琵甲属）具有更接近的祖先特征）不一致。

图 1 中余下的节点u_3（贞琵甲属）在表 1 的背景下为与节点u_4（小琵甲属）、u_6（贝琵甲属）特征相差最大的节点。尽管如此，由于所选案例的实际背景，节点u_3（贞琵甲属）与其余全部节点仍归属于同一族内，因此此时由图 1 得到的结论与图 4 显示的聚类结果的第六层（即所有节点均归于同一族内）一致。

通过上面的对比可以发现，本文得到的结果与文献[25]中得到的结果有 5 层是一致的，故本文方法的准确率为

5/6=83%。但事实上,因为 SPSS 是一个统计软件,其结果会带有一些误差,因此本文方法的准确率至少是 83%。

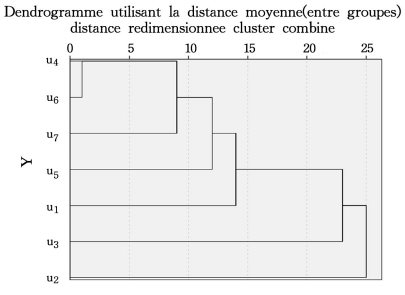

图 4 基于表 1 中特征的对象聚类图

Fig. 4 Object clustering diagram based on features in Table 1

(1)此外,本文后续提出的近似超图的相关算法在表 1 的数据背景下得到的可定义超边集合 $E_{def}=\{e_1,e_3,e_4,e_5\}$ 以及不可定义超边集合 $E_{undef}=\{e_2\}$ 其实是对这一聚类过程进行分析,揭示了聚类过程中不同的特征对聚类结果的影响。对于表 1 中的数据,以生物特征为聚类对象,利用 SPSS27 软件得到的结果如图 5 所示。从图 5 可以发现,特征 a_1(即为特征贮液囊呈卵形)和特征 a_5(即为贮液囊外表具稀疏花纹)聚类分析结果的从左向右的第一层中起到了最重要的作用,因为这两个特征相同的个体数是最多的。其次,特征 a_3(即为贮液囊末端超过第Ⅳ可见腹板基部)和特征 a_4(即为贮液囊壁厚)则是在聚类分析结果的从左向右的第二层到第四层起到了最重要的作用。相比之下,特征 a_2(即为囊体着生于第Ⅴ可见腹板近基部 2/3 之前)在聚类分析结果中的体现仅仅出现在第六层之中。这与本文所得的结果相互印证,说明了本文提出的关于近似超图算法的有效性。

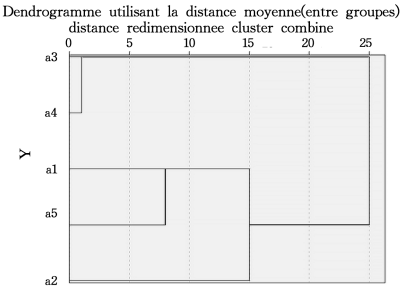

图 5 表 1 中特征的聚类图

Fig. 5 Clustering diagram of features in Table 1

5.3 基于超图的知识提取方法

本节总结了算法 1—算法 8 这 8 种算法,提出了一种由原始信息系统直接得到可定义知识及潜在知识的方法,如图 6 所示。

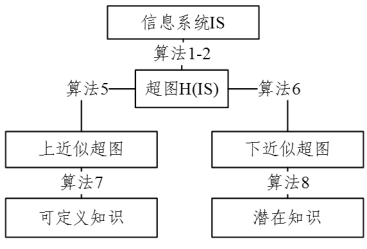

图 6 知识提取框架

Fig. 6 Framework of knowledge extraction

5.3.1 算法生成过程

图 5 对应的知识提取方法的具体过程见算法 9。

算法 9 基于超图的知识提取算法

输入:信息系统 $IS=(U=\{u_j|j=1,\cdots,n\},A=\{a_i|i=1,\cdots,m\},Z,f)$,近似参数 s

输出:可定义超边集合 $E_{def}=\{e|e\in E,e^{\vee_s}=e^{\wedge_s}\}$

不可定义超边集合 $E_{undef}=\{e|e\in E,e^{\vee_s}\neq e^{\wedge_s}\}$

1. 初始化:E_{def},E_{undef};
2. 通过算法 1 将信息系统 IS 转化为超图 H(IS);
3. 通过算法 5 得到超图 H(IS)的上近似超图 $H(IS)^{\vee_s}$;
4. 通过算法 6 得到超图 H(IS)的下近似超图 $H(IS)^{\wedge_s}$;
5. 通过算法 5 及算法 7 得到可定义超边集合 E_{def};
6. 通过算法 6 及算法 8 得到不可定义超边集合 E_{undef};
7. Return E_{def},E_{undef}

由于算法 9 是对全文算法的总结,由算法 1—算法 8 的分析过程可知,算法 9 的时间复杂度为:

$$O(n+m^2n)+O(mn^2\log_2 n)+O(m^2n^2\log_2 n+mn)=$$
$$O(m^2n^2\log_2 n+m^2n+mn^2\log_2 n+mn+n)$$

5.3.2 方法对比

将第 1 章中提及的在知识提取进展方面的 4 个具有代表性的成果[2-5]与本文方法进行对比,从中可以发现本文方法(即算法 9)的一些优势,对比结果如表 10 所列。

表 10 知识提取算法对比

Table 10 Comparison of knowledge extraction algorithms

方法	可视化提取	潜在知识提取	可定义知识提取	算法实现
本文	√	√	√	√
文献[2]	×	√	√	√
文献[3]	√	×	√	√
文献[4]	×	√	√	√
文献[5]	√	×	√	√

注:√表示方法具有这一性质;×表示方法不具有这一性质。

由表 10 可以得出以下结论:

1)5 种方法都有相应的算法实现过程,但是由表 10 可知,除本文方法以外,不同的方法在知识提取的范围上各有千秋,因此没有必要进行算法复杂度的对比,只需确定相应方法是否具有算法实现的性质即可。

2)文献[2]利用粗糙集作为知识提取的工具,首先给出了可定义知识及潜在知识的定义,并对两类知识的提取给出了对应的算法设计,但是并未实现知识提取的可视化。相比之下,本文方法在可视化方面具有优势。

3)文献[3]利用超图作为工具,在一类信息系统——形式背景下进行知识提取并进行了属性约简,提取到了可定义知识,并且实现了知识提取的可视化,但是在知识提取的范围上却不够广,对于潜在知识的提取并未涉及。相比之下,本文方法不仅具有可视性,而且能够完成可定义知识及潜在知识的提取,因此在知识提取范围上有一定优势。

4)文献[4]利用粗糙集和三支决策理论来进行知识提取,实现了对可定义知识及潜在知识的提取,但是并未做到知识提取的可视化。相比之下,本文方法在可视化方面具有一定优势。

5)文献[5]利用知识图谱进行可视化知识提取,并且将其

推广至大数据模型之中，实现了知识提取的可视化，但是却并未涉及潜在知识的提取。相比之下，本文方法不仅与文献[5]一样具有良好的可视性，而且在知识提取范围上较文献[5]具有一定优势。

6)综合1)－5)，本文提出的知识提取方法实现了具有可视性的知识提取，并且在提取范围上兼顾了可定义知识和潜在知识的提取，拓广了知识提取的范围。

结束语　本文针对如何通过可视化的方法对信息系统提取可定义知识及潜在知识这一问题，给出了不同种类的信息系统与超图之间相互转化的方法。

事实上，因为现有的知识提取过程大多是基于IS进行的，相对来说较为抽象，因此需要引入更为直观的图示法，让更多的研究人员参与其中。另外，根据本文算法复杂度分析的结论，IS与超图的转换，其复杂度也都是多项式级的，在可接受的范围内。虽然转换本身会占用一点时间，似乎会给研究速度带来一些不利影响，然而，鉴于其相关研究成果的使用人群数量的增加和成果应用范围的扩展，利用超图进行IS相关研究利大于弊，因此其仍然是一项值得去做的工作。

进一步地，给出超图上E-距离的定义，从而刻画了近似超图的框架。通过近似超图来提取可定义知识及潜在知识，并且从理论上分析了所提算法的正确性及效率。另外，本文使用MS算法完成了$Z\to\{0,1\}$的转化。事实上，用户可以根据自己的喜好，用任何一种由$Z\to\{0,1\}$的转化方式或算法完成这一过程，算法2和算法4的其余过程保持不变，完全可以实现提取过程。在定义7中，E-距离也可以用其他方式加以定义，不必都取为$|v_i-v_j|$，例如可以通过对每对顶点之间给予权值的方式定义，或者通过对每个顶点赋予权值，$|v_i-v_j|$表示两点之间权值之差等等。

总的来说，根据实际背景的不同，研究人员可以选择其他不同于E-距离的距离定义来替换此处的E-距离，作为研究中距离的定义。替换之后，本文其他研究方法的框架不变，也就是说，研究人员可以根据自己研究问题所需使用本文的E-距离或者其他喜好的距离定义方式，结合本文的研究方法和内容，得出自己所需的成果。事实上，根据用户的需求替换E-距离的定义，其余内容不变，本文的其余结论仍然成立。

尽管本文提出的算法完成了潜在知识的提取过程，但是定义的E-距离是否适用于全部的实际场景仍有待验证。未来的研究工作主要有以下几个方向：

1)在大数据时代，实际问题的论域规模和属性规模较之以往更大，因此，如何用本文方法在大数据时代下实现知识的有效提取，有待进一步深入研究。

2)由定义7和定义8以及推论5可知，当信息系统中的论域规模和属性规模较大时，得到E-距离的复杂度会很高，这会导致提取知识的复杂度进一步提升。如何改变这一状态是未来工作的方向之一。

3)三支决策[36]是近年来提出的一种更符合人类思维方式的理论。如何将三支决策理论与超图理论进行结合，分析三支超图的可行性及相关性质，并探究三支超图在知识提取领域中与粗糙集等理论的区别与联系也是未来研究工作的另一方向。

参 考 文 献

[1] PAWLAK Z. RoughSets: Theoretical Aspects of Reasoning about Data[M]. Dordrecht: Kluwer Academic Publishers, 1991.

[2] PAWLAK Z. Rough Sets[J]. International Journal of Computer & Information Sciences, 1982, 11: 341-356.

[3] MAO H, WANG S Y, LIU C, et al. Hypergraph-Based Attribute Reduction of Formal Contexts in Rough Sets[J]. Expert Systems with Applications, 2023, 234: 121062.

[4] YAO Y Y, YANG J. Granular Fuzzy Sets and Three-way Approx-imations of Fuzzy Sets[J]. International Journal of Approximate Reasoning, 2023, 161: 109003.

[5] JARADEH M Y, SINGH K, STOCKER M, et al. Information Extraction Pipelines for Knowledge Graphs[J]. Knowledge and Information Systems, 2023, 65(5): 1989-2016.

[6] BRETTO A. Hypergraph Theory: an Introduction[M]. Cham: Springer, 2013.

[7] VINAS R, JOSHI C K, GEORGIEV D, et al. Hypergraph Factorization for Multi-tissue Gene Expression Imputation[J]. Nature Machine Intelligence, 2023, 5(7): 739-753.

[8] PRAJNANASWAROOPA S, GEETHA J, SOMASUNDARAM K. Total Chromatic Number for Some Classes of Cayley Graphs [J]. Soft Computing, 2023: 1-9.

[9] DALCENGIO S, LECOMTE V, POLETTINI M. Geometry of Nonequilibrium Reaction Networks [J]. Physical Review X, 2023, 13(2): 021040.

[10] BERGE C. Graphs and Hypergraphs[M]. English translation, Amsterdam: North-Holland Publishing Company, 1973.

[11] CHVATAL V. Hypergraphs and Ramseyian Theorems[J]. Proceedings of the American Mathematical Society, 1971, 27(3): 434-440.

[12] LOVASZ L. Normal Hypergraphs and The Perfect Graph Conjecture[J]. Discrete Mathematics, 1972, 2(3): 253-267.

[13] ERDOS P, LOVASZ L. Problems and Results on 3-Chromatic Hypergraphs and Some Related Questions[J]. Infinite and Finite Sets, 1975, 10(2): 609-627.

[14] OWRANG O M M, MILLER L L. Query Translation in a Heterogeneous Distributed Database Based on Hypergraph Models [C]// Proceedings of the 1986 ACM Fourteenth Annual Conference on Computer Science. New York: Association for Computing Machinery, 1986: 412.

[15] GOLDSTEIN A J. Database Systems: A Directed Hypergraph Database: A Model for The Local Loop Telephone Plant[J]. Bell System Technical Journal, 1982, 61(9): 2529-2554.

[16] SACCA D. Closures of Database Hypergraphs[J]. Journal of the ACM, 1985, 32(4): 774-803.

[17] LANDE D, FU M, GUO W, et al. Link Prediction of Scientific Collaboration Networks Based on Information Retrieval [J]. World Wide Web, 2020, 23: 2239-2257.

[18] ZHANG F, YUAN N J, LIAN D, et al. Collaborative Knowledge Base Embedding for Recommender Systems[C]// Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. New York: Association for Compu-

ting Machinery, 2016: 353-362.
[19] JI S, FENG Y, JI R, et al. Dual Channel Hypergraph Collaborative Filtering[C]// Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. New York: Association for Computing Machinery, 2020: 2020-2029.
[20] YANG B, MITCHELL T. Leveraging Knowledge Bases in LSTMs for Improving Machine Reading[C]// Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). Vancouver: Association for Computational Linguistics, 2017: 1436-1446.
[21] RITAL S, CHERIFI H, MIGUET S. Weighted Adaptive Neighborhood Hypergraph Partitioning for Image Segmentation[C]// Pattern Recognition and Image Analysis: Third International Conference on Advances in Pattern Recognition (ICAPR 2005). Berlin Heidelberg: Springer, 2005: 522-531.
[22] TAN S L, GUAN Z Y, CAI D, et al. Mapping Users Across Networks by Manifold Alignment on Hypergraph[C]// Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence. California: AAAI Press, 2014: 159-165.
[23] ZHOU H Y, ZHANG D Q. Multi-site Hypergraph Convolutional Neural Networks and Application[J]. Computer Science, 2022, 49(3): 129-133.
[24] CUI B J, ZHANG Y P, WANG B. Multimodal Data Fusion Algorithm Based on Hypergraph Regularization[J]. Computer Science, 2023, 50(6): 167-174.
[25] LIU C, REN G D. Phylogenetic Analysis of Genera of the Tribe Blaptini Based on The Characteristics of Defensive Glands (Coleoptera: Tenebrionidae)[J]. Acta Entomologica Sinica, 2012, 55(10): 1205-1220.
[26] GRATZER G. Lattice Theory: First Concepts and Distributive Lattices[M]. New York: Dover Publications, 2009.
[27] LIANG J Y, QU K S, XU Z B. Reduction of Attribute in Information Systems[J]. Systems Engineering-Theory & Practice, 2001, 21(12): 76-80.
[28] GONG X, HIGHAM D J, ZYGALAKIS K. Generative Hypergraph Models and Spectral Embedding[J]. Scientific Reports, 2023, 13(1): 540.
[29] YOU L, JIANG H, HU J, et al. GPU-accelerated Faster Mean Shift with Euclidean Distance Metrics[C]// 2022 IEEE 46th Annual Computers, Software, and Applications Conference (COMPSAC). Los Alamitos: IEEE, 2022: 211-216.
[30] ZHAO L. A Theory of Spatial Granular Computing[D]. Regina: University of Regina, 2023.
[31] GAO Y, FENG Y, JI S, et al. HGNN+: General Hypergraph Neural Networks[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2022, 45(3): 3181-3199.
[32] HUANG J, HUANG X, YANG J. Residual Enhanced Multi-Hypergraph Neural Network[C]// 2021 IEEE International Conference on Image Processing (ICIP). Anchorage: IEEE, 2021: 3657-3661.
[33] CHEN L H. Research on Modeling and Representation of Heterogeneous Hypergraphs for Academic Recommendations[D]. Jilin: Jilin University, 2023.
[34] VELDT N. Optimal LP Rounding and Linear-Time Approximation Algorithms for Clustering Edge-Colored Hypergraphs[C]// Proceedings of the 40th International Conference on Machine Learning. New York: PMLR, 2023: 34924-34951.
[35] ZHAO X, YU Y, ZHOU G, et al. Fast Hypergraph Regularized Nonnegative Tensor Ring Decomposition Based on Low-rank Approximation[J]. Applied Intelligence, 2022, 52(15): 17684-17707.
[36] YAO Y Y. Three-way Decision: an Interpretation of Rules in Rough Set Theory[C]// Rough Sets and Knowledge Technology: 4th International Conference, RSKT 2009, Gold Coast, Australia, July 14-16, 2009. Proceedings 4. Berlin Heidelberg: Springer, 2009: 642-649.

LIU Chuan, born in 1997, master candidate. His main research interests include hypergraph and formal concept analysis.

MAO Hua, born in 1963, Ph.D, professor. Her main research interests include rough sets, formal concept analysis and hypergraph.

(责任编辑:杨雪敏)