# 模糊决策粗糙集代价敏感属性约简研究

刘 偲 秦亮曦
(广西大学计算机与电子信息学院 南宁 530004)

**摘 要** 针对决策中普遍存在的代价问题,在模糊理论和决策粗糙集的基础上,对其代价敏感属性约简方法进行了研究。在模糊决策粗糙集属性约简中引入了包含误分类代价和测试代价的总代价。因此约简的目标不再只是考虑正域的大小,而是寻找使得总代价最小的最优属性子集。提出了一种模糊决策粗糙集代价敏感属性约简(COSAR)算法,该算法采用启发式方法搜索最优属性子集。给出了算法的步骤,并将该算法与已有的模糊粗决策粗糙集属性快速约简(QuickReduct)算法进行了性能对比。实验结果表明,COSAR 算法比 QuickReduct 算法具有更强的属性约简能力、更低的分类总代价、更短的运行时间,且随着测试样本的增加,分类总代价差值也越来越大。

**关键词** 模糊决策粗糙集,代价敏感,属性约简
**中图法分类号** TP181 **文献标识码** A

## Study on Cost Sensitive Attribute Reduction for Fuzzy Decision Theoretic Rough Sets

LIU Cai QIN Liang-xi
(School of Computer, Electronics and Information, Guangxi University, Nanning 530004, China)

**Abstract** Aiming at the cost problem that generally exists in desicion-making, on the basis of fuzzy theory and decision theoretic rough sets, we studied the method of cost sensitive attribute reduction. We introduced the total cost including misclassification cost and test cost into attribute reduction for fuzzy decision theoretic rough sets (FDTRS). Thus the target of reduction is not only to considere the size of positive region, but also to find the optimal subset of attributes with the minimum total cost. We proposed a cost sensitive attribute reduction (named as COSAR) algorithm for FDTRS. The algorithm uses a heuristic method to search the optimal subset. We provided the procedure of the algorithm and compared the performance of the algorithm with the existing FDTRS attribute reduction algorithm, called QuickReduct. The experimental results show that COSAR algorithm has stronger attribute reduction capability, lower total classification cost, shorter running time than QuickReduct algorithm, and with the increasing of test samples, the difference of total classification cost between two methods is growing larger.

**Keywords** Fuzzy decision theoretic rough sets(FDTRS), Cost sensitive, Attribute reduction

## 1 引言

Z. Pawlak 在 1982 年提出了经典粗糙集模型[1],它是一种处理模糊性和不确定性的数学工具[2]。经典粗糙集由于建立在严格代数包含关系的基础上,因此容错性较差。Yao 将经典粗糙集的严格代数包含关系改为概率包含关系,使得决策类的负域不恒为空,并将贝叶斯风险理论引入概率粗糙集赋予 3 个决策域新的语义,重新定义了一个新的模型即决策粗糙集模型[3]。

决策粗糙集模型属于概率粗糙集模型,其条件概率是基于不可分辨的等价关系定义的,只能处理离散型的数据。然而在实际应用中,绝大部分数据都是连续型的数据,用基于不可分辨的等价关系粗糙集来处理时,都需要先将连续型的数据离散化。数据的离散化不可避免地会造成部分信息的丢失[4]。所以将模糊集与粗糙集结合起来可以直接处理连续型的数据。L. A. Zadeh 于 1965 年提出了模糊集合的概念[5],随后 Dubois 和 Prade 于 1990 年提出模糊粗糙集理论[6,7](下称为 Dubois 模型)。

属性约简是粗糙集研究的核心问题,将冗余的属性去掉之后可以发现数据的本质信息。Yao 在经典粗糙集的基础上提出了一种决策粗糙集的正域约简方法[8]。R. Jensen 在 2009 年提出了基于 Dubois 模型的属性约简方法。近两年郭敏等人提出了基于模糊化的决策粗糙集正域约简方法[4],王莉等人将模糊粗糙集与决策粗糙集相结合提出一种模糊决策粗糙集模型及其正域约简方法[9]。

上述各模型都是以精度为目标的分类方法,属于精度敏感分类,而在实际分类问题中不仅需要关注分类精度,误分类代价也非常重要[10,16]。一般而言,样本的分类结果与测试属性集紧密相关。在一定范围内,随着测试属性的增加,样本的分类精度越高,错误分类结果越少,误分类代价越小[10]。因此,包含更多属性的测试属性集通常具有较小的误分类代价。但在实际问题中,获取样本的属性值本身具有一定的代价,即

本文受国家自然科学基金(61363027),广西自然科学基金(2013GXNSFAA253003,2015GXNSFAA139292)资助。
刘 偲(1991-),男,硕士生,主要研究方向为决策粗糙集、数据挖掘、机器学习,E-mail:liucaitc@163.com;秦亮曦(1963-),男,博士,教授,主要研究方向为数据挖掘、决策粗糙集等。

测试代价[11,16]。测试属性集越多，虽然分类精度越高即误分类代价越低，但是同时测试代价也越高。所以，需要将误分类代价和测试代价的总代价降到最低，才能适应实际问题。然而现在研究的模糊决策粗糙集都没有考虑代价敏感[15]，李华雄等人曾在 2013 年将代价敏感分类成功引入了决策粗糙集，但是由于决策粗糙集本身的局限性，其并不能处理连续型数据且会造成部分信息的丢失。所以，本文将代价敏感分类引入模糊决策粗糙集，通过找到误分类代价与测试代价总和最小的属性约简，使其在实际应用中可以直接处理连续型的数据，并能在更短的时间内找到一个属性约简，其误分类代价与测试代价总和远远小于已有模糊决策粗糙集属性约简方法。

## 2 相关理论介绍

为便于理解，本节先回顾模糊集理论及模糊决策粗糙集模型、模糊粗糙集属性约简的相关知识。

### 2.1 模糊集理论

模糊集[5]是用隶属度函数来描述模糊概念的一种形式，隶属度函数是表示一个对象 $x$ 隶属于一个集合 $A$ 的程度的函数，模糊集可以用如下定义描述。

**定义 1**[4] 设 $F$ 是集合 $X$ 到 $[0,1]$ 上的一个隶属度函数：

$F: X \to [0,1], x \to F(x)$

则称 $F$ 是 $X$ 上的一个模糊集，$F(x)$ 是模糊集 $F$ 的隶属度函数，或称 $F(x)$ 是 $X$ 对模糊集 $F$ 的隶属度。

常用隶属度函数有三角形、梯形、钟形、高斯类型和多项式类型[4]。$F(x)$ 的值越接近 1，表示对象 $x$ 隶属于模糊集 $F$ 的程度越高；$F(x)$ 的值越接近 0，表示对象 $x$ 隶属于模糊集 $F$ 的程度越低。

Dubois 和 Prade 最早将模糊集和粗糙集结合，提出了模糊粗糙集，该模型的模糊下、上近似集定义如下。

**定义 2**[7]

$$\mu_{\underline{P}X}(F_i) = \inf_x \max\{1-\mu_{F_i}(x), \mu_X(x)\}, \forall i$$

$$\mu_{\overline{P}X}(F_i) = \sup_x \min\{\mu_{F_i}(x), \mu_X(x)\}, \forall i$$

其中，$F_i \subseteq U/P$，$U$ 为论域，$P$ 为属性集，$F_i$ 是由 $P$ 得到的模糊集，$X$ 是需要被近似描述的概念，$\mu_{F_i}(x)$ 是 $x$ 对 $F_i$ 的隶属度，$\mu_X(x)$ 是 $x$ 对 $X$ 的隶属度。

模糊等价类是模糊决策粗糙集的核心，与经典的粗糙集等价类不同，它可以通过一个模糊等价关系 $R$ 来确定两个对象关于 $R$ 相似。

**定义 3** 设 $U$ 是一个非空集合，称 $U$ 上的模糊二元关系 $R$ 是相似关系当且仅当 $R$ 满足

(1) **自反性**：$\mu_R(x,x)=1, x \in U$；

(2) **对称性**：$\mu_R(x,y)=\mu_R(x,y), x,y \in U$；

(3) **传递性**：$M_R(x,z) \geqslant \mu_R(x,y) \wedge \mu_R(y,z), x,y,z \in U$。

对于单个属性 $a$，$U/IND(a)$ 可以看成该属性的模糊等价类集合。如果把属性 $a$ 模糊化成 $N_a$ 和 $Z_a$ 两个模糊集，则 $U/IND(\{a\})=\{N_a, Z_a\}$。假设 $R=\{a\}$，则 $U/R=\bigotimes\{a \in R: U/IND(\{a\})\}$，其中 $A \otimes B=\{X \otimes Y: \forall X \in A, \forall Y \in B, X \cap Y=\varnothing\}$。如果 $R$ 为多个条件属性的集合，那么它所对应的模糊等价关系就是它所包含的单个属性的模糊等价关系的一个笛卡尔积[12,13,16]。

例如，设 $R=\{a,b\}$，$U/IND(\{a\})=\{N_a, Z_a\}$，$U/IND(\{b\})=\{N_b, Z_b\}$，则 $U/IND(R)=\{N_a \cap N_b, N_a \cap Z_b, Z_a \cap N_b, Z_a \cap Z_b\}$。

如果 $(x,y) \in IND(R)$，则称 $(x,y)$ 是模糊 $R$ 等价关系，将该等价关系记作 $[x]_R$。

### 2.2 模糊决策粗糙集模型

设 $\Omega=\{w_1, w_2, w_3, \cdots, w_s\}$ 表示 $s$ 个状态的集合；$A=\{a_1, a_2, a_3, \cdots, a_m\}$ 表示 $m$ 个可能的决策；$x$ 为论域中的对象；$\widetilde{x}$ 表示 $x$ 的某种描述（比如 $x$ 关于某个属性集的模糊等价类可以看作是 $x$ 的一种描述）；$F(x)$ 是 $x$ 对 $\widetilde{x}$ 的隶属度；$P(w_j|x)$ 表示在 $\widetilde{x}$ 描述下的对象 $x$ 具有状态 $w_j$ 的条件概率；$\lambda(a_i|w_i)$ 表示在状态 $w_i$ 的情况下作出决策 $a_i$ 的风险代价，$\lambda$ 经常由经验得出，也可以根据实际数据得到。对于具有 $\widetilde{x}$ 描述的对象 $x$ 来说，假设采取 $a_i$ 决策可能带来的决策风险期望为：

$$R(a_i|\widetilde{x}) = \sum_{j=1} \lambda(a_i|w_j) P(w_j|\widetilde{x})$$

这里为了方便描述，只考虑只有 2 种状态的状态集合 $\Omega=\{X, \sim X\}$，状态 $X$ 和 $\sim X$ 为互补的 2 种状态。给定决策集 $A=\{a_P, a_N, a_B\}$，其中 $a_P, a_N, a_B$ 分别表示决策为正域 $POS(X)$、决策为负域 $NEG(X)$ 和决策为边界域 $BND(X)$ 3 种决策。$\lambda_{PP}, \lambda_{BP}, \lambda_{NP}$ 分别表示当 $x$ 属于概念 $X$ 时，作出 $a_P, a_N, a_B$ 3 种决策所对应的代价函数值。由此可计算出 3 种决策的期望风险为：

$$R(a_P|[x]_R) = \lambda_{PP} P(X|[x]_R) + \lambda_{PN} P(\sim X|[x]_R)$$

$$R(a_N|[x]_R) = \lambda_{NP} P(X|[x]_R) + \lambda_{NN} P(\sim X|[x]_R)$$

$$R(a_B|[x]_R) = \lambda_{BP} P(X|[x]_R) + \lambda_{BN} P(\sim X|[x]_R)$$

根据贝叶斯最小风险决策原则，可以得到如下形式的决策规则：

IF $R(a_P|[x]_R) \leqslant R(a_N|[x]_R)$ and $R(a_P|[x]_R) \leqslant R(a_B|[x]_R)$, decide $POS(X)$

IF $R(a_N|[x]_R) \leqslant R(a_P|[x]_R)$ and $R(a_N|[x]_R) \leqslant R(a_B|[x]_R)$, decide $NEG(X)$

IF $R(a_B|[x]_R) \leqslant R(a_P|[x]_R)$ and $R(a_B|[x]_R) \leqslant R(a_N|[x]_R)$, decide $BND(X)$

对于决策代价函数值的大小，显然有如下关系：

$\lambda_{PP} \leqslant \lambda_{BP} < \lambda_{NP}$，$\lambda_{NN} \leqslant \lambda_{BN} < \lambda_{PN}$

另外，由于状态集合由互补的 $X$ 和 $\sim X$ 组成，可得：$P(X|[x]_R)=1-P(\sim X|[x]_R)$，则决策规则可以化为：

IF $P(X|[x]_R) \geqslant \gamma$ and $P(X|[x]_R) \geqslant \alpha$, decide $POS(X)$

IF $P(X|[x]_R) \leqslant \beta$ and $P(X|[x]_R) \leqslant \gamma$, decide $POS(X)$

IF $P(X|[x]_R) \geqslant \beta$ and $P(X|[x]_R) \leqslant \alpha$, decide $POS(X)$

令：

$$\alpha = \frac{(\lambda_{PN}-\lambda_{BN})}{(\lambda_{PN}-\lambda_{BN})+(\lambda_{BP}-\lambda_{PP})}$$

$$\beta = \frac{(\lambda_{BN}-\lambda_{NN})}{(\lambda_{BN}-\lambda_{NN})+(\lambda_{NP}-\lambda_{BP})}$$

$$\gamma = \frac{(\lambda_{PN}-\lambda_{NN})}{(\lambda_{NP}-\lambda_{PP})+(\lambda_{PN}-\lambda_{NN})}$$

易知当下式成立时：

$$(\lambda_{PN}-\lambda_{BN})(\lambda_{NP}-\lambda_{BP}) > (\lambda_{BP}-\lambda_{PP})(\lambda_{BP}-\lambda_{NN})$$

可以得到 $\alpha > \beta$；进一步推导，仅仅用 $\alpha, \beta$ 得到化简之后的决策规则：

IF $P(X|[x]_R) \geqslant \alpha$, decide $POS(X)$

IF $P(X|[x]_R)\leqslant\beta$, decide $NEG(X)$

IF $\beta<P(X|[x]_R)<\alpha$, decide $BND(X)$

其中，$P(X|[x]_R)$可以用 Dubois 模型中的下近似集概念求得[12,16]，可将 $P(X|[x]_R)$定义为：

$$P(X|[x]_R)=\underset{F\in U/R}{\mathrm{SUP}}\,\mu_{\underline{PX}}(F_i)$$

且 $\mu_{F_1\cap\cdots\cap F_n}(x)=\min(F_1(x),F_2(x),\cdots,F_n(x))$。

### 2.3 模糊决策粗糙集正域约简

目前已知求模糊决策粗糙集的属性约简都是指快速约简算法即 The Fuzzy-rough QuickReduct Algorithm[12,16]，下简称为 QuickReduct 算法。

**算法 1** QuickReduct Algorithm

```
1. R←{}, γ'best←0, γ'prev←0
2. Do
3. T←R
4. γ'prev←γ'best
5. ∀ x∈(C−R)
6.    IF γ'R∪{x}(D)>γT'(D)
7.       T←R∪{x}
8.       γ'best←γT'(D)
9. R←T
10. until γ'best=γ'prev
11. return R
```

## 3 模糊决策粗糙集代价敏感属性约简

### 3.1 代价敏感

将数据集表示为如下四元组形式的决策信息表：

$S=\{U,C\cup D,f,V\}$

这里模糊化选取三角型函数，选取的三角隶属度函数可以定义为：

$$\mu_1(x)=\begin{cases}0, & 0\leqslant x\leqslant x_1\\ a_1x+b_1, & x_1<x\leqslant x_2\\ c_1x+d_1, & x_2<x\leqslant x_3\\ 0, & x_3<x\leqslant x_4\end{cases}$$

$$\mu_2(x)=\begin{cases}0, & 0\leqslant x\leqslant y_1\\ a_2x+b_2, & y_1<x\leqslant y_2\\ c_2x+d_2, & y_2<x\leqslant y_3\\ 0, & y_3<x\leqslant y_4\end{cases}$$

考虑论域 $U$ 中的每个样本 $x\in U$ 和指定的条件属性子集 $B\subseteq C$，根据模糊决策粗糙集的决策规则能够得到最优分类结果和相应的误分类代价，由此可以确定一个误分类代价映射函数：$Miscost(x,B)$。

该函数的函数值由决策结果确定，即有如下形式：

1)当决策为 $POS(X)$时：

$Miscost(x,B)=\lambda_{PN}P(\sim X|[x]_R)$

2)当决策为 $BND(X)$时：

$Miscost(x,B)=\lambda_{BP}P(X|[x]_R)+\lambda_{BN}P(\sim X|[x]_R)$

3)当决策为 $NEG(X)$时：

$Miscost(x,B)=\lambda_{NP}P(X|[x]_R)$

在考虑样本 $x$ 在属性子集 $B$ 上的测试代价时，假设各个样本在同一属性上的测试代价是相等的，所以 $x$ 在属性子集 $B$ 上的测试代价为属性子集 $B$ 中所有属性测试代价的和，测试代价设为一个非负实数。因此可以得到一个计算测试代价函数[9]：

$$Testcost(x,B)=\sum_{i=1}^{|B|}F(c_i)$$

其中，$F(c_i)$为 $B$ 中单个属性的测试代价。

那么，样本 $x$ 在条件属性子集 $B$ 上的分类代价为误差代价 $Miscost(x,B)$与测试代价 $Testcost(x,B)$之和记为 $Sumcost(x,B)$，即：

$Sumcost(x,B)=Miscost(x,B)+Testcost(x,B)$

### 3.2 代价敏感属性约简

现有的模糊决策粗糙集属性约简中，不考虑误差代价和测试代价，只考虑正域的变化。而代价敏感属性约简的目标是寻找一个最优约简属性子集 $B\subseteq C$，使得所有待分类样本 $x\in U$ 在最优约简属性子集 $B$ 上具有最小的分类总代价 $SC=\sum Sumcost(x,B)$。要使分类总代价 $SC$ 最小，即需要每一个待分类的样本分类代价 $Sumcost(x,B)$最小。这里使用启发式算法搜索最优属性子集[10]。

**算法 2** Cost sensitive Attribute reduction Algorithm for Fuzzy Decision Theoretic Rough Sets（以下简称 COSAR 算法）

输入：一个决策表 S={U,C∪D,f,V}，训练样本 x，误差代价矩阵，测试代价集

输出：局部最优属性约简集合 B*

Step1 计算初始属性集：

$$B=\arg\min_{c_i\in C}\mathrm{Sumcost}(x,\{c_i\})$$

令待选属性集 Btest=C−B。

Step2 计算当前分类总代价 SC：

SC=Miscost(x,B)+Testcost(x,B)

Step3 Do

Step4 对每个属性 $b_i\in$Btest，计算分类总代价：

SC′=Miscost(x,B∪{$b_i$})+Testcost(x,B∪{$b_i$})

Step5 对每个属性 $b_i\in$Btest 计算属性重要度：

Sig(x,B,$b_i$)=SC−SC′

Step6 选出重要度最高的属性 $b_i^*$：

$$b_i^*=\arg\max_{b_i\in B}\mathrm{Sig}(x,B,b_i)$$

Step7 如果 Sig(x,B,$b_i^*$)≤0，则转至 Step10。

Step8 更新 B=B∪{$b_i^*$}；更新 Btest=Btest−{$b_i^*$}；更新 SC=SC′。

Step9 until Btest=∅；

Step10 输出约简属性集 B*=B 及分类总代价 SC。

## 4 实验结果及分析

本节主要验证模糊决策粗糙集代价敏感属性约简算法的有效性，并将其与已有的模糊决策粗糙集属性约简算法进行比较。实验的机器为 Intel(R) Xeon(R) 的 3.50GHz CPU，8GB 内存，64 位 WINDOWS8 操作系统，算法在 Matlab 平台上实现。实验数据取自 UCI 数据库中的 4 个数据集 Heart，Iono，WDBC(Breast Cancer Wisconsin (Diagnostic))和 WPBC (Breast Cancer Wisconsin (Prognostic))。数据集中有少量缺失的数据，使用该属性在其他所有对象中的最频值来补齐该缺失的属性值，WDBC 和 WPBC 有数据 ID 列，将 ID 列删除，然后使用 WEKA 标准化、归一化。预处理之后的实验数据基本信息如表 1 所列。

表 1　实验数据基本信息

| 名称 | 类别数 | 属性个数 | 处理后属性数 | 样本数 |
|---|---|---|---|---|
| Heart | 2 | 13 | 13 | 270 |
| Iono | 2 | 34 | 34 | 351 |
| WDBC | 2 | 31 | 30 | 569 |
| WPBC | 2 | 34 | 33 | 198 |

在实验中，假定误分类代价满足 $\lambda_{PP} \leqslant \lambda_{BP} \leqslant \lambda_{NP}$，$\lambda_{NN} \leqslant \lambda_{BN} \leqslant \lambda_{PN}$，$\lambda_{NN} = \lambda_{PP} = 0$[14]。因为 UCI 数据集上没有给出误差代价矩阵，为使实验具有可重复性，指定 4 个数据集的误差代价矩阵，如表 2 所列[9]。

表 2　误分类代价矩阵

| 名称 | POS | BND | NEG |
|---|---|---|---|
| X | 0 | 8 | 20 |
| ~X | 15 | 7 | 0 |

数据集也没有给出各属性的测试代价，因为实验需要，本文假定各测试代价服从正态分布 $N(\mu,\sigma)$，并设数据集的属性测试代价服从 $N(0.02,0.01)$，使用正态随机函数随机生成属性的测试代价。为使实验具有可重复性，指定 4 组数据的测试代价矩阵，如表 3－表 6 所列。

表 3　Heart 各属性测试代价

| 测试代价 | 0.0175 | 0.0231 | 0.0231 | 0.0113 | 0.0196 |
|---|---|---|---|---|---|
| 0.0183 | 0.0262 | 0.0309 | 0.0310 | 0.0113 | 0.0207 |
| 0.0078 | 0.0088 | | | | |

表 4　Iono 各属性测试代价

| 测试代价 | 0.0093 | 0.0435 | 0.0138 | 0.0274 | 0.0180 |
|---|---|---|---|---|---|
| 0.0288 | 0.0123 | 0.0059 | 0.0057 | 0.0248 | 0.0182 |
| 0.0180 | 0.0341 | 0.0229 | 0.0219 | 0.0358 | 0.0119 |
| 0.0269 | 0.0283 | 0.0175 | 0.0221 | 0.0083 | 0.0085 |
| 0.0210 | 0.0272 | 0.0458 | 0.0133 | 0.0218 | 0.0191 |
| 0.0006 | 0.0156 | 0.0020 | 0.0284 | 0.0111 | |

表 5　WDBC 各属性测试代价

| 测试代价 | 0.0074 | 0.0364 | 0.0152 | 0.0249 | 0.0138 |
|---|---|---|---|---|---|
| 0.0259 | 0.0267 | 0.0200 | 0.0371 | 0.0329 | 0.0215 |
| 0.0222 | 0.0307 | 0.0317 | 0.0112 | 0.0348 | 0.0197 |
| 0.0214 | 0.0279 | 0.0268 | 0.0169 | 0.0183 | 0.0137 |
| 0.0290 | 0.0289 | 0.0322 | 0.0306 | 0.0129 | 0.0317 |
| 0.0102 | | | | | |

表 6　WPBC 各属性测试代价

| 测试代价 | 0.0083 | 0.0124 | 0.0256 | 0.0238 | 0.0310 |
|---|---|---|---|---|---|
| 0.0210 | 0.0263 | 0.0257 | 0.0095 | 0.0256 | 0.0386 |
| 0.0233 | 0.0328 | 0.0142 | 0.0077 | 0.0080 | 0.0351 |
| 0.0174 | 0.0138 | 0.0222 | 0.0355 | 0.0169 | 0.0204 |
| 0.0234 | 0.0188 | 0.0179 | 0.0138 | 0.0353 | 0.0172 |
| 0.0124 | 0.0241 | 0.0099 | 0.0093 | | |

实验先随机选取 10 组不同的 50 个数据样本作为训练样本，分别采用 QuickReduct 算法和 COSAR 算法计算局部最优属性集 $B^*$ 以及总代价 $SC$，然后比较 10 组结果的平均值。最后通过 WEKA 中的 NaiveBayes 算法计算测试样本的分类精度并取平均值。实验中需要使用一组三角隶属度函数模糊化，在实际应用中模糊化选取的隶属度函数由专家根据经验给出，不同的隶属度函数在同一个数据集上会有不同的约简结果，分类精度也不尽相同，这从后面的实验结果中可以看出。为了让实验具有可重复性，下面给出本文实验使用的一组三角隶属度函数：

$$\mu_1(x)=\begin{cases}0, & 0\leqslant x\leqslant 0.18\\ 50x-9, & 0.18<x\leqslant 0.2\\ -2.5x+1.5, & 0.2<x\leqslant 0.6\\ 0, & 0.6<x\leqslant 1\end{cases}$$

$$\mu_2(x)=\begin{cases}0, & 0\leqslant x\leqslant 0.2\\ 2.5x-0.5, & 0.2<x\leqslant 0.6\\ -5x+4, & 0.6<x\leqslant 0.8\\ 0, & 0.8<x\leqslant 1\end{cases}$$

表 7 和图 1 展示了在 4 个数据集上 QuickReduct 算法和 COSAR 算法得到的局部最优属性约简集合平均个数的比较结果。

表 7　约简属性集合平均个数

| 数据集 | 处理后全属性集 | QuickReduct 算法 | COSAR 算法 | 约简率（%） |
|---|---|---|---|---|
| Heart | 13 | 2.2 | 1 | 54.55 |
| Iono | 34 | 2.5 | 1 | 60.00 |
| WDBC | 31 | 2.9 | 1.6 | 44.83 |
| WPBC | 33 | 3.8 | 1.2 | 68.42 |

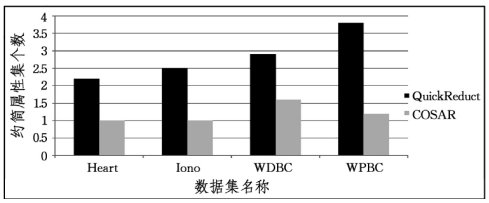


图 1　约简属性集平均个数

由表 7 和图 1 上可以清楚地看出，COSAR 算法在 4 组数据集上约简得到的最优属性集个数都比 QuickReduct 算法得到的精简了许多；在 WPBC 数据集上，COSAR 算法在 QuickReduct 算法的基础上属性个数更是约简了 68.42%，证明 COSAR 算法比 QuickReduct 算法的属性约简能力更强。从表 7 中也可以发现，代价敏感属性约简的大部分属性都是冗余的。

以下约简率皆为 COSAR 算法相对 QuickReduct 算法的约简率，定义为：

$$\text{约简率}=\frac{(\text{QuickReduct 算法}-\text{COSAR 算法})}{\text{QuickReduct 算法}}$$

表 8 和图 2 展示了在 4 个数据集上实验的平均总代价 SC。

表 8　平均总代价 $SC$

| 数据集 | QuickReduct 算法 | COSAR 算法 | 约简率（%） |
|---|---|---|---|
| Heart | 140.99 | 0.8857 | 99.37 |
| Iono | 137.7600 | 0.4777 | 99.65 |
| WDBC | 181.74 | 3.3425 | 98.16 |
| WPBC | 230.58 | 4.4088 | 98.09 |

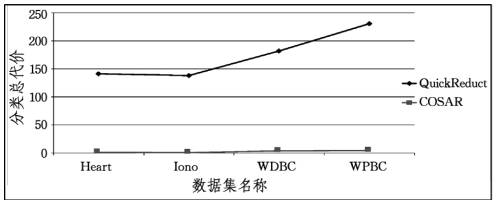


图 2　平均总代价 $SC$

从表 8 和图 2 上可以很明显看出，COSAR 算法能大大降低测试代价总和。在 4 个数据集上，COSAR 算法在

QuickReduct 算法的基础上总代价约简率都超过了 98%；特别是在 Iono 数据集上，COSAR 算法能在 QuickReduct 算法的基础上降低 99.65%的总代价。以上证明了 COSAR 算法约简得到的最优属性集总代价要远小于 QuickReduct 算法得到的最优属性集总代价，验证了 COSAR 算法的有效性。

表 9 和图 3 展示的是 4 个数据集在 10 组实验中所用的平均时间，这里记录的时间是使用的 Matlab 自带的记录运行时间方式统计的。

表 9　平均运行时间(s)

| 数据集 | QuickReduct 算法 | COSAR 算法 | 约简率(%) |
|---|---|---|---|
| Heart | 25.1503 | 5.1874 | 79.37 |
| Iono | 78.5488 | 13.7996 | 82.43 |
| WDBC | 120.24 | 25.2751 | 78.98 |
| WPBC | 375.75 | 19.0351 | 94.93 |

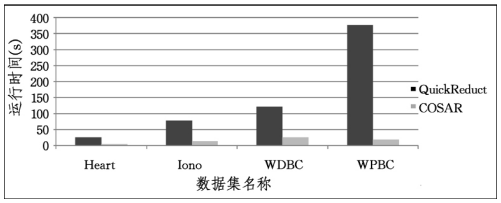


图 3　平均运行时间(s)

从图 3 中能看到，COSAR 算法比 QuickReduct 算法在运行时间要短得多，而且通过比较表 7 和表 9 能发现，QuickReduct 算法在约简属性集合平均个数增多时，平均运行时间也会增加。这是因为 QuickReduct 算法的运行时间会随着约简属性集合个数的增加呈指数级增长，但是 COSAR 算法并不存在这样的情况，它总是能在短时间内得到结果。

图 4—图 7 分别用图像的方式展示 QuickReduct 算法和 COSAR 算法在 4 个数据集上的分类总代价比较。从图 4—图 7 也能观察得到，与 QuickReduct 算法相比较，COSAR 算法有更低的分类总代价，且随着测试样本的增加，差值也越来越大。这也是代价敏感分类所追求的目标，即找到一个属性约简具有最小的分类总代价。

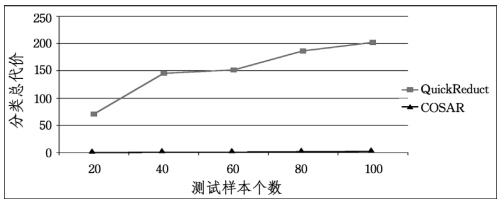


图 4　Heart 分类总代价 *SC* 比较

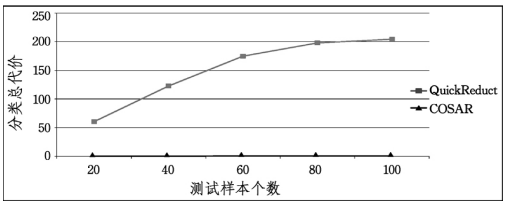


图 5　Iono 分类总代价 *SC* 比较

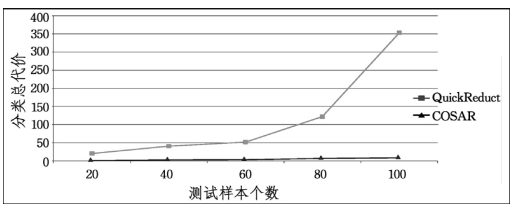


图 6　WDBC 分类总代价 *SC* 比较

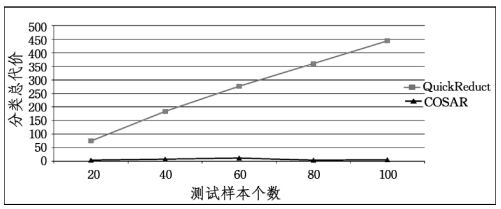


图 7　WPBC 分类总代价 *SC* 比较

在实际应用中，如果各属性的测试代价均较高，COSAR 算法可以帮助决策者平衡误分类代价和测试代价，在获得具有较小误分类代价的分类属性前提下节省大量花费在获取数据上的成本。

最后用表 10 和图 8 展示 4 组数据集上两种算法得到的最优约简在 WEKA 的 NaiveBayes 算法计算的平均分类精度。

表 10　平均分类精度(%)

| 数据集 | QuickReduct 算法 | COSAR 算法 |
|---|---|---|
| Heart | 69.77 | 39.20 |
| Iono | 72.06 | 82.00 |
| WDBC | 91.12 | 83.93 |
| WPBC | 63.89 | 61.99 |

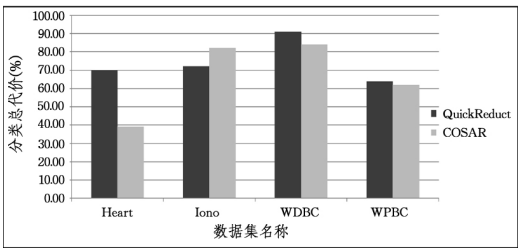


图 8　平均分类精度比较

观察表 10 和图 8 可知，COSAR 算法可能在个别数据集上比 QuickReduct 算法的分类精度要低很多，但是在大部分数据集上只是略低一点，而在某些数据集上 COSAR 算法的分类精度会更高。造成这种不稳定的原因取决于模糊化用到的隶属度函数的好坏，单个隶属度函数并不能适用于所有情况，应该根据实际问题选择恰当的隶属度函数，但是如何确定隶属度的适用性还有待研究，这也是我们下一步的工作。

结束语　本文对模糊决策粗糙集的代价敏感属性约简问题进行了研究，通过将误分类代价和测试代价引入模糊决策粗糙集，提出了 COSAR 算法，并通过实验验证了该算法的有效性。同时，将此方法与已有的 QuickReduct 算法做了对比实验，证明了 COSAR 算法比 QuickReduct 算法具有更强的属性约简能力，所需的误分类代价与测试代价之和远小于后者，所需的运行时间远少于后者，并且随着测试样本的增加，两种方法所需的分类总代价差值也越来越大。由于实验中模糊化所使用的隶属度函数并不能适用于所有数据集，因此在不同的数据集上的分类精度有高有低。下一步将研究分析使用不同的隶属度函数对数据进行模糊化的结果，希望能找到最优隶属度函数的形式，或者确定各种隶属度的适用情形。

由于在 UCI 上获取的实验数据中并不包含误分类代价和测试代价，因此本文讨论的方法只在理论上证明可行，并未用于真实环境，我们也将进一步研究如何将其用于实际应用。

## 参 考 文 献


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

[2] 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,2009,32(7):1229-1246

[3] Yao Y Y. Decision-theoretic rough set models[M]// Yao J, Lingras P, Wu W Z, et al. Rough Sets and Knowledge Technology. Lecture Notes in Computer Science 4481, Heidelberg: Springer, 2007:1-12

[4] 郭敏,贾修一,商琳. 基于模糊化的决策粗糙集属性约简和分类[J]. 模式识别与人工智能,2014(8):701-707

[5] Zadeh L A. Fuzzy sets [J]. Information & Control, 1965,8(65):338-353

[6] Dubois D, Prade H. Rough fuzzy sets and fuzzy sets[J]. International Journal of General Systems,1990,17(2):191-209

[7] Dubois D, Prade H. Putting Rough Sets and Fuzzy Sets Together [M]// Intelligent Decision Support. Springer Netherlands,1992, 11:203-232

[8] Yao Y, Zhao Y. Attribute reduction in decision-theoretic rough set models[J]. Information Sciences,2008,178(17):3356-3373

[9] 王莉,周献中,李华雄. 模糊决策粗糙集模型及其属性约简[J]. 上海交通大学学报,2013,47(7):1032-1035

[10] 李华雄,周献中,黄兵,等. 决策粗糙集与代价敏感分类[J]. 计算机科学与探索,2013(2):126-135

[11] Min Fan, He Hua-ping, Qian Yu-hua, et al. Test-cost-sensitive Attribute Reduction[J]. Information Sciences, 2011, 181(22): 4928-4942

[12] Jensen R, Shen Q. Fuzzy-rough attribute reduction with application to Web categorization[J]. Fuzzy Sets & Systems,2004,141(3):469-485

[13] 朱江华,李海波,潘丰. 基于遗传算法和模糊粗糙集的知识约简[J]. 计算机仿真,2007,24(1):86-89

[14] 李华雄,刘盾,周献中. 决策粗糙集模型研究综述[J]. 重庆邮电大学学报(自然科学版),2010,22(5):624-630

[15] 刘家彬,闵帆. 代价敏感粗糙集研究综述[J]. 漳州师范学院学报(自然科学版),2011,24(4):17-22

[16] Jensen R, Shen Q. New approaches to fuzzy-rough feature selection[J]. IEEE Transactions on Fuzzy Systems, 2009, 17(4): 824-838


(上接第 66 页)

(4)误差分析

常用的误差分析方法主要有过分依赖原始数据的绝对误差、体现误差比例的相对误差以及评价变化程度的均方误差等。本文采用更能反映预测的可信度的相对误差来分析结果。相对误差的公式为:

$$\theta=\frac{|V-V'|}{V}*100\%=\frac{\Delta V}{V}*100\% \tag{10}$$

其中,$V$ 表示实际输出的判定结果,$V'$为期望输出的结果。误差对比如图 9 所示。

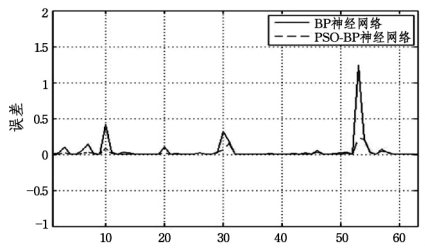


图 9 BP 与 PSO-BP 神经网络误差对比图

从图 9 可以看出 BP 神经网络模型误差波动较大,最大误差已经达到 1;PSO-BP 神经网络的相对误差幅度小而且稳定,在 0.2 左右波动。

结束语 本文采用 PSO-BP 神经网络算法将粒子群算法和 BP 神经网络算法结合在一起,发挥了粒子群算法良好的搜寻能力,解决了 BP 神经网络收敛速度慢、存在多个局部极值点的问题。通过 PSO 算法与 BP 神经网络的结合找到了合适的初始权值以及阈值,并通过自学习以及训练的过程得出较优的 PSO-BP 神经网络模型。对模型进行测试,结果表明本文方法误差较小,可得到较好的预期结果,对调养肠胃的饮食食谱起指导作用。

## 参 考 文 献


[1] 秦翔. 健康饮食生活方式影响下的烹饪家电设计研究[D]. 江南大学,2011

[2] 徐大明,周超,孙传恒,等. 基于粒子群优化 BP 神经网络的水产养殖水温及 pH 预测模型[J]. 渔业现代化,2016(1):24-29

[3] Xie G, Zhang J. Variable precision rough set for groupdecision-making:an application[J]. International Journal of Approximate Reasoning,2008,49(2):331-343

[4] 闫驰. 基于 PSO-BP 神经网络的无线传感器网络定位算法[J]. 电子科技,2016(4):56-58,62

[5] Wan Hong-bo, Zhao Xiao-qi, Tu Xu-yan, et al. Cooperative Velocity Updating Model based Particle Swarm Optimization [J]. Applied Intelligence,2014,3(40):322-342

[6] 张彩凤. PSO-BP 神经网络股价预测[J]. 经营管理者,2016(9):185

[7] 徐以山,曾碧,尹秀文,等. 基于改进粒子群算法的 BP 神经网络及其应用[J]. 计算机工程与应用,2009,35:233-235

[8] 徐兰,方志耕,刘思峰. 基于粒子群 BP 神经网络的质量预测模型[J]. 工业工程,2012(4):17-20,27

[9] 邓涛,黄希光. 基于 PSO—BP 算法的农业机械数据预测分析研究[J]. 中国农机化学报,2016(4):269-273,284

[10] 李希. 基于 K 均值聚类和 BP 神经网络的耐火材料损伤模式识别[D]. 武汉:武汉科技大学,2012

[11] 龙泉,刘永前,杨勇平. 基于粒子群优化 BP 神经网络的风电机组齿轮箱故障诊断方法[J]. 太阳能学报,2012(1):120-125

[12] 李松,刘力军,翟曼. 改进粒子群算法优化 BP 神经网络的短时交通流预测[J]. 系统工程理论与实践,2012(9):2045-2049

[13] 孙亚基于粒子群 BP 神经网络人脸识别算法[J]. 计算机仿真,2008(8):201-204