粗糙集与粒子群算法结合的属性离散化方法

沈中林,范敬德,樊 玮
SHEN Zhong-lin,FAN Jing-de,FAN Wei

中国民航大学 计算机学院,天津 300300
Software Technology Research Center,School of Computer,Civil Avitation University of China,Tianjin 300300,China

SHEN Zhong-lin,FAN Jing-de,FAN Wei.Method of discretization of continuous attributes based on rough set and Particle Swarm Optimization.Computer Engineering and Applications,2009,45(12):129-130.

Abstract: This paper presents an algorithm for discretization based on rough set and particle swarm optimization.It mainly inducts a modified PSO (MPSO) and improves the ability to break away from the local optimum.Simultaneously,the algorithm obtains a satisfactory discretization result.Authors do many tests for different data sets and the experimental results prove that the algorithm has better validity for discretization of continuous attributes.

Key words: Particle Swarm Optimization(PSO);rough set;discretization

摘 要:提出了一种结合粗糙集和粒子群的连续属性离散化算法,采用了 MPSO 算法的思想,提高了粒子群摆脱局部极值的能力,得到了较好的离散化效果。对不同的数据集进行了多次测试,结果表明该算法在对数据离散化时有较好的性能。

关键词:粒子群优化;粗糙集;离散化
DOI:10.3778/j.issn.1002-8331.2009.12.042 **文章编号:**1002-8331(2009)12-0129-02 **文献标识码:**A **中图分类号:**TP311

1 引言

粗糙集是一种处理不精确和不确定知识的有效工具。以不可分辨关系为核心的粗糙集理论[1]只能处理离散化的数据,而现实应用中数据往往是连续型的,因此连续属性的离散化是粗糙集的主要问题之一,离散化的效果影响了粗糙集的实用性。

连续属性离散化的方法有多种,但是每个离散化方法都应该使得连续属性被离散化后的分割区间尽可能少而且尽量减少整个数据的信息丢失。在粗糙集中进行属性的离散化时,决策表中决策属性对条件属性集的依赖度在离散化前后应该保持一致。针对每个连续属性不同的分割断点集,决策属性对条件属性集的依赖度是不同的。为了保持离散化前后决策表的相容性,需要寻找每个属性合适的分割断点。而基于鸟群捕食行为的粒子群优化算法恰恰适合于在解空间中寻找最优解。因此,研究了基于 PSO 的粗糙集连续属性离散化方法(PSORSAD)。

2 粗糙集理论及离散化问题描述

连续属性的离散化是指给定该连续属性某些断点,用这些断点来分割该属性的取值区间,处于同一个区间内的数值被认为是相同的。

设决策表 $S=(U,C\cup D,V,f)$,$b\in C$,$b\in(V_{b_{\min}},V_{b_{\max}})$, 给定 b 的一个分割断点集$\{b_0,b_1,b_2,\cdots,b_l\}$,那么 $b_0=V_{b_{\min}}$,$b_l=V_{b_{\max}}$,属性 b 被划分成如下几个区间:

$P_c=\{[b_0,b_1],[b_1,b_2],\cdots,[b_{l-1},b_l]\}=\{0,1,\cdots,l-1\}$

定理 设决策表 $S=(U,C\cup D,V,f)$,$x,y\in U$,若 x、y 的条件属性取值完全相同而它们的决策属性取值不同,那么 $POS_C(D)<1$。

证明 设 $U/C=\{C_1,C_2,\cdots,C_m\}$,$U/D=\{D_1,D_2,\cdots,D_n\}$,则必 $\exists i(i=1,\cdots,m)$满足 $x,y\in C_i$。因为 x、y 的决策属性值不同,所以,$\neg\ \exists j(j=1,\cdots,n)$使得 $C_i\subseteq D_j$。故有$|POS_C(D)|<|U|$,那么必有 $\gamma_C(D)=|POS_C(D)|/|U|<1$。证毕。

数据离散化前后决策表的相容性应该保持一致,由上述定理可知,$POS_C(D)=1$ 保证了离散化后的决策表是相容的。同时,尽可能使离散化后连续属性的区间数目较少。数据离散化的方法有多种,而寻找最优的离散化结果已被证明是 NP 完全问题[2]。本文拟结合 PSO 算法这一有效工具来进行连续属性的离散化。

3 粒子群优化算法

粒子群优化算法(PSO)是基于群体的演化算法,它源于模拟鸟群捕食行为来求解优化问题。所有的粒子均在解空间中进行搜索,每个粒子都有自己的位置和飞行速度,二者均用向量来表示。每个粒子的位置向量代表优化问题的一个解,将其代入目标函数即可得到一适应值。基本的 PSO 算法参见文献[3]。文献[4]中在粒子速度的更新公式中添加了一个惯性因子 ω,提出了 APSO(Adaptive Particle Swarm Optimization)算法,即随着迭代次数的增加,ω 线性减小。

基本的 PSO 算法和 APSO 算法的共同缺点是易于陷入局

基金项目:国家自然科学基金委员会与中国民用航空总局联合资助项目(No.60672173)。
作者简介:沈中林(1966-),男,副教授,主要研究方向为数据库与信息工程;范敬德(1984-),男,硕士生,主要研究方向为数据挖掘。
收稿日期:2008-03-03 **修回日期:**2008-06-02

部最优而无法跳出。

在上述两种算法中,每个粒子只向自身当前最优解和群体当前最优解学习,但是每个粒子都可能有好的维度被其他粒子作为学习的榜样。鉴于此,文献[5]提出了广泛学习粒子群优化算法(CLPSO)。CLPSO 本质上采用一种动态拓扑模型,其核心思想为:在每次的迭代过程中,每个粒子的飞行速度向量有 m 维向群体当前极值学习,在剩下的维数中随机选择几维向随机选择的其他粒子的当前个体极值学习,最后剩余的几维向该粒子自身的个体极值学习。该算法的优点,粒子的飞行速度向量的每一维都是相互独立学习的,因此具有很强的防止陷入局部最优的能力。

文献[6]在粒子飞行速度的公式中引入高斯分布的系数,并证明采用高斯分布系数能够提高算法的收敛性,但是它抛弃了惯性因子,使得算法易于陷入局部最小。文献[7]在 CLPSO 算法的基础上引入文献[6]中的思想,提出了 MPSO 算法。本文拟采用 MPSO 算法的思想对连续属性进行离散化。

4 PSORSAD 算法

PSORSAD 算法的基本思想:对于给定的一个决策表,给定连续属性的分割区间数,初始化每个连续属性的断点集,以决策属性对所有条件属性的依赖度作为 PSO 算法的适应度函数,然后改变每个连续属性分割区间的端点值并计算决策属性对所有条件属性的依赖度,如此反复执行下去直至算法满足终止条件。

PSORSAD 算法是以 CLPSO 算法为基础并引入 MPSO 算法的思想。PSORSAD 算法具体实现过程如下:

粒子群体的初始化:设 $S=(U,C\cup D,V,f)$ 为一决策表,$C=\{C_1,C_2,\cdots,C_d\}$,n 为每个连续属性的分割区间数。每个粒子 $\boldsymbol{X}=\{X_1,X_2,\cdots,X_d\}$ 是一个 d 维向量,其中 $\boldsymbol{X}$ 的每一维 $\boldsymbol{X}_i=\{l_0,l_1,\cdots,l_n\}$ 是 $n+1$ 维向量,它表示了属性 C_i 的断点集合,其中 $l_0=C_{i-\min}$,$l_n=C_{i-\max}$,$C_{i-\min}$ 和 $C_{i-\max}$ 分别为属性 C_i 的最小值和最大值。在初始化 X_i 时,随机选择 $C_{i-\min}$ 和 $C_{i-\max}$ 之间的 $n-1$ 个值并升序排序然后依次赋值给 $l_1,\cdots,l_{n-1}$。粒子 X 的每一维的飞行速度 $V_i\in[-(C_{i-\max}-C_{i-\min})/4,(C_{i-\max}-C_{i-\min})/4]$。

飞行速度和位置的变化公式:

算法初始设定每个粒子的飞行速度的 m 维向群体极值学习。每个粒子的飞行速度变化的伪代码如算法 1 所示。

变量 *attr*[]存放该粒子向群体极值学习的维。

算法 1

```
for i=1 to d   //i 代表了粒子的维数
  if(attr.contains(i))
      该维的飞行速度向群体极值的相应维度学习;
  else if(i==rand(1,d)) //rand(1,d)表示 1 和 d(包含)之间的一个随机整数,该维的飞行速度向随机选择的粒子的个体极值的相应维度学习;
  else
      该维的飞行速度向自身个体极值的相应维度学习;
end for
```

粒子的每一维 X_i 中的任一断点的取值范围为 $[C_{i-\min},C_{i-\max}]$,但是在算法执行过程中难免有断点超越边界的情况,针对越界情况,采取的策略是不让该粒子参与当前迭代次数中的决策属性对条件属性集依赖度的计算。

惯性因子的变化公式:

设定惯性因子 $\omega\in[\omega_{\min},\omega_{\max}]$,算法的最大迭代次数为 *iterationmax*。当前迭代过程中 $\omega=\omega-\frac{\omega_{\max}-\omega_{\min}}{iterationmax}*iteration$,其中 *iteration* 为当前的迭代次数。

粒子适应度函数的计算:

每次迭代过程中,每个粒子的个体极值存储了一组连续属性的断点集,用这些断点来离散化原始数据集,在离散化后的数据集的基础上计算决策属性对条件属性集的依赖度,把该依赖度作为粒子的适应度。

算法的终止条件:

当迭代次数达到最大迭代次数 *iterationmax* 或决策属性对条件属性集的依赖度达到 1 时,算法终止。

5 实验

为了验证算法的有效性,选择了 UCI 机器学习库中的 6 个数据集进行测试,另外还对作者所做项目中的一部分气象数据集(weather)进行了测试。用离散化后决策属性对条件属性集的依赖度作为衡量标准。表 1 是待测试数据集的特征描述。算法初始设定每个连续属性的分割区间数,表 2 显示了在分割区间数不同时对各数据集进行 10 次测试所得到的离散化后决策属性对条件属性集依赖度的平均值,表 3 是对离散化后的数据集进行分类正确率的测试,采用 CPAR 算法[8],每个数据集采用 80%作为训练集,20%作为测试集。实验参数设置如下:粒子群体大小为 30,最大迭代次数为 500,惯性因子初始值为 0.7,它随着迭代次数的增加线性减小至 0.2。

表 1 数据集的特征描述

DataSet	*D*	*C*	*N*	*Continuous*
austra	690	2	15	6
glass	214	6	10	9
heart	270	2	14	13
iris	150	3	5	4
vehicle	846	4	19	18
wine	178	3	14	13
weather	756	7	160	81

注:*D* 是记录数,*C* 是类别数,*N* 是属性个数,*Continuous* 是连续属性的个数。

表 2 各数据集的离散化结果

区间数	austra	glass	heart	iris	vehicle	wine	weather
2	0.960 58	0.356 54	0.999 26	0.667 33	0.433 81	1.000 00	0.985 18
3	0.980 43	0.613 55	1.000 00	0.957 33	0.852 60	1.000 00	1.000 00
4	0.986 81	0.796 26	1.000 00	0.994 00	0.970 68	1.000 00	1.000 00
5	0.992 31	0.861 68	1.000 00	0.998 66	0.999 03	1.000 00	1.000 00
6	0.995 07	0.908 88	1.000 00	1.000 00	1.000 00	1.000 00	1.000 00

表 3 离散后的分类准确率

区间数	austra	glass	heart	iris	vehicle	wine	weather
2	0.843	0.493	0.776	0.893	0.554	0.897	0.458
3	0.849	0.578	0.770	0.876	0.534	0.908	0.502
4	0.851	0.579	0.813	0.970	0.548	0.866	0.468
5	0.866	0.633	0.778	0.950	0.638	0.883	0.472
6	0.864	0.540	0.802	0.970	0.599	0.894	0.466

由表 2 可知,随着每个属性分割区间数目的增加,离散化

(下转 159 页)