

# 基于自适应粒子群算法的特征选择

李 策,王保云,高 浩

(南京邮电大学 自动化学院,江苏 南京 210003)

**摘 要:**在模式分类问题中,数据往往存在不相关或冗余的特征,从而影响分类的准确性。特征选择的提出,很好地解决了这一问题。特征选择的关键在于利用最少的特征获得最佳的分类效果。为了达到这一目的,一种基于自适应粒子群的特征选择的理论被提出。相比于原始的粒子群算法,在初始过程中引入混沌模型增加其初始粒子的多样性,在更新机制中引入自适应因子增加其全局搜索能力。同时将特征数目引入到适应度函数中,在迭代前期通过惩罚因子调节分类准确率和特征数目对于适应度函数的影响,在迭代中后期惩罚因子恒定,使特征数目对于适应度函数的影响趋于稳定。自适应粒子群算法具有很好的全局收敛性,能够避免陷入局部最优,尤其适合高维数据的降维问题。大量的理论分析和仿真实验的结果表明,与其他粒子群算法(PSO)的特征选择结果相比,在数据特征数目各异的情况下,该算法具有更好的分类效果,同时表明了所提算法的可行性以及优越性。

**关键词:**特征选择;粒子群算法;分类;自适应;封装

中图分类号:TP391

文献标识码:A

文章编号:1673-629X(2017)04-0089-05

doi:10.3969/j.issn.1673-629X.2017.04.020

## Feature Selection Based on Adaptive Particle Swarm Optimization

LI Ce, WANG Bao-yun, GAO Hao

(College of Automation, Nanjing University of Posts and Telecommunications,  
Nanjing 210003, China)

**Abstract:** In pattern classification problems, there is often irrelevant or redundant features in data, thus affecting the accuracy of the classification. Feature selection is proposed to be a good solution to this problem. The key of feature selection is to use the least feature for the best classification results. In order to achieve this object, a theory based on adaptive particle swarm feature selection is presented. Compared to the original particle swarm optimization, chaos model is introduced in the initial process of increasing its diversity of primary particles, the introduction of adaptive factor to increase its global search capability in the update mechanism. At the same time the number of features will be introduced to the fitness function, in the early iterations adjustment classification accuracy and the number of features by penalizing factor for adapting to the impacts of the function, in the latter part of the penalty factor constant iteration, bringing the number of features of the fitness function tends to affect stable. Adaptive particle swarm algorithm has good global convergence and can avoid falling into local optimum, especially for lower-dimensional problem of high dimensional data. A large number of theoretical analysis and simulation results show that compared with other PSO feature of the election results, in the case where the number of different data characteristics, this algorithm has better classification results. Also it shows that the proposed algorithm is feasible and superior.

**Key words:** feature selection; PSO; classification; adaptive; wrapper

## 0 引言

特征选择也叫特征子集选择。指从已有的特征中选择  $N$  个特征使系统的特定目标最优化,从输入特征中选择出一些最有效特征以降低数据集维度的过程<sup>[1]</sup>,是提高学习算法性能的一个重要手段,也是模式

识别中关键的数据预处理步骤。特征选择应用在分类,对于机器学习和数据挖掘是一项重要的任务。分类过程中,数据集往往包含大量的特征,但是并不是所有的特征对于分类都是有用的,其中很多无关和冗余的特征会导致分类效果的不佳,同时导致维度的复杂

收稿日期:2016-05-08

修回日期:2016-09-08

网络出版时间:2017-03-07

基金项目:国家自然科学基金资助项目(61271232,61372126);东南大学移动通信国家重点实验室开放研究基金(2012D05)

作者简介:李 策(1991-),男,硕士研究生,研究方向为特征选择;王保云,博士,教授,博导,研究方向为香农信息论,无线通信中的博弈与协作、信号处理技术,视频信息的分析与理解;高 浩,博士,副教授,研究方向为优化算法、图像处理的理论和实际应用。

网络出版地址: <http://cnki.net/kcms/detail/61.1450.TP.20170307.0921.024.html>

性,因而特征选择是一种非常有效的解决方式<sup>[2]</sup>。

特征选择算法根据评价函数的不同可以分为封装模式和过滤模式<sup>[3]</sup>。其中过滤模式通过分析特征子集内部的特点来衡量其好坏,一般用作预处理,与分类器的选择无关。封装模式实质上是一个分类器,用选取的特征子集对样本集进行分类,分类精度作为衡量特征子集好坏的标准。在过滤模式中,学习算法作为独立的一部分,而在封装模式中是作为评价功能的一部分,可以取得较好的分类效果。选择以粗糙集作为分类器的封装模型。特征选择的任务实际是一个组合优化问题<sup>[4]</sup>,特征选择过程中,往往特征数目繁多,搜索空间较大,所以需要搜索算法去获得最优的选择方案,存在的搜索方法有序列前向选择 (SFS)<sup>[5]</sup> 和序列后向选择 (SBS)<sup>[6]</sup>。但是这些算法不仅需要很大的计算代价,同时容易陷入局部最优。因此需要具有全局搜索能力的算法应用到特征选择中。例如,粒子群算法 (PSO)<sup>[7-10]</sup>、遗传算法 (GA)<sup>[11]</sup> 和遗传编程算法 (GP)<sup>[12]</sup>。三种算法相比,粒子群算法具有快速、简单等优势,因而基于粒子群算法的特征选择往往具有更好的应用效果。而基于自适应粒子群的特征选择方法,相比于原始的粒子群算法,在初始过程引入混沌模型增加其初始粒子的多样性,在更新机制引入自适应因子增加其全局搜索能力。同时将特征数目引入到适应度函数中,在迭代前期通过惩罚因子调节分类准确率和特征数目对于适应度函数的影响,在迭代中后期惩罚因子恒定,使特征数目对于适应度函数的影响趋于稳定。实验结果表明,该算法具有优越性。

## 1 特征选择和粒子群算法

### 1.1 封装模式的特征选择

特征选择是选取一系列的特征构成一个特征子集,能够使得这个子集有比特征全集相同或者更好的分类功能。一般分为四个过程:产生过程、评价函数、停止准则、验证过程。一般流程如图 1 所示<sup>[2]</sup>。

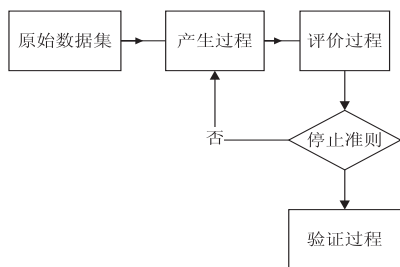


图 1 特征选择的一般流程

特征选择中的封装模式最早由 John 等于 1994 年提出<sup>[1]</sup>。在该模式中,分类学习算法封装在特征选择的过程中,并以特定学习算法的性能作为子集的评价标准。在特征空间中搜索出对分类器具有较高性能的

特征子集,直接构造分类模型。当满足一定条件(一般设定的迭代次数)时,将最优的特征子集输出,同时获得一个具有较高分类性能模型。在封装模式的特征选择方法中,有很多用来评价特征子集的学习算法,如贝叶斯分类器、近邻法、支持向量机及神经网络等,这些方法都是直接利用分类器的分类性能来评价特征子集的优劣。

### 1.2 粒子群算法

粒子群算法是于 1995 年由 Kennedy 和 Eberhart 提出的,通过模拟鸟群觅食行为而发展起来的一种基于群体协作的随机搜索算法<sup>[13]</sup>。其描述如下:

假设在一个  $D$  维的目标搜索空间,  $m$  个粒子组构成一个群落,其中第  $i(i = (1, 2, \dots, m))$  个粒子表示一个  $D$  维向量  $X_i = (x_{i1}, x_{i2}, \dots, x_{id})$ 。第  $i$  个粒子在目标搜索空间的位置是  $x_i$ 。每一个粒子所在的位置就是一个潜在的备选解,将  $x_i$  带入到目标函数中,计算其适应度值。每个粒子  $i$  还有一个速度决定它们飞翔的方向和距离,  $V_i = (v_{i1}, v_{i2}, \dots, v_{id})$  也是一个  $D$  维向量。第  $i$  粒子搜索到的最优位置记为  $p_b$  (局部最优解),整个粒子群迄今为止搜索到的最佳位置记为  $p_g$  (全局最优解)。

粒子速度和位置的更新公式如下:

$$V_{id}^{t+1} = w * V_{id}^t + c_1 * \text{rand} * (p_b - X_{id}^t) + c_2 * \text{rand} * (p_g - X_{id}^t) \quad (1)$$

$$X_{id}^{t+1} = X_{id}^t + V_{id}^{t+1} \quad (2)$$

其中,  $d = 1, 2, \dots, D, i = 1, 2, \dots, m, m$  为种群规模;  $c_1$  和  $c_2$  为加速常数;  $\text{rand}$  为  $[0, 1]$  之间任意的随机数;  $t$  为迭代次数。

上述的粒子群算法,只适应于连续问题的求解。1997 年, Kennedy 和 Eberhart 提出了离散二进制粒子群算法 (BPSO)<sup>[14]</sup>,这种算法采用的是二进制的编码形式。在二进制的粒子群算法中,粒子的位置  $X_i$ ,是用二进制的字符串表示 (10110011) 而速度向量不做这种要求。粒子的位置更新公式变为:

$$X_{id} = \begin{cases} 1 & \text{rand} > S(v_{id}) \\ 0 & \text{otherwise} \end{cases} \quad (3)$$

$$\text{其中, } S(v_{id}) = \frac{1}{1 + e^{-v_{id}}}。$$

## 2 自适应粒子群算法

在原始粒子群算法的基础上提出自适应粒子群算法,舍弃粒子的速度,采用高斯模型,利用粒子局部的全局最优解来更新粒子的位置公式:

$$x_{id} = \begin{cases} N(\mu, \sigma) & \text{rand} < 0.5 \\ P_{\text{best}}(t)_{id} & \text{otherwise} \end{cases} \quad (4)$$

其中,  $\mu = \alpha * \text{Pbest}_{id}(t) + (1 - \alpha) * \text{Gbest}_d(t)$  ;  
 $\sigma = |\text{Pbest}_{id} - \text{Gbest}_d| + \Delta, \Delta = \text{rand} * |\text{Pbest}_{r1} - \text{Pbest}_{r2}| * e^{\frac{f(\text{Gbest}) - f(x_d)}{1}}$ 。

与其他粒子群算法相比,引入自适应机制,“ $\Delta$ ”值根据当前全局最优值与当前适应度值的差以及随机选择的某一维数的两个局部最优解的差进行调整。在随机局部最优解差值一定的情况下,干扰值“ $\Delta$ ”随着  $x_i$  与  $\text{Gbest}$  的适应度差值越小而增大,如果粒子的适应度值达到其全局最优时,则产生的干扰值最大,粒子将被最大幅度的干扰影响。在这种情况下,干扰就像一个重新初始化的操作,从而使得粒子跳出局部最优。随机局部最优解差值的引入提高了当前邻域内细化搜索的能力,在迭代结束时,所有的个体极值收敛到一个位置,即  $|\text{Pbest}_{r1} - \text{Pbest}_{r2}|$  将收敛到零,干扰也将收敛为零,作为一个必要条件,保证了种群的收敛。在改进粒子位置更新公式的同时,将混沌模型引入种群的初始化过程,一般的种群初始化是随机,即  $x_{id} = \text{rand}$ 。混沌模型的初始化进程为:

$$\begin{aligned} c_r &= \text{rand} \\ x_{id} &= \begin{cases} 4 * c_r * (1 - c_r) & \text{otherwise} \\ c_r + 0.1 * \text{rand} & c_r = \{0, 0.25, 0.5, 0.75\} \\ c_r - 0.1 * \text{rand} & c_r = 1 \end{cases} \end{aligned} \quad (5)$$

该初始化模式增加了种群的多样性,避免粒子陷入局部最优。

粒子的局部和全局的位置更新公式分别为:

$$\begin{aligned} P_{bi}(t+1) &= \begin{cases} P_{bi}(t) & \text{其他} \\ X_i(t+1) & F(P_{bi}(t)) < F(X_i(t+1)) \\ X_i(t+1) & F(P_{bi}(t)) = F(X_i(t+1)) \&\& \\ & N(P_{bi}(t)) \geq N(X_i(t+1)) \end{cases} \quad (6) \\ P_{gi}(t+1) &= \begin{cases} P_{gi}(t) & \text{其他} \\ P_{bi}(t+1) & P_{gi}(t) < P_{bi}(t+1) \\ P_{bi}(t+1) & P_{gi}(t) = P_{bi}(t+1) \&\& N(P_{gi}(t)) \geq N(P_{bi}(t+1)) \end{cases} \end{aligned} \quad (7)$$

其中,  $P_{bi}(t+1)$  为粒子个体的最优位置;  $P_{gi}(t+1)$  为粒子整体的最优位置;  $F$  为适应度函数;  $X_i(t+1)$  为粒子的当前位置;  $N$  为特征数目。

粒子的位置更新是与一般的位置更新方式不同,当适应度值与最优值相等时,通过比较特征数目来更新当前位置。特征选择的目标是利用较少的特征达到最佳的优化效果,位置更新公式的改进有利于利用更

少的特征数目。

### 3 适应度函数

传统的基于粒子群算法的特征选择方法采用封装模式的评价函数,由于追求高的分类效果,一般设计:

$$\text{Accuracy} = \frac{\# \text{features}}{\# \text{allfeatures}} \quad (8)$$

其中,  $\text{Accuracy}$  为最终特征子集的分类精确率;  $\# \text{features}$  为准确预测的样本特征;  $\# \text{allfeatures}$  为输入的特征样本。

#### 3.1 改进的适应度函数

每个特征子集包含一定数量的特征,如果两个特征子集取得准确率相同,包含特征数目较少的特征子集就该被选中。所以,在适应度函数中引入被选择的特征子集的数目是很有必要的,所以适应度函数为:

$$\text{Fitness} = (1 - \alpha) * \text{Accuracy} + \alpha * \frac{1}{\#S} \quad (9)$$

$$\alpha = \begin{cases} t/T & t < 20 \\ 0.8 & t \geq 20 \end{cases}$$

其中,  $\#S$  为被选择的特征子集的特征数目;  $\alpha$  为一个惩罚因子,是控制分类精确率和特征数目对于适应度函数的影响,其取值范围为  $[0, 1]$ ;  $T$  为迭代次数;  $t$  为当前迭代次数。

在封装模式中,适应度函数用来评价特征子集的好坏,数据分类的准确率在适应度函数中应该起到主导作用。该实验中,迭代次数为 100,随着迭代次数的增加,迭代的后期  $\alpha$  不断增大,导致特征数目成为适应度函数的主导,直接导致迭代后期分类精确率的下降。所以应该控制  $t$  的取值范围。实验数据表明,当  $t$  最大值取 20 时,数据的分类精确率相比于其他算法有一定的提高,还能降低被选择特征子集中的特征数目。当  $t$  为 20 时,即前 20 次的迭代过程中,迭代开始分类准确率主导适应度函数,随着迭代次数的增加被选特征子集中的特征数目对于适应度函数的影响增加,当迭代到达 20 次时,  $\alpha$  的取值固定为 0.8。迭代的中期以及后期,特征数目分类准确率对于适应度函数的影响达到平衡。

#### 3.2 邻近算法

封装模式的算法都需要学习算法对特征子集进行评估。采用的算法是邻近算法<sup>[15]</sup>。邻近算法也称  $K$  最近分类算法,是数据挖掘分类技术中最简单的方法之一。KNN 算法的核心思想是,如果一个样本在特征空间中的  $K$  个最相邻的样本中的大多数属于某一个类别,则该样本也属于这个类别,并具有这个类别上样本的特性。KNN 算法中,所选择的邻居都是已经正确分类的对象。同时它依赖于极限定理,在类别决策时,只



与极少量的相邻样本有关。邻近算法具有简单,易于理解,易于实现,无需估计参数,无需训练等特点。

由于邻近算法具有的优势,将其作为特征选择的分类评价函数。采用留一验证(LOOCV)的一阶邻近算法( $K$ 取1)。LOOCV 指只使用原本样本中的一项来当作验证资料,而剩余的则留下来当作训练资料。这个步骤一直持续到每个样本都被当作一次测试资料。实验中采用留一验证,将输入数据分成训练集和测试集,并用邻近算法对选出的特征子集进行评价。这个过程一直持续到迭代结束。

4 解 码

基于离散粒子群算法的特征选择中,粒子的位置向量  $X_i = (x_{i1}, x_{i2}, \cdots, x_{id})$  代表一个特征子集,其中  $x_{i1}$  就代表一个特征。在离散粒子群算法中,  $x_{i1}$  是二进制,当  $x_{i1}$  取0时,就代表这个特征没有被选中,当  $x_{i1}$  为1时,就代表该特征被选中。例如,  $X_i = (01000111)$  代表该特征子集中有八个特征,1 的个数为4,代表4个特征被选中。在以往的特征选择算法中,都是采用的离散粒子群算法。粒子的位置向量由十进制转换为二进制,采用的是 Sigmoid 函数:  $S(v_{id}) = \frac{1}{1 + e^{-v_{id}}}$ 。但是在自适应粒子群算法中,舍弃了粒子的速度向量,无法使用 Sigmoid 函数。所以需要引入解码机制。将特征被选进特征子集的可能性作为解码的原理,把粒子群中的粒子作为解决问题的一个潜在方法。粒子  $X_i = (x_{i1}, x_{i2}, \cdots, x_{id})$ ,  $x_{i1} \in [0,1]$  代表第一个特征被选择的可能性:

$$e_{ij} = \begin{cases} 0 & \text{其他} \\ 1 & x_{ij} > \text{rand} \end{cases} \quad (10)$$

当  $e_{ij} = 1$  时,就代表第  $j$  个特征被选进特征子集。如果为0,不被选中。在统计被选中的特征数目时,可直接统计  $e_{ij}$  中1的个数,直接对其求和就可以计算出被选中的特征数目。

5 实验设计

实验采用UCI<sup>[16]</sup>机器学习库中的数据,选择其中的 Glass, Sonar, Wine, Vowel, WDBC, Segmentation, Satellite, Ionosphere 这些最为常用的数据进行分类比较。

5.1 数据表格

数据表格如表1所示。

5.2 参数设定

粒子群种群数目为20,迭代次数为100,邻近算法  $K$  值取1。

为了更好地说明自适应粒子群算法对于数据分类的优越性,选择离散粒子群算法(BPSO)<sup>[9]</sup>和混沌映射

离散粒子群算法(CBPSO)<sup>[7]</sup>;还有 Bare bones 离散粒子群算法<sup>[10]</sup>、Catfish BPSO(基于鲶鱼效应的离散粒子群算法<sup>[17]</sup>)、APSO(改进的适应度函数的算法)。比较不同算法作用在相同数据集上的分类效果,同时比较被选特征子集中的特征数目。上述其他算法的迭代次数同样设置为100。

表 1 实验数据

数据	特征数目	类别数目	样本数目
Wine	13	3	178
Vowel	10	11	990
WDBC	30	2	569
Vehicle	18	4	846
Satellite	36	6	6 435
Sonar	60	2	208
Segmentation	19	7	2 310
Ionosphere	34	2	351
Glass	10	7	214

6 实验结果分析

在表2中,可以明显发现,基于自适应粒子群算法的特征选择方法相比于其他算法能够取得很好的分类效果,同时还减少了被选择特征的数目。数据集特征数目比较小时,例如 Glass(  $D = 10$  ),基于粒子群算法的特征选择分类方法都能取得百分之百的分类效果。自适应粒子群算法分类效果也能够达到百分之百,被选特征子集中的特征数目多一个。在数据集特征数目较少时,自适应粒子群算法不具有优势。在数据集 Vowel 中,自适应粒子群算法相比于其他算法的效果会好很多,不仅分类的精确度高,相比于 BPSO 提高了接近0.5%,相比于其他算法提高了0.3%。而且被选的特征数目也比其他粒子群算法要少,相比 Bare bones 粒子群算法特征数目少了一半。在数据 Wine 中,自适应粒子群算法分类效果能够达到百分之百,相比于 Bare bones 粒子群算法提高了0.6%。同时被选的特征数目相比于其他粒子群算法要少。在数据集 Vehicle 中,各种算法对于该数据集的分类效果普遍不高,APSO 相比于其他算法分类的准确率有明显提高,相比于 Bare bones 粒子群算法提高了近1%,同时被选特征的数目与 Bare bones 粒子群算法接近。当在数据集特征数目较多,例如在数据集 Satellite 和 Sonar 中,APSO 相比于 Bare bones 粒子群算法的优势更加明显,不仅能够获得很好的分类效果,同时有效减少了被选特征的数目。

将特征子集的特征数目引入到适应度函数中,对于特征子集进行评价,不仅能够降低特征子集的特征

数目,同时对于分类效果也产生了有益的影响。改进的适应度函数相比于改进算法更新机制对于特征选择有更加直接的效果。特别是在特征数目较多的特征子集进行分类的过程中,影响体现的更加明显。

表2 各算法分类效果对比

数据	BPSO		CBPSO		Cat BPSO		Bare PSO		APSO	
	特征数	分类准确率/%	特征数	分类准确率/%	特征数	分类准确率/%	特征数	分类准确率/%	特征数	分类准确率/%
Glass	3	100	3	100	3	100	3	100	4	100
Vowel	9	99.49	8	99.44	8	99.7	8	99.7	4	99.9
Wine	8	98.88	8	99.44	8	99.44	5	99.45	3	100
Vehicle	11	74.7	12	75.06	12	75.06	6.8	75.54	7	76.5
Segmentation	11	97.88	10	97.92	10	97.92	11.2	98.24	6	98.27
WDBC	13	97.72	15	97.54	14.2	98.24	12.9	98.36	13	98.64
Ionosphere	10	93.73	15	96.02	8	96.01	11.4	96.21	13	96.87
Satellite	25	91.45	NA	NA	26.8	91.62	NA	NA	22	92.17
Sonar	31	96.15	26	95.57	30.2	96.92	28.2	96.08	27	96.63

7 结束语

基于自适应粒子群的特征选择的算法,在初始过程中引入混沌模型增加了初始粒子的多样性,在更新机制中引入自适应因子增加其全局搜索能力。同时将特征数目引入到适应度函数中,在迭代前期通过惩罚因子调节分类准确率和特征数目对于适应度函数的影响,在迭代中后期惩罚因子恒定,使特征数目对于适应度函数的影响趋于稳定。自适应粒子群算法具有很好的全局收敛性,能够避免陷入局部最优,尤其适合高维数据的降维问题。实验结果表明,与其他粒子群算法的特征选择结果相比,在数据特征数目各异的情况下,该算法具有更好的分类效果。

参考文献:

[1] 张丽新,王家钦,赵雁南,等. 机器学习中的特征选择[J]. 计算机科学,2004,31(11):180-184.

[2] Dash M,Liu H. Feature selection for classification[J]. Intelligent Data Analysis,1997,1(1):131-156.

[3] 周 城,葛 斌,唐九阳,等. 基于相关性和冗余度的联合特征选择方法[J]. 计算机科学,2012,39(4):181-184.

[4] 陈 彬,洪家苯. 最优特征子集选择问题[J]. 计算机学报,1997,20(2):133-138.

[5] Marill T,Green D M. On the effectiveness of receptors in recognition systems[J]. IEEE Transactions on Information Theory,1963,9(1):11-17.

[6] Whitney A W. A direct method of nonparametric measurement selection[J]. IEEE Transactions on Computers,1971,20(9):1100-1103.

[7] Heidari-Bateni G,McGillem C D. A chaotic direct-sequence spread-spectrum communication system[J]. IEEE Transactions on Communications,1994,42(234):1524-1527.

[8] Unler A,Murat A. A discrete particle swarm optimization method for feature selection in binary classification problems[J]. European Journal of Operational Research,2010,206(3):528-539.

[9] Yang C S,Chuang L Y,Ke C H,et al. Boolean binary particle swarm optimization for feature selection[C]//Proceedings of CEC. [s.l.]:IEEE,2008:2093-2098.

[10] Zhang Y,Gong D,Hu Y,et al. Feature selection algorithm based on bare bones particle swarm optimization[J]. Neuro-computing,2015,148:150-157.

[11] Yuan H,Tseng S S,Gangshan W,et al. A two-phase feature selection method using both filter and wrapper[C]//IEEE international conference on systems,man, and cybernetics. [s.l.]:IEEE,1999:132-136.

[12] Neshatian K,Zhang M. Dimensionality reduction in face detection;a genetic programming approach[C]//24th international conference on image and vision computing. New Zealand:IEEE,2009:391-396.

[13] Kennedy J. Particle swarm optimization[C]//International conference on neural networks. [s.l.]:IEEE,1995:1942-1948.

[14] Kennedy J,Eberhart R C. A discrete binary version of the particle swarm algorithm[C]//International conference on systems,man, and cybernetics. [s.l.]:IEEE,1997:4104-4108.

[15] Cover T M,Hart P E. Nearest neighbor pattern classification[J]. IEEE Transactions on Information Theory,1967,13(1):21-27.

[16] Asuncion A, Newman D. UCI machine learning repository[R]. California:University of California Irvine,2007.

[17] Chuang L Y,Tsai S W,Yang C H. Improved binary particle swarm optimization using catfish effect for feature selection[J]. Expert Systems with Applications,2011,38(10):12699-12707.