基于粒计算的属性约简改进算法

唐 孝[1,2] 舒 兰[1]
(四川师范大学数学与软件科学学院 成都 610068)[1] (电子科技大学数学科学学院 成都 611731)[2]

摘 要 粒计算是基于问题求解、模式分类及信息处理的多层次粒结构分析方法,它是粗糙集、模糊集、数据挖掘以及人工智能等多领域交叉的一门新学科。在讨论知识粒度的基本概念和性质后,介绍了通过计算属性对约简核的重要度 $Sig_{Core(A)}(a)$ 来进行信息系统约简的方法。考虑到有的信息系统没有约简核,提出了基于粒计算的约简算法的改进。改进后的算法既可以用于有约简核的系统,也可以用于没有约简核的系统。数值实验证实了算法的有效性。

关键词 粒计算,知识粒度,属性重要度,属性约简
中图法分类号 TP18 **文献标识码** A

Improved Algorithm of Attribute Reduction Based on Granular Computing

TANG Xiao[1,2] SHU Lan[1]
(School of Mathematics, Sichuan Normal University, Chengdu 610068, China)[1]
(School of Mathematical Sciences, University of Electronic Science and Technology of China, Chengdu 611731, China)[2]

Abstract Granular computing is a method of multilayer granular structure analysis based on problem solving, pattern classification and information processing. It is a new multidisciplinary cross discipline between rough sets, fuzzy sets, data mining and artificial intelligence. Some important properties of granular computing were discussed as well as the reduction algorithm. The traditional reduction algorithm based on granular computing is gradually calculated with reduction *Core*(*A*), but some information systems may have no reduction core. In this case, an improved reduction algorithm based on attribute significance of granular computing was proposed. The algorithm can be used in system with both reduction core and no reduction core. Finally, experiments show the feasibility of the algorithm.

Keywords Granular computing, Knowledge granule, Attribute significance, Attribute reduction

1 引言

随着科学技术的飞速发展,各行业领域的数据日趋猛增,如何从海量数据中得到关键信息是形式概念分析重要的研究内容。粒计算是处理这类信息的有力工具,它已在机器学习、数据分析、智能数据处理、知识发现等方面得到了很好的应用[1-5]。

知识发现就是从海量的数据集中找到潜在其中的、人们想得到的有用的知识的过程,它是数据挖掘的一种更广义的说法。知识是人们通过观察、度量和推理等方式得到的一种定义,而这些方式的实体就是粒度。由人类活动所形成的世界里,粒度无处不在,任何复杂信息的简化和复杂问题的求解都可以通过粒化的思想将它分解成知识的"片"和"块",然后在这些小的"片"和"块"上再逐个求解,这就是粒计算的基本思想。粒计算(Granular Computing, GrC)是由美国圣荷西州立大学林早阳教授(L. Y. Lin)于 1996 年首次提出来的,它是信息处理的一种新的概念和计算范式,是处理模糊的、不精确的、不完整的和海量信息的新工具。

目前基于粒计算的属性约简方法大多以约简核为基础,首先计算出系统的约简核,然后确定剩余属性对于核的重要度的大小来获取系统的约简集[6-11]。然而有些系统是没有约简核的,经典算法在这类问题上就无能为力了,因此本文提出一个改进算法,它既可以处理系统有约简核的情况,也可以处理没有约简核的情况。

2 基础知识

为了方便后面的讨论,我们首先简单介绍基于粗糙集的知识粒度的基本概念与性质,详细知识参见文献[12-15]。

定义 1 设 U 是非空有限论域,R 是 U 上的一个等价关系,知识库 $K=\langle U,R\rangle$ 也被称为近似空间(Approximation Space),称等价关系 R 生成的等价类$[u]_R$ 为基本知识颗粒,称商集 $U/R=\{[u]_R\mid u\in U\}$ 为论域 U 的 R-粒划分。

定义 2 设 $P,Q\in R$ 是 U 上的等价关系,

若对 $\forall u,v\in U$ 有 $uPv\Leftrightarrow uQv$,则称 P 与 Q 相等,记作 $P=Q$;

若对 $\forall u,v\in U$ 有 $uPv\Rightarrow uQv$,则称 P 比 Q 细或 Q 比 P 粗,记作 $P\leq Q$;

若 $P\leq Q$ 且 $P\neq Q$,也即是 $P\prec Q$,则称 P 比 Q 严格细或

本文受四川师范大学科研项目(13KYL15)资助。
唐 孝(1981-),男,博士生,讲师,主要研究方向为不确定性分析、数据挖掘,E-mail:80651177@163.com;**舒 兰**(1965-),女,硕士,教授,博士生导师,主要研究方向为不确定性分析、数据挖掘、信号处理。

Q 比 P 严格粗。

由 R 中的等价关系 P 生成的不可区分关系 $IND(P)=\bigcap_{P\in R}[x]_P$，表示了利用知识库中的部分知识 P 所能达到的最高认知程度，而 $IND(R)$ 表达了知识库 $K=\langle U,R\rangle$ 的最高分辨和认知程度。

定义 3 设 $K=\langle U,R\rangle$ 是一个知识库，$P\in R$ 是论域 U 上的等价关系，称为知识。我们把知识 $P\in R$ 的粒度记作 $GD(P)$，则

$$GD(P)=\frac{|P|}{|U\times U|}=\frac{|P|}{|U|^2} \tag{1}$$

其中，$|P|$ 表示 $P\subseteq U\times U$ 的基数。

知识 P 的粒度可以表示它的分辨能力，对 $\forall u,v\in U$，若 $(u,v)\in P$，表示对象 u,v 在 P 下不可分辨，属于 P 的同一个等价类；否则它们可分辨，属于不同的 P-等价类。一般情况下，若 $GD(P)$ 越大，表示 P 的分辨能力越弱；反之若 $GD(P)$ 越小，表示 P 的分辨能力越强。

定理 1 若 $P\in R$ 是知识库 $K=\langle U,R\rangle$ 中的知识，$U/P=\{X_1,X_2,\cdots,X_n\}$，则

$$GD(P)=(\sum_{i=1}^{n}|X_i|^2)/|U|^2 \tag{2}$$

证明：对于 $\forall(u,v)\in P$，存在 $u,v\in$ 某一 X_i，故 P 的元是通过在 $\frac{U}{P}$ 的每个等价类 X_i 中任取二元（可以相同）组成的，于是有 $P=\sum_{i=1}^{n}|X_i|^2$，故

$$GD(P)=\frac{|P|}{|U|^2}=\frac{(\sum_{i=1}^{n}|X_i|^2)}{|U|^2}$$

性质 1 设 $S=(U,A,V,f)$ 是一个信息系统，$P,Q\subseteq A$，则有

(1)若 $P\Rightarrow Q$，则 $GD(P)\leqslant GD(Q)$；

(2)若 $P\Leftrightarrow Q$，则 $GD(P)=GD(Q)$。

证明：

(1)由 $P\Rightarrow Q$ 知，$IND(P)\subseteq IND(Q)$，则：$|IND(P)|\leqslant|IND(Q)|$，又

$$GD(P)=GD(IND(P))=|IND(P)|/|U|^2$$

$$GD(Q)=GD(IND(Q))=|IND(Q)|/|U|^2$$

因此 $GD(P)\leqslant GD(Q)$。

(2)由 $P\Leftrightarrow Q$ 知，$P\Rightarrow Q$ 且 $Q\Rightarrow P$，由(1)知

$$GD(P)\leqslant GD(Q) \text{ 且 } GD(Q)\leqslant GD(P)$$

可得 $GD(P)=GD(Q)$。

3 知识粒度在属性约简中的应用

信息系统 $S=(U,A,V,f)$ 中属性集 A 的每一个属性 a 的重要性都可以通过知识粒度来分析，若在属性集 A 中去掉属性 a 后引起的粒度的较大变化，那么 a 对于属性集 A 来讲就重要。

定义 4 设 $S=(U,A,V,f)$ 是一个信息系统，属性集 A 中的属性 a 的重要度表示为：$Sig_{A-\{a\}}(a)$，定义为：

$$Sig_{A-\{a\}}(a)=GD(A-\{a\})-GD(A) \tag{3}$$

特别地，当 $A=\{a\}$ 时，用 $Sig(a)$ 表示 $Sig_\phi(a)$，且 $Sig(a)=Sig_\phi(a)=GD(\phi)-GD(A)=1-GD(\{a\})$，其中 $GD(\phi)=1$（由于 $U/IND(\phi)=\{U\}$）。

性质 2 $0\leqslant Sig(a)\leqslant 1/|U|$。

定义 5 设 $S=(U,A,V,f)$ 是一个信息系统，C 是属性集 A 中的子集，$C\subseteq A$，$\forall a\in A-C$ 对于属性集 C 的重要度表示为 $Sig_C(a)$，定义为：

$$Sig_C(a)=GD(C)-GD(C\cup\{a\}) \tag{4}$$

定义 5 表明，属性 a 对于属性集 C 的重要性可以通过在 C 中添加 a 后引起的知识粒度的变化来度量，变化越大，认为 a 对于 C 越重要。

定义 6 设 $S=(U,A,V,f)$ 是一个信息系统，$a\in A$，如果 $GD(A-\{a\})=GD(A)$，则称 a 在 A 中是不必要的（可省去的）；否则 a 在 A 中是必要的（不可省去的）。如果每个 $a\in A$ 都为 A 中必要的，则称 A 为独立的，否则称 A 为依赖的。

定义 7 设 $P\subseteq A$，若 P 是独立的，且 $GD(P)=GD(A)$，则称 P 是 A 中的一个约简，A 的所有约简记为 $red(A)$；A 中所有必要属性组成的集合称为 A 的核，记为 $Core(A)$，$Core(A)=\bigcap red(A)$。

性质 3 属性 a 在 A 中是必要的（不可省去的），当且仅当 $Sig_{A-\{a\}}(a)>0$。

性质 4 $Core(A)=\bigcup\{a\in A|Sig_{A-\{a\}}(a)>0\}$。

属性的重要度从知识粒度的角度提供了一个属性约简的方法：通过计算 $GD(A-\{a\})$ 与 $GD(A)$ 是否相等来判断属性 a 是否可去；若 $GD(A-\{a\})=GD(A)$，表示属性 a 是可去的，若不等，表示 a 不可去。在一个属性集合中那些不可去的属性的集合就是约简的核 $Core(A)$，接下来确定剩余属性对于核的重要度 $Sig_{Core(A)}(a)$，取重要度中最大值 $\max_{a\in A-Core(A)} Sig_{Core(A)}(a)$ 与核组成最终约简 $red(A)$。

算法 1

输入：信息系统 $S=(U,A,V,f)$，其中 $U=\{x_1,x_2,\cdots,x_n\}$，$A=\{a_1,a_2,\cdots,a_m\}$；

输出：该信息系统的最小约简 red(A) 和核 Core(A)。

步骤 1 计算属性集 A 的知识粒度 GD(A)；

步骤 2 计算属性集 A 中每一个属性 a 的重要度 $Sig_{A-\{a\}}(a)$；

步骤 3 计算核 Core(A)，$Core(A)=\{a\in A|Sig_{A-\{a\}}(a)>0\}$；

步骤 4 判断 GD(Core(A)) 与 GD(A) 是否相等，若 GD(Core(A))=GD(A)，输出最小约简 red(A)=Core(A)。若 GD(Core(A))>GD(A)，转下一步；

步骤 5 计算 B=A−Core(A) 中每一个属性对于 Core(A) 的重要度；

步骤 6 取 $\max_{b\in B}Sig_{Core(A)}(b)$；

步骤 7 判断 $GD(Core(A)\cup b)$ 与 GD(A) 是否相等；若 $GD(Core(A)\cup a')=GD(A)$，输出 $red(A)=Core(A)\cup b$ 和 Core(A)。若 $GD(Core(A)\cup b)>GD(A)$，重复步骤 5 的过程，转下一步；

步骤 8 计算 $C=A-Core(A)\cup b$ 每一个属性对于 $Core(A)\cup b$ 的重要度，取重要度中最大值 $\max_{c\in C}Sig_{Core(A)\cup b}(b)$ 的那个属性与 $Core(A)\cup b$ 合并；重复步骤 7 直到相等或者取遍 C 中所有属性；

步骤 9 输出最终约简 $red(A)=Core(A)\cup b\cup c\cup\cdots Core(A)$。

例 1 给定一个信息系统 $S=(U,A,V,f)$，如表 1 所列，该信息系统有 6 个对象及 4 个属性：$U=\{x_1,x_2,x_3,x_4,x_5,x_6\}$，$A=\{a_1,a_2,a_3,a_4\}$，求系统的约简和核。

表 1 一个信息系统

U	a_1	a_2	a_3	a_4
x_1	1	0	2	2
x_2	0	1	1	1
x_3	2	0	0	1
x_4	1	1	0	2
x_5	2	2	0	0
x_6	2	1	1	1

解：

(1)求 $GD(A)$

$U/A=\{x_1,x_2,x_3,x_4,x_5,x_6\}$，

$GD(A)=(\sum_{i=1}^{n}|X_i|^2)/|U|^2=6/36=1/6$；

(2)求 A 中各属性对于 A 的重要度

$U/A-\{a_1\}=\{x_1,\{x_2,x_6\},x_3,x_4,x_5\}$，

$GD(A-\{a_1\})=8/36$；

$Sig_{A-\{a_1\}}(a_1)=GD(A-\{a_1\})-GD(A)=2/36$；

$U/A-\{a_2\}=\{x_1,x_2,x_3,x_4,x_5,x_6\}$，

$GD(A-\{a_2\})=6/36$；

$Sig_{A-\{a_2\}}(a_2)=GD(A-\{a_2\})-GD(A)=0$；

$U/A-\{a_3\}=\{x_1,x_2,x_3,x_4,x_5,x_6\}$，

$GD(A-\{a_3\})=6/36$；

$Sig_{A-\{a_3\}}(a_3)=GD(A-\{a_3\})-GD(A)=0$；

$U/A-\{a_4\}=\{x_1,x_2,x_3,x_4,x_5,x_6\}$，

$GD(A-\{a_4\})=6/36$；

$Sig_{A-\{a_4\}}(a_4)=GD(A-\{a_4\})-GD(A)=0$；

(3)$Core(A)=\{a\in A|Sig_{A-\{a\}}(a)>0\}=\{a_1\}$，

$U/Core(A)=\{\{x_1,x_4\},x_2,\{x_3,x_5,x_6\}\}$，

$GD(Core(A))=14/36>6/36=GD(A)$；

(4)求 $B=A-Core(A)=\{a_2,a_3,a_4\}$ 中每个属性对于 $Core(A)$ 的重要度

$Sig_{Core(A)}(a_2)=GD(Core(A))-GD(Core(A)\cup a_2)$
$=14/36-6/36=8/36$；

$Sig_{Core(A)}(a_3)=GD(Core(A))-GD(Core(A)\cup a_3)$
$=14/36-8/36=6/36$；

$Sig_{Core(A)}(a_4)=GD(Core(A))-GD(Core(A)\cup a_4)$
$=14/36-10/36=4/36$；

(5)取 $\max_{b\in B} Sig_{Core(A)}(b)=8/36$

求 $U/Core(A)\cup a_2=\{x_1,x_2,x_3,x_4,x_5,x_6\}$，求得 $GD(Core(A)\cup a_2)=GD(A)$，则信息系统的约简为 $red(A)=Core(A)\cup a_2=\{a_1,a_2\}$，核 $Core(A)=\{a_1\}$。

4 一个改进算法

信息系统的属性约简结果不是唯一的，有的可能会有多个约简，而且这些约简不一定都能得到约简核。对于那些没有约简核的情况，上面的算法就无能为力了，下面我们给出一个新的约简算法，该算法既能够处理有约简核的也可以处理没有约简核的情况。

算法 2

输入：信息系统 S=(U,A,V,f)，其中 $U=\{x_1,x_2,\cdots,x_n\}$，$A=\{a_1,a_2,\cdots,a_m\}$；

输出：该信息系统的最小约简 red(A)和核 Core(A)。

步骤 1 计算属性集 A 的知识粒度 GD(A)；

步骤 2 计算属性集 A 中每一个属性 a 的重要度 $\mathrm{Sig}_{A-\{a\}}(a)$，若 $\mathrm{Sig}_{A-\{a\}}(a)\neq0$，表示系统有约简核，转步骤 5；
若 $\mathrm{Sig}_{A-\{a\}}(a)=0$，表示系统没有约简核，转下一步；

步骤 3 计算属性的两两组合对于 A 的重要度 $\mathrm{Sig}_{A-\{a_i,a_j\}}(a_i,a_j)$，$1\leqslant i\neq j\leqslant m$；

步骤 4 取 $\max_{1\leqslant i\neq j\leqslant m}\mathrm{Sig}_{A-\{a_i,a_j\}}(a_i,a_j)$ 的属性集作为系统的最小约简，算法终止；
输出 $\mathrm{red}(A)=\{(a_i,a_j)|\max_{1\leqslant i\neq j\leqslant m}\mathrm{Sig}_{A-\{a_i,a_j\}}(a_i,a_j)\}$。

步骤 5 计算核 Core(A)，$\mathrm{Core}(A)=\{a\in A|\mathrm{Sig}_{A-\{a\}}(a)>0\}$；

步骤 6 判断 GD(Core(A))与 GD(A)是否相等，若 GD(Core(A))=GD(A)，输出最小约简 red(A)=Core(A)。
若 GD(Core(A))>GD(A)，转下一步；

步骤 7 计算 B=A−Core(A)中每一个属性对于 Core(A)的重要度；

步骤 8 取 $\max_{b\in B}\mathrm{Sig}_{\mathrm{Core}(A)}(b)$；

步骤 9 判断 GD(Core(A)∪b)与 GD(A)是否相等；若 GD(Core(A)∪a′)=GD(A)，输出 red(A)=Core(A)∪b 和 Core(A)。
若 GD(Core(A)∪b)>GD(A)，重复步骤 5 的过程，转下一步；

步骤 10 计算 C=A−Core(A)∪b 每一个属性对于 Core(A)∪b 的重要度，取重要度中最大值 $\max_{c\in C}\mathrm{Sig}_{\mathrm{Core}(A)\cup b}(b)$ 的那个属性与 Core(A)∪b 合并，重复步骤 7 直到相等或者取遍 C 中所有属性后终止算法；

步骤 11 输出最终约简 red(A)=Core(A)∪b∪c∪…和 Core(A)。

例 2 给定一个信息系统 $S=(U,A,V,f)$，如表 2 所列，该信息系统有 5 个对象及 4 个属性：$U=\{x_1,x_2,x_3,x_4,x_5\}$，$A=\{a_1,a_2,a_3,a_4\}$，求系统的约简。

表 2 一个信息系统

U	a_1	a_2	a_3	a_4
x_1	1	0	2	1
x_2	2	1	2	1
x_3	0	1	2	0
x_4	2	0	1	1
x_5	1	2	2	0

解：

(1)求 $GD(A)$

$U/A=\{x_1,x_2,x_3,x_4,x_5\}$，

$GD(A)=(\sum_{i=1}^{n}|X_i|^2)/|U|^2=5/25=1/5$；

(2)求 A 中各属性对于 A 的重要度

$U/A-\{a_1\}=\{x_1,x_2,x_3,x_4,x_5\}$，

$GD(A-\{a_1\})=1/5$，

$Sig_{A-\{a_1\}}(a_1)=GD(A-\{a_1\})-GD(A)=0$；

$U/A-\{a_2\}=\{x_1,x_2,x_3,x_4,x_5\}$，

$GD(A-\{a_2\})=1/5$，

$Sig_{A-\{a_2\}}(a_2)=GD(A-\{a_2\})-GD(A)=0$；

$U/A-\{a_3\}=\{x_1,x_2,x_3,x_4,x_5\}$，

$GD(A-\{a_3\})=1/5$，

$Sig_{A-\{a_3\}}(a_3)=GD(A-\{a_3\})-GD(A)=0$；

$U/A-\{a_4\}=\{x_1,x_2,x_3,x_4,x_5\}$，

$GD(A-\{a_4\})=1/5$，

$Sig_{A-\{a_4\}}(a_4)=GD(A-\{a_4\})-GD(A)=0$；

说明系统没有约简核，每一个单个属性都可省。

(3)求属性的两两组合对于 A 的重要度

$Sig_{A-\{a_i,a_j\}}(a_i,a_j)$，$1\leqslant i\neq j\leqslant 4$

(下转第 346 页)

分区和计算用户旅行代价。在真实与合成的数据集上的实验表明，PMCF 算法可以有效利用用户和物品的位置信息来产生位置感知推荐，相比于传统推荐算法，PMCF 算法显著提高了推荐精度。

参考文献

[1] 项亮，陈义，王益. 推荐系统实践 [M]. 河北：人民邮电出版社，2012：121-143，179-195

[2] Park M-H，et al. Location-Based Recommendation System Using Bayesian User's Preference Model in Mobile Devices [C]// Proceedings of the Ubiquitous Intelligence and Computing（UIC）. Hong Kong，China，2007：1130-1139

[3] Netflix News and Info-Local Favorites [EB/OL]. [2013-09-10]. http://tinyurl.com/4qt8ujo

[4] Bao J，Chow C-Y，Mokbel M F，et al. Efficient evaluation of k-range nearest neighbor queries in road networks [C]// Proceedings of the International Conference on Mobile Data Management. Kansas City，MO，USA，2010：115-124

[5] Mouratidis K，Yiu M L，Papadias D，et al. Continuous nearest neighbor monitoring in road networks [C]// Proceedings of the Very Large Date Bases（VLDB）. Seoul，Korea，2006：43-54

[6] Bruno N，Gravano L，Marian A. Evaluating Top-k Queries over Web-Accessible Databases [C]// Proceedings of the 18th International Conference on Data Engineering（ICDE）. San Jose，California，USA，2002：369-380

[7] Venetis P，Gonzalez H，Jensen C S，et al. Hyper-Local，Directions-Based Ranking of Places [C]// Proceedings of the 27th International Conference on Data Engineering（ICDE）. Hannover，Germany，2011：290-301

[8] Takeuchi Y，Sugimoto M. An Outdoor Recommendation System based on User Location History [C]// Proceedings of the Ubiquitous Intelligence and Computing（UIC）. Wuhan and Three Gorges，China，2006：625-636

[9] Zheng V W，Zheng Y，Xie X，et al. Collaborative Location and Activity Recommendations with GPS History Data[C]// Proceedings of the 10th International Conference on World Wide Web. New York，USA，2010：1029-1038

[10] Ye M，Yin P，Lee W-C. Location Recommendation for Location-based Social Networks [C]// Proceedings of the ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems. San Jose，California，USA，2010：458-461

[11] Sarwar B，Karypis G，Konstan J，et al. Item-based Collaborative Filtering Recommendation Algorithms [C]// Proceedings of the 10th International Conference on World Wide Web. New York，USA，2001：285-295

[12] Fagin R，Lotem A，Naor M. Optimal Aggregation Algorithms for Middleware [C]// Proceedings of the twentieth ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems. New York，USA，2001：102-113

（上接第 315 页）

$GD(A-\{a_1,a_2\})=9/25$,

$Sig_{A-\{a_1,a_2\}}(a_1,a_2)$

$=GD(A-\{a_1,a_2\})-GD(A)=4/25$;

$GD(A-\{a_1,a_3\})=7/25$,

$Sig_{A-\{a_1,a_3\}}(a_1,a_3)$

$=GD(A-\{a_1,a_3\})-GD(A)=2/25$;

$GD(A-\{a_1,a_4\})=7/25$,

$Sig_{A-\{a_1,a_3\}}(a_1,a_4)$

$=GD(A-\{a_1,a_4\})-GD(A)=2/25$;

$GD(A-\{a_2,a_3\})=5/25$,

$Sig_{A-\{a_2,a_3\}}(a_2,a_3)$

$=GD(A-\{a_2,a_3\})-GD(A)=0$;

$GD(A-\{a_2,a_4\})=7/25$, $Sig_{A-\{a_2,a_4\}}(a_2,a_4)$

$=GD(A-\{a_2,a_4\})-GD(A)=2/25$;

$GD(A-\{a_3,a_4\})=5/25$,

$Sig_{A-\{a_3,a_4\}}(a_3,a_4)$

$=GD(A-\{a_3,a_4\})-GD(A)=0$;

由上面讨论得，系统的最小约简为 $red(A)=\{a_1,a_2\}$，次小约简为 $red(A)=\{a_1,a_3\}$ or $\{a_1,a_4\}$ or $\{a_2,a_4\}$。

结束语 本文主要针对没有约简核的信息系统，提出了基于粒计算的属性约简算法的改进。传统的基于粒计算的约简算法大多以约简核 $Core(A)$ 为基础来逐步计算属性对于核的重要度，从而确定出系统的约简集。然而有的信息系统可能没有约简核，因此，基于核的约简算法就失效了。针对这一情况，本文对约简算法进行了改进，改进后的算法既可以用于有约简核的系统也可以用于没有约简核的系统，并通过实验证明了该算法的可行性。

参考文献

[1] 苗夺谦，王国胤，刘清，等. 粒计算：过去、现在与展望[M]. 北京：科学出版社，2007

[2] 王国胤，张清华，胡军. 粒计算研究综述[J]. 智能系统学报，2007，6(2)：8-26

[3] 徐伟华，刘士虎，张文修. 一般二元关系下信息系统知识的粒度描述[J]. 计算机工程与应用，2011，47(18)：40-44

[4] 胡峰，黄海，王国胤. 不完备信息系统的粒计算方法[J]. 小型微型计算机系统，2005，26(8)：1335-1339

[5] 刘清，刘群. 粒及粒计算在逻辑推理中的应用[J]. 计算机研究与发展，2004，41(4)：546-551

[6] 徐久成，史进玲，孙林. 一种基于相对粒度的决策表约 简算法[J]. 计算机科学，2009，36(3)：205-207

[7] 冯林，刘照鹏，方丹. 信息系统中粒计算模型及其属性约简方法[J]. 重庆邮电大学学报：自然科学版，2008，22(5)：652-655

[8] 赵敏，罗可，秦哲. 基于粒计算的属性约简算法[J]. 计算机工程与应用，2008，44(30)：157-159

[9] 陈玉明，苗夺谦，焦娜. 基于二进制粒与粒计算的属性约简[J]. 广西师范大学学报：自然科学版，2008，26(2)：81-84

[10] 王红霞，王志伟，程艳慧. 一种基于粒计算属性约简算法的改进及应用[J]. 微计算机信息，2010，26(5)：33-35

[11] 苗夺谦，范世栋. 知识的粒度计算及其应用[J]. 系统工程理论与实践，2002，22(1)：48-56

[12] 张文修，吴伟志，梁吉业，等. 粗糙集理论与方法[M]. 北京：科学出版社，2001

[13] 梁吉业，李德玉. 信息系统中的不确定性与知识获取[M]. 北京：科学出版社，2005

[14] Pawlak Z. Rough Sets：Theoretical Aspects of Reasoning about Data [M]. Dordrecht，Boston：Kluwer Academic Publishers，1991

[15] Liang J Y，Shi Z Z. The Information Entropy，Rough Entropy and Knowledge Granulation in Rough Set Theory[J]. International Journal of Uncertainty，Fuzziness and Knowledge-Based Systems，2004，12(1)：37-46