一种基于层次聚类的子系统划分方法研究

朱 锐[1] 廖鸿志[1,2] 李 彤[1,2] 代 飞[1,2] 王一荃[1] 莫 启[1] 林雷蕾[1]
(云南大学软件学院 昆明 650091)[1] (云南省软件工程重点实验室(云南大学) 昆明 650091)[2]

摘 要 信息领域中常常会涉及到子系统的划分问题,而U/C矩阵法是信息系统划分的一种常用方法,但是系统的复杂性以及人为的参与常常导致子系统划分产生低效率、不确定性以及错误划分等问题。因此深入剖析了系统与子系统、子系统与功能、功能与数据等之间的关系与性质,通过对U/C矩阵按照功能相似度进行层次聚类,并引入结构熵和 H_{pal} 熵来对聚类形成的子系统进行度量,给出了具体的计算公式,提出了一种新的划分子系统的方法,将原本需要人为参与的事情转变为通过计算来完成。同时,实现了一个原型系统来对所提出的方法进行验证,并给出了具体实例。

关键词 子系统划分,层次聚类,结构熵,U/C矩阵,功能相似度
中图法分类号 TP311 **文献标识码** A **DOI** 10.11896/j.issn.1002-137X.2015.12.025

Approach to Subdividing Systems Based on Hierarchical Clustering

ZHU Rui[1] LIAO Hong-zhi[1,2] LI Tong[1,2] DAI Fei[1,2] WANG Yi-quan[1] MO Qi[1] LIN Lei-lei[1]
(School of Software, Yunnan University, Kunming 650091, China)[1]
(Key Laboratory in Software Engineering of Yunnan Province, Yunnan University, Kunming 650091, China)[2]

Abstract System subdivision always is involved in the information technology domain, and a commonly used method is the U/C Matrix. However, the system complexity and the factor of people will cause some critical problems, such as inefficiency, uncertainty and mistakenly division. Consequently, the relationship between system and subsystems, subsystems and functions, functions and data, as well as properties of these relationships were discussed. The subsystem hierarchically clustered according to the simulation of functions was measured by structure entropy and H_{pal} entropy, and a series of computational formula were given. This new way of system subdivision changes the things finished by people to computation. Meanwhile, a prototype system was accomplished and a case study was analyzed to verify the theory.

Keywords System subdivision, Hierarchical clustering, Structure entropy, U/C matrix, Function similarity

1 引言

子系统划分在信息领域中是非常重要及复杂的。子系统划分常常应用于软件工程领域,当采用结构化方法时,往往需要根据需求,对功能模块进行子系统划分。在工程领域,子系统划分常用的方法为U/C矩阵法。目前,在研究领域,对子系统的划分往往集中在如何利用计算机来模拟U/C矩阵法的过程,比如文献[1]和文献[2]对UC矩阵建立子系统的步骤进行模拟与优化,而对UC矩阵的研究都没有涉及到子系统划分的本质——如何进行划分?依靠什么原则划分?陈智高等人[3]提出一种元素填入式的MIS子系统划分方法,该方法由确定元素、计算元素耦合度、不同子系统个数的元素划分、不同划分结果的评判、子系统个数与元素划分方案的选定等5个步骤组成。文献[4]提出了一种基于模糊聚类的MIS子系统划分的方法,该方法是基于模糊等价关系以及模糊相似关系"最大树"的。这些方法只介绍了一种方法而没有进行实例验证。同时U/C矩阵方法还被应用于数据库[5,6],这些方法尝试利用关系数据库实现U/C矩阵的方法,并对其存储、正确性检验、表上作业等进行了分析,同时利用结果关系进行了子系统划分。但其同样也只是说明了U/C矩阵应用面的广泛,没有从理论的角度进行系统的分析。通过这些文献可以看出,对子系统划分的研究主要集中于两个方面:1)应用,即对新系统进行划分;2)改进,即对已经存在的系统进行度量与评价,并进行重组。

本文认为U/C矩阵是一个蕴含了系统与子系统间较多信息的优秀的数学模型。传统的方法是通过人为的方式来观

到稿日期:2015-02-10 返修日期:2015-03-28 本文受国家自然科学基金资助项目(61379032,61262024,61262025,61462091,61462095),云南省自然科学青年基金(2014FD006),云南省教育厅科研重点项目(2013Z057),云南省软件工程重点实验室开放基金(2012SE401),云南省科技厅面上项目(2012FB119),云南大学研究生科研课题项目(ynuy201375,ynuy201425),云南省博士研究生学术新人奖资助。

朱 锐(1987-),男,博士生,CCF学生会员,主要研究领域为软件工程、形式化方法,E-mail:ray0209055@gmail.com;廖鸿志(1946-),男,教授,主要研究领域为系统科学;李 彤(1963-),男,教授,博士生导师,CCF高级会员,主要研究领域为软件工程、形式化方法,E-mail:tli@ynu.edu.cn(通信作者);代 飞 男,讲师,主要研究领域为软件过程;王一荃 女,助理研究员,主要研究领域为软件过程;莫 启 男,博士生,主要研究领域为软件过程;林雷蕾 男,硕士生,主要研究领域为软件过程。

察和调整矩阵，这种方式存在有以下缺点：1）低效率；2）调整结果根据人员不同而不同；3）无法进行定量的描述；4）正确性无法保证。根据上述原因，本文考虑采用数据挖掘中聚类分析以及结构熵的概念，根据已经建立好的 U/C 矩阵，基于矩阵中功能间结构的相似性进行层次聚合，聚合完毕后针对每种聚合结果，再根据结构熵来对形成的划分的内聚度进行度量，最终取出内聚度高的划分作为子系统的划分。通过该方法可以有效地进行定量计算与分析，即采用数学的方法来进行划分，而非通过人为的方式来操作和变化矩阵，这样就降低了人为划分的不确定性，为子系统的划分提供了一定的数学基础。

本文第 1 节简要介绍了当前对于子系统划分的一些研究成果与不足；第 2 节为相关研究基础，简单介绍了 U/C 矩阵，并对聚类方法做了简要总结；第 3 节主要介绍了本文所要使用的一些基础概念与凝聚的层次聚类方法；第 4 节为本文核心，给出了子系统划分的算法流程，讨论了如何对相似性进行度量，如何利用相似性矩阵进行聚类，最后再进行度量；第 5 节为案例研究，给出了一个具体案例，以及该案例使用传统方法的解决结果与通过本方法聚类的结果，并做了简要对比；最后对本文进行总结与展望。

2 相关研究基础

2.1 U/C 矩阵

U/C 矩阵是 MIS 开发中用于系统分析阶段的一个重要工具。“企业系统规划方法”和“信息工程”都推荐建立表示数据类和过程之间的 U/C 矩阵(M)。企业系统规划(business system planning)认为数据类和过程是定义企业信息系统总体结构的基础，应该建立它们之间的内在联系，并可清除在考虑定义和内容时所产生的问题。过程/数据类矩阵(M)是建立二者联系的工具。其中列表示数据，行表示过程(功能)，并以字母 C 和 U 来表示过程(功能)对数据类的产生和使用。在矩阵中，首先按关键资源的生命周期顺序放置过程，开始是计划过程，然后是度量和控制过程，以及直接涉及产品的过程，最后是管理支持资源的过程；其次是根据过程产生数据的顺序来安排数据，开始是由计划过程产生的数据，接着将其所有数据列入矩阵，并在适当的行列交叉处填上 C 和 U。矩阵(M)按照一定的规则进行调整后，可以给出划分系统的子系统方案，并可确定每个子系统相关的共享数据库和专业(私有)数据库，同时也可了解子系统之间的数据通信。根据其对数据类的产生和使用特点，可将子系统分类如下：产生数据类但不使用其他数据类的子系统；使用其他数据类来产生一个数据类的子系统；使用数据类但不产生数据类的子系统。

U/C 矩阵是一张表格，它可以表示数据/功能系统化分析的结果。它的左边第一列列出系统中各功能的名称，上面第一行列出系统中各数据类的名称。表中在各功能与数据类的交叉处，填写功能与数据类的关系。U/C 矩阵的正确性，可由 3 方面来检验：

(1)完备性检验。这是指每一个数据类必须有一个产生者(即“C”)和至少一个使用者(即“U”)；每个功能必须产生或者使用数据类。否则这个 U/C 矩阵是不完备的。

(2)一致性检验。这是指每一个数据类仅有一个产生者，即在矩阵中每个数据类只有一个“C”。如果有多个产生者的情况出现，则会产生数据不一致的现象。

(3)无冗余性检验。这是指每一行或每一列必须有“U”或“C”，即不允许有空行空列。若存在空行空列，则说明该功能或数据的划分是没有必要的、冗余的。

利用 U/C 矩阵方法划分子系统的步骤如下：

(1)用表的行和列分别记录下企业住处系统的数据类和过程。表中功能与数据类交叉点上的符号 C 表示这类数据由相应功能产生，U 表示这类功能使用相应的数据类。

(2)对表做重新排列，按功能组排列。然后调换“数据类”的横向位置，使得矩阵中 C 最靠近对角线。

(3)将 U 和 C 最密集的地方框起来，并为框起名，就构成了子系统。落在框外的 U 说明了子系统之间的数据流。这样就完成了划分系统的工作。

2.2 基本聚类方法概述

聚类就是将物理或抽象的对象，按照对象间的相似性进行区分和分类的过程[7]。聚类分析简称聚类，是一个把数据对象划分成子集的过程。每个子集是一个簇，使得簇中的对象彼此相似，但与其它簇中的对象不相似。由聚类分析产生的簇的集合称为一个聚类[8]。在相同的数据集上，不同的聚类方法可能产生不同的聚类。划分是通过聚类算法进行而非人为的。聚类分析广泛用于许多应用领域，如商务智能、图像模式识别、Web 搜索、生物学和安全。在商务智能应用中，聚类可以用来把大量客户分组，其中组内的客户具有非常类似的特征[8]。常见的聚类方法可以分为基于划分的方法、基于分层的方法、基于密度的方法和基于网格的方法，其中基于划分的聚类算法在模式识别里是最常用的聚类算法类型[7]。

(1)划分方法：给定一个 n 个对象的集合，划分方法构建数据的 k 个分区，其中每个分区表示一个簇，并且 $k\leqslant n$。最常见的方法是 K 均值法，但是该方法需要指定 K，即对于子系统划分而言，要指定子系统的个数，故未采用。

(2)层次方法：层次方法创建给定数据对象间的层次分解，分为凝聚和分裂。通过一层层合并或分解可以形成层次化的聚类结果。该方法易于操作和分析，对子系统分析较为合适。

(3)基于密度的方法：大部分划分方法基于对象之间的距离进行聚类。该方法只能发现球状簇，未被选用是因为 U/C 矩阵的对象之间离散性强，不易衡量距离。

(4)基于网格的方法：基于网格的方法把对象空间量化为有限个单元，形成一个网络结构。该方法所需规模较大，即 n 的个数较大，但是 U/C 矩阵的规模不可能太大，并且不易给出坐标，所以也未采用。

综上，本文即考虑使用层次化的聚类方法进行子系统的划分。

3 基本概念与性质

定义 1(系统)[9] 系统 $S=\langle A,R\rangle$，其中 A 为系统中全部元素构成的集合，R 为元素之间关系的集合。

定义 2(子系统)[9] 子系统 S_i 为系统 S 的子系统，当且仅当 $S_i\subset S$，且 S_i 也是系统。

定义 3(划分)[10] 设 A 为非空集合,若存在 A 的一个子集族 θ 满足:(1)$\varnothing\notin\theta$;(2)$\forall x,y\in\theta$ 且 $x\neq y$,则 $x\cap y=\varnothing$;(3)$\bigcup\theta=A$,则称 θ 为 A 的一个划分。θ 中元素称为划分块。例如系统 S 包含的元素集合为 $\{a,b,c,d,e\}$,那么可能存在的划分 $\theta=\{x,y\}$,$x=\{a,b\}$,$y=\{c,d,e\}$,则 x,y 为划分 θ 的划分块。

U/C 矩阵实际就是对子系统的一种划分,只因为人为的因素过多,因此会存在一定的人为干扰而导致不准确、不全面。对于求子系统的全部划分问题,其实际用到的数学模型为:将 $n(n\geqslant 1)$ 个不同的球放入 r 个相同的盒中去,并且要求无空盒,问有多少种不同的放法?这里要求 $n\geqslant r$。解决该问题主要使用组合数学中的第二类 Stirling 数[10]。随着 n 的增长,其划分的数量以指数级增长。举个简单例子,把 n 个对象放到 2 个互斥的盒子里的数目就有 $2^{n-1}-1$ 种。由于数量太大,如果能够对如此大数量的划分进行遍历来衡量哪种划分最好,当然是最准确的。但是由于数量过于巨大,当系统中功能数量太大后,就无法采用穷举法进行遍历,因此必须要考虑一种合适的划分方法来进行子系统划分。

定义 4(UC 矩阵) $ucMatrix=\begin{bmatrix} & D_j \\ F_i & v_{ij}\end{bmatrix}$,$i\in[1,m]$,$j\in[1,n]$;$i,j\in\mathrm{N}$。其中,$F_i$ 表示功能,D_j 表示数据,$v_{ij}\in\{1,0,-1\}$。为了在程序中易于计算,用 1 代表 U,-1 代表 C,0 代表空 Null。

定义 5(中间矩阵) $matrix=\begin{bmatrix} & D_j \\ F_i & v_{ij}\end{bmatrix}$,$i\in[1,m]$,$j\in[1,n]$;$i,j\in\mathrm{N}$。其中,$F_i$ 表示功能,D_j 表示数据,$v_{ij}\in\{-1,0,1,2\}$。为了在程序中易于计算,用 1 代表 U,-1 代表 C,0 代表空 Null,2 代表功能 F_i 对 D_j 既依赖又创建,这种情况在 $UCMatrix$ 中不会存在,但是在层次合并产生的中间矩阵中会出现。

定义 6(结构熵)[11] 如果一个系统 X 有 n 个子系统 $X_1,X_2,\cdots,X_n$,子系统 X_i 与 X_j 有关联,则 $g_{ij}=1$,否则 $g_{ij}=0$。令 $N(i)=\sum_{j=1}^{n}g_{ij}$,连接强度 $\rho(i)=N(i)/\sum_{j=1}^{n}N(j)$。定义系统 X 的结构熵 $H=-\sum_{i=1}^{n}\rho(i)\ln\rho(i)$,其中 $\sum_{i=1}^{n}\rho(i)=1$。

4 子系统划分

4.1 子系统划分算法流程

本文主要的算法思想是利用聚类方法。本质上来说,聚类算法是一类非监督学习算法。在有监督的学习中,学习的目标是要在两类样本中找出它们的分界,训练数据是给定标签的,要么属于正类,要么属于负类。而非监督学习,它的目的是在一个没有标签的数据集中找出这个数据集的结构并把它自动聚成两类或者多类。层次聚类算法分为凝聚的和分裂的。凝聚的层次聚类算法 AGNES(Agglomerative NESting)主要考虑将每个对象自成一族,然后这些族根据某种准则逐步合并[8]。因此,本文中子系统划分主要涉及到 4 个主要的过程:构建相似性矩阵,进行层次聚类,进行内聚度度量以及求平均内聚度最高的划分。整体的流程如图 1 所示。

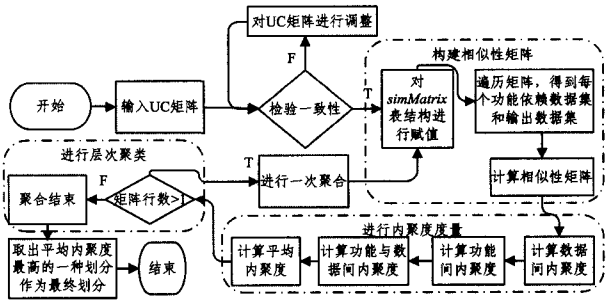

图 1 子系统划分算法流程

4.2 相似性度量

世界是由系统组成的,而系统与系统间存在着普遍的相似性。社会组织间存在着相似性,在社会网络中,如果两位顾客在社会网络中有相似近邻,那么他们是相似性的。自然界中的相似性更加普遍,比如不同品牌的汽车间存在着一定的相似性,这种相似性使得我们将这些事物聚合为一个“汽车”类。因此可以将问题进行抽象,在信息系统中,功能间的相似性可以通过功能对于数据的依赖进行刻画。文献[12]为了解决过程间的并行性问题,而采用“伯恩斯坦条件”的 3 种功能对数据的依赖关系,分别是正依赖(True Dependence)、反依赖(Anti Dependence)以及输出依赖(Output Dependence)。本文认为一个功能依赖的数据主要来自两部分:所依赖的数据和产生的数据。因此两个功能间的数据相似性就有 4 种可能:所依赖数据间的相似性,所产生数据间的相似性,功能 1 所依赖数据与功能 2 所产生数据间的相似性,功能 1 所产生与功能 2 所依赖数据间的相似性。从而,可以对两个功能对数据依赖的相似性进行度量。

定义 7(功能相似度) 当且仅当

$$Sim(F_a,F_b)=\alpha\frac{|FD(a)\cap FD(b)|}{|FD(a)\cup FD(b)|}+\beta\frac{|FC(a)\cap FC(b)|}{|FC(a)\cup FC(b)|}+\gamma\frac{|FD(a)\cap FC(b)|}{|FD(a)\cup FC(b)|}+\delta\frac{|FC(a)\cap FD(b)|}{|FC(a)\cup FD(b)|}\geqslant\varepsilon$$

时,认为 F_a 和 F_b 相似。其中 $\alpha+\beta+\gamma+\delta=1$ 为权值,ε 为阈值,一般 $\alpha=\beta=\gamma=\delta$,即认为同样重要。$FD(F_i)$ 表示功能 F_i 所依赖的数据集合,记为 $FD(F_i)=\{D_j\mid F_i\rightarrow D_j\}$;$FC(F_i)$ 表示功能 F_i 所产生的数据集合,记为 $FC(F_i)=\{D_j\mid F_i\Rightarrow D_j\}$,$a,b,i$ 为自然数。有了功能相似度,就可以考察两个功能是否相似了。假设当前只考虑功能的输入依赖之间的相似性,即设 $\alpha=1,\beta=\gamma=\delta=0$,$FD(1)=\{a,b,c,d,e\}$,$FD(2)=\{a,b,d,e,f\}$,$FD(3)=\{a,c,h\}$,那么根据定义 7 可知 $Sim(F_1,F_2)=4/6\approx 0.66$,$Sim(F_1,F_3)=2/6\approx 0.33$,$Sim(F_2,F_3)=2/7\approx 0.29$。如果 $\varepsilon=0.7$,可知 F_1,F_2,F_3 之间两两不相似;如果 $\varepsilon=0.5$,可知 F_1 和 F_2 相似,其它不相似;如果 $\varepsilon=0.3$,可知 F_1 和 F_2 相似,F_1 和 F_3 相似其它不相似;如果 $\varepsilon=0.2$,可知 F_1,F_2,F_3 之间两两相似。功能间的相似性可以用来衡量两个功能间的相似程度。因此考虑将功能间相似的功能进行聚类而达到子系统的划分。

定理 1 功能相似度 $Sim(F_a,F_b)\geqslant 0$ 且 $Sim(F_a,F_b)\leqslant 1$。

证明:设 $A=\frac{|FD(a)\cap FD(b)|}{|FD(a)\cup FD(b)|}$,$B=\frac{|FC(a)\cap FC(b)|}{|FC(a)\cup FC(b)|}$,$C=\frac{|FD(a)\cap FC(b)|}{|FD(a)\cup FC(b)|}$,$D=\frac{|FC(a)\cap FD(b)|}{|FC(a)\cup FD(b)|}$,对 A 而言,当

$FD(a)\cap FD(b)=\varnothing$ 时，A 取最小值为 0，当 $FD(a)=FD(b)$ 时，A 取最大值为 1，因此 $0\leqslant A\leqslant 1$；同理 $0\leqslant B,C,D\leqslant 1$。同时有 $Sim(F_a,F_b)=f(A,B,C,D)=\alpha A+\beta B+\gamma C+\delta D$。对 $f(A,B,C,D)$ 而言，其显然是递增的，也就是当 A,B,C,D 均为 1 时取最大值，即 $Sim(F_a,F_b)_{max}=1$；当 A,B,C,D 均为 0 时取最小值，即 $Sim(F_a,F_b)_{min}=0$，故 $0\leqslant Sim(F_a,F_b)\leqslant 1$。证毕。

通过定理 1 可知，功能间的相似度的阈值 ε 越小（越靠近 0），功能越容易两两相似。因为 $Sim(F_a,F_b)\geqslant\varepsilon$，$\varepsilon$ 越接近 0$Sim(F_a,F_b)$的可取值范围就越大。

定义 8（相似性矩阵） n 个功能间的相似性矩阵 $simMatrix=\begin{bmatrix} & F_j \\ F_i & s_{ij}\end{bmatrix}_{n\times n}=\begin{bmatrix} & F_j \\ F_i & Sim(F_i,F_j)\end{bmatrix}_{n\times n}$，其中 $F_i,F_j\in ucMatrix$。

定义 9（功能） 功能 F 为三元组（$funName,DD,CD$），其中 $funName$ 表示功能名，DD 为依赖数据集，CD 表示输出数据集。

算法 1 构建相似性矩阵 *buildTheSimMatrix*(*UCMatrixucMatrix*$_{mn}$)

```
Input:U/C 矩阵 ucMatrix;对应的功能集 Functions={f1,f2,…,fn}
Output:simMatrix
BEGIN
  //对 simMatrix 表结构进行赋值
  For i:=1 To m Begin
    simMatrix0i=fi.funName;
    simMatrixi0=fi.funName;
  End
  //遍历 UC 矩阵,得到每个功能依赖数据集和输出数据集
  For i:=1 To m Begin
    For j:=1 To n Begin
      If 1==ucMatrixij Then Begin
        fi.CD:=fi.CD ∪ j
      End
      Else If -1==ucMatrixij Then Begin
        fi.DD:=fi.DD ∪ j
      End
    End
  End
  //计算相似性矩阵
  For i:=1 To m Begin
    For j:=1 To n Begin
      float A:=|fi.DD∩fj.DD| / |fi.DD∪fj.DD|
      float B:=|fi.CD∩fj.CD| / |fi.CD∪fj.CD|
      float C:=|fi.DD∩fj.CD| / |fi.DD∪fj.CD|
      float D:=|fi.CD∩fj.DD| / |fi.CD∪fj.DD|
      simMatixij:=α*A+β*B+γ*C+δ*D
    End
  End
End
```

定义 10 UC 矩阵的一个划分 *divide* 为二元组，表示为（*matrix*，*simMatrix*），其中 *matrix* 表示中间矩阵（可以为初始的 UC 矩阵），*simMatrix* 表示该矩阵对应的相似性矩阵。

定义 11 UC 矩阵的层次聚类划分结果 $Divides=\{divide_1,divide_2,\cdots,divide_n\}$。

定义 12（最相似的功能对） 最相似的功能对 *simFunctions* 为三元组（*function*，*function*，*value*），其中 $function\in matrix.Functions$ 为功能，值 $value\in\{-1,0,1,2\}$。

根据聚类方法的要求，需要将两个相似的功能对进行合并，层层合并也就形成了聚类，而每层合并的标准就是将最相似的两个功能对根据功能间的相似性进行合并。功能与功能间的数值也应该进行合并，而根据定义 4 和定义 5 可知，最开始为 UC 矩阵，进行一次聚类后就产生了中间矩阵，而使用 1 代表 U，+1 代表 C，0 代表空 Null，2 代表功能 F_i 对 D_j 既依赖又创建。因此就产生了合并的规则，如表 1 所列。如果合并的两个功能的值均为 −1，说明两个功能对某个数据 D 均是创造，那么合并后依然只是创造；同理，两个功能的值均为 1 时，说明都是使用，所以结果还是 1；如果一个是 −1 创造，一个是 1 使用，则叠加后为 2，因为规定了 2 为既依赖又创建。

表 1 功能合并规则

v_1 \ v_2	−1	0	1	2
−1	−1	−1	2	2
0	−1	0	1	2
1	2	1	1	2
2	2	2	2	2

4.3 基于相似性矩阵的层次聚类算法

本文利用相似性矩阵来寻找层次聚类中每次需要合并的最为相似的两个功能，具体的算法见算法 2 和算法 3。算法 2 是将一个中间矩阵（初始时是 UC 矩阵）进行聚合，直到中间矩阵的大小为 1 才停止，每次聚合就会形成原系统的一种划分。算法 3 是进行一次聚合，为算法 2 所调用。

算法 2 基于相似性矩阵的层次聚类算法 *buildDivides*(*Object*[][] *matrix*，*SimMatrix simMatrix*)

```
INPUT:中间矩阵 matrix(可以为初始的 UC 矩阵),matrix 对应的相
      似性矩阵 simMatrix,临时矩阵_matrix,临时相似性矩阵
      _simMatrix。
OUTPUT:层次聚类划分结果 Divides
Begin
  //行数大于 2
  If matrix.length > 2 Then Begin
  //进行一次构建划分
  _matrix:=buildDivide(matrix,simMatrix)
  //跳出迭代
  If _matrix.length <=2 Then Begin
    divide:={result,nil}
    divides:=divides ∪ {divide}
    Exit
  End
  _simMatrix:=buildTheSimMatrix(_matrix);
  divide:={result,_simMatrix}
  divides:=divides ∪ {divide};
  buildDivides(result,_simMatrix)
End
```

算法 3 进行一次构建划分算法 *buildDivide*(*Object*[][] *matrix*,*SimMatrix simMatrix*)

```
INPUT:中间矩阵 matrix(可以为初始的 UC 矩阵),matrix 对应的相
      似性矩阵 simMatrix。
OUTPUT:一次构建划分结果矩阵 result
Begin
  int simRowNum=simMatrix. getRowNum()
  //两行以内都不算
  If simColumnNum < 3 Begin
    Exit
  End
  //两行数据,进行合并
  If simColumnNum ==3 Begin
    result10:=matrix10+"_"+matrix20
      For i:=1 to matrix. columnNum Begin
        result1i:=ucAdd(matrix1i,matrix2i)
      End
    Exit
  End
  //找最大值对应的功能对
  simFunctions:=getMostSimFunctions(simMatrix)
  int a:=simFunctions. function1
  int b:=simFunctions. function2
  //合并行
  result a0:=matrix a0+"_"+ matrix b0;
  For i:=1 to matrix. columnNum Begin
    result1i:=ucAdd(matrix ai,matrix bi)
  End
  For i:=a+1 to simFunctions. b Begin
    For int j:=0 to matrix. columnNum Begin
      result ij:=matrix ij
    End
  End
  For i:=b to matrix. rowNum-1 Begin
    For int j:=0 to matrix. columnNum Begin
      int c:=i+1
        result ij:=matrix cj
    End
  End
End
```

4.4 基于结构熵的系统划分内聚度度量

在进行划分后,另一个核心问题就是如何找出哪种划分最好,如何评价划分的"优"的问题。一般而言,我们认为一个系统其内聚度越高就越"优"。因此,需要对系统的内聚度进行度量。陈等人[13]指出内聚度是指模块内各成分之间的联结强度。在面向对象程序中,内聚度主要是指类内部各成分之间的联结强度。郁等人[14]在类依赖图的基础上,提出了一种基于结构熵的类内聚度度量方法,从类的属性与属性、属性与方法和方法与方法之间的依赖关系 3 个方面对类的内聚度进行度量的方法。一般而言,一个内聚性高的模块应当只完成软件过程中的单一任务,而不与其它模块发生联系[15]。内聚度越高,越容易理解、修改和维护,所以如果要划分子系统,就必须要对其内聚度进行度量。本文借鉴了这些文献的思想,同时引入结构熵的概念,将结构熵和 H_{pal} 熵进行结合,给出了具体内聚度度量的公式,并提出了平均内聚度的概念。熵是系统有序程度的度量。如果网络是随机连接的,各个节点的重要度大致相当,那么认为网络是"无序的"。反之,如果网络是非标度的,网络中有少量"核心节点"和大量"末梢节点",节点的重要度存在差异,则认为这种网络是"有序的"[11]。而在系统划分中,希望网络是无序的,即希望结构熵值较大,这是因为无序的网络分布会较为均匀,减少了孤立节点。而另一方面,本文引入 H_{pal} 熵,通过经典信息熵定义(以结构熵的为例进行讨论,因为本质是一样的,结构熵的定义参见定义 6),可以看出 $N(i)$是不能为 0 的,则连接强度 $\rho(i)$也不能为 0。而事实上,在 UC 矩阵的划分过程中,只要有功能与其他数据或功能没有关系,在计算时 $N(i)$就可能为 0,这就要求必须考虑 $\rho(i)$为 0 的情况。这种情况的解决方案有两种,一种是在计算时对这种情况单独考察,另一种就是引入一种新的计算方案。本文采用了后者,通过本文实验部分也可看出引入 H_{pal} 熵可以有效地避免单独考察引起的不必要的逻辑判断,提高程序的效率。同时 Pal 等人[17]成功地将其应用于图像分割处理;文献[16]使用 H_{pal} 熵来度量系统不确定性的合理性,并进行了理论分析。

定义 13(H_{pal} 熵)[17] $H=\sum_{i=1}^{n} p_i e^{1-p_i}$。

定义 14(指数形式的结构熵) 如果一个系统 X 有 n 个子系统 $X_1,X_2,\cdots,X_n$,子系统 X_i 与 X_j 有关联,则 $g_{ij}=1$,否则 $g_{ij}=0$。令 $N(i)=\sum_{j=1}^{n} g_{ij}$,连接强度 $\rho(i)=N(i)/\sum_{j=1}^{n} N(j)$。定义系统 X 的结构熵 $H=\sum_{i=1}^{n}\rho(i)e^{1-\rho(i)}$,其中 $\sum_{i=1}^{n}\rho(i)=1$。

4.4.1 数据与数据间的依赖

系统中数据间的内聚度主要是指系统中数据间关系的密切程度。对系统中每个数据 $D_i(i=1,2,3,\cdots,m)$,引入一个集合 $DD(D_i)$记录数据 D_i 所依赖的其它数据的集合:$DD(D_i)=\{D_j \mid D_i \rightarrow D_j$,且 $D_i \neq D_j\}$,数据间的联接强度 $\rho(D_i)=\dfrac{|DD(D_i)|}{\sum_{k=1}^{m}|DD(D_k)|}$,则数据间的内聚度公式为:

$$Cohesion(D\_D)=\begin{cases}0, & m=0\\ 1, & m=1\\ \dfrac{1}{m}\sum_{i=1}^{m}\dfrac{|DD(D_i)|}{m-1}\times\dfrac{\sum_{i=1}^{m}\rho(D_i)e^{1-\rho(D_i)}}{e^{1-\frac{1}{m}}}, & m>1\end{cases}$$

其中,m 表示系统中涉及到的数据的个数。当 $m=0$ 时,表示系统中没有数据,那么其内聚度也为 0;当 $m=1$ 时,自身内聚度为 1;当 $m>1$ 时,其中 $\dfrac{|DD(D_i)|}{m-1}$表示 D_i 所依赖的数据的个数占除去自身的其它数据的总个数的比例。$\dfrac{\sum_{i=1}^{m}\rho(D_i)e^{1-\rho(D_i)}}{e^{1-\frac{1}{m}}}$对结构熵进行了归一化处理,这是根据 $E'=\dfrac{E-E_{\min}}{E_{\max}-E_{\min}}$而 H_{pal} 熵的最小值为 0 所得。

4.4.2 功能与功能间的依赖

系统中功能间的内聚度主要是指系统中功能间关系的密切程度。对系统中每个功能 $F_i(i=1,2,3,\cdots,n)$,引入一个集合 $FF(F_i)$记录功能 F_i 所依赖的其它功能集合:$FF(F_i)=$

$\{F_j \mid F_i \rightarrow F_j$，且 $F_i \neq F_j\}$，功能间的联接强度 $\rho(F_i)=\frac{|FF(F_i)|}{\sum_{k=1}^{n}|FF(F_k)|}$，则功能间的内聚度公式为：

$$Cohesion(F\_F)=\begin{cases}0, & n=0\\ 1, & n=1\\ \frac{1}{n}\sum_{i=1}^{n}\frac{|FF(F_i)|}{n-1}\times\frac{\sum_{i=1}^{n}\rho(F_i)e^{1-\rho(F_i)}}{e^{1-\frac{1}{n}}}, & n>1\end{cases}$$

其中，n 表示系统中涉及到的功能的个数，$\frac{|FF(F_i)|}{n-1}$ 表示 F_i 所依赖的功能的个数占除去自身的其它功能的总个数的比例。$\frac{\sum_{i=1}^{n}\rho(F_i)e^{1-\rho(F_i)}}{e^{1-\frac{1}{n}}}$ 对结构熵进行了归一化处理。

4.4.3 功能与数据间的依赖

系统中数据与功能的内聚度主要是指系统中功能对数据的依赖程度。对系统中每个功能 $F_i(i=1,2,3,\cdots,n)$，有 $FD(F_i)=\{D_j \mid F_i \rightarrow D_j\}$，$FD\_Out$ 记录一个功能 F_i 所涉及到的数据集(包括依赖和产生)中与其它功能有关系的数据集。$FD\_Out(F_i)=\{D_j \mid (F_i \rightarrow D_j \vee F_i \Rightarrow D_j) \wedge (\exists k(k\neq i), D_j \in FD(F_k) \vee D_j \in FC(F_k))\}$，功能与数据间的联接强度 $\rho(FD_i)=\frac{|FD(F_i)|}{\sum_{k=1}^{n}|FD(F_k)|}$，则功能与数据间的内聚度计算公式为：

$$Cohesion(F\_D)=\begin{cases}0, & n=0\\ \frac{|FD(F_i)|}{m}, & n=1\\ \frac{1}{n}\sum_{i=1}^{n}\frac{|FD\_OUT(F_i)|}{m}\times\frac{\sum_{i=1}^{n}\rho(F_i)e^{1-\rho(F_i)}}{e^{1-\frac{1}{n}}}, & n>1\end{cases}$$

其中，m 表示子系统中所有数据的个数；n 为所有功能的个数，若 $n=1$，则认为其内聚度由其所依赖的数据数目来决定。$\frac{|FD\_OUT(F_i)|}{m}$ 表示与 F_i 有关的所有数据中与其他功能也有关的数据占总数据的比重。

4.4.4 子系统内聚度

综上，子系统的内聚度可以用公式 $Cohesion(S)=\alpha Cohesion(D\_D)+\beta Cohesion(F\_F)+\gamma Cohesion(F\_D)$ 表示，其中 $\alpha+\beta+\gamma=1$ 为权值。一般情况下，$\alpha=\beta=\gamma$。

4.4.5 划分评价

一个划分下可以有多个子系统，每个子系统都有自己的内聚度，那么就应该对不同划分下的内聚度进行考察。假设系统 T 存在划分 $\overline{X}=\{S_1,S_2,S_3,\cdots,S_t\}$，共有 t 个子系统，那么这种划分下的加权平均内聚度为 $\overline{Cohesion}=\frac{\sum_{i=1}^{t}Cohesion(S_i)*|S_i|}{\sum_{i=1}^{t}|S_i|}$。

5 案例研究

本文采用 Java 语言实现了所提出方法的一个原型系统，通过该原型系统可以快速地将 U/C 矩阵进行聚合，并给出聚合的每个步骤以及相应的平均内聚度。以网络上最为常见的 U/C 教学案例为例，具体的 U/C 矩阵如表 2 所列，通过表上作业法得到的子系统结果如表 3 所列。

表 2 原始 U/C 矩阵

功能 \ 数据	个人信息	教师/考生	教师/考生库	试卷	题库	答案	考生答案库	分数	标准答案库	成绩库	处理结果
登录	U	U	C								
抽题				U	C						
答题	U		U			C					
浏览	U			U		U	U				
修改		U	U			U	C				
交卷				C	U						U
判分						U	U	C	U		
登分	C		U					U		C	
统计		U	U					U	C	U	U
打印		C						U	U	U	U
查询	U	U	U					U			C

表 3 表上作业法调整结果

功能 \ 数据	教师/考生库	题库	答案	考生答案库	试卷	分数	成绩库	标准答案库	教师/考生	处理结果	个人信息
登录	C								U		U
抽题		C			U						
答题	U		C								U
浏览			U	U	U						U
修改	U		U	C					U		
交卷		U			C					U	
判分			U	U		C		U			
登分	U					U	C				U
统计	U					U	U	C	U	U	
打印						U	U	U	C	U	
查询	U					U			U	C	C

本文所提出算法的计算结果如图 2、图 3 所示，其中图 2 表示在 U/C 矩阵聚类的每一步都应该对应一个相似性矩阵，图 3 为最终聚类形成的划分的加权平均内聚度。可以看出，8 号对应的划分的加权平均内聚度为最高的。图 4 为 8 号划分：{{1,3,5,7,8,9,10,11},{2,6},{4}}，对应的各功能名字为{{登录，答题，修改，判分，登分，统计，打印，查询}，{抽题，交卷}，{浏览}}。通过该结果可以发现，划分的结果比表上作业法划分的结果更加有依据，比如：通过该算法“登录”和“抽题”功能被划分开，这是因为“登录”是创建“教师/考生库”和依赖“教师/考生”以及“个人信息”数据，而“抽题功能”却主要是创建“题库”和依赖于“试卷”数据，两个功能几乎不相干，应该划分到不同的子系统中。通过这个简单例子可以看出，表上作业法存在较大的缺陷，同时也证明了本方法的有效性和可靠性。

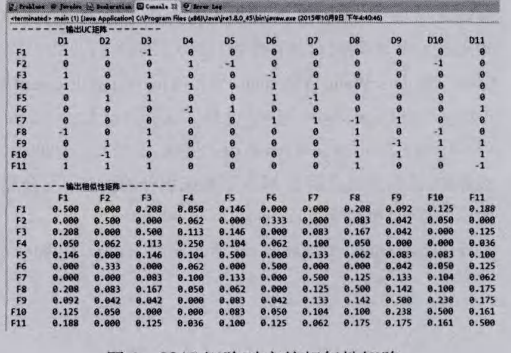

图 2 U/C 矩阵对应的相似性矩阵

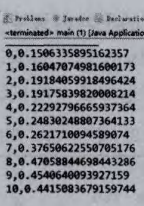

图 3　聚类形成的划分的加权平均内聚度

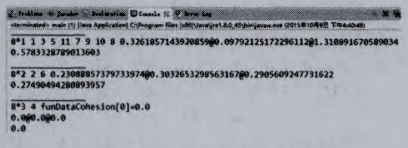

图 4　对应 8 号划分所形成的子系统

结束语　本文提出了一种基于层次聚类的子系统划分方法。该方法首先根据功能间的相似性构建矩阵所对应的相似性矩阵，同时利用层次聚类方法对 U/C 矩阵进行聚合，在聚合过程中对所形成的划分进行度量，度量方法主要是利用 H_{pal} 熵对传统结构熵进行改进，同时借鉴了依赖性分析的一些方法。最后通过实验证明了所提出方法的正确性和有效性。我们认为本方法不仅可用于子系统划分，对于某些图结构的相似性分析也具有一定的参考意义；同时，该方法可以发布为服务，以供企业在系统分析和设计时进行参考。

如同所有理论一样，本方法也存在一定的不足，比如层次凝聚的方法无法进行回溯，因此未来考虑对聚类方法改进；另外，内聚度的度量对于子系统划分极其重要，不同的内聚度计算而得到最终的划分结果也不尽相同。目前还没有一种公认的比较好的内聚度度量方法，因此可以将度量方法封装为构件，如果存在某些更好的度量方法，只需要对度量方案进行替换即可。

参 考 文 献

[1] 张建. U/C 矩阵在信息系统功能划分中的优化与研究[J]. 贵州大学学报(自然科学版)，2008，25(1)：1-5
Zhang Jian. Optimizing Research on U/C Matrix in Information Management System[J]. Journal of Guizhou University (Natural Science)，2008，25(1)：1-5

[2] 任小琦. U/C 矩阵对角线带状聚合的研究[J]. 现代电子技术，2012，34(22)：4-6
Ren Xiao-qi. Research on U/C Matrix Diagonal Strip－shaped Polymerization[J]. Modern Electronics Technique，2012，34(22)：4-6

[3] 陈智高，王延清，牟援朝. 元素填入式 MIS 子系统划分方法[J]. 华东理工大学学报(自然科学版)，2000，26(2)：198-200
Chen Zhi-gao，Wang Yan-qing，Mou Yuan-chao. Element－filling System Subdivision Method[J]. Journal of East China University of Science and Technology，2000，26(2)：198-200

[4] 袁兆山. 模糊聚类应用于 MIS 子系统划分的研究[J]. 合肥工业大学学报(自然科学版)，1995，18(4)：48-52
Yuan Zhao-shan. Research on Applying Fuzzy Clustering on MIS Subsystem Division[J]. Journal of Hefei University of Technology，1995，18(4)：48-52

[5] 牛忠慈. uc 矩阵用于数据库设计[J]. 甘肃科学学报，1998，10(3)：42-44
Niu Zhong-ci. U/C Matrix Used in Database Design[J]. Journal of Gansu Science，1998，10(3)：42-44

[6] 张凤林，皮德常，丁宇红. 关系数据库实现 U/C 矩阵的方法[J]. 计算机应用，2000，20(7)：36-37
Zhang Feng-lin，Pi De-chang，Ding Yu-hong. A Method of Implementing the U/C Matrix on the Basis of Relational Database[J]. Computer Applications，2000，20(7)：36-37

[7] 张敏，于剑. 基于划分的模糊聚类算法[J]. 软件学报，2004，15(6)：858-868
Zhang M，Yu J. Fuzzy partitional clustering algorithms[J]. Journal of Software，2004，15(6)：858-869

[8] Kamber J H M. 数据挖掘概念与技术[M]. 范明，孟小峰，等译. 北京：机械工业出版社，2001

[9] 许国志，顾基发，车宏安. 系统科学[M]. 上海：上海科技教育出版社，2000
Xu Guo－zhi，Gu Ji-fa，Che Hong-an. Systems Science[M]. Shanghai Scientific & Technological Education Press，2000

[10] 耿素云，屈婉玲，王捍贫. 离散数学教程[M]. 北京：北京大学出版社，2002

[11] 谭跃进，吴俊. 网络结构熵及其在非标度网络中的应用[J]. 系统工程理论与实践，2004，24(6)：1-3
Tan Yue-jin，Wu Jun. Network Structure Entropy and Its Application to Scale-free Networks[J]. System Engineering-Theory & Practice，2004，24(6)：1-3

[12] Li T. An Approach to Modelling Software Evolution Processes[M]. Beijing：Tsinghua University Press，2008

[13] 陈振强，徐宝文. 一种基于依赖性分析的类内聚度度量方法[J]. 软件学报，2003，14(11)：1849-1856
Chen Z Q，Xu B W. An approach to measurement of class cohesion based on dependence analysis[J]. Journal of Software，2003，14(11)：1849-1856

[14] 郁湧，唐家华，李文宏，等. 一种基于结构熵的类内聚度度量方法[J]. 系统工程与电子技术，2009，31(3)：702-704
Yu Yong，Tang Jia-hua，Li Wen-hong，et al. Approach to measurement of class cohesion based on structure entropy[J]. Systems Engineering and Electronics，2009，31(3)：702-704

[15] 姚爱群. 信息系统开发方法[M]. 北京：清华大学出版社，2004

[16] 雷芳，黄进. 一种新信息熵及其若干性质[J]. 重庆邮电学院学报(自然科学版)，2007，18(6)：778-780
Lei Fang，Huang Jin. A New Information Entropy Definition and Properties[J]. Journal of Chongqing University of Posts and Telecommunications (Natural Science)，2007，18(6)：778-780

[17] Pal N R，Pal S K. Entropy：a new definition and its applications[J]. IEEE Transactions on Systems，Man and Cybernetics，1991，21(5)：1260-1270