# 不完备信息系统下的属性约简算法

何 伟[1] 刘春亚[1] 赵 军[2] 李 华[1]

(重庆大学计算机学院 重庆400030)[1] (重庆邮电学院计算机系 重庆400065)[2]

**摘 要** 传统的粗糙集模型是处理完全信息系统的有力工具,但对于不完全信息系统却显得无能为力。因此对不完备信息系统的研究也是粗集理论研究领域之一。本文在M. Kryszkiewcz提出的一个容差关系的基础上提出改进,使之更加具有灵活性。然后以该改进模型为基础,运用属性重要性理论,给出属性约简算法,并通过一个实例来验证。最后将该算法和经典算法进行了比较。

**关键词** 粗集,容差关系,属性重要性,属性约简

## An Algorism of Attributes Reduction in Incomplete Information System

HE Wei[1] LIU Chun-Ya[1] ZHAO Jun[2] LI Hua[1]

(College of Computer Science, Chongqing University, Chongqing 400030)[1]

(Department of Computer Science, Chongqing University of Posts and Teleommunications, Chongqing 400065)[2]

**Abstract** Traditional rough set model is a powerful tool for processing complete information systems. But it cannot handle incomplete information systems, so the research of incomplete information is one of the research fields in rough set. The paper improves the tolerance relation model proposed by M. Kryszkiewcz to make it more flexible. Then significance of attributes theory is applied to propose a algorism of attributes reduction based on the improved model. Then an example is given to verify the algorism. At last, we give a comparison between improved algorism and traditional algorisms.

**Keywords** Rough set, Tolerance relation, Significance of attributes, Attributes reduction

## 1 引言

粗集理论是波兰数学家Pawlak于1982年提出的研究不确定和不精确知识的数学工具[1]。由于它不依赖于人们的先验知识和外界的影响,而是反映数据本身所隐藏的知识,因而逐渐被应用到KDD中。目前,粗集理论得到了广泛的研究与应用,取得了大量的成果,提出了许多数学模型[2,3]。但是在现实世界中,有些信息由于遗漏或者根本无法得到而变得不完全,对于这样的信息系统,传统的粗集模型就存在局限性。因为传统的粗集模型是基于等价类的不确定关系,而在不完全信息系统中,这种等价关系不复存在,因而无法判断两个对象在某个属性集合上是否可分辨。因此需要对传统的粗集模型进行扩展,目前已经有人提出了一些扩展模型,如容差关系模型、基于量化容差关系的模型等[2,6]。

本文重点放在对M. Kryszkiewcz提出的容差关系加以改进,并结合属性重要性度量,提出基于不完备信息系统的属性约简算法。为不完全信息系统的数据处理提供一些解决问题的思路以供探讨。

## 2 基于不完全信息的粗糙集模型的改进

### 2.1 容差关系及其不足

给定信息表S=(U,AT,V,f),其中AT=C∪D。C是条件属性集合,D是决策属性集合。U是对象集合,假设存在对象x∈U,x在条件属性上的取值有遗漏或缺失,但在决策属性上没有遗漏或缺失。f是映射。对于不完全信息,M. Kryszkiewcz提出了一个容差关系[2],该容差关系定义为:

**定义1**[2] 给定信息表S=(U,C∪D,V,f),对于有遗漏的属性子集B⊆C,记遗漏属性值为*,

$$T=\{(x,y)\mid x\in U\wedge y\in U\wedge (c_i\in B\Rightarrow (c_i(x)=c_i(y)\vee c_i(x)=*\vee c_i(y)=*))\}$$

但是容差关系存在一些不足:

1 容差关系在对论域中的对象进行分类时并不令人满意,但有些凭直觉就可以分类的在容差关系中还不能分类。

2. 根据容差关系的定义,在关系表中两个对象在某些属性上相似只要它们在这些属性上要么相同,要么有一个属性为遗漏值,这样定义并不一定符合客观实际。

3. 缺乏灵活性。当用容差关系得到的结果不令人满意时,没有办法进行调节。

### 2.2 改进容差关系

事实上,如果根据量化容差关系可知,两个对象在某个属性上是否相同应该取决于该属性的可取值的数目[2]。假设两个对象为x. y∈U,属性$c_i\in C$,$|Vc_i|$是属性$c_i$的可取值的个数。假设x,y在$c_i$上等概率地取$Vc_i$的每个值,那么对象在$c_i$上取每一个值的可能性为$1/|Vc_i|$,由此定义两个对象在属性$c_i$上相同的概率。

**定义2** 给定信息表S=(U,C∪D,V,f),x∈U,y∈U,$c_i\in C$,$Vc_i$是$c_i$的取值范围,对象x,y在属性$c_i$上相同的概率$Rc_i(x,y)$为:

$$Rc_i(x,y)=\begin{cases}1/|Vc_i| & c_i(x)\text{与}c_i(y)\text{不同时为}*\\ 1/|Vc_i|^2 & c_i(x)\text{与}c_i(y)\text{同时为}*\\ 1 & c_i(x)\text{与}c_i(y)\text{都不为}*\text{且}c_i(x)=c_i(y)\\ 0 & c_i(x)\text{与}c_i(y)\text{都不为}*\text{且}c_i(x)\neq c_i(y)\end{cases}\quad(1)$$

何 伟 硕士生,研究方向:粗集理论,现代远程教育.刘春亚 硕士生,研究方向:粗集理论,数据挖掘.李 华 副教授,研究方向:现代远程教育.

在此基础上还可以定义两个对象相同的概率 γ，它等于这两个对象在给定属性集合中的每一个属性上相同的概率之积，即：

$$\gamma=\prod_{i=1}^{n}R_{ri}(x,y) \tag{2}$$

为了增加灵活性，可以定义阈值 β。当两个对象相同的可能性 γ≥β 时，两个对象在给定的属性空间中相同，否则就不相同。基于上述观点，信息表的容差关系可修改为：

$$T'=\{(x,y)\mid x\in U\wedge y\in U\wedge\gamma\geqslant\beta\} \tag{3}$$

其中 γ 是两个对象的相同的可能性测度，β 是设定的阈值。于是可以定义相容类和相对正域：

**定义3** 在上述关系中，给定属性集合 P 下 x 的相容类定义为：

$$S_p(x)=\{y\mid y\in U,(x,y)\in T'\} \tag{4}$$

相对正域定义为：

$$POS_PQ=\bigcup_{Y\in U/IND(Q)}\{S_p(x)\mid S_p(x)\in Y\} \tag{5}$$

下面举例说明，该实例来自于文[5]，其中 C=(a,b,c,d) 是条件属性集合；D={d}是决策属性集合。

表1

| Car | Price(a) | Mileage(b) | Size(c) | Max-Speed (d) | D |
|---|---|---|---|---|---|
| 1 | high | Low | full | low | good |
| 2 | low | * | full | low | good |
| 3 | * | * | compact | low | poor |
| 4 | high | * | full | high | good |
| 5 | low | High | full | high | excellent |
| 6 | low | High | full | * | good |

在这个信息表中，有6个对象，6个属性。其中 a,b,c,d 为条件属性，a,b,d 的取值为{low,high}，c 的取值为{full,compact}，D 为决策属性，取值为{good,poor,excellent}并且没有不完全信息。考察对象在条件属性 C={a,b,c,d}上的相同关系。给定 β=0.4

考察对象1和对象2在属性 b 上的取值。b(1)=low，b(2)=*，由于 b 只取两个值，那么 b(1)=b(2)的可能性为0.5，a(1)=high，a(2)=low，c(1)=c(2)=full。由(2)式得，γ=0，对象1和对象2在属性 C 上不相同。

考察对象2和6，对象2在属性 b 上有遗漏值，对象6在属性 d 上有遗漏值，那么 b(2)=b(4)的可能性为 Rb(x,y)=0.5，d(2)=d(4)的可能性 Rd(x,y)也为0.5，对象2和对象4相同的概率为0.25<0.4，认为二者不相同。如果 β 取0.2，那么就可以认为它们相同。

## 3 基于属性重要性算法

### 3.1 属性重要性

目前属性约简广泛采用的算法是利用属性重要性来求得属性约简[3,5]。它是通过计算每一个属性对信息表产生的信息量的大小或者计算属性的重要性程度来寻找对信息表影响最大的属性，去掉对信息表产生影响较小或不产生影响的属性，从而求得属性约简。它可以不直接求核值，使得算法更为通用，因而得到了广泛运用。本文利用属性重要性来求决策表的属性约简。这里先给出属性重要性的定义。

**定义4**[7] 给定不完全信息表 S={U,C∪D,V,f}，决策属性 D 和条件属性 C 的依赖度定义为：

$$\gamma(C,D)=card(POS_CD)/card(U) \tag{6}$$

根据依赖度的定义，任意属性 a∈C−R 的重要度定义为：

$$SIG(a,R,D)=\gamma(RY\{a\},D)-\gamma(R,D) \tag{7}$$

### 3.2 基于改进容差关系的分类和求正域算法

在给出属性约简算法之前，先给出基于上述改进关系 T′ 的分类和求正域的算法。这两个算法在后面的约简算法中都要用到，与经典模型下基于等价关系的算法不同，改进容差关系下只有分类而没有划分。因此传统分类算法不再适用，因此需要重新给出算法。

**算法1**

```
输入：决策表 S=(U,C∪D,V,f)，属性集合 A⊆C，域值 β
输出：A 在 U 上的分类
1. T_A(u_1)={u_1}; T_A(u_2)={u_2},…,T_A(u_n)={u_n}
   i=1,j=0,β=1
2. for i=1to |U|−1 do {
        for j=i+1 to|U| do{
            for each a_i∈A do {
              计算 R a_i(u_i,u_j);
              β=β* R a_i(u_i,u_j);}
            if (β≥γ)then {
            T_A (u_i)= T_A(u_i)∪{ u_j};
              T_A(u_j)= T_A(u_j)∪{ u_i};}
        }
        }
3. 输出 T_A(u_1); T_A(u_2),…,T_A(u_n)
```

算法性能分析：算法第一步是初始化，大约需要 O(n)；第二步是分类，假定属性集合 A 中有 K 个属性，即 card(A)=K，则三重循环共需要 O(Kn²)，最后输出需要 O(n)，所以该算法一共需要 O(n)+O(Kn²)+O(n)=O(Kn²)。

**算法2** 求相对正域的算法

```
输入：U|IND(D)={D_1,D_2,…,D_m}
      T_A(u_1); T_A(u_2),…,T_A(u_n)
输出：A 的 D 正域 POS_AD。
  1. i=1,j=1,POS=0;C[k]=0(k=1,2,…,n);
  2. while(i(=m){
       while(j<=n){
          if((T_A(u_j)⊆D_i)and(C[j]≠0)){
             POS=POS∪T_A(u_j);
             C[j]=1;}
          j=j+1;}
     i=i+1;j=1;
     }
  3. 输出 POS_AD。
```

算法说明：这里只说明一下 C[j]的作用。C[j]是一个标志，它标志 $u_j$ 是否已经被划分到某个决策类当中。若 $u_j$ 已经被划分到某个决策类中，那么其余的决策类就不必再判断 $u_j$ 是否是它们的子集，这样可以减少运算时间，提高算法效率。

算法性能分析：从算法的描述来看，它的时间复杂度为 O(mn)。

### 3.3 基于属性重要性的约简算法

算法首先考虑没有遗漏值或遗漏值较少的属性，然后再考虑遗漏值较多的属性，通过计算属性的相对重要性，选取相对重要性较大的属性或加入到约简中，然后判断当前所选属性的重要性是否为零，如果是，则找到一个约简，否则继续寻下一个重要性较大的属性，直到两者相等为止。在给出算法之前，先引入两个辅助元素：

$M(c_i)$：统计每个属性中有多少个对象在该属性上有遗漏值。

MCi：是存放具有相同 $M(c_i)$ 的属性集合。MCi={c|c∈C∧M(c)=i}

下面给出利用属性重要性的约简算法。

**算法3**

```
输入：不完全信息表 S={U,C∪D,V,f}
输出：约简属性 RED
1. RED=0; M(Ci)=0,(i=1,2,….);MCj=0,(i=1,2……);
2. 对每一个 c_i∈C 记录每个 c_i 的缺失值数量；
3. 把具有相同 M(c_i)值的 c_i 加入到集合 MCj(j=0,1,2……)中；
4. k=0；
```

5. 对每一个 $c \in MC_k$
   计算 $SIG(\{c\},RED,D)=\gamma(RED\cup\{c\},D)-\gamma(RED,D)$;
6. $MC_k=0$? 如果是,转(9)
7. 选取 SIG 最大的 $c_m$,$MC_k=MC_k\backslash\{c_m\}$;
   $RED=RED\cup\{c_m\}$
8 如果 $SIG(c_m,RED,D)=0$,找到一个属性约简,转(10);否则转(6)
9. $k=\max(j)$? 如果是,转(10),
   否则,$k=k+1$,转(5)
10. 输出 RED,算法终止。

算法性能分析:算法第一步的时间耗费是 $O(|C|)$,算法第二步最坏时间复杂度是 $O(|C|n)$,算法第三步时间复杂度是 $O(|C|)$,算法第四步最坏时间复杂度为 $O(|C|^2)$,整个算法时间复杂度为 $O(|C|^2+|C|n)$。

## 4 实例

以表1为例来验证算法。选取域值 $\beta=1/3$。由表1可知,$M(C)=0$,$M(a)=M(d)=1$,$M(b)=3$,则首先选取 c,但 c 不满足条件(*),就继续选取 a 或者 d,由于 $SIG(a,\{c\},D)=1/3$,$SIG(d,\{c\},D)=0$,因此优先选取 a,$\{a,c\}$还是不满足条件(*),于是选取 d,条件满足,算法中止。于是得到一个约简 $\{a,c,d\}$。b 是可省略的。

## 5 与传统算法的比较

### 5.1 算法性能的比较

属性约简算法由于涉及的数据量比较大,算法的时间开销和空间开销也比较大。因此算法的时间复杂度应尽量控制在多项式范围内,而且次数越低越好。在经典算法中,许多算法一般控制在 $O(n^2)$或者 $O(n^3)$这个数量级上。本论文提出的算法大约是 $O(|C|^2+|C|n)$,依然在多项式范围内,是可以接受的。

### 5.2 适用范围比较

传统算法都是基于经典模型,以等价关系为基础的。它们在处理信息系统的时候有一定的局限性,基于改进模型的算法则给予了较大的灵活性。它实际上是经典模型的扩展,当 $\beta=1$ 时,就变成传统的等价关系;$\beta=0$ 时就变成 M. Kryszkiewcz 提出的容差关系。它借鉴了量化容差关系的思想,使算法既可以处理完备信息系统,也可以处理不完备信息系统,只需要改变 β 值即可,而不用更换算法,因此具有较大的灵活性。

## 参考文献


1 曾黄麟. 粗集理论极其应用——关于数据推理的新方法. 重庆:重庆大学出版社,1988
2 王国胤. Rough 集理论与知识获取. 西安:西安交通大学出版社,2001
3 张文修,等. 粗糙集理论与方法. 北京:科学出版社,2001
4 Liang Jiye, Xu Zongben. Uncertainty Measures of Roughness of Knowledge and Rough Sets in Incomplete Information Systems. In:IEEE Proc. of the 3rd World Congress on Intelligent and Automation,2000. 526～2629
5 赵卫东,吴明赞. 不完全信息下的粗集拓展. 计算机科学,2002(9. 专刊)
6 Pawlak Z. Rough set theory and its applications to data analysis [J]. Cybernetics and System,1998,29(7):661～688
7 石峰,娄臻亮,张永清. 一种改进的粗糙集属性约简启发式算法. 上海交通大学学报,2002,26(4):478～481


(上接第110页)

$k=\sum_{i=1}^{n}a_ik_i$,其中 $k_1$为线性核函数 $k_1(\chi_1,\chi_2)=\chi_1^T,\chi_2$,$k_2$为多项式核函数 $k_2(\chi_1,\chi_2)=(1+\chi_1^T\chi_2)^d$,$k_3$为高斯核函数 $k_3(\chi_1,\chi_2)=\exp(-0.5(\chi_1-\chi_2)^T(\chi_1-\chi_2)/\sigma)$,$a_i$ 为各个核函数的权重。这时的 SVM 的参数有六个,分别为:$a_1,a_2,a_3,d,\sigma,C$。这时的计算结果如下:

表2 线性组合核的 SVM 参数选择结果

| | 代数 | 时间 | 结果 | 错分数 |
|---|---|---|---|---|
| 遗传方法 | 116～132代 | 351～423秒 | $a_1=0.18,a_2=0.23,$ $a_3=0.59,\sigma=0.5,C=1$ | 0 |
| 免疫方法 | 14～23代 | 37～49秒 | $a_1=0.19,a_2=0.22,$ $a_3=0.59,\sigma=0.5,C=1$ | 0 |

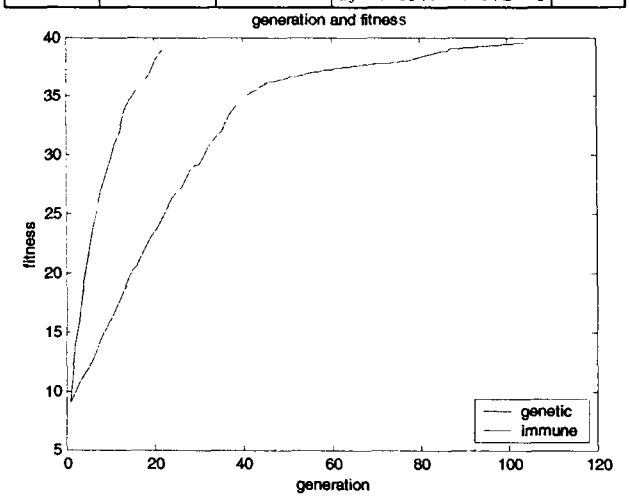


图3 遗传和免疫方法适应度的对比

从这个结果可以看出,对于复杂问题,免疫方法比遗传方法有较大的优势。

**结论** 从实验结果可以看出,免疫算法在遗传操作的过程中能够自动地找出局部的最优解作为疫苗注射给子代的个体,使得搜索空间在算法的中后部分大大减小,问题也就能够以较快的速度收敛,但又保持了种群的多样性,不会使结果落入局部最优解,较好地解决了复杂核函数的多参数选择的问题。

## 参考文献


1 边肇祺,张学工. 模式识别. 清华大学出版社
2 Vapnik V N. The Nature of Statistical Learning Theory. New York: Springer- Verlag
3 王磊. 免疫进化计算及应用:[西安电子科技大学博士论文]
4 Genton M G. Classes of Kernels for Machine Learning: A Statistics Perspective. Department of Statistics North Carolina State University
5 Christopher J.Burges C. A Tutorial on Support Vector Machines for Pattern Recognition. Microsoft Research
6 Chapelle O.Vapnik V. Choosing Multiple Parameters for Support Vector Machines. AT&T Research Labs
7 Liang Xue-feng, Liu Fang. Choosing Multiple Parameters for SVM Based on Genetic Algorithm. In: Intl. Conf. on Signal Processing Proc.
8 刘芳,梁雪峰. 一种基于线性组合核的 SVM 算法. 计算机科学
9 程云鹏,张凯院. 矩阵论. 西北工业大学出版社