粒度搜索：完备不协调决策表下简单、高效的启发式属性约简算法

**摘要：**属性约简是粗糙集理论中的核心内容之一。属性约简包含信息系统中目标对象集的定义及求解保持目标对象集分类的最小属性子集的算法两方面内容。完备不协调决策表的属性约简目标函数分为五类。从目标对象集的定义形式上看，五类属性约简目标函数的目标对象集的定义复杂抽象不便于计算。从求解保持目标对象集分类的最小属性子集的算法上看，最大分布保持属性约简的启发式算法存在正确性问题，相对不可辨识关系保持属性约简目标函数的启发式算法仍没有被设计出来，另外三类属性约简的启发式算法不够高效。考虑到以上问题，本文提出可以表达五种属性约简目标函数的粒度空间的概念，基于粒度空间，我们设计了称为粒度搜索的启发式属性约简算法，该算法可用以求取五种属性约简。属性约简的表示对比分析表明使用粒度空间阐述属性约简更加简单、直观、便于计算，算法复杂度分析、算例对比和实验对比结果表明本文所提算法的更加简单、直观、高效。

1. **引言**

粗糙集[4-12]（Rough Sets）理论是波兰数学家Pawlak教授在1982年提出的一种定量分析处理不精确、不一致、不完备信息与知识的数学工具。相较于其他软计算理论，粗糙集无需被处理数据之外的先验信息，对问题的描述和处理更为客观，因此广泛应用于模式识别、数据挖掘、机器学习等领域[1-3, 13-18]。属性约简[19-21]是粗糙集的核心内容之一，其包含两方面的内容——信息系统的目标对象集的定义及求解保持目标对象集分类的最小属性子集的算法。信息系统可分为信息表和决策表，决策表可分为完备决策表和不完备决策表，完备决策表可分为协调决策表和不协调决策表，本文主要关注完备不协调决策表下属性约简定义的直观、简洁性问题与相应求解算法的简单直观、高效性不足。

在目标对象集的定义方面较有代表性的工作如下：Pawlak提出了正域保持属性约简的概念；苗夺谦等[22]提出了信息论观点下的属性约简目标函数，在此基础上，王国胤等[23]进一步提出了条件信息熵属性约简概念；Slezak[24]基于属性频度信息提出了分布约简概念；张文修等[25]提出了最大分布约简和分配约简概念；Kryszkiewicz比较分析了针对单个对象和整体决策表的属性约简目标函数，提出了近似约简和可能约简概念[26]；苗夺谦和姚一豫[27]定义了相对不可分辨关系描述，提出了相对可分辨关系保持的属性约简目标函数；钱宇华等[28]定义了组合熵的概念并提出了组合熵保持属性约简，更多属性约简目标函数可查阅[29-33]。考虑到基于等价关系导出对象的分类存在容噪能力较差、规则提取利用的信息有限的特点，学者们提出多种合理的论域分类方法拓展出多种更加合理的模型，如模糊粗糙集模型、粗糙模糊集模型[34]、覆盖粗糙集模型[35,36]、概率粗糙集模型[37-40]等，由于本文关注经典粗糙集模型，其相应的目标对象集不在此处一一列述。

求解保持目标对象集分类的最小属性子集的算法可分为可辨识矩阵属性约简算法和启发式属性约简算法两类。1992年Skowron[41]提出了使用可辨识矩阵求取所有正域保持属性约简的算法， Slezak在2000年提出了使用可辨识矩阵求取分布保持属性约简的算法，Kryszkiewicz在2001年提出了使用可辨识矩阵求取广义决策保持属性约简的算法，张文修在2003年提出了求取最大分布保持属性约简的可辨识矩阵的算法，苗夺谦在2009年提出了使用可辨识矩阵求取相对不可分辨关系求取属性约简的算法，借助可辨识矩阵获取属性约简成为设计属性约简算法的重要思路[42-49]。启发式属性约简算法由学者的直观理解或经验构造出来，可以快速地计算得到一个属性约简。较具有代表性的有：1995年Hu等[4]提出了更加高效的正域保持属性约简的启发式算法；1997年苗夺谦等[11]从信息论的角度出发提出了基于互信息的分布保持属性约简启发式算法；2014年黎敏等[59]提出了广义决策保持属性约简启发式算法、分布保持属性约简启发式算法、最大分布保持属性约简启发式算法。文献[50]总结了目前构建启发式属性约简算法策略：增加策略、删除策略和增加—删除策略，并指出增加策略的启发式属性约简算法得到的结果不一定是属性约简。钱宇华等[58]从粒度的角度观察增加策略的属性约简算法迭代过程中正域的变化，提出使用去除正域加快增加策略启发式属性约简算法的速度，更多启发式属性约简算法相关内容可查阅[51-57]。

启发式属性约简算法与可辨识矩阵属性约简算法被提出后，二者关系的不清楚成为设计更加高效的算法的阻碍，针对这个问题，文献[29-33]做了较为系统的归纳与总结。文献[33]在分析已有工作的基础上，归纳出5种较为典型的属性约简：正域保持属性约简、广义决策保持属性约简、分布保持属性约简、最大分布保持属性约简、相对不可分辨关系保持属性约简。目前可辨识矩阵算法可以求解5种属性约简，启发式属性约简算法可以求解正域保持属性约简、广义决策保持属性约简、分布保持属性约简、最大分布保持属性约简。

目前五种属性约简的定义存在复杂抽象不便于理解、计算的缺点，可辨识矩阵属性约简算法求解属性约简的原理符合人的直觉但其求解效率低下，启发式属性约简算法求解速度较好，但仅有4种属性约简目标函数的启发式算法被提出，其中最大分布保持属性约简的启发式算法存在正确性问题，相对不可分辨关系保持属性约简的启发式算法仍未被提出。且目前数据规模量级飞速增长，启发式属性约简算法的效率并不能完全满足现实需求。考虑到以上问题，本文从粒度角度审视完备不协调决策表下的五种属性约简的定义，考察五种属性约简的粒度的可辨识关系与不可辨识关系，引入并借助粒度空间的概念来定义五种属性约简，相较原有的属性约简定义，使用粒度空间来阐述属性约简更加简单、直观、便于计算。基于粒度空间的概念，我们设计了可以求取五种属性约简的启发式的算法——粒度搜索算法，修正了最大分布保持属性约简算法，填补了相对不可分辨关系保持属性约简启发式算法的空白。相较已有的三种启发式属性约简算法，粒度搜索算法的重要度函数更加简单、直观、高效。

文章的结构如下：第二节回顾了粗糙集的基础知识。第三节引入了粒度、粒度空间、粒度的可辨识关系、粒度的不可辨识关系的概念；分析了五种属性约简粒度的可辨识关系与不可辨识关系，提出属性约简的粒度都属于粒度空间中的元素。第四节基于粒度观点下的属性约简定义，提出了一种简单、直观、高效的启发式属性约简算法，该算法能够求解5种属性约简。第五节为算例对比、理论分析和实验对比验证粒度搜索算法的简单直观、高效。

1. **基础知识**

本部分共分两小节，第一小节回顾粗糙集中的基础概念——上下近似、五种属性约简的定义；第二小节回顾启发式属性约简算法。

**2.1五种属性约简**

定义 2.1 是一个决策表，是一个非空有限的对象集合，称为论域。为属性集合，分别表示条件属性集合、决策属性集合，并且。，表示属性可取值的集合，称作属性的域；是一个对应论域、属性的函数，即。

需要说明的是，决策表元素均为离散值，默认，即决策表只有一个决策属性。

定义 2.2 对，记 ，为论域上定义的一个等价关系，称为不可分辨关系。 为导出的不可分辨关系的等价类全集，它构成了的一个完备划分，简记为。 是下对象所属的等价类。

定义2.3 信息系统，，概念关于的上、下近似分别定义为：

；

。

利用上下近似逼近不确定概念是粗糙集中处理不确定性的一个重要思路。属性约简是保持决策表中不确定性概念集合的最小属性子集，姚一豫等[61]对属性约简做了概括性定义：

定义 2.4给定决策表和分类特征，若满足：

（1）分类特征的度量函数，使得，即将属性子集映射为偏序集的元素

（2）；

（3），。

根据周杰等在文献[33]的总结，完备不协调决策表现有正域保持属性约简、广义决策保持属性约简、分布保持属性约简、最大分布保持属性约简、相对不可分辨关系保持属性约简5中属性约简。

定义 2.5 决策表，决策属性集相对条件属性集的正域为：

。

决策属性集相对条件属性集的边界域为：

。

定义 2.6 决策表，为属性集的正域保持属性约简，当且仅当：

（1） ；

（2） 。

定义 2.7 决策表，在属性集下关于决策属性集的广义决策为：

，

为属性集的一个广义决策保持属性约简，当且仅当：

（1） ；

（2） 。

定义 2.8 决策表，，在属性集下关于决策属性集的隶属度分布函数定义为：

。

为属性集的一个分布保持属性约简，当且仅当：

（1） ；

（2） 。

定义 2.9 决策表，，在属性集下关于决策属性集的最大隶属度分布函数定义为：

。

为属性集的一个最大分布保持属性约简，当且仅当：

（1） ；

（2） 。

定义 2.10决策表，由确定的关于决策属性集的相对不可分辨关系定义为:



为属性集的一个相对不可分辨关系保持属性约简，当且仅当满足：

（1）；

（2），。

**2.2属性约简算法**

考虑到研究内容的限制，此处只介绍求解4种属性约简的启发式属性约简算法。启发式算法的策略大致分为三种：增加策略、删除策略、增加—删除策略。文献[61]指出增加策略并不保证得到的一定是属性约简，而删除策略的启发式属性约简算法目前没有较好的提速方法，故本文使用增加—删除策略来构建启发式属性约简算法，故本文使用增加—删除策略来介绍已有的启发式属性约简算法。

算法 2.1 增加—删除策略的启发式属性约简算法

输入：决策表

输出：属性约简

初始化：

Step 1：计算停止点。

Step 2：如果，执行Step 3；否则执行Step 2.1。

Step 2.1：计算，其满足对。

Step 2.2：令，执行Step 2。

Step 3：对，若，则。

Step 4：，返回。

需要求取不同类型的属性约简，使用不同的评价函数，外部重要度函数、内部重要度函数即可，以下介绍四种类型启发式属性约简算法所用函数。

定义 2.16 给定决策表， 那么定义正域保持属性约简评价函数



正域保持属性约简算法重要度如下：





定义 2.17 给定决策表，那么定义广义决策保持属性约简评价函数



广义决策保持属性约简算法重要度如下：





定义 2.18 给定决策表，那么定义分布保持属性约简评价函数



分布保持属性约简算法重要度如下：





定义 2.19 给定决策表，那么定义最大分布保持属性约简评价函数



最大分布保持属性约简算法重要度如下：





需要说明的是，最大分布保持属性约简的评价函数和重要度函数疑似存在问题，考虑各部分内容结构的安排，该问题详细的说明放在了第5大节。

1. **属性约简的粒度空间表示**

本节共分两小节，第一小节介绍粒度、粒度空间、粒度的可辨识关系、粒度的不可辨识关系的关系，并指出粒度的可辨识关系满足对称性，粒度的不可辨识关系为等价关系。第二小节介绍了五种属性约简粒度的可辨识关系、不可辨识关系，并指出五种属性约简粒度的可辨识关系满足对称性，五种属性约简粒度的不可辨识关系为等价关系。第三小节主要证明属性约简的可辨识关系与不可辨识关系与属性约简目标函数的关系。结合粒度空间的可辨识关系、不可辨识关系，指出五种属性约简粒度均处于一个可被定义的粒度空间中，并给出了五种属性约简粒度的粒度空间定义。

**3.1 粒度与粒度空间**

此处介绍论域由等价关系导出的等价类簇构成的粒度。

定义 3.1.1概念表示论域的任意非空子集，一个粒度是*n*个概念构成的集合。考虑到研究内容的特性，增加限制

。

为清楚的表示，约定使用“(*e1*，*e2*)”表示*e1*，*e2*构成一个概念，在文中将使用的形式表述一个粒度。

定义 3.1.2给定粒度，则该粒度的可辨识关系。

粒度的可辨识关系表现了一个概念的外延，体现了一个粒度的分类能力。根据定义可知粒度的可辨识关系满足：

**对称性**：若，则。

对应粒度的可辨识关系定义粒度的不可辨识关系。

定义 3.1.3给定粒度，则该粒度的相应的不可辨识关系。

粒度的不可辨识关系描述了概念的内涵（即一个概念的内在组成信息）与粒度的泛化能力。另外需要注意的是该关系满足：

**自反性**：若，必有

**传递性**：若有，即属于同一个概念，属于同一个概念，故属于同一个概念，

**对称性**：若有，即属于同一个概念，故属于同一个概念，

定理 3.1.1 给定决策表，粒度，其中。则必有

证明：对于，若，若，故得证。

定义 3.1.4 粒度空间可以用二元组表示。为一个粒度。为概念的运算符集合。由基本概念空间中的概念做中的运算次得到的粒度的集合为粒度空间。

考虑到研究内容的限制，本文只关注由定义3.1.1所描述的粒度导出的祖先粒度空间、子粒度空间。祖先粒度空间与子粒度空间定义如下。

定义3.1.5给定粒度，（合并运算符、二目运算符），称粒度空间为的祖先粒度空间。（分裂运算符、二目运算符，该运算符右边的概念必须为左边的子集），称粒度空间为的子粒度空间。对于，称是的子粒度，是的祖先粒度。

粒度的祖先粒度空间规模表现了粒度的分类能力，粒度的子粒度空间规模表现了粒度的泛化能力。

为更好的理解上述概念，此处给出一个小例子。

例3.1.1 给定粒度。则祖先粒度空间为：



只有两个概念，最多能做一次合并运算)，的子粒度空间为

。

定理 3.1.2 对于，则对于，使得。

定理 3.1.3 对于，则对于，使得。

定理 3.1.4 是一个决策表。给定，若，则，。

定理 3.1.5 给定粒度，则必有。

证明：根据定理3.1.2知满足。由粒度可辨识关系的定义知有，已知，所以。

定理 3.1.6 给定粒度，则必有

证明：根据定理3.1.3知满足。由粒度可辨识关系的定义知有，已知，所以。

定义 3.1.6给定基本概念空间、粒度，两个粒度的拼接结果为在两个粒度中均属于同一个概念**}**

粒度的拼接结果可以由基本概念空间中的概念做若干次合并运算得到，可以由中任意一个粒度做若干次分裂运算得到，从各自的角度观察，拼接是一种分裂运算。

例3.1.2 给定基本概念空间和具体粒度，。则。可以由中的概念做分裂得到。

定义 3.1.7 是一个决策表。粒度，，使 ，则称在决策表中可被表达（具有可表达性），称为的表达属性，为的表达粒度；否则，称粒度在下不能被表达（不具有可表达性）。

例子3.1.3设决策表，，，，。则在决策表中可被表达，属性表达为，在决策表中不可被表达。

定理 3.1.7给定决策表，则对于任意分类特征,属性约简表达的粒度一定是的祖先粒度。

证明：由定理3.1.4结合定义3.1.6可证。

**3.2 属性约简粒度的可辨识关系与不可辨识关系**

本节主要分析属性约简的粒度的可辨识关系与不可辨识关系。

定义 3.2.1 给定决策表，那么不同目标粒度的可辨识关系如下：

正域保持(*POS*)目标粒度的可辨识关系：

。

广义决策保持(*GEN*)目标粒度的可辨识关系：

。

分布保持(*DIS*)目标粒度的可辨识关系：

。

最大分布保持(*MDS*)目标粒度的可辨识关系：

。

相对不可分辨关系保持(*IND*)目标粒度的可辨识关系：

。

为方便叙述，使用表示属性约简目标下粒度的可辨识关系。需要注意以上所有二元关系满足

**对称性：**若，则。

定理 3.2.1给定决策表与下的属性约简，则有。

证明：可由定理3.1.5、3.1.7证明成立，此处只证明，使用反正法如下。

**正域保持属性约简**：设，，则。

情况1： ，则有，与是正域保持属性约简矛盾。

情况2：如果 ，则有， 则有，与是正域保持属性约简矛盾。

**广义决策保持属性约简**：假设，使得，已知，而，那么，与是广义决策保持属性约简矛盾。

**分布保持属性约简**：假设，使得。已知，但由于，则有。则有，与是分布保持属性约简矛盾。

**最大分布保持属性约简**：假设，使得。已知，。若，则必定有；若，则必定有。与是最大分布保持属性约简矛盾。

**相对不可分辨关系保持属性约简**：假设，使得，也即有，也即有。

情况1：，由于，则使得。， 。而与是相对不可分辨关系保持属性约简矛盾。

情况2：。由于，则使得。， 。而，与是相对不可分辨关系保持属性约简矛盾。

情况3：。由可知，也即是，而根据假设有，这与是相对不可分辨关系保持属性约简矛盾。

定义 3.2.2 给定决策表，为论域中的对象，那么不同目标粒度的不可辨识关系表示如下

正域保持(*POS*)目标粒度的不可辨识关系：

。

广义决策保持(*GEN*)目标粒度的不可辨识关系：

。

分布保持(*DIS*)目标粒度的不可辨识关系：

。

最大分布保持(*MDS*)目标粒度的不可辨识关系：

。

相对不可分辨关系保持(*IND*)目标粒度的不可辨识关系：

。

根据以上定义，可知具有以下性质：

**自反性：**对于。

**传递性：**对于，则。

**对称性：**对于，则。

证明：由于满足对称性，而，所以满足自反性、传递性、对称性。

定理 3.2.2给定决策表与下的属性约简，则有。

证明：由， ，可知定理成立，由定理3.1.5、3.1.7可知显然成立。

**3.3 属性约简的粒度空间**

定理 3.3.1 给定决策表和属性约简目标，，如果一个粒度的可辨识关系满足，且该粒度在决策表中能够被属性集表达，那么一定是属性约简目标下的属性约简或属性约简超集。

证明：此处证明对于必满足2.6-2.10属性约简定义中的第一个条件，分不同类型使用反证法证明。

**正域保持属性约简**：分两种情况：

情况1：设，。由于可知。所以。，则有，与相悖。

情况2：设，，由可知。由可知，与相悖。

**广义决策保持属性约简**：设，可知，由可知，与相悖。

**分布保持属性约简**：设，可知，由可知，则有，与相悖。

**最大分布保持属性约简**：设，可知，由可知，则有，与相悖。

**相对不可分辨关系保持属性约简**：分两种情况讨论：

情况1：。由可知。由可知。此处证明逻辑式为假，与假设中相悖。

情况2：。由可知。由可知。则可以得到下式。对假设中的分情况讨论:

I、，可知，根据可知，与假设导出的矛盾。

II、，可知，根据可知，与假设导出的矛盾。

III、，可知，根据可知，与假设导出的矛盾。

综合以上所有情况可得定理3.3.1成立。

定理 3.3.2 给定决策表和属性约简目标，，如果粒度的不可辨识关系满足，且该粒度在决策表中能够被属性集表达，也就是，那么一定是属性约简目标下的属性约简或属性约简的超集。

证明：由， ，结合定理3.3.1可知成立，由定理3.1.5、3.1.7可知显然成立。

定义 3.3.3给定决策表属性约简目标，，那么五种属性约简的目标粒度定义如下：

正域保持属性约简目标粒度：；

广义决策保持属性约简目标粒度：；

分布保持属性约简目标粒度：；

最大分布保持属性约简目标粒度：；

相对不可分辨关系保持属性约简目标粒度：；

定理 3.3.3给定决策表属性约简的目标粒度，所有目标下属性约简的粒度一定为的子粒度。

证明：由定理3.3.1和定理3.3.2可知显然成立。

定理 3.3.4给定决策表与属性约简的目标粒度。为目标下属性约简的充分必要条件为：

1. ；
2. 。

证明：由定理3.3.1和定理3.3.2结合五种属性约简的定义可知显然成立。

1. **粒度搜索算法**

本节主要介绍属性约简目标粒度的求解、粒度的评价函数和重要度函数，并探究了重要度函数的一些性质，最后借助评价函数和重要度函数构建了粒度搜索算法。

此处给出根据定理3.3.2计算属性约简目标粒度的算法。

算法 4.1 **属性约简目标粒度构建算法**

输入：决策表，属性约简目标。

输出：属性约简目标粒度

初始化：

Step 1：计算，并令

Step 2：检查，其中：

若存在，则，继续执行Step 2。

否则执行Step 3.

Step 3：返回

以下介绍粒度近似的概念。

定义 4.1 给定决策表与属性约简目标粒度。粒度评价函数定义为

。

称为属性集下的粒度近似。

定理 4.1 给定决策表与属性约简的目标粒度。若有，那么。

证明： 已知，则，根据定理3.1.1、3.1.2可知。

定理 4.2 给定决策表与属性约简的目标粒度。为下的属性约简的充分必要条件为：

1. ；
2. 。

证明：：设满足，。由可知。由于可知对于任意。

：设为目标下的属性约简，那么根据定义3.3.3可知，根据定理3.1.1、3.1.2可知。由于对于任意必须满足，即，那么对于，，故得证。

定义 4.3给定决策表与属性约简目标粒度。外部属性重要度为

。

内部属性重要度定义为

。

定理 4.3 给定决策表和属性约简目标粒度，，令

；

；

。

对于，有

。

证明：已知





结合，

，

，，

可得

也就是。

定理 4.4 给定决策表与属性约简的目标粒度。为属性约简的充分必要条件为：

1. ；
2. 。

其中。

根据以上内容构建以下粒度搜索算法

算法 4.2 **粒度搜索算法**

输入：决策表，属性约简目标

输出：属性约简

初始化：

Step 1：使用算法4.1计算。

Step 2：如果，执行Step 3；否则执行Step 2.1。

Step 2.1：计算，其满足

。

Step 2.2：令，执行Step 2。

Step 3：对，若，则。

Step 4：，返回。

算法Step 2所做工作为在粒度空间搜索一个尽可能小的属性集合其粒度尽可能接近，Step 3所做工作为在粒度空间中搜索一个可表达粒度，的属性表达满足为的属性表达的真子集，两个步骤均可解释为在特定的粒度空间里搜索满足条件的粒度，故该算法命名为粒度搜索算法。

1. **算法复杂度分析与实验对比**

本节主要分四部分内容，第一部分为使用算例指出已有最大分布保持属性约简启发式算法的正确性问题，另外使用算例来验证粒度搜索算法求取最大分布保持属性约简与相对不可分辨关系保持属性约简的正确性.。第二部分主要分析属性约简定义、启发式属性约简算法重要度的直观简洁性。第三部分为算法复杂度分析，该部分将给出已有算法和粒度搜索算法的复杂度上界。第四部分为已有启发式属性约简算法、粒度搜索算法在现实环境中求解属性约简耗费时间、得到的属性约简结果的对比，该节主要体现粒度搜索算法的高效性、正确性。

* 1. **算法的正确性验证**

考虑到正域保持属性约简、广义决策保持属性约简、分布保持属性约简已有启发式算法，故正确性验证部分主要放在实验部分，此处主要展现使用粒度求取最大分布保持属性约简、相对不可分辨关系保持属性约简的正确性，其余三种类型属性约简的求取正确性读者可以自行验证。

算例 4.1 对于以下决策表，求取相应目标的属性约简

表 5.1 决策表

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 属性  对象 | 条件属性 | | | | | | 决策属性 |
|  |  |  |  |  |  |  |
|  | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
|  | 1 | 0 | 0 | 0 | 0 | 0 | 2 |
|  | 1 | 1 | 0 | 0 | 0 | 0 | 1 |
|  | 1 | 1 | 0 | 0 | 0 | 0 | 2 |
|  | 1 | 1 | 0 | 0 | 1 | 1 | 1 |
|  | 1 | 1 | 0 | 0 | 1 | 1 | 2 |
|  | 1 | 1 | 0 | 1 | 1 | 1 | 1 |
|  | 1 | 1 | 0 | 1 | 1 | 1 | 2 |
|  | 1 | 1 | 0 | 1 | 1 | 1 | 2 |
|  | 1 | 1 | 1 | 1 | 1 | 1 | 2 |
|  | 1 | 1 | 1 | 1 | 1 | 1 | 3 |

以上决策表论域由属性导出的划分如下：



。

此处给出计算正域保持属性约简目标粒度的过程：

Step 1： ；

Step 2： 由于，所以，故合并两个概念后得到。

由于，故合并两个概念得到。

由于，故合并两个概念得到

由于不存在两个概念可以合并，结束Step 2。

Step 3：返回。

考虑篇幅限制，其余属性约简目标粒度的计算过程不在此展现，直接罗列如下：

广义决策保持属性约简目标粒度



分布保持属性约简目标粒度



最大分布保持属性约简目标粒度



相对不可分辨关系保持属性约简目标粒度



**已有最大分布保持属性约简评价函数正确性问题**：此处指出定义2.19最大分布保持属性约简的评价函数与重要度函数的问题，使用该重要度函数求取决策表5.1的最大分布保持属性约简过程如下：

初始化：。

Step 1：计算迭代停止点。

Step 2：,计算,,,,,,故。

由于,,,,,,故。

由于，执行Step 3。

Step 3：检查是否有，无冗余属性，执行Step 4。

Step 4：返回。

事实上对于决策表5.1属性集合不是最大分布保持属性约简，对象。所以已有的最大分布保持属性约简启发式算法用于判断属性约简的定理不正确。

考虑到最大分布保持属性约简的启发式算法存在问题，相对不可分辨关系保持属性约简的启发式算法尚未被提出，此处使用粒度搜索算法计算决策表5.1的最大分布保持属性约简与相对不可分辨关系保持属性约简来验证所提算法的正确性。

**使用粒度搜索算法求取最大分布保持属性约简的正确性验证**：计算过程如下

初始化：

Step 1：使用算法4.1计算。

Step 2：由于。计算属性的外部重要度，，，。故令，，。

由于。计算属性的外部重要度，，，，故令，，。

由于。计算属性的外部重要度，，，故，，。

由于。计算属性的外部重要度，，，故，，。

由于，执行Step 3。

Step 3：对，检查是否 ，无冗余属性。

Step 4：返回。

**使用粒度搜索算法求取相对不可分辨关系保持属性约简的正确性验证**：计算过程如下

Step 1：使用算法4.1计算。

Step 2：由于。计算属性的外部重要度，，，。故令，，。

由于。计算属性的外部重要度，，，，故令，，。

由于。计算属性的外部重要度，，，故，，。

由于。计算属性的外部重要度，，，故，，。

由于。计算属性的外部重要度，，，故，，。

由于，执行Step 3。

Step 3：对，检查是否 ，无冗余属性。

Step 4：返回。

* 1. **属性约简的表达、重要度函数的直观性、简洁性对比**

此处从理论直观和算法复杂程度两个角度比较已有属性约简的定义上的差异。

在理论直观方面，使用已有的属性约简定义表达决策表5.1的广义决策保持数属性约简，，且。使用粒度空间表达如下，如图5.1所示，从左到右的粒度分别为，可以看出，是的子粒度，不是的子粒度，所以、是属性约简或者属性约简的超集。考察属性必要性后我们可以知道为广义决策保持属性约简。

C:\Users\bricklee\AppData\Local\Temp\1541646514(1).png C:\Users\bricklee\AppData\Local\Temp\1541494336(1).png C:\Users\bricklee\AppData\Local\Temp\1541497371(1).png C:\Users\bricklee\AppData\Local\Temp\1541497300(1).png

图 5.1 的粒度分布图

可以看出使用粒度来描述属性约简的优势在于直观、简单。对于正域保持属性约简的表达，使用粒度的形式与已有的定义在直观、简单方面没有太大差异，但对于广义决策保持属性约简、分布保持属性约简、最大分布保持属性约简、相对不可分辨关系保持属性约简，使用粒度来表达相应的属性约简将更加简单、直观。

在此比较不同启发式算法所使用重要度函数的差异。依旧以决策表5.1的广义决策保持属性约简计算为例，，当时，,,,,，到此我们知道属性重要度最大。我们来看看对于相同的情况下粒度近似的重要度计算。当时，目标粒度和论域由各属性导出的粒度如图5.2所示，通过分布图可以看出目标粒度与之间关联最为紧密。具体的计算,,,,，到此我们知道属性重要度最大。

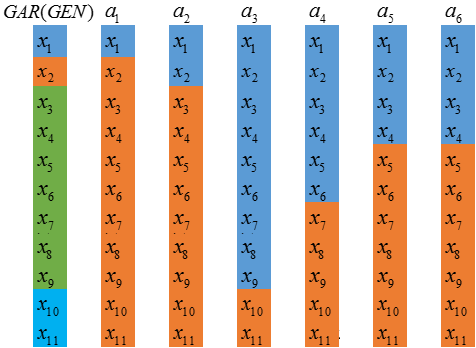


图 5.2 与各个属性导出粒度的粒度分布图

可以察觉本文使用的重要度函数——粒度近似不仅展现了重要度的量的区别，也展现了重要度的质的区别，相比原有重要度的函数抽象形式，粒度近似只用了元素包含于集合之中这种简单基础的概念，它可以使用或者说更便于使用图来表现，因此更加直观简单。

从算法复杂度的角度看，未完待续。。。

|  |  |  |
| --- | --- | --- |
| 属性约简定义 | 算法复杂度下界 | 算法复杂度上界 |
| *POSB(D)= POSC(D)* |  |  |
| *Gran(B,GAR(POS),U)=U* |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
| *IND(B|D)=IND(C|D)* |  |  |
|  |  |  |

* 1. **算法复杂度分析与对比**

算法复杂度上界

表 5.1 启发式算法复杂度

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 算法名称 | 保持的分类特征 | *Step 1* | *Step 2* | *Step 3* |
| ***PR*** | 正域保持 | *O*(*|U||C|*) |  | *O*(*|U||C|*) |
| ***Q-ARA*** | 广义决策保持 | *O*(*|U||C|*) |  | *O*(*|U||C|*) |
| ***Q-DRA*** | 分布保持 | *O*(*|U||C|*) |  | *O*(*|U||C|*) |
| ***GS*** | — | *O*(*|U||C|*) |  | *O*(*|U||C|*) |

表 5.2 粒度近似与正域近似加速后启发式算法复杂度分析

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 算法名称 | 保持分类特征 | *Step 1* | *Step 2* | *Step 3* |
| *FSPA-PR* | 正域保持 | *O*(*|U||C|*) |  | *O*(*|U||C|*) |
| *FSPA-Q-ARA* | 广义决策保持 | *O*(*|U||C|*) |  | *O*(*|U||C|*) |
| *FSPA-Q-DRA* | 分布保持 | *O*(*|U||C|*) |  | *O*(*|U||C|*) |
| *GA-GR* | — | *O*(*|U||C|*) |  | *O*(*|U||C|*) |

增加粒度近似与正域近似的比较关系的证明。

* 1. **实验分析与对比**

本文所有程序均使用python3实现运行在搭载Intel(R) Core(TM) i5-4200U 1.6GHz的CPU、4G容量的内存、Windows8.1（64bit）系统的PC上。

实验数据集选用

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Data sets | Cases | attributes | classes |  |
| shuttle(int) | 58000 | 9 | 19 | 0.23 |
| pid | 768 | 8 | 2 | 0.519 |
| mushroom | 5644 | 22 | 2 | 0.534 |
| heart | 270 | 6 | 2 | 0.935 |
| glass | 214 | 9 | 7 | 0.937 |
| Ions | 351 | 34 | 2 | 0.94 |
| vehicle | 846 | 18 | 4 | 0.946 |
| tic | 9822 | 85 | 2 | 0.968 |
| wave | 5000 | 21 | 3 | 0.981 |
| wdbc | 569 | 30 | 2 | 0.989 |
| segment | 2310 | 19 | 7 | 0.991 |
| sat | 6435 | 36 | 6 | 0.993 |
| wpbc | 198 | 34 | 2 | 1 |
| wine | 178 | 13 | 3 | 1 |
| sonar | 208 | 60 | 2 | 1 |
| tic-tac-toe | 958 | 9 | 2 | 1 |
| dermatology | 358 | 34 | 6 | 1 |
| kr-vs-kp | 3196 | 36 | 2 | 1 |
| breast-cancer-wisconsin | 683 | 9 | 2 | 1 |
| letter-recognition | 20000 | 9 | 26 | 1 |

正域保持属性约简算法时间效率对比

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Data sets | PR | FSPA-PR | GS-POS | GRGS-POS |
| shuttle(int) | 19.51871845 | 28.37645248 | 19.49334013 | 22.57602781 |
| pid | 0.21044269 | 0.27446003 | 0.21378673 | 0.26306346 |
| mushroom | 4.93032414 | 3.98189738 | 3.15866418 | 1.64034666 |
| heart | 0.1808601 | 0.16185427 | 0.18102496 | 0.16035641 |
| glass | 0.06783481 | 0.07948604 | 0.06692005 | 0.06931086 |
| Ions | 0.6934107 | 0.40526528 | 0.69633393 | 0.40440826 |
| vehicle | 1.04120111 | 0.7618517 | 1.03974174 | 0.74703924 |
| tic | 130.8836339 | 33.57291727 | 130.8192021 | 30.88906722 |
| wave | 9.47202753 | 8.56440673 | 9.48463907 | 8.45996831 |
| wdbc | 1.14312427 | 0.62455529 | 1.13541623 | 0.60752138 |
| segment | 1.220725 | 0.61304326 | 1.2294158 | 0.60897947 |
| sat | 33.08210971 | 17.41063641 | 32.85905035 | 17.0447036 |
| wpbc | 0.40205403 | 0.59012759 | 0.39943934 | 0.58742823 |
| wine | 0.0771921 | 0.05417958 | 0.07709909 | 0.05386782 |
| sonar | 0.58271976 | 0.38142011 | 0.57827493 | 0.37237201 |
| tic-tac-toe | 0.34608536 | 0.33855756 | 0.34589868 | 0.33692692 |
| dermatology | 0.49882609 | 0.24088165 | 0.49903265 | 0.24279712 |
| kr-vs-kp | 14.67861748 | 7.86734151 | 14.65027295 | 7.86477109 |
| breast-cancer-wisconsin | 0.14676725 | 0.09313101 | 0.14575435 | 0.09293536 |
| letter-recognition | 18.91574454 | 13.25145199 | 18.89526265 | 13.3798921 |

广义决策保持属性约简算法时间效率对比

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Data sets | PR | FSPA\_PR | GA | GRGA |
| shuttle(int) | 19.72652721 | 23.81085837 | 19.72736114 | 21.12343976 |
| pid | 0.2125506 | 0.27891513 | 0.21019572 | 0.26050267 |
| mushroom | 4.26813502 | 3.60696457 | 3.72761645 | 2.0674512 |
| heart | 0.18188262 | 0.16397244 | 0.17902867 | 0.16039361 |
| glass | 0.06921399 | 0.06315519 | 0.06594949 | 0.06280879 |
| Ions | 0.72238901 | 0.40748289 | 0.69351847 | 0.40304447 |
| vehicle | 1.04353354 | 0.78042582 | 1.01870626 | 0.73846583 |
| tic | 137.329063 | 33.26651207 | 130.2013208 | 29.95058664 |
| wave | 9.50896728 | 8.31530384 | 9.31469443 | 8.30828795 |
| wdbc | 1.16005555 | 0.61312601 | 1.14343924 | 0.59299245 |
| segment | 1.1915427 | 0.58346067 | 1.19404589 | 0.58188839 |
| sat | 33.85444002 | 17.69415225 | 32.33591402 | 17.42929649 |
| wpbc | 0.38038989 | 0.2077151 | 0.36612397 | 0.20351211 |
| wine | 0.07043602 | 0.5861241 | 0.06957322 | 0.58507528 |
| sonar | 0.55461835 | 0.37838911 | 0.53419228 | 0.37061178 |
| tic-tac-toe | 0.32012907 | 0.33558814 | 0.31445452 | 0.333257 |
| dermatology | 0.4257895 | 0.2442366 | 0.45879696 | 0.25057445 |
| kr-vs-kp | 14.41426079 | 8.11095716 | 14.18205839 | 7.87343879 |
| breast-cancer-wisconsin | 0.13471958 | 0.09376351 | 0.12922721 | 0.09212581 |
| letter-recognition | 20.51935223 | 14.22786919 | 18.48431961 | 13.26495968 |

分布保持属性约简算法时间效率对比(s)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Data sets | PR | FSPA\_PR | GA | GRGA |
| shuttle(int) | 19.81904047 | 24.01229208 | 19.47651342 | 21.84878728 |
| pid | 0.21938238 | 0.28107756 | 0.21644568 | 0.27310458 |
| mushroom | 3.13270532 | 2.89773429 | 3.70388995 | 2.14403197 |
| heart | 0.18655197 | 0.16110694 | 0.18094734 | 0.16064764 |
| glass | 0.07243423 | 0.06560052 | 0.06914343 | 0.06409047 |
| Ions | 0.68539345 | 0.45110816 | 0.68142589 | 0.42654068 |
| vehicle | 1.11044205 | 0.81017198 | 1.02362579 | 0.75949746 |
| tic | 151.5381804 | 35.52771716 | 130.4260673 | 30.00328503 |
| wave | 10.00735652 | 8.49216494 | 9.24546761 | 8.33527253 |
| wdbc | 1.14908749 | 0.71237226 | 1.10377512 | 0.58710749 |
| segment | 2.04907595 | 0.65749027 | 1.58372111 | 0.60879729 |
| sat | 36.19289174 | 18.68179217 | 32.6945102 | 16.7990134 |
| wpbc | 0.34326283 | 0.19080756 | 0.3659527 | 0.20874533 |
| wine | 0.06100366 | 0.04434117 | 0.06977529 | 0.05505584 |
| sonar | 0.58080301 | 0.37357158 | 0.52809627 | 0.37326688 |
| tic-tac-toe | 0.32827012 | 0.34122485 | 0.31302144 | 0.3359833 |
| dermatology | 0.51883006 | 0.25672241 | 0.46391278 | 0.24561002 |
| kr-vs-kp | 14.61533784 | 7.72532315 | 14.19964205 | 7.80233435 |
| breast-cancer-wisconsin | 0.13694039 | 0.0979928 | 0.13061858 | 0.09500799 |
| letter-recognition | 22.55785142 | 14.83163854 | 19.38712866 | 14.35084579 |

1. **总结与展望**

本文从粒度的角度研究了属性约简的粒度上的特点，并指出所有属性约简的粒度均处于可定义的粒度空间中，借助粒度空间，我们构建了统一的正域保持属性约简、广义保持属性约简、分布保持属性约简、最大分布保持属性约简、相对不可分辨关系保持属性约简的启发式算法，理论分析和实验对比表明，本文所提的属性约简定义更加简单直观，所提算法更加高效！

本文指出所有属性约简所表达的粒度均处于一个粒度空间，根据这个定理设计更加高效的求取所有属性约简的算法、求取最小属性约简的算法是未来工作的一部分。另外，使用二元关系分析拓展模型中的属性约简定义与粒度的关系，借助粒度与粒度空间设计更加高效的属性约简算法是未来的方向之一。

**致谢**

本文想法萌芽于2017年，期间张楠老师一直给予我学术上的指导和精神上、生活上的帮助、支持、鼓励，在此特别感谢张楠老师；和高学义同学的交流讨论是这篇文章的想法成型外在条件之一，特别感谢高学义同学，你的认真、刻苦激励着我继续前行；感谢陈曼如同学给予论文上的修改建议。特别感谢黎敏老师为我提供文献[59]的实验数据，这对本文的实验工作非常重要。未完待续…

**参考文献**

1. K. Kira, L.A. Rendell, The feature selection problem: traditional methods and a new algorithm, Proc. AAAI 92 (1992) 129–134.
2. R. Jensen, Q. Shen, Computational Intelligence and Feature Selection: Rough and Fuzzy Approaches, IEEE Press/Wiley & Sons, 2008.
3. M. Modrzejewski, Feature selection using rough set theory, in: Proceedings of European Conference on Machine Learning, 1993, pp. 213–226.
4. Pawlak Z. Rough sets[J]. International Journal of Information and Computer Science, 1982(11): 314-356
5. Koczkodaj W W, Orlowski M, Marek V W. Myths about rough set theory[J]. Communications of the ACM, 1998, 41(11): 102-103
6. Pawlak Z, Skowron A. Rudiments of rough sets[J]. Information Sciences, 2007, 177: 3-27.
7. Pawlak Z. Rough Sets: Theoretical Aspects of Reasoning about Data[M]. The Netherlands: Kluwer Academic Publishers, 1992
8. Pawlak Z. Rough set theory and its applications to data analysis. Cybernetics and Systems, 1998, 29(7): 661-688.
9. Z. Pawlak, Rough set approach to multi-attribute decision analysis, European Journal of Operational Research 72 (1993) 443–459.
10. Z. Pawlak, Rough sets and intelligent data analysis, Information Sciences 147 (2002) 1–12.
11. Z. Pawlak, A. Skowron, Rough membership functions, in: R.R. Yager, M. Fedrizzi, J. Kacprzyk (Eds.), Advances in the Dempster–Shafer Theory of Evidence, John Wiley and Sons, New York, 1994, pp. 251–271.
12. A. Skowron, S.K. Pal, Rough Sets, Pattern Recognition, and Data Mining, Pattern Recognition Letters 24 (6) (2003) 829–933.
13. Jelonek J, Krawiec K, Slowinski R. Rough set reduction of attributes and their domains for neural networks. International Journal of Computational Intelligence, 1995, 11: 339-347.
14. Shen Q, Chouchoulas A. A rough-fuzzy approach for generating classification rules. Pattern Recognition, 2002, 35: 2425-2438.
15. Dimitras A I, Slowinski R, Susmaga R, et al. Business failure using rough sets. European Journal of Operational Research, 1999, 114: 263-280.
16. Anantaram C, Nagaraja G, Nori K V. Verification of accuracy of rules in a rule based system. Data & Knowledge Engineering, 1998, 27: 115-138.
17. Mushrif M M, Ray A K. Color image segmentation: rough-set theoretic approach. Pattern Recognition Letters, 2008, 29: 483-493.
18. Hu X H, Cercone N. Learning in relational databases: A rough set approach. International Journal of Computational Intelligence, 1995, 11(2): 323-338.
19. Wang Guoyin, Yao Yiyu, Yu Hong. A Survey on Rough Set Theory and Applications. Chinese Journal of Computer, 2009,32(7):1229-1246(in Chinese).
20. 张文修, 吴伟志. 粗糙集理论介绍和研究综述[J]. 模糊系统与数学, 2000, 14(4): 1-12
21. 黄正华, 胡宝清. 模糊粗糙集理论研究进展[J]. 模糊系统与数学, 2005, 19(4): 125-134

启发式属性约简算法

1. 苗夺谦, 胡桂荣. 知识约简的一种启发式算法[J]. 计算机研究与发展, 1999, 36(6): 681-684

Miao Duoqian, Hu Guirong. A heuristic algorithm for reduction of knowledge. Journal of Computer Research & Development, 1999,26(6):681-684.

1. 王国胤, 于洪, 杨大春. 基于条件信息熵的决策表约简[J]. 计算机学报, 2002, 25(7): 759-765

G.Y. Wang, H. Yu, D. Yang, Decision table reduction based on conditional information entropy, Chinese Journal of Computers 25 (7) (2002) 759–766.

1. Slezak D. Various approaches to reasoning with frequency based decision reducts: a survey[J]. In: Rough Set Methods and Applications: New Developments in Knowledge Discovery in Information Systems. Physica-Verlag Gmbh, Heidelberg, Germany, 2000: 235-285
2. 张文修, 米据生, 吴伟志. 不协调目标信息系统的知识约简[J]. 计算机学报, 2003, 26(1): 12-18

W.X. Zhang, J.S. Mi, W.Z. Wu, Knowledge reduction in inconsistent information systems, Chinese Journal of Computers 26 (2003) 12–18.

1. Kryszkiewicz M. Comparative study of alternative types of knowledge reduction in inconsistent systems[J]. International Journal of Intelligent Systems, 2001, 16: 105-120
2. Miao D Q, Zhao Y, Yao Y Y, et al. Relative reducts in consistent and inconsistent decision tables of the Pawlak rough set model[J]. Information Sciences, 2009, 179(24): 4140-4150
3. Y.H. Qian, J.Y. Liang, Combination entropy and combination granulation in rough set theory, International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems 16 (2) (2008) 179–193.

约简类型

1. Li D Y, Zhang B. On knowledge reduction in inconsistent decision information systems. International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems, 2004, 12(5): 651-672.
2. Qin K Y, Pei Z, Du W F. The relationship among several knowledge reduction approaches. In: Proceedings of FSKD 2005, LNAI 3613. Berlin: Springer-Verlag, 2005: 1232-1241.
3. 邓大勇, 黄厚宽, 李向军. 不一致决策系统中约简之间的比较. 电子学报, 2007, 35(2): 252-255.
4. 徐章艳, 杨炳儒, 宋威, 等. 几种不同属性约简的比较研究. 小型微型计算机系统, 2008, 29(5): 848-853.
5. Zhou J, Miao D Q, Pedrycz W, Zhang H Y. Analysis of alternative objective functions for attribute reduction in complete decision tables[J]. Soft Computing, 2011(15): 1601-1616
6. Dubois D, Prade H. Rough fuzzy sets and fuzzy rough sets[J]. International Journal of General Systems, 1990, 17(2): 191-209
7. Zhu W, Wang F Y. Reduction and axiomization of covering generalized rough sets[J]. Information Sciences, 2003, 152(1): 217-230
8. 魏莱, 苗夺谦, 徐菲菲, 等. 基于覆盖的粗糙模糊集模型研究[J]. 计算机研究与发展, 2006, 43(10): 1719-1723
9. 王基一, 许黎明. 概率粗糙集模型[J]. 计算机科学, 2002, 29(8): 76-78.
10. Yao J T, Yao Y Y, Ziarko W. Probabilistic rough sets: Approximations, decision-makings, and applications[J]. International Journal of Approximation Reasoning, 2008, 49(2): 253-254.
11. Slezak D, Ziarko W. Bayesian rough set model. In: Proceedings of the International Workshop on Foundation of Data Mining (FDM'2002). Maebashi, Japan, 2002: 131-135.
12. W. Ziarko, Variable precision rough set model, Journal of Computer and System Sciences 46 (1993) 39–59.
13. Skowron A, Rauszer C. The discernibility matrices and functions in information systems[J]. In: Slowinski R, (Eds.). Intelligent Decision Support Handbook of Applications and Advances of the Rough Sets Theory. Dordrecht: Kluwer Academic Publishers, 1992(11): 331-362
14. Wang Jue, Wang Ju. Reduction algorithms based on discernibility matrix: the ordered attributes method[J]. Journal of Computer Science & Technology, 2001, 16(6): 489-504
15. Yao Y, Zhao Y. Discernibility matrix simplification for constructing attribute reducts[J]. Information Sciences, 2009, 179(7): 867-882
16. Yang M. An Incremental Updating Algorithm for Attribute Reduction Based on Improved Discernibility Matrix[J]. Chinese Journal of Computers, 2007, 30(5): 815-822
17. Wang R Z, Miao D Q, Hu G R. Discernibility matrix based algorithm for reduction of attributes. In: Butz C J, Nguyen N T, Takama Y, (Eds.). Web Intelligence and Intelligent Agent Technology. Hong Kong: IEEE Computer Society, 2006: 477-480.
18. Dong-Yi YE, Chen Z J. A New Discernibility Matrix and the Computation of a Core[J]. Acta Electronica Sinica, 2002, 30(7):1086-1088.
19. Yang M. An Incremental Updating Algorithm for Attribute Reduction Based on Improved Discernibility Matrix[J]. Chinese Journal of Computers, 2007, 30(5):815-822.
20. Zhang W X, Wei L, Qi J J. Attribute Reduction in Concept Lattice Based on Discernibility Matrix[J]. Rough Sets Fuzzy Sets Data Mining & Granular Computing Pt Proceedings, 2005, 3642:157-165.
21. Yao Y, Zhao Y, Wang J. On Reduct Construction Algorithms[C] //International Conference on Rough Sets and Knowledge Technology. Springer Berlin Heidelberg, 2006: 297-304
22. Wang J, Miao D Q. Analysis on attribute reduction strategies of rough set. Journal of Computer Science and Technology, 1998, 13(2): 189-193.
23. Miao D Q, Wang J. Information-based algorithm for reduction of knowledge. In: Proceedings of the 1997 IEEE International Conference on Intelligent Processing Systems (ICIPS’97). Beijing, China, 1997: 1155-1158.
24. Shi H J, Qin C, Chen H J, et al. Heuristic algorithm of attribute reduction in condition entropy[J]. Computer Engineering & Design, 2008.
25. Hua-Xiong LI, Zhou X Z. Heuristic attribute reduction based on 0-1 discernibility matrix[J]. Journal of Central South University, 2009, 40: 304-308
26. Nguyen S Hoa, Nguyen H Son. Some efficient algorithms for rough set methods. In: Proceedings of the Conference of Information Processing and Management of Uncertainty in Knowledge-Based Systems (IPMU'96). Granada, Spain, 1996: 1451-1456.
27. S.H. Liu, Q.J. Sheng, B. Wu, Z.Z. Shi, F. Hu. Research on Efficient Algorithms for Rough Set Methods[J]. Chinese Journal of Computers, 2003, 40(5):637-642.
28. S.X. Wu, M.Q. Li, W.T. Huang, S.F. Liu, An improved heuristic algorithm of attribute reduction in rough set, Journal of System Sciences and Information, 2 (3) (2004) 557–562.
29. Z.Y. Xu, Z.P. Liu, B.R. Yang, W. Song, A quick attribute reduction algorithm with complexity of max*(O(*|*C*||*U*|*), O(*|*C*|2|*U/C*|*))*, Chinese Journal of Computer, 29 (3) (2006) 391–398.
30. Qian Y H, Liang J Y, Pedrycz W, Dang C Y. Positive approximation: An accelerator for attribute reduction in rough set theory. Artificial Intelligence, 2010, 174(9-10): 597-618.
31. Li M, Shang C, Feng S, et al. Quick attribute reduction in inconsistent decision tables[J]. Information Sciences An International Journal, 2014, 254:155-180.
32. Frank A, Asuncion A. UCI Machine Learning Repository [http://www.ics.uci.edu/ml], Irvine, CA: University of California, School of Information and Computer Science, 2010.
33. Zhao Y, Luo F, Wong S K M, et al. A general definition of an attribute reduct[C] //Proceedings of the Second Rough Sets and Knowledge Technology (RSKT’07), 2007: 101-108