一种基于拆分的基因选择算法

王永全[1] 焦 娜[1] 苗夺谦[2]
(华东政法大学信息科学与技术系 上海 201620)[1] (同济大学计算机科学与技术系 上海 201804)[2]

摘 要 基因表达数据是由成千上万个基因及几十个样本组成的,有效的基因选择算法是基因表达数据研究的重要内容。粗糙集是一个有效的去掉冗余特征的工具。然而,对于含有成千上万特征、几十个样本的基因表达数据,现有基于粗糙集的特征选择算法的计算效率会变得非常低。为此,将拆分方法应用于特征选择,提出了一种基于拆分的特征选择算法。该算法把一个复杂的表拆分成简单的、更容易处理的主表与子表形式,然后把它们的结果连接到一起解决初始表的问题。实验结果表明,该算法在保证分类精度的同时,能明显提高计算效率。

关键词 特征选择,拆分,主表,子表,粗糙集,基因选择

Gene Selection Method Based on Decomposition

WANG Yong-quan[1] JIAO Na[1] MIAO Duo-qian[2]
(Department of Information Science and Technology, East China University of Political Science and Law, Shanghai 201620, China)[1]
(Department of Computer Science and Technology, Tongji University, Shanghai 201804, China)[2]

Abstract Efficient gene selection is a key issue for classifying microarray gene expression data, since the data typically consist of a huge number of genes and a few dozens of samples. Rough set theory is an efficient tool for further reducing redundancy. However, when handling numerous genes, most existing methods based on rough set theory gain worse performance. A gene selection method based on decomposition was presented. The idea of decomposition is to break a complex task down into a master-task and several sub-tasks that are simpler, more manageable and more solvable by using existing induction methods, then joining them together to solve the original task. To evaluate the performance of the proposed approach, we applied it to four benchmark gene expression data sets and compared our results with those obtained by conventional methods. Experimental results illustrate that our algorithm improve computational efficiency significantly while keeping classification accuracy.

Keywords Feature selection, Decomposition, Master-table, Sub-table, Rough set theory, Gene selection

1 引言

微阵列技术的发展为癌症的深入研究带来了曙光。癌症基因表达数据分析成为基因表达数据分析的重要研究内容之一,使得人们能够对正常和疾病两种状态下的基因表达状况进行比较研究,从而帮助人们更好地对癌症的诊断与分类以及癌症的防治等相关课题进行研究。

基因表达数据主要有以下两个特点:1)基因数量大(从几千个到几万个)。换句话说,数据是高维的。2)样本数量少(从几个到几十个,最多几百个)。目前基因表达数据只能收集到较少的实验样本。

在DNA微阵列实验中,虽然基因的数目成千上万,但实际上影响样本分类的,往往只有很少一部分关键基因,其他的基因往往是不相关的或是冗余的。过多的基因会导致噪声的增加,增加分类算法的复杂性,并影响到分类的效果。而且与分类无关的基因将会干扰相关基因的信息。因此通过微阵列数据找到在不同类别中有显著不同表达的基因子集,亦即找到与分类相关的关键基因,成为微阵列数据挖掘研究的一个主要问题。这一过程叫做基因选择。

粗糙集是一个有效的去除冗余特征的工具[1-7]。自1982年Z. Pawlak提出粗糙集以来,它已经被广泛应用在特征选择和分类领域中。很多研究工作者致力于使用粗糙集理论进行特征选择(或者称为属性约简)的研究[8]。然而,对于特征个数非常多的基因表达数据,常用的基于粗糙集的特征选择算法的计算效率往往会很低,甚至得不到结果。

本文的研究动机是为了更便于处理决策表属性约简问题,以至于能够有效处理大规模基因表达数据。因此,本文提出一种拆分的方法。拆分方法[9]的思想是把一个大型的复杂的任务拆分成若干个小的更容易处理的子任务,这些子任务能够用现有的方法来进行处理。将它们的结果合并到一起,

到稿日期:2011-02-10 返修日期:2011-04-19 本文受国家自然科学基金(60775038,60775036),国家社科基金(06BFX051),上海市教委重点学科(第五期)司法鉴定建设项目(J51102),教育部博士学科点专项科研基金(20060247039)资助。

王永全(1964—),男,博士,教授,主要研究方向为智能计算、网络与信息安全、模糊集、粗糙集等;焦 娜(1977—),女,博士,讲师,主要研究方向为人工智能、模式识别、粗糙集等,E-mail:zdx.jn@163.com;苗夺谦(1964—),男,教授,博士生导师,主要研究方向为人工智能、模式识别、粗糙集、数据挖掘等。

便能解决原来的初始问题。拆分的方法能够使原来的初始任务变得更容易解决并且消耗的时间较少。拆分方法被用在统计学的研究[10]、图形学的研究[11]和工程学的研究上[12]。拆分的思想已经用在了一些数据挖掘的研究上，例如不完备信息系统的拆分[13,14]、多 agent 系统的拆分[15]。

本文在前人工作启发下，对一个经典粗糙集特征选择算法进行改进，把拆分的思想用到特征选择算法当中，提出一种基于拆分的基因选择算法，并对时间复杂度进行了分析。实验结果表明，本文提出的算法不仅能找到高区分能力的特征子集，还能有效提高计算效率。

2 背景知识

本部分介绍以下基本概念和一个经典的粗糙集特征选择算法。

2.1 基本概念

基因表达数据中有一个类别属性，因此可以把基因表达数据看成粗糙集里面的决策表。

定义 1 决策表 $T=(U,C\cup D,V,f)$，其中 U 为非空有限论域；C 为条件属性集；D 为决策属性集；$V=\bigcup_{a\in C\cup D}V_a$ 为属性值域，其中 V_a 为属性 a 的值域；$f:U\times(C\cup D)\to V$ 为信息函数。

定义 2 给定决策表 $T=(U,C\cup D,V,f)$，$B\subseteq C\cup D$，论域 U 关于不可分辨关系 B 定义为：

$$IND(B)=\{(x,y)\in U\times U\mid \forall b\in B, b(x)=b(y)\}$$

显然，$IND(B)$ 是一个等价关系。

定义 3 给定决策表 $T=(U,C\cup D,V,f)$，$B\subseteq C\cup D$，论域 U 关于不可分辨关系 B 的划分定义为：

$$\frac{U}{IND(B)}=\{[x_i]_B:x_i\in U\}$$

式中，$[x_i]_B$ 表示 U 中所有与 x_i 在关系 $IND(B)$ 下是等价的元素构成的集合。

定义 4 给定决策表 $T=(U,C\cup D,V,f)$，$X\subseteq U$，$B\subseteq C$，X 关于 B 的上、下近似集分别记为 $\overline{B}X$，$\underline{B}X$：

$$\underline{B}X=\bigcup\{[x_i]_B\mid [x_i]_B\subseteq X\}$$

$$\overline{B}X=\bigcup\{[x_i]_B\mid [x_i]_B\cap X\neq\emptyset\}$$

上近似集由可能属于 X 的对象集构成，下近似集由肯定属于 X 的对象集构成。上、下近似集从两个侧面对概念 X 进行逼近，即概念 X 可用两个精确的概念近似描述，若概念 X 本身模糊或不确定，这种近似描述具有重要意义。

定义 5 给定决策表 $T=(U,C\cup D,V,f)$，$X\subseteq U$，$B\subseteq C$，决策属性集 D 相对条件属性集 B 的正区域定义为：

$$POS_B(D)=\bigcup_{X\in U/IND(D)}\underline{B}X$$

定义 6 给定决策表 $T=(U,C\cup D,V,f)$，$B\subseteq C$，若

(1) $POS_B(D)=POS_C(D)$

(2) $\forall a\in B, POS_{(B-\{a\})}(D)\neq POS_C(D)$

则称属性集 B 为决策表 T 的一个属性约简。

2.2 一般特征选择算法

特征选择算法在粗糙集里也被称为属性约简。找所有属性约简是 NP 难问题。然而，对于绝大多数的实际应用来说找到一个属性约简已经足够了。本文提出的属性约简算法是找一个约简。

一般约简算法[6]是对每一个特征进行判断，删除掉冗余的特征，直到满足终止条件，即得到一个约简。

一般约简算法的具体步骤如下。

算法 1 GAR 算法

输入：决策表 $T=\langle U,C\cup D,V,f\rangle$。

输出：属性约简 Red。

(1)令 $Red=C$；

(2)While 每个连接属性 $a\in C$ do

(3)Begin

(4) If $POS_{(Red-\{a\})}(D)=POS_{(Red)}(D)$, then

(5) Begin

(6) a 是不必要的属性并且在 T 中删除该属性，即 $Red=Red-\{a\}$；

(7) 合并相同的对象；

(8) End

(9)End

(10)输出 Red。

一般特征选择算法是一个很常用的经典属性约简算法，算法简单而且容易理解。当属性数相对较少时，该算法的运算时间是很短的。但在属性个数较多的情况下，例如基因表达数据，该算法的计算效率就会降低很多，因为一般特征选择算法必须对每一个特征进行判断。

为了解决这个问题，本文对一般特征选择算法进行了改进，使其可以处理大规模的基因表达数据。首先把一个大规模决策表转化成一个主表和几个子表的形式。主表是由决策属性集和连接属性集组成的，每个连接属性对应各个子表中的关键字。子表是由条件属性子集及连接属性组成的。然后把它们的结果连接起来解决原来表的问题。此算法使原来的大规模决策表变得更容易解决并且消耗的时间较少。本文根据这种主表，从表的拆分方法提出一种基于拆分的特征选择算法。基于拆分的特征选择算法的描述将在下一部分具体介绍。

3 基于拆分的基因选择算法

本部分首先介绍主表、子表的定义和若干性质，然后提出基于拆分的特征选择算法。

3.1 基本定义

把一个决策表拆分成一个主表和几个子表。主表是由决策属性集和几个连接属性组成的，连接属性是子表中的关键字；子表由条件属性的子集及连接属性构成。

定义 7 给定决策表 $T=(U,C\cup D,V,f)$。

(1)子表 $T^{B_i}=\langle U^{B_i},B_i\cup\{b_i\},V^{B_i},f^{B_i}\rangle$，$i=1,2,\cdots,m$，其中 U^{B_i} 为非空有限论域；$B_i\subseteq C$，$i=1,2,\cdots,m$，$C=\bigcup_{i=1}^{m}B_i$，$B_i\cap B_j=\emptyset$，$i\neq j$，b_i 是子表与主表的连接属性，它是 T^{B_i} 的关键字；$V^{B_i}=\bigcup_{a\in B_i}V_a^{B_i}$，其中 $V_a^{B_i}$ 为属性 a 的值域；$f^{B_i}:U^{B_i}\times B_i\to V^{B_i}$ 为信息函数。

(2)主表 $T^S=\langle U,S\cup D,V^S,f^S\rangle$，其中 U 为非空有限论域；$S=\bigcup_{i=1}^{m}\{b_i\}$ 是连接属性集，D 为决策属性集；$V^S=\bigcup_{a\in S\cup D}V_a^S$，其中 V_a^S 为属性 a 的值域；$f^S:U\times(S\cup D)\to V^S$ 为信息函数。

(3)中间表 $T^{M_i}=\langle U,M_i\cup D,V^{M_i},f^{M_i}\rangle$，$i=1,2,\cdots,m$，其中 U 为非空有限论域；$M_i=(S-\{b_i\})\cup B_i$，$i=1,2,\cdots,m$，D 为决策属性集；$V^{M_i}=\bigcup_{a\in M_i\cup D}V_a^{M_i}$，其中 $V_a^{M_i}$ 为属性 a 的值

域；$f^{M_i}:U\times(M_i\cup D)\to V^{M_i}$ 为信息函数。

例 1　表 1 是一个决策表，把它拆分成一个主表和两个子表，合并表 2 和表 3 构成中间表 5。类似地，表 6 是由表 2、表 4 组成的。图 1 表示了决策表、主表、子表和中间表之间的关系。决策表 1 通过子表 3 和子表 4 被压缩成主表 2。在运算过程中能够用到中间表 5 和表 6。事实上，从图 1 中能够看到主表 2、中间表 5 和表 6 与决策表 1 是完全等价的。下面给出它们之间的性质。

表 1　决策表

U	a_1	a_2	a_3	a_4	d
1	1	1	1	0	1
2	1	0	1	1	1
3	0	0	0	1	0
4	1	0	1	0	1

表 2　主表

U	b_1	b_2	d
1	b_1^1	b_2^1	1
2	b_1^2	b_2^2	1
3	b_1^3	b_2^3	0
4	b_1^2	b_2^1	1

表 3　第一个子表

b_1	a_1	a_2
b_1^1	1	1
b_1^2	1	0
b_1^3	0	0

表 4　第二个子表

b_2	a_3	a_4
b_2^1	1	0
b_2^2	1	1
b_2^3	0	1

表 5　第一个中间表

U	a_1	a_2	b_2	d
1	1	1	b_2^1	1
2	1	0	b_2^2	1
3	0	0	b_2^3	0
4	1	0	b_2^1	1

表 6　第二个中间表

U	b_1	a_3	a_4	d
1	b_1^1	1	0	1
2	b_1^2	1	1	1
3	b_1^3	0	1	0
4	b_1^2	1	0	1

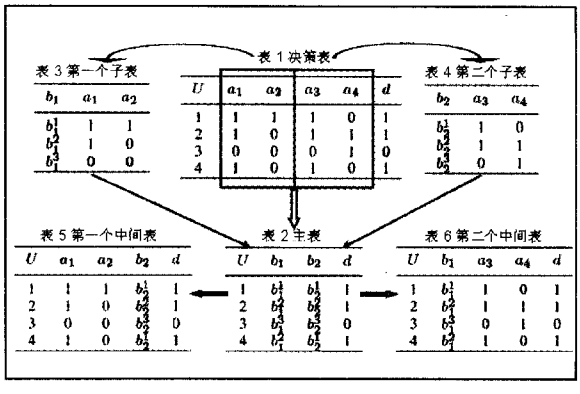

图 1　表之间的关系

3.2　基本性质

根据上面的定义，有如下性质：

给定决策表 $T=\langle U,C\cup D,V,f\rangle$，子表 $T^{B_i}=\langle U^{B_i},B_i\cup\{b_i\},V^{B_i},f^{B_i}\rangle$，$i=1,2,\cdots,m$，主表 $T^S=\langle U,S\cup D,V^S,f^S\rangle$，中间表 $T^{M_i}=\langle U,M_i\cup D,V^{M_i},f^{M_i}\rangle$，$i=1,2,\cdots,m$。

性质 1　主表 T^S 的正区域与决策表 T 的正区域是等价的，即 $POS_{(S)}(D)=POS_{(C)}(D)$。

证明：等价关系（不可分辨关系）的交集仍然是等价关系。S 是 C 的条件属性的组合。论域 U 关于等价关系 S 的划分等价于论域 U 关于等价关系 C 的划分，即 $U/IND(S)=U/IND(C)$，因此 $POS_{(S)}(D)=POS_{(C)}(D)$。

性质 1 表明主表与决策表正区域之间是等价关系。类似地，中间表与决策表正区域之间的等价关系见推论 1。

推论 1　中间表 T^{M_i} 的正区域与决策表 T 的正区域是等价的，即 $POS_{(M_i)}(D)=POS_{(C)}(D)$。

性质 2　主表 T^S 的连接属性 b_i 是不必要的，即 $POS_{(S-\{b_i\})}(D)=POS_{(S)}(D)$，当且仅当决策表 T 与连接属性 b_i 相对应的条件属性集 B_i 是不必要的，即 $POS_{(C-B_i)}(D)=POS_{(C)}(D)$。

证明（必要性）：设 b_i 在 T^S 中是不必要的，即 $POS_{(S-\{b_i\})}(D)=POS_{(S)}(D)$；由定义 7，$b_i$ 是 B_i 的一个组合，论域 U 关于等价关系 $S-\{b_i\}$ 的划分等价于论域 U 关于等价关系 $C-B_i$ 的划分，即 $U/IND(S-\{b_i\})=U/IND(C-B_i)$，因此 $POS_{(S-\{b_i\})}(D)=POS_{(C-B_i)}(D)$。由性质 1 可知，$POS_{(S)}(D)=POS_{(C)}(D)$，故 $POS_{(C-B_i)}(D)=POS_{(C)}(D)$。

（充分性）：设 B_i 在 T 中是不必要的，即 $POS_{(C-B_i)}(D)=POS_{(C)}(D)$；由定义 7，$B_i$ 的组合由 b_i 表示，论域 U 关于等价关系 $C-B_i$ 的划分等价于论域 U 关于等价关系 $S-\{b_i\}$ 的划分，即 $U/IND(C-B_i)=U/IND(S-\{b_i\})$，因此 $POS_{(C-B_i)}(D)=POS_{(S-\{b_i\})}(D)$。由性质 1 可知，$POS_{(C)}(D)=POS_{(S)}(D)$，故 $POS_{(S-\{b_i\})}(D)=POS_{(S)}(D)$。

性质 2 表明主表中的不必要连接属性与决策表中的不必要条件属性集之间的关系。类似地，推论 2 表明中间表中的不必要条件属性与决策表中不必要条件属性之间的关系；推论 3 和推论 4 分别是主表中的必要连接属性与决策表中必要条件属性子集之间的关系和中间表中必要条件属性与决策表中必要条件属性之间的关系。

推论 2　中间表 T^{M_i} 的条件属性 a 是不必要的，即 $\exists a\in B_i$，$POS_{(M_i-\{a\})}(D)=POS_{(M_i)}(D)$，当且仅当决策表 T 的条件属性 a 是不必要的，即 $POS_{(C-\{a\})}(D)=POS_{(C)}(D)$。

推论 3　若主表 T^S 的连接属性 b_i 是必要的，即 $POS_{(S-\{b_i\})}(D)\neq POS_{(S)}(D)$，那么在决策表 T 中一定有与连接属性 b_i 相对应的条件属性集 B_i 的子集 A 是必要的，即 $A\subseteq B_i$，$POS_{(C-A)}(D)\neq POS_{(C)}(D)$。

推论 4　若中间表 T^{M_i} 的条件属性 a 是必要的，即 $POS_{(M_i-\{a\})}(D)\neq POS_{(M_i)}(D)$，那么决策表 T 的条件属性 a 是必要的，即 $POS_{(C-\{a\})}(D)\neq POS_{(C)}(D)$。

上面的性质与推论就能够保证把决策表拆分成子表、把决策表转变成主表和中间表进行运算时结果的一致性和不变性。

3.3　基于拆分的特征选择算法

决策表的拆分原则是随机原则，即决策表中的条件属性集被随机地近似平均分配到各个子表当中。决策表中的决策属性不被拆分。拆分后的每个子表中都有一个关键字，决策

表中被拆分的条件属性集则用关键字代替，这样决策表变成了主表。如果只有一部分条件属性被关键字代替，决策表就变成了中间表。

设子表的个数是 k，首先把决策表拆分为一个主表和 k 个子表。决策表的条件属性集被平均分配到 k 个子表中。一个连接属性和一个条件属性子集构成一个子表。主表由决策属性集与 k 个连接属性组成。每个连接属性分别是各自子表的关键字。

如果主表中的连接属性是不必要的，则可以删除主表中的这个连接属性并且合并相同的对象，继续判断下一个连接属性。否则，将一个子表与主表合并成中间表。如果中间表的条件属性是不必要的，可以删除中间表的条件属性并且合并相同的对象。否则，继续下一次循环。最后得到一个约简。

依据上述分析，基于拆分的属性约简算法描述如下。

算法 2 DAR 算法

输入：决策表 $T=\langle U,C\cup D,V,f\rangle$；子表个数为 k。

输出：属性约简 Red。

(1)将决策表 T 拆分成一个主表 $T^S=\langle U,S\cup D,V^S,f^S\rangle$ 和 k 个子表 $T^{B_i}=\langle U^{B_i},B_i\cup\{b_i\},V^{B_i},f^{B_i}\rangle$，$i=1,2,\cdots,k$；

令 $Red=C$；

(2)While 每个连接属性 $b_i\in S$ do

(3)Begin

(4) If $POS_{(S-\{b_i\})}(D)=POS_{(S)}(D)$，then

(5) Begin

(6) b_i 是不必要的属性并且在 T^S 中删除该连接属性，即 $S=S-\{b_i\}$；

(7) $Red=Red-B_i$；

(8) 合并相同的对象；

(9) End

(10) Else

(11) Begin

(12) 将一个子表 T^{B_i} 与主表 T^S 合并成中间表 T^{M_i}；

(13) While 每个属性 $a\in B_i$ do

(14) Begin

(15) If $POS_{(M_i-\{a\})}(D)=POS_{(M_i)}(D)$，then

(16) Begin

(17) a 是不必要的属性并且在 T^{M_i} 中删除该属性，即 $B_i=B_i-\{a\}$；

(18) $Red=Red-\{a\}$；

(19) 合并相同的对象；

(20) End

(21) End

(22) End

(23)End

(24)输出 Red。

实际上，决策表的条件属性集被分成几个部分。算法处理每个部分而不是每个条件属性。每部分被一个连接属性所代替。换句话说，决策表中的 $|C|$ 个条件属性被压缩成主表中的 k 个连接属性。如果连接属性是不必要的，则与连接属性相对应的条件属性子集是不必要的，这个条件属性子集能够被一次性全部删掉，在条件属性子集中的每个属性不必再逐一进行判断。即使连接属性是必要的，也需要把主表与子表转化为中间表，中间表的比例也被压缩很多。对于 $|C|$ 个条件属性而言，中间表的属性的数量是 $(\frac{|C|}{k})+k$，这个数量已经非常小了。因此算法节省了很多计算时间。在最好情况下，只需要判断主表中的 k 个连接属性就可以得到一个属性约简。DAR 的最小时间复杂度是 $O(|N|^2*(k+|D|)^2)(k\ll|C|)$。在最差情况下，要判断中间表的 $(|C|/k)+k$ 个属性来获得一个属性约简。DAR 的最大时间复杂度是 $O(|N|^2*((|C|/k+k)+|D|)^2)$，GAR 的平均时间复杂度是 $O(|N|^2*(|C|+|D|)^2)$。DAR 的最大时间复杂度远远低于 GAR 的平均时间复杂度。因此，DAR 的性能要优于 GAR。

4 实验结果与分析

基因表达数据集可以归为决策表，其中每一个记录对应一个样本。条件属性集包含每一个基因，决策属性对应癌症的子类。选用 4 个公开的基因表达数据集：Leukemia 数据集(http://www.broad.mit.edu/MPR)包含有 7129 个基因、72 个样本，其中 47 个是 ALL 子类，25 个是 AML 子类；Colon-cancer 数据集(http://microarray.princeton.edu/ oncology)包含有 2000 个基因、62 个样本，其中 40 个 Colon-cancer 子类，22 个 Normal 子类；Lymphoma 数据集(http://llmpp.nih.gov/lymphoma)包含了 4026 个基因、96 个样本，其中 54 个 Other type 子类，42 个 B-cell lymphoma 子类；Liver cancer 数据集(http://genome-www.stanford.edu/hcc/)包含有 1648 个基因、156 个样本，其中 82 个 HCCs 子类，74 个 nontumor livers 子类。实验数据集见表 7。实验运行在 Intel Dual E2140 1.60GHz (处理器)，1G (内存)，Windows XP (操作系统)的个人计算机上。实验程序语言是 VC++ 6.0 及 SQL server 数据库。

表 7 实验所用数据集

数据集	属性数	决策数	样本数
Leukemia	7129	ALL	47
		AML	25
Colon-cancer	2000	Colon-cancer	40
		Normal	22
Lymphoma	4026	Other type	54
		B-cell	42
Liver cancer	1648	HCCs	82
		nontumor livers	74

粗糙集方法只能处理离散的数据，在约简之前先要对数据集进行离散化。实验中使用了文献[16]中的简单离散化算法对基因表达数据进行离散化，对离散化后的基因表达数据集应用上述约简算法进行特征选择，找出对疾病分类有较强区分能力的基因特征子集。根据本文提出的基于拆分的特征选择算法，假设子表的个数为 6。实验结果见表 8。前 3 列是数据集的基本描述，第 4，5 列是约简后的属性数。后两列表示两个算法的运行时间(单位 s)，实验结果是 10 次实验的平均值。

表 8 运行时间实验结果

数据集	属性数	样本数	GAR 约简后属性数	DAR 约简后属性数	GAR 运行时间	DAR 运行时间
Leukemia	7129	72	5	5	35s	3s
Colon-cancer	2000	62	7	6	12s	1s
Lymphoma	4026	96	7	7	48s	5s
Liver cancer	1648	156	6	5	133s	8s

正如表 8 所列，两种算法挑选出的特征子集均包含不到 10 个属性。在 Colon-cancer 和 Liver cancer 数据集上，DAR 算法约简后的属性数比 GAR 约简后的属性数要少。从这个对比结果可以看出，从 GAR 算法得到的约简要包含比 DAR

算法更多的冗余属性。在找出相对最优约简方面，DAR 要优于 GAR。同时，从表 8 的结果发现，DAR 算法的运行时间明显比 GAR 的运行时间要少。DAR 的计算效率远远高于 GAR 的计算效率。

为验证选择出的基因子集对疾病分类的有效性，将两种算法约简对应的规则集分别用作各自分类器，来预测未知样本的类别。10-折交叉检验用于检验分类器的有效性。实验结果见表 9。

表 9　预测正确率实验结果

数据集	GAR 预测正确率	DAR 预测正确率
Leukemia	94.9%	94.9%
Colon-cancer	93.3%	94.3%
Lymphoma	94.5%	95.8%
Liver cancer	92.2%	94.1%

从表 9 可以看出，两种算法均达到了高于 90 %的预测准确率，并且 DAR 算法的预测率要高于 GAR 算法。

为了与现有基因表达数据的选择算法进行有效比较，选用 SVM[17]算法、NSGA[19]算法和 Cho[20]算法对 4 个基因表达数据集进行基因选择。表 10 和表 11 分别表示用 DAR 算法、SVM 算法、NSGA 算法和 Cho 算法选择的基因数和运行时间。

表 10　选择基因个数实验结果

数据集	DAR 选择基因数	SVM 选择基因数	NSGA 选择基因数	Cho 选择基因数
Leukemia	5	10	8	20
Colon-cancer	6	15	12	32
Lymphoma	7	16	12	35
Liver cancer	5	12	9	24

表 11　运行时间实验结果

数据集	DAR 运行时间	SVM 运行时间	NSGA 运行时间	Cho 运行时间
Leukemia	3s	201s	57s	122s
Colon-cancer	1s	123s	25s	58s
Lymphoma	5s	315s	74s	152s
Liver cancer	8s	526s	125s	289s

如表 10 中所列，DAR 算法选择的基因数明显少于其他 3 种现有的基因选择算法。从这个结果可以看出其他 3 种基因选择算法选择出的基因可能包含较多的冗余基因。DAR 算法能够找出重要度较高的基因，并且选择的基因数量较少。

从表 11 的实验结果中发现，DAR 算法的运行时间比 SVM 算法、NSGA 算法和 Cho 算法的运行时间要少。DAR 算法的时间效率明显比其他 3 种基因选择算法的时间效率要高。

表 12　预测正确率实验结果

数据集	分类器	DAR	SVM	NSGA	Cho
Leukemia	KNN	96.3%	95.6%	93.3%	96.2%
	Naïve Bayes	95.5%	95.8%	91.4%	96.1%
	C5.0	97.8%	96.4%	94.9%	97.2%
Colon-cancer	KNN	95.4%	94.2%	92.5%	95.9%
	Naïve Bayes	94.3%	93.7%	90.2%	95.1%
	C5.0	96.4%	96.1%	93.4%	96.4%
Lymphoma	KNN	97.1%	96.2%	94.7%	98.0%
	Naïve Bayes	96.6%	95.7%	92.6%	97.3%
	C5.0	98.3%	97.6%	95.2%	97.8%
Liver cancer	KNN	93.3%	91.5%	88.6%	95.1%
	Naïve Bayes	91.1%	91.2%	85.3%	92.4%
	C5.0	95.6%	94.8%	93.3%	94.6%

为了检验基因选择算法对未知样本的预测精度，选择 KNN、Naïve Bayes 和 C5.0 分别作为分类器，对未知样本的类别进行预测。同样选用 10-折交叉检验用于分类器的有效性的检验。实验结果见表 12。

从表 12 的结果能够看出，选择 KNN 和 Naïve Bayes 作分类器时，对 DAR 算法选择的基因进行类别预测时，预测正确率低于 Cho 算法，大部分高于 SVM 算法和 NSGA 算法。选择 C5.0 作分类器时，DAR 算法选择的基因进行类别预测时，预测正确率是最高的。并且本文的 DAR 算法的预测正确率高于或部分高于文献[17-20]中的结果。

上述的实验结果说明，本文提出的算法不仅能用很短的时间选择出高区分能力的特征子集，而且能保证较高的分类精度。

本文提出的基于拆分的特征选择算法需给定子表的个数，子表的个数将会影响运算性能。4 个数据集再次被用到测试上。分别选用 2,4,6,8,10,12,14,16 个子表对本文算法进行重复实验。实验结果见图 2。

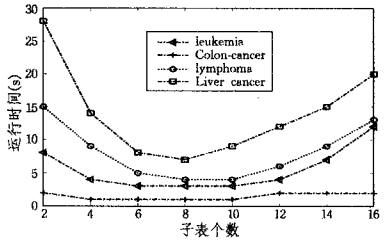

图 2　DAR 算法对不同个数的子表运算的性能对比

从图 2 中能够看出，子表的个数在 4 与 12 之间时，执行时间较少。即使不能选择最适合的子表个数，本文算法的执行时间也是较少的(见表 8，表 11)。

结束语　粗糙集理论提供了寻找特征子集的有效数学工具，但是现有的粗糙集约简算法的低效性限制了粗糙集理论的广泛应用，特别是在基因表达数据上的应用。针对粗糙集一个经典的属性约简算法，本文引入拆分的思想，即把一个复杂的问题拆分成小的、简单的、更容易处理的子问题。这些子问题可以用现有的算法进行处理，然后把它们的结果连接到一起解决初始问题。给出了一种基于拆分的基因选择算法，它可有效地降低计算属性约简的时间复杂度。在 4 个公开的基因表达数据集上的实验结果表明，本文提出的算法不仅有较高的计算性能，而且取得了较高的预测未知样本的正确率。

参 考 文 献

[1] Pawlak Z. Rough sets[J]. International Journal of Computer and Information Sciences, 1982, 11(5): 341-356

[2] 吴伟志，张文修，徐宗本. 粗糙模糊集的构造与公理化方法[J]. 计算机学报，2004，27(2)：197-203

[3] 徐章艳，刘作鹏，杨炳儒，等. 一个复杂度为 max(O(|C||U|)，O(|C|²|U/C|))的快速属性约简算法[J]. 计算机学报，2006，29(3)：391-399

[4] 史忠植，常亮. 基于动态描述逻辑的语义 Web 服务推理[J]. 计算机学报，2008，31(9)：1599-1611

[5] 王国胤，姚一豫，于洪. 粗糙集理论与应用研究综述[J]. 计算机学报，2009(07)：1229-1247

[6] 王国胤. Rough 集理论与知识获取[M]. 西安:西安交通大学出版社,2001

[7] 苗夺谦,周杰,张楠,等. 基于代数方程组的属性约简研究[J]. 电子学报,2010,38(5):1021-1028

[8] Maria S,Maite L. Rough set based approaches to feature selection for case-based reasoning classifiers[J]. Pattern Recognition Letters,2011,32(2):280-292

[9] Rokach L. Decomposition methodology for classification tasks:a meta decomposer framework[J]. Pattern Analysis and Applications,2006,9:257-271

[10] Fischer B. Decomposition of time series—comparing different methods in theory and practice[R]. Eurostat Working Paper,1995

[11] Wang M,liu B,Tang J H,et al. Metric learning with feature decomposition for image categorization [J]. Neurocomputing,2010,73(10-12):1562-1569

[12] Kusiak A. Decomposition in data mining:an industrial case study [J]. IEEE Trans Electron Packag Manuf,2000,23(4):345-353

[13] Bazan J G,Latkowski R,Szczuka M. Missing template decomposition method and its implementation in rough set exploration system[C]// Proceedings of the Fifth International Conference on Rough Sets and Current Trends in Computing. Kobe,LNAI,2006:254-263

[14] Zhang Q Z. An Approach to Rough Set Decomposition of Incomplete Information Systems[C]// IEEE Conference on Industrial Electronics and Applications. Harbin,IEEE,2007:2455-2460

[15] Nguyen H S, Nguyen S H, Skowron A. Decomposition of task specification problems[C]// Proceedings of the 11th International Symposium on Foundations of Intelligent Systems. Warsaw, LNAI,1999:310-318

[16] Fayyad U M,Irani K B. Multi-Interval Discretization of Continuous-valued Attributes for Classification Learning[C] // Proceedings of the 13th International Joint Conference of Artificial Intelligence. France,1993:1022-1027

[17] Wang Y,Tetko I V,HallMark A,et al. Gene selection from microarray data for cancer classification-a machine learning approach[J]. Computation Biology and Chemistry,2005,29(1):37-46

[18] Au A,Chan K C C,Wong A K C,et al. Attribute clustering for grouping, selection, and classification of gene expression data [J]. IEEE/ACM Transactions on Computational Biology and Bioinformatics,2005,2(2):83-101

[19] Deb K,Reddy A R. Reliable Classification of Two Class Cancer Data Using Evolutionary Algorithms[J]. BioSystems,2003,72(1/2):111-129

[20] Cho S B,Ryu J. Classification gene expression data of cancer using classifier ensemble with mutually exclusive features[C]// Proceedings of the IEEE,Special Issue on Bioinformatics Part-I: Advances and Challenges. New York: IEEE Press,2002:1744-1753

(上接第 222 页)

通信高效可靠,以及如何选择更优的解,以达到最佳态势,并未做深入探讨。近年来,分布式约束满足问题和分布式约束最优化问题(DCOP)[11]越来越受到关注,理论渐趋成熟,多飞行器即时路径规划问题也开始利用其理论框架。然而,针对具体的算法优化和最优解的选择还有待研究,这是未来的研究重点。

参 考 文 献

[1] Brosh E, Shavitt Y. Approximation and Heuristic Algorithms for Minimum Delay Application-layer Multicast Trees [C]// IEEE INFOCOM. 2004

[2] Hart P E,Nilsson N J,Raphel B. A Formal Basis for the Heuristic Determination of Minimum Cost Paths[J]. IEEE Transations on Systems Science and Cybernetics,1968,4(2):100-107

[3] Bellman R. Dynamic Programming[M]. Princeton,New Jersey: Princeton University Press,1957

[4] Bertsekas D P. Dynamic Programming and Optimal Control [M]. volumes 1 and 2,Belmont,Massachusetts:Athena Scientific,1995

[5] Wu Z,Yu Z Y. Dynamic Programming Principle for One Kind of Stochastic Recursive Optimal Control Problem and Hamliton-Jacobi-Bellman Equation[J]. Siam Journal on Control and Optimization,2008,47 (5):2616

[6] Hernández A G,Fallah-Seghrouchni A E,Soldano H. Distributed Learning in Intentional BDI Multi-agent Systems[C]// Proceedings of the Fifth Mexican International Conference in Computer Science (ENC'04). 2004

[7] Asenstorfer J,Cox T,Wilksch D. Tactical Data Link Systems and the Australian Defence Force (ADF)-Technology Developments and Interoperability Issues[M]. Commonwealth of Australia,February 2004

[8] Petcu A. A Class of Algorithms for Distributed Constraint Optimization[M]. ISBN 978-1-58603-989-9. 2009

[9] Chan Y M,Cruz J B J R. Decentralized stochastic adaptive nash games[J]. Optimal Control Applications&Methods, 1983, 4: 163-178

[10] Wang X,Lu C,Koutsoukos X. Enhancing the robustness of distributed real-time middleware via end-to-end utilization control [C]// IEEE RTSS. 2005

[11] Yokoo M, Durfee E H, Ishida T, et al. The Distributed Constraint Satisfaction Problem:Formalization and Algorithms[J]. IEEE Transctions on Knowledge and Data Engineering,1998,10 (5)

[12] Lee D,Arana I,Ahriz H,et al. Multi-Hyb:A Hybrid Algorithm for Solving DisCSPs with Complex Local Problems[C]// IEEE/WIC/ACM International Joint Conference on Web Intelligence and Intelligent Agent Technology Workshops. 2009

[13] U. S. Department of Transportation Federal Aviation Administration. Pilot's Handbook of Aeronautical Knowledge[M]. ISBN-13:978-1560277507. 2009