ISSN 1004-9037,CODEN SCYCE4
Journal of Data Acquisition and Processing Vol. 37,No. 3,May 2022,pp. 668－679
DOI:10. 16337/j. 1004-9037. 2022. 03. 016

http://sjcj. nuaa. edu. cn
E-mail:sjcj@nuaa. edu. cn
Tel/Fax:＋86-025-84892742

基于粗糙超立方体和离散粒子群的特征选择算法

王思朝[1]，罗　川[1]，李天瑞[2]，陈红梅[2]

(1. 四川大学计算机学院，成都 610065;2. 西南交通大学计算机与人工智能学院，成都 611756)

摘　要：特征选择指在保持数据分类性能不变的同时，选出不含冗余特征的特征子集。粗糙超立方体方法可从特征相关度、依赖度和重要度这3方面对特征子集进行综合评估，已成功用于特征选择。特征子集组合的计算是一个NP-难问题，而传统的前向搜索策略只能得到局部最优结果。因此，本文设计了一种新的离散粒子群优化与粗糙超立方体方法相结合的算法。该算法首先引入相关度用以生成一组粒子，然后对粗糙超立方体方法的目标函数改进后作为优化函数，最后由粒子群迭代优化，找到最优的特征子集。实验结果表明，相比传统粗糙超立方体方法和采用粒子群优化的粗糙集方法，本文算法能够得到具有更小特征数量和更高分类性能的特征子集。

关键词：粗糙集；特征选择；组合优化；粗糙超立方体；离散粒子群

中图分类号：TP301.6　　**文献标志码：**A

Feature Selection Based on Rough Hypercuboid and Binary PSO

WANG Sizhao[1], LUO Chuan[1], LI Tianrui[2], CHEN Hongmei[2]

(1. College of Computer Science, Sichuan University, Chengdu 610065, China;2. School of Computing and Artificial Intelligence, Southwest Jiaotong University, Chengdu 611756, China)

Abstract: Feature selection is to choose a subset without containing redundant features, while keeping the classification performance of the data unchanged. Rough hypercuboid approaches can comprehensively evaluate the feature subsets from the three aspects of the relevance, dependency and significance of features, which have been used for feature selection successfully. However, calculating the combination of all feature subsets is NP-hard, and the results obtained by traditional forward search methods is locally optimal. Therefore, a new algorithm based on the rough hypercuboid approach is designed by integrating binary particle swarm optimization. The algorithm first introduces the feature relevance to generate a set of particles, then sets the improved objective function of the rough hypercuboid method as the optimization function, and finally finds the optimal feature subset by iterative optimization of binary particle swarm. By comparing with traditional rough hypercuboid methods and the rough set method based on particle swarm optimization, etc, experimental results demonstrate the proposed algorithm is able to acquire a feature subset with fewer features and higher classification performance.

Key words: rough set; feature selection; combinatorial optimization; rough hypercuboid; binary paticle swarm

基金项目：国家自然科学基金(62076171, 61573292, 61976182)。
收稿日期：2021-04-30；**修订日期：**2021-06-04

引 言

面对日益增长的数据维度和数据量,特征选择以其能够过滤冗余特征,对原始数据进行降维,进而提高学习效率和性能的特点,在文本挖掘[1]、图像处理[2]和信息检索[3]等诸多领域中都有着不可或缺的重要作用。一般来说,过滤式、封装式和嵌入式是目前特征选择主要的3种方法。过滤式特征选择方法在评估特征质量时,往往以某种评价准则为依据排序特征,然后进行挑选,该过程与学习算法无关,如Laplacian得分[4]、Constraint得分[5]等,或是基于某种搜索策略对优化目标进行迭代求解,如基于前向搜索策略的mRMR[6]等。由于过滤式方法在特征选择过程中并不需要构建学习器对特征子集进行评估,因此拥有较高的选择效率。封装式方法会先通过某种搜索方法获取到特征子集,再由评价函数选取,最后采用学习算法对得到的特征子集进行评估。封装式方法中学习算法的选择不尽相同,如常用的决策树、贝叶斯等。该方法的求解结果比较好,主要是有学习算法的介入,但容易出现“过拟合”现象,即由贝叶斯选择出来的特征子集在决策树上的分类效果往往不如人意。此外,构建学习器对特征子集评估的开销较大,相对于过滤式会增加额外的时间消耗,从而效率低下。在嵌入式特征选择方法中,伴随着学习算法的构建,最优特征子集的求解也会一并进行,即把特征选择过程嵌入其中。但是由于嵌入式算法依赖于具体的学习算法,导致其通用性不佳[7]。

粗糙集理论以其在处理不精确和不完备数据的独特优势,为不确定性特征选择问题提供了一套系统的基于决策语义保持的理论框架[8]。经典粗糙集模型无法直接处理含有数值型数据的特征选择任务,所以需要离散数据,但这个操作不可避免地会丢失部分特征信息。针对这一问题,邻域粗糙集[9]、模糊粗糙集[10]以及粗糙超立方体[11]等扩展模型及方法被相继提出,并被引入到面向数值型数据处理的特征选择问题中。然而,目前基于上述模型和方法的特征选择算法中搜索策略均为前向搜索或后向消除的启发式策略,无法得到全局最优的特征子集。近年来,越来越多的元启发式(Meta-heuristic)算法与粗糙集理论相结合被应用于解决特征选择问题,特别是集群智能算法,包括粒子群优化(Particle swarm optimization,PSO)[12]、人工蚁群优化[13]等。Chen等[14]通过条件与决策特征间的互信息定义出特征重要程度,进而提出了基于蚁群优化的特征选择算法;Yamany等[15]采用灰狼优化算法作为搜索优化方法,设计了基于粗糙集正区域的特征选择算法;Chen等[16]在邻域粗糙集模型框架下选择鱼群优化算法寻找最优特征子集,可以很好地处理数值型数据。

粒子群优化算法是集群智能算法中较为常用的方法之一。Wang等[17]提出了一种基于粗糙正域的粒子群优化算法,对比采用基因算法的粗糙集特征选择算法有更好的表现;Bae等[18]对传统的粒子群优化算法进行了改进,提出了一种新的称为智能动态集群的演化算法,同样也是基于粗糙正域,但平均效率高于文献[17]算法;Inbarani等[19]设计了一种基于粗糙集和粒子群优化算法的相对约简和快速约简算法,并应用于医学诊断;Zhang等[20]提出了一种基于邻域粗糙集的离散粒子群优化算法,用于解决基因特征选择问题。由于基于前向搜索策略的粗糙超立方体方法只能得到局部最优结果,而计算所有特征子集组合开销过大,针对这一问题,本文将离散粒子群优化算法和粗糙超立方体方法相结合,提出了一种新颖的基于粗糙超立方体和离散粒子群的特征选择算法(Feature selection based on rough hypercuboid and binary PSO,FSRHBPSO)。该算法在粒子生成阶段引入了特征重要度这一先验知识,以提高收敛效率;并改进了粗糙超立方体的目标函数,消除了特征数量较多时导致特征相关度和重要度值过小的影响,用于优化函数;最后引入了线性递增的变异机制,使粒子不至于困入局部最优解而无法逃脱。

1 相关基础知识

本节阐述了粗糙超立方体方法的基本概念,并且介绍了离散粒子群的相关理论。

1.1 粗糙超立方体

假设论域$\boldsymbol{U}=\{u_1,u_2,\cdots,u_n\}$是一个包含$n$个对象的集合,条件特征集$\boldsymbol{C}=\{A_1,A_2,\cdots,A_m\}$和决策特征集$\boldsymbol{D}=\{d\}$是一个非空的有限集合。由决策特征集$\boldsymbol{D}$对论域$\boldsymbol{U}$进行划分,得到等价类$\boldsymbol{U}/\boldsymbol{D}=\{\beta_1,\beta_2,\cdots,\beta_c\}$。条件特征$A_k\in\boldsymbol{C}$相对于第$i$个等价类$\beta_i$的值域表示为区间$[L_i,U_i]$,该区间包含了所有属于等价类$\beta_i$的对象在$A_k$下的特征取值。

给定特征A_k,假设论域$\boldsymbol{U}$相对于决策特征集$\boldsymbol{D}$有c个等价类,则定义超立方体等价划分矩阵为$\boldsymbol{H}(A_k)=[h_{ij}(A_k)]$,其中

$$h_{ij}(A_k)=\begin{cases}1 & L_i\leqslant x_j(A_k)\leqslant U_i\\ 0 & \text{其他}\end{cases}\tag{1}$$

式中:$h_{ij}(A_k)$表示对象u_j在特征A_k下被分类到等价类β_i的隶属程度,当对象u_j在特征A_k下的特征取值属于区间$[L_i,U_i]$,则$h_{ij}(A_k)$值为1;否则$h_{ij}(A_k)$值为0。式(1)满足以下两个条件:$1\leqslant\sum_{j=1}^{n}h_{ij}(A_k)\leqslant n,\forall i$; $1\leqslant\sum_{i=1}^{c}h_{ij}(A_k)\leqslant c,\forall j$。特征$A_k$下的误分类对象可由矩阵$\boldsymbol{H}(A_k)$诱导出的1个$n$维混淆向量$\boldsymbol{V}(A_k)=[v_1(A_k),v_2(A_k),\cdots,v_n(A_k)]$所辨识,其中

$$v_j(A_k)=\min\left\{1,\ \sum_{i=1}^{c}h_{ij}(A_k)-1\right\}\tag{2}$$

等价类β_i在特征A_k下的粗糙近似集可由A_k的超立方体等价划分矩阵和混淆向量表示为

$$\begin{cases}\underline{A}(\beta_i)=\{u_j|h_{ij}(A_k)=1\text{ and }v_j(A_k)=0\}\\ \overline{A}(\beta_i)=\{u_j|h_{ij}(A_k)=1\}\end{cases}\tag{3}$$

则等价类β_i的边界域定义为

$$\mathrm{BND}_A(\beta_i)=\{u_j|h_{ij}(A_k)=1\text{ and }v_j(A_k)=1\}\tag{4}$$

条件属性A_k和决策特征集$\boldsymbol{D}$的依赖度定义为

$$\gamma_{A_k}(\boldsymbol{D})=\frac{1}{n}\sum_{i=1}^{c}\sum_{j=1}^{n}h_{ij}(A_k)\wedge[1-v_j(A_k)]=1-\frac{1}{n}\sum_{j=1}^{n}v_j(A_k)\tag{5}$$

式中$0\leqslant\gamma_{A_k}(\boldsymbol{D})\leqslant1$。

给定两个条件特征A_k、A_l,特征子集$\{A_k,A_l\}$的超立方体等价划分矩阵可以计算为$\boldsymbol{H}(\{A_k,A_l\})=\boldsymbol{H}(A_k)\wedge\boldsymbol{H}(A_l)$,其中$h_{ij}(\{A_k,A_l\})=h_{ij}(A_k)\wedge h_{ij}(A_l)$。

因此特征A_k相对于特征子集$\{A_k,A_l\}$的重要度为

$$\sigma_{\{A_k,A_l\}}(\boldsymbol{D},A_k)=\frac{1}{n}\sum_{j=1}^{n}[v_j(\{A_l\})-v_j(\{A_k,A_l\})]\tag{6}$$

式中$0\leqslant\sigma_{\{A_k,A_l\}}(\boldsymbol{D},A_k)\leqslant1$。

根据上述定义,可以得到一些特征评估准则,以选择出特征间具有高重要度,特征与决策类别具备高相关度和依赖度的最优特征子集。

假设$\boldsymbol{S}(\boldsymbol{S}\subseteq\boldsymbol{C})$为已选特征子集,则它与决策特征集$\boldsymbol{D}$的平均相关度为

$$J_{\mathrm{relev}}=\frac{1}{|\boldsymbol{S}|}\sum_{A_i\in\boldsymbol{S}}\gamma_{A_i}(\boldsymbol{D})\tag{7}$$

依赖度为

$$J_{\mathrm{depen}}=\gamma_{\boldsymbol{S}}(\boldsymbol{D})\tag{8}$$

特征子集$\boldsymbol{S}$的平均重要度为

$$J_{\text{signf}}=\frac{\sum_{A_i\neq A_j\in \boldsymbol{S}}\{\sigma_{\{A_i,A_j\}}(\boldsymbol{D},A_i)+\sigma_{\{A_i,A_j\}}(\boldsymbol{D},A_j)\}}{|\boldsymbol{S}|(|\boldsymbol{S}|-1)} \tag{9}$$

通过结合上述3个特征评估准则可以构建出以下目标函数,用于挑选最优的特征子集,即

$$J=\tilde{\omega}J_{\text{relev}}+\lambda(1-\tilde{\omega})J_{\text{depen}}+(1-\lambda)(1-\tilde{\omega})J_{\text{signf}} \tag{10}$$

式中$\tilde{\omega}$和λ为2个权重参数。

特征选择过程可以采用一种较为流行的前向搜索策略。依据目标函数启发式地挑选特征,直到所选特征数量满足要求[11]。然而该方法搜索范围仅限于部分特征空间,所得结果也只能看做是局部最优。为了充分发挥粗糙超立方体方法的优势,依赖于离散粒子群优化算法在整个特征空间中的搜索能力,将两者相结合,提出了一个新的用于解决特征选择问题的组合优化算法,以挑选出全局最优的特征子集。

1.2 离散粒子群优化算法

PSO算法是Kennedy等在1995年提出[12]。该算法中粒子通过不断学习自身和群体行为从而实现迭代优化,具体过程为:在D维的问题空间中,随机产生1组粒子,每个粒子可以认为是该问题的1种可行的解决方案,并且用向量$\boldsymbol{X}_i=(x_{i1},x_{i2},\cdots,x_{iD})$和$\boldsymbol{V}_i=(v_{i1},v_{i2},\cdots,v_{iD})$分别描述第$i$个粒子的位置和速度。另外,第$i$个粒子和整个群体在优化搜索过程中,由目标优化函数计算得到的个体以及全局的最佳位置分别记作$\boldsymbol{P}_i=(p_{i1},p_{i2},\cdots,p_{iD})$和$\boldsymbol{P}_g=(p_{g1},p_{g2},\cdots,p_{gD})$。搜索过程中,第$i$个粒子会依据式(11)和式(12)对其当前位置和速度进行重新计算,随机移动以寻找全局最优的解决方案,即

$$v_{ij}^{t+1}=w\times v_{ij}^{t}+c_1\times \text{rand}_1\times(p_{ij}^{t}-x_{ij}^{t})+c_2\times \text{rand}_2\times(p_{gj}^{t}-x_{ij}^{t}) \tag{11}$$

$$x_{ij}^{t+1}=x_{ij}^{t}+v_{ij}^{t+1} \tag{12}$$

式中:w为惯性权重;j为指位置或速度向量的第j维度,$j=1,2,\cdots,D$;t为当前迭代次数,$t=1,2,\cdots,M$,M为最大迭代次数;c_1,c_2为学习因子,通常$c_1=c_2=2$;rand_1、rand_2为介于[0,1]的随机数。此外,$v_{ij}^{t}\in[V_{\min},V_{\max}]$,以避免粒子飞出搜索空间。粒子群优化算法虽然操作简单,易于实现,但是算法随机生成1组粒子,缺乏先验知识,并且容易困入到局部最优的位置。

为了将PSO算法从连续空间应用到离散搜索空间中的优化问题。Kennedy等[21]进一步设计出Binary PSO (BPSO)算法。其中,粒子的位置向量可以用二进制变量表示,即向量中的每1位为1或0。速度向量保持原有形式,不过其数值含义代表了粒子位置的某1位将变为"1"的概率,由Sigmoid函数式(13)将速度向量的连续值映射到[0,1],有

$$s(v_{ij})=\frac{1}{1+\text{e}^{-v_{ij}}} \tag{13}$$

粒子位置更新公式为

$$x_{ij}^{t+1}=\begin{cases}1 & \text{rand}<s(v_{ij}^{t})\\ 0 & \text{其他}\end{cases} \tag{14}$$

式中rand为介于[0,1]间的随机数。将BPSO应用于特征选择问题中,粒子位置向量的每一维就是1个特征,值为1表示特征被选择。因此某一维速度越大,该对应特征被选择的概率也就越高。

2 本文方法

本文方法中每1个粒子代表1种特征选择子集。针对随机生成粒子,缺乏先验知识这一问题,充分考虑了特征与决策类的相关度作为粒子初始化的依据。此外本文还结合特征相关度、依赖度和重要度这3种标准在实验过程中的实际表现情况,对评价标准式(10)进行了改进,作为优化函数。鉴于合适的

优化函数可以帮助优化算法挑选出性能最好的特征子集。本文还引入了遗传算法中的变异机制,进一步加强了粒子的搜索能力。

2.1 粒子编码

粒子群中每个粒子的位置向量对应1个特征子集,那么分量就是1个特征。分量的值仅为1或0。当特征子集包含某个特征时,相应地分量值设为1,否则为0。所以二进制位串代表1种特征选择模式,它的长度应该等于m,即原始特征的总个数。

2.2 粒子初始化

该阶段会初始化I个粒子,I代表粒子群中粒子个数。粒子初始化对于PSO算法收敛速度和结果质量非常重要。不考虑先验知识,随机生成1组粒子虽然在一定程度上有益于寻找最优结果,特别是处理一些高维的优化问题。但是不排除会出现一些粒子距离最优特征子集过远,使得优化算法收敛速度过慢。为解决该问题,本文在粒子初始化阶段考虑了特征与决策特征之间的相关度,并采用了概率的策略,位置向量的生成方法具体为

$$x_{ij}=\begin{cases}1 & \text{rand}\leqslant 1-\dfrac{\text{rank}(\gamma_{A_j}(\boldsymbol{D}))}{m}\\ 0 & \text{其他}\end{cases} \tag{15}$$

式中:x_{ij}为第i个粒子的第j维位置分量;$\gamma_{A_j}(\boldsymbol{D})$为特征$A_j$与决策特征集$\boldsymbol{D}$之间的相关度;$\text{rank}(\gamma_{A_j}(\boldsymbol{D}))$是指对所有特征的相关度降序排列后特征$A_j$相关度的序值。式(15)表明特征的相关度越高,那么该特征序值越小,被选择的可能性越高,该特征对应的位置分量为1的概率也就越大。

2.3 参数设置

惯性权重w可以调节前一时刻的运动状态与现在时刻速度的关系,同时决定了粒子的局部和全局搜索能力。一般来说,希望粒子在前期能在特征解空间中快速找到最优特征子集的大致范围,然后再聚焦该范围,找出最优的特征选择子集。因此,本文选取了线性递减方法,其计算方式为

$$w=w_{\max}-\frac{w_{\max}-w_{\min}}{M}\times t \tag{16}$$

式中$w_{\min}$和$w_{\max}$为w的最小和最大值,需预先设定。式(16)说明粒子在迭代初始运动速度较快,全局搜索能力较强。当处于中后期时,速度会越来越小,有利于粒子很好地进行局部范围探索。

2.4 优化函数

优化函数决定了特征选择的质量,合适的优化函数能帮助BPSO选择出更好的特征子集,即保证分类能力的同时保留更少的特征规模。式(10)中粗糙超立方体方法的目标函数虽然综合考虑了特征子集的平均相关度、依赖度和平均重要度,适合作为优化函数,但实验过程中,本文发现在处理实际数据集时,多数特征子集计算得到的平均相关度和平均重要度要远小于1,特别是所选择的特征子集中特征的个数较多时。这就使得目标函数中依赖度对特征子集的评估影响远大于平均相关度和重要度。出现这种情况的原因是3种评价指标的取值范围不相同,结合式(7),平均相关度的取值范围为

$$0\leqslant J_{\text{relev}}=\frac{1}{|\boldsymbol{S}|}\sum_{A_i\in\boldsymbol{S}}\gamma_{A_i}(\boldsymbol{D})\leqslant\max_{A_i\in\boldsymbol{S}}\{\gamma_{A_i}(\boldsymbol{D})\}$$

式中:$0\leqslant\max\limits_{A_i\in\boldsymbol{S}}\{\gamma_{A_i}(\boldsymbol{D})\}\leqslant 1$。结合式(8),可以看出依赖度的取值范围为[0,1]。同样地,结合式(9),平均重要度的取值范围为

$$0 \leqslant J_{\text{signf}} \leqslant \max_{A_i, A_j \in \boldsymbol{S}} \left\{ \frac{\sigma_{\{A_i, A_j\}}(\boldsymbol{D}, A_i) + \sigma_{\{A_i, A_j\}}(\boldsymbol{D}, A_j)}{2} \right\}$$

式中$0 \leqslant \max\limits_{A_i, A_j \in \boldsymbol{S}} \left\{ \frac{\sigma_{\{A_i, A_j\}}(\boldsymbol{D}, A_i) + \sigma_{\{A_i, A_j\}}(\boldsymbol{D}, A_j)}{2} \right\} \leqslant 1$。

因此,为了保证3种评价标准取值范围相同,提高平均相关度和重要度对评价特征子集的作用,本文选择将两者进行归一化。另外,为了减少特征重要度在迭代时的重复计算,保证算法效率,本文选择在迭代前计算出两两特征的重要度之和,用大小为$m \times m$的特征重要度矩阵$\mathbf{Sig} = \{\text{sig}_{ij}\}$表示,其中

$$\text{sig}_{ij} = \begin{cases} \sigma_{\{A_i, A_j\}}(\boldsymbol{D}, A_i) + \sigma_{\{A_i, A_j\}}(\boldsymbol{D}, A_j) & i \neq j \\ 0 & i = j \end{cases} \tag{17}$$

式中:$1 \leqslant i, j \leqslant m$,$\text{sig}_{ij}$表示特征$A_i$相对于特征子集$\{A_i, A_j\}$的重要度与特征$A_j$相对于特征子集$\{A_i, A_j\}$的重要度之和。

因此,结合粗糙超立方体方法的目标函数式(10),粒子群的优化函数为

$$\text{Fitness} = \tilde{\omega} \frac{J_{\text{relev}}}{\max\limits_{A_i \in \boldsymbol{S}} \{\gamma_{A_i}(\boldsymbol{D})\}} + \lambda(1 - \tilde{\omega}) J_{\text{depen}} + (1 - \lambda)(1 - \tilde{\omega}) \frac{2 \times J_{\text{signf}}}{\max\{\mathbf{Sig}\}} \tag{18}$$

2.5 变异机制

BPSO算法容易出现迭代后期陷入到局部最优的情形,而且该算法本身缺少逃脱局部最优解的机制,为了增加粒子搜索过程中的多样性,本文引入了线性递增的变异机制,有

$$x_{ij} = \begin{cases} \tilde{x}_{ij} & \text{rand} < r_{\text{mut}} \\ x_{ij} & \text{其他} \end{cases} \tag{19}$$

式中

$$r_{\text{mut}} = r_{\min} + \frac{r_{\max} - r_{\min}}{M} \times t \tag{20}$$

式中:r_{mut}为变异率;$r_{\max}$、$r_{\min}$为2个预先设定值。可以看出,粒子的变异概率在优化过程中会越来越大,相应的局部搜索能力也会越来越强。

2.6 时间复杂度分析

算法1概括了本文算法的主要步骤。在其运行过程中,第1步需要计算每个特征的粗糙超立方体等价划分矩阵,该矩阵大小为$n \times c$,c表示类别数,那么其时间复杂度为$O(mnc)$。同样地,第2步中每个特征相关度的计算仍然基于粗糙超立方体等价划分矩阵,其时间复杂度同样为$O(mnc)$。第3步中由于式(6)的计算只涉及两个粗糙超立方体等价划分矩阵的计算,时间复杂度为$O(nc)$,而式(17)中特征重要度矩阵中有m^2个元素,则其时间复杂度为$O(m^2nc)$。在第(5)~(13)步中,主要计算部分为步骤6中的优化函数式(18),由于相关度和重要度已计算出,只需计算依赖度式(8)。而依赖度的计算与所选特征个数有关。最坏情况下,每个特征均被选中,这时对于每个粒子,式(18)所需要的时间复杂度为$O(mnc)$,共有I个粒子,并进行了M次迭代,所以步骤(5)~(13)的时间复杂度为$O(MImnc)$。由以上分析可知,本文算法的时间复杂度为$O(MImnc)$。

算法1 FSRHBPSO算法

输入:决策信息系统$DT = <\boldsymbol{U}, \boldsymbol{C} \cup \boldsymbol{D}>$,原始特征个数$m$,最大迭代次数$M$,粒子个数$I$,学习因子$c_1$,$c_2$,粒子速度、惯性权重和变异率的最小和最大值,即$V_{\min}$,$V_{\max}$,$w_{\min}$,$w_{\max}$和$r_{\max}$,$r_{\min}$,权重参数$\tilde{\omega}$和$\lambda$;

输出:特征子集$\boldsymbol{S}$

(1) 根据式(1)计算出每个特征的粗糙超立方体等价划分矩阵;
(2) 根据式(5)计算出每个特征与决策特征集间的相关度;
(3) 根据式(6)和式(17)得到特征重要度矩阵Sig;
(4) 根据式(15)初始化粒子;
(5) for $t=1$ to M
(6) 根据式(18)为所有粒子计算优化函数值;
(7) 更新所有粒子的个体和全局最优位置
(8) for $j=1$ to I
(9) 依据式(11)计算粒子速度;
(10) 依据式(14)计算粒子位置;
(11) 依据式(19)对粒子执行变异操作;
(12) endfor
(13) endfor
(14) 选择全局最优位置作为最终的特征子集$\boldsymbol{S}$

3 实验与结果

本节选取了多个数据集和对比算法进行特征选择,并对所得子集用两种不同的分类器比较它们的平均性能。最终结果显示,本文算法在大多数情况上,可在保证分类质量的同时具备更少的特征数量。

3.1 数据集

为了更好地测试本文算法性能,本文从UCI数据库选择了6个数据集,它们规模和维度不一,类别数目也有所差异,但均只包含数值型特征,具体描述如表1所示。

表1 数据集描述

Table 1 Details of datasets

No.	Dataset	Instance	Feature	Class
1	Spectfheart	267	44	2
2	Ionosphere	351	33	2
3	Wdbc	569	30	2
4	Segment	2 310	19	7
5	Texture	5 500	40	11
6	Satimage	6 435	36	7

3.2 参数$\tilde{\omega}$和λ的取值分析

将本文算法运行在表1中不同的数据集上,实验中参数$\tilde{\omega}$和λ的取值均从0.0开始,以0.1为间隔递增至1.0,共有11×11种组合,例如$\tilde{\omega}=0.5$和$\lambda=0.5$,其余的实验参数如表2所示。由于BPSO算法运行结果具有随机性,为了避免偶然误差对实验结论的影响,本文算法在每个数据集和每种参数组合都进行了10次实验。此外,本文还选用了Weka[22]中的C4.5和Naive Bayes两种分类算法,采用10次十折交叉验证的方法用于分类精度的评估。图1和图2分别描绘了本文算法在不同的权重参数组合以及C4.5和Naive Bayes两种分类器下的平均分类精度和特征选择个数。从图1、2中可以看出,随着参数$\tilde{\omega}$逐渐减小,λ逐渐增大,除了图1(a)和图2(a)外,其余子图的分类精度都表现出逐步升高的整体趋势,并在接近$\tilde{\omega}=0$和$\lambda=1$该点处又呈现下降趋势。同样地,当参数$\tilde{\omega}$逐渐减小,λ逐渐增大时,所有子图的颜色也在由蓝色逐步过渡到橙色,表明特征选择的个数也在逐渐增加。结合优化函数式(18)可发现,$\tilde{\omega}$减小、λ增大时,特征子集依赖度的权重变大,会引起分类精度的提高,同时特征数量也会增加。虽然本文算法在不同数据集同一分类器或同一数据集不同分类器上取得最优分类精度的权重参数值都不统

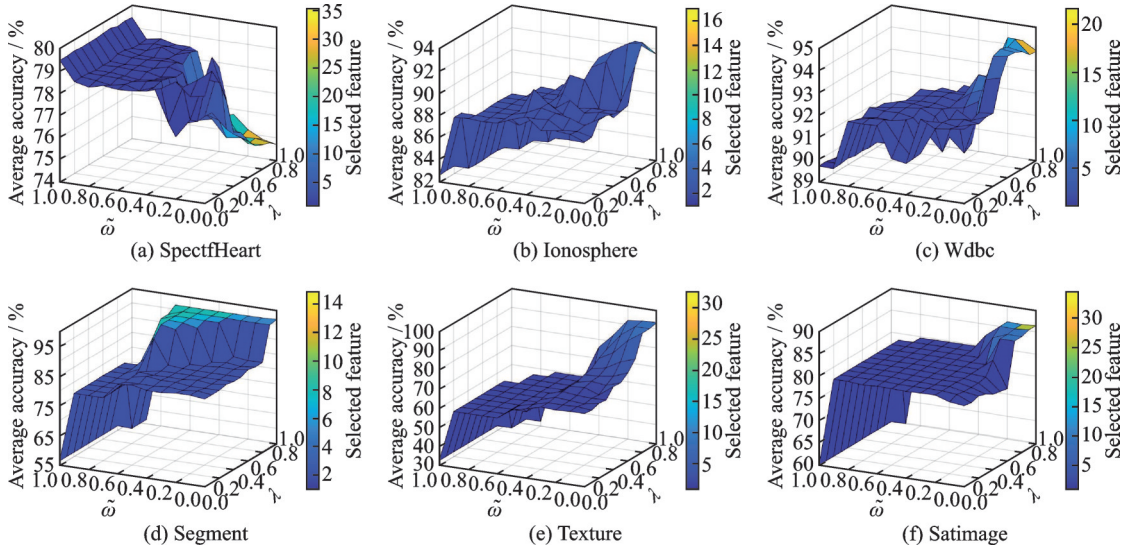

图1 FSRHBPSO算法在6个数据集(C4.5分类器)不同的参数$\tilde{\omega}$和λ组合下的平均分类精度和特征选择个数

Fig. 1 Average classification accuracy and the number of selected features over 6 datasets with C4.5 classifier and the different combinations of parameters $\tilde{\omega}$ and λ

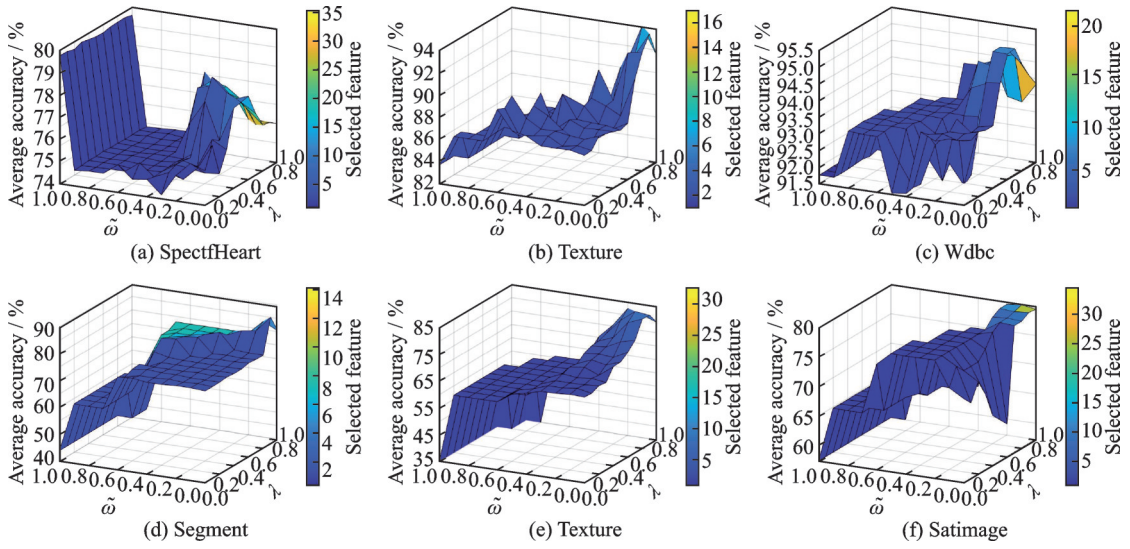

图2 FSRHBPSO算法在6个数据集(Naive Bayes分类器)不同的参数$\tilde{\omega}$和λ组合下的平均分类精度和特征选择个数

Fig.2 Average classification accuracy and the number of selected features over six datasets with Naive Bayes classifier and the different combinations of parameters $\tilde{\omega}$ and λ

一,但是在$\tilde{\omega}\in[0.0, 0.3]$和$\lambda\in[0.6, 0.9]$时,大多数数据集在特征个数较少的同时,拥有较高的平均分类精度。

即使在SpectfHeart数据集上,本文算法取得最佳平均分类精度的位置为$\tilde{\omega}=1$,但是在$\tilde{\omega}\in[0.0, 0.3]$和$\lambda\in[0.6, 0.9]$时,本文算法也可以得到比较好的结果。因为在全部特征下SpectfHeart在C4.5和Naive Bayes分类器上的分类精度分别为74.406 7%和68.935 3%。

3.3 分类精度实验比较与分析

本文在$\tilde{\omega}\in[0.0,0.3]$和$\lambda\in[0.6,0.9]$范围内与其他5种对比算法从特征个数和分类精度两方面进行比较，包括粗糙超立方体加前向搜索策略的RH算法[11]、粗糙集结合粒子群的RS-PSO算法[17]、粗糙集结合智能动态群体优化的RS-IDS算法[18]、邻域粗糙集与离散粒子群相结合的NRS-DPSO算法[20]以及GBNRSFS颗粒球邻域粗糙集方法[23]。权重参数$\{\tilde{\omega},\lambda\}$在6个数据集的具体取值依次为$\{0.3,0.6\}$、$\{0.0,0.8\}$、$\{0.0,0.7\}$、$\{0.0,0.9\}$、$\{0.1,0.9\}$以及$\{0.2,0.9\}$。RS-PSO，RS-IDS和NRS-DPSO三种算法中粒子群参数参考了原文设置，具体数值如表2所示。NRS-DPSO算法的邻域半径为0.16。

表2 4种算法的参数设置

Table 2 Parameter settings of four algorithms

Parameter	M	I	c_1	c_2	$w_{\min}$	$w_{\max}$	$V_{\min}$	$V_{\max}$	$r_{\min}$	$r_{\max}$	α	β	C_w	C_p	C_g
RS-PSO	300	30	2	2	0.4	1.4	$-m/3$	$m/3$	—	—	0.9	0.1	—	—	—
RS-IDS	300	30	2	2	0.4	1.4	$-m/3$	$m/3$	—	—	0.9	0.1	0.1	0.4	0.9
NRS-DPSO	300	30	2	2	0.4	0.9	$-m/3$	$m/3$	—	—	—	—	—	—	—
FSRHBPSO	300	30	2	2	0.9	1.4	−6	6	0.001	0.01	—	—	—	—	—

由于RS-PSO和RS-IDS算法无法直接处理数值特征，本文选择Weka工具中有监督的Kononenko离散化方法，对实验所用数据集进行离散。同样地，为了避免随机误差对实验结果的影响，除RH算法外的5种算法在每个数据集上均进行10次独立地特征选择。表3比较了在所有数据集上，5种算法10次选择结果的平均特征个数(avg)和不同结果个数(diff)，其中平均个数最小的结果加粗表示。此外，由于NRS-DPSO算法的优化函数是基于正区域的，所以当生成的粒子群在迭代优化过程中不存在与条件特征集合的正区域相等的粒子时，便会无法得到有效的特征子集，结果为空集，这里用“—”表示。同时由于RH算法需要指定特征选择的个数，除了数据集Texture设置为7外，其余数据集上均与本文算法所选特征个数保持一致，以验证在相同特征子集大小下，本文算法是否有更好的性能。

表3 特征子集大小比较

Table 3 Comparison of the number of feature subsets

Dataset	RS-PSO		RS-IDS		NRS-DPSO		GBNRSFS		FSRHBPSO	
	avg	diff	avg	diff	avg	diff	avg	diff	avg	diff
SpectfHeart	16	1	27.1	9	10.9	10	8.5	10	**4**	1
Ionosphere	7.8	10	11.3	10	7.6	6	12.1	10	**4**	1
Wdbc	7.7	10	9.7	9	11.3	7	**2**	1	7	1
Segment	5.1	3	6.5	10	—	—	10.3	10	**5**	2
Texture	7.6	10	10.6	10	—	—	22.5	10	**7.2**	2
Satimage	9.4	10	11.7	10	—	—	25.1	10	**6**	1
Average	8.93	7.33	12.82	9.67	9.93	7.67	13.42	8.50	**5.53**	1.33

从表3可以看出，就平均特征个数而言，本文算法在6个数据集的特征个数都要少于RS-PSO、RS-IDS和NRS-DPSO算法，特别是在SpectfHeart数据集上，本文算法的约简率为9.09%，而RS-PSO、RS-IDS和NRS-DPSO的约简率分别为36.36%，61.59%和24.77%。以外，除了在Wdbc数据集上，本文算法的特征选择个数均小于GBNRSFS算法，尤其是在数据集Texture和Satimage上，GBNRSFS算

法的约简率分别是56.25%和69.72%,而本文算法的约简率仅为18%和16.67%。就10次选择结果中不同特征子集个数而言,RS-PSO、RS-IDS和NRS-DPSO三种粒子群算法的平均值分别为7.33,9.67和7.67。这是由于它们很容易陷入局部最优解,从而导致10次独立实验最终收敛的结果各不相同。而GBNRSFS算法的值为8.5,这是其中*k*-means算法聚类结果的不确定造成的,从而使得特征选择结果并不统一。而本文算法除了在数据集Segment和Texture上得到2种不同的特征子集外,在其余数据集上均只有1种结果,这是因为本文算法中的变异机制允许粒子以一定概率逃脱局部最优,从而可以收敛到全局最优的情况。总的来说,本文算法相对于其余算法具有较强的稳定性和全局搜索能力。本文进一步对6种算法的特征选择结果进行平均分类精度的比较,同样用到了Weka中的C4.5和Naive Bayes两种分类器和10次十折交叉验证的方法。表4、5列出了6种算法的特征子集分别在C4.5和Naive Bayes分类器上平均分类精度的比较结果。

表4 特征子集在C4.5分类器上平均分类精度的比较

Table 4 Comparison of average classification accuracy of feature subsets using C4.5 classifier

Dataset	RH	RS-PSO	RS-IDS	NRS-DPSO	GBNRSFS	FSRHBPSO
SpectfHeart	76.179 8	76.464 4	75.142 3	75.868 9	76.209 7	**76.689 1**
Ionosphere	91.763 5	89.928 8	89.339 0	91.079 8	87.498 6	**93.367 5**
Wdbc	93.103 7	93.882 3	94.277 7	94.045 7	92.933 2	**94.409 5**
Segment	96.043 7	95.332 9	94.846 3	—	95.788 3	**96.302 6**
Texture	84.700 4	90.954 5	89.879 1	—	**93.145 1**	92.266 5
Satimage	86.219 1	84.873 3	85.549 5	—	**86.348 2**	86.310 0
Average	88.001 7	88.572 7	88.172 3	86.998 1	88.653 9	**89.890 9**

表5 特征子集在Naive Bayes分类器上平均分类精度的比较

Table 5 Comparison of average classification accuracy of feature subsets using Naive Bayes classifier

Dataset	RH	RS-PSO	RS-IDS	NRS-DPSO	GBNRSFS	FSRHBPSO
Wine	71.254 6	71.374 5	68.307 1	67.558 0	68.760 3	**72.430 7**
Ionosphere	87.498 6	84.532 8	87.601 2	88.208 0	75.755 0	**90.618 2**
Wdbc	93.864 7	93.885 8	93.984 2	93.251 3	90.845 4	**95.295 2**
Segment	81.268 4	81.789 2	76.848 1	—	76.433 3	**87.015 6**
Texture	76.162 9	80.454 2	75.453 1	—	78.902 2	**83.755 1**
Satimage	79.571 9	78.350 4	78.446 3	—	79.589 9	**79.622 4**
Average	81.603 5	81.731 2	80.106 7	83.005 8	78.381 0	**84.789 5**

从表4结果来看,在C4.5分类器的平均分类精度上,本文算法只有在Texture和Satimage数据集上略低于GBNRSFS算法,但是结合特征选择个数来看,本文算法在特征选择个数约为GBNRSFS算法特征个数的1/3和1/4的情况下,与其平均分类精度仅相差0.878 6%和0.038 2%,不足1%。在6个数据集上,相对于其余4个对比算法,本文算法的平均分类精度都要更高。具体来说,本文算法在Ionosphere上相对于RH、RS-PSO、RS-IDS、NRS-DPSO和GBNRSFS算法的平均分类精度分别提高了1.604 0%,3.438 7%,4.025 8%,2.287 7%和5.868 9%。从平均结果上看,本文算法的平均分类精度相对于RH、RS-PSO、RS-IDS和GBNRSFS算法分别提高了1.889 2%,1.318 2%,1.718 6%和1.237%;在前3个数据集上,本文算法比NRS-DPSO也有1.157 2%的提升。

同样地,表5结果显示本文算法在Naive Bayes分类器上仍然优于其他算法,尤其是在Segment和Texture两个数据集上。在Segment上,本文算法相对于RH、RS-PSO、RS-IDS和GBNRSFS算法的平均分类精度分别有5.747 2%,5.224 6%,10.167 5%和10.582 3%的提高;在Texture上,相对其他算法分别提高了7.592 2%,3.300 9%,8.302%和4.852 9%。最后从所有数据集的平均结果来看,本文算法相对于RH、RS-PSO、RS-IDS和GBNRSFS算法仍有较大幅度的优势,具体为3.186 0%,3.058 4%,4.682 8%和6.408 5%;在前3个数据集上,NRS-DPSO的平均分类精度要低于本文算法3.108 9%。总体而言,在多数数据集上,FSRHBPSO算法在C4.5和Naive Bayes分类器的表现都要优于另外5种算法。

4 结束语

本文采用了适用于二元空间的离散粒子群优化算法,引入粗糙超立方体方法中相关度的概念在粒子初始化阶段加入了先验知识;同时添加变异机制,丰富了优化算法搜索过程中的多样性。此外还根据实验过程中粗糙超立方体3种评估标准的实际表现情况,对目标函数进行了改进,并将改进后的目标函数与离散粒子群优化算法相结合,设计了新的基于离散粒子群和粗糙超立方体的特征选择算法。最后实验结论表明该算法能在保证分类质量的同时选择出数量更少的特征子集。同时相比前向搜索策略的粗糙超立方体方法及其他粒子群算法具有更好的性能。下一步工作考虑将该算法与云计算相结合,以增强算法面对大规模数据的计算能力。

参考文献:

[1] KOU G, YANG P, PENG Y, et al. Evaluation of feature selection methods for text classification with small datasets using multiple criteria decision-making methods[J]. Applied Soft Computing, 2020, 86: 105836.

[2] LIANG Y, ZHANG M, BROWNE W N. Image feature selection using genetic programming for figure-ground segmentation [J]. Engineering Applications of Artificial Intelligence, 2017, 62: 96-108.

[3] JI X, SHEN H W, RITTER A, et al. Visual exploration of neural document embedding in information retrieval: semantics and feature selection[J]. IEEE Transactions on Visualization and Computer Graphics, 2019, 25(6): 2181-2192.

[4] HE X, CAI D, NIYOGI P. Laplacian score for feature selection[C]//Proceedings of Advances in Neural Information Processing Systems. Cambridge: MIT Press, 2006: 507-513.

[5] ZHANG D, CHEN S, ZHOU Z H. Constraint score: A new filter method for feature selection with pairwise constraints[J]. Pattern Recognition, 2008, 41(5): 1440-1451.

[6] PENG H, LONG F, DING C. Feature selection based on mutual information: Criteria of max-dependency, max-relevance, and min-redundancy[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2005, 27(8): 1226-1238.

[7] ZHANG X, WU G, DONG Z, et al. Embedded feature-selection support vector machine for driving pattern recognition[J]. Journal of the Franklin Institute, 2015, 352(2): 669-685.

[8] PAWLAK Z. Rough sets[J]. International Journal of Computer & Information Sciences, 1982, 11(5): 341-356.

[9] HU Q, YU D, XIE Z. Neighborhood classifiers[J]. Expert Systems with Applications, 2008, 34(2): 866-876.

[10] JENSEN R, SHEN Q. Semantics-preserving dimensionality reduction: Rough and fuzzy-rough-based approaches[J]. IEEE Transactions on Knowledge and Data Engineering, 2004, 16(12): 1457-1471.

[11] MAJI P. A rough hypercuboid approach for feature selection in approximation spaces[J]. IEEE Transactions on Knowledge and Data Engineering, 2014, 26(1): 16-29.

[12] KENNEDY J, EBERHART R. Particle swarm optimization[C]//Proceedings of IEEE International Conference on Neural Networks. Washington D C: IEEE Press, 1995: 1942-1948.

[13] DORIGO M, BLUM C. Ant colony optimization theory: A survey[J]. Theoretical Computer Science, 2005, 344(2): 243-278.

[14] CHEN Y, MIAO D, WANG R. A rough set approach to feature selection based on ant colony optimization[J]. Pattern

Recognition Letters, 2010, 31(3): 226-233.
[15] YAMANY W, EMARY E, HASSANIEN A E. New rough set attribute reduction algorithm based on grey wolf optimization [M]. Cham: Springer International Publishing, 2016: 241-251.
[16] CHEN Y, ZENG Z, LU J. Neighborhood rough set reduction with fish swarm algorithm[J]. Soft Computing, 2017, 21(23): 6907-6918.
[17] WANG X, YANG J, TENG X, et al. Feature selection based on rough sets and particle swarm optimization[J]. Pattern Recognition Letters, 2007, 28(4): 459-471.
[18] BAE C, YEH W C, CHUNG Y Y, et al. Feature selection with intelligent dynamic swarm and rough set[J]. Expert Systems with Applications, 2010, 37(10): 7026-7032.
[19] INBARANI H H, AZAR A T, JOTHI G. Supervised hybrid feature selection based on PSO and rough sets for medical diagnosis[J]. Computer Methods & Programs in Biomedicine, 2014, 113(1): 175-185.
[20] 张哲, 孙丽君. 基于离散粒子群优化和邻域约简的基因特征选择算法[J]. 计算机工程, 2016, 42(3): 188-191, 197.
ZHANG Zhe, SUN Lijun. Gene feature selection algorithm based on discrete particle swarm optimization and neighborhood reduction[J]. Computer Engineering, 2016, 42(3): 188-191,197.
[21] KENNEDY J, EBERHART R C. A discrete binary version of the particle swarm algorithm[C]//Proceedings of IEEE International Conference on Systems, Man, and Cybernetics. Washington D C: IEEE Press, 1997: 4104-4108.
[22] WAIKATO M L G, Weka 3-data mining with open source machine learning software in Java [EB/OL]. (2020-01-16)[2021-03-06]. https://www.cs.waikato.ac.nz/ml/weka/.
[23] XIA S, ZHANG Z, LI W, et al. GBNRS: A novel rough set algorithm for fast adaptive attribute reduction in classification[J]. IEEE Transactions on Knowledge and Data Engineering, 2020,34(3): 1231-1242.

作者简介:

王思朝(1996-),男,硕士研究生,研究方向:特征选择、并行计算等,E-mail: wangsiczhao@stu. scu. edu. cn。

罗川(1987-),**通信作者**,男,副教授,博士生导师,研究方向:数据挖掘、粒计算等,E-mail:cluo@scu.edu.cn。

李天瑞(1969-),男,教授,博士生导师,研究方向:数据挖掘、粒计算等。

陈红梅(1971-),女,教授,博士生导师,研究方向:数据挖掘、粒计算等。

(编辑:刘彦东)