doi:10.3969/j.issn.1001-4616.2021.01.017

基于神经网络树和人工蜂群优化的数据聚类

吉珊珊

(东莞职业技术学院计算机工程系,广东 东莞 523808)

[摘要] 针对高维数据引起的“维数灾难”问题,设计了一种基于神经网络树和人工蜂群优化的高维数据聚类算法. 首先,设计了改进的二元人工蜂群优化算法,以封装式方法最大化径向基函数网络的准确率,以过滤式方法最小化特征的冗余度;然后,基于每个特征子集的样本集训练径向基函数网络,构建以径向基函数网络为节点的神经树;最终,采用门网络将连接的类簇分离,获得最终的聚类结果. 基于高维数据集和低维数据集均完成了仿真实验,结果表明本算法对于高维数据集实现了较高的聚类准确率.

[关键词] 高维数据,神经网络树,人工蜂群优化,聚类算法,特征选择
[中图分类号]TP391 [文献标志码]A [文章编号]1001-4616(2021)01-0119-09

Neuron Network Tree and Artificial Bee Colony Optimization Based Data Clustering Algorithm

Ji Shanshan

(Department of Computer Enginneering, Dongguan Polytechnic, Dongguan 523808, China)

Abstract: Focusing on the “curse of dimensionality” problem caused by high dimensional data, a neuron network tree and artificial bee colony optimization based clustering algorithm for high dimensional data is designed. Firstly, an improved binary artificial bee optimization algorithm is designed, the accuracy of radial basis function network is maximized by a wrapper method, the feature redundancy is minimized by a filter method; then, a radial basis function network is trained by samples corresponding to each feature, a neuron tree that each node consists of a radial basis function network is constructed; finally, the gating network is adopted to separate the jointed clusters to output the final results. Simulation experiments are done based on both high dimensional datasets and low dimensional datasets, the results show that the proposed algorithm realizes good clustering accuracy to high dimensional datasets.

Key words: high dimensional data, neuron network tree, artificial bee optimization, clustering algorithm, feature selection

高维数据聚类算法存在“维数灾难”问题,导致数据聚类的准确率和时间效率均无法满足实际的应用要求[1]. 特征选择处理是解决“维数灾难”问题的一个有效方法,能够有效地加快聚类处理的速度[2]. 高维数据的特征选择过程中,容易误删除与目标类相关的特征,保留冗余的特征,将此类特征子集应用于聚类程序,不仅降低聚类的准确率,也增加聚类的处理时间[3]. 因此,特征选择的性能对聚类处理的性能具有巨大的影响.

特征选择方法可以选出信息量丰富的特征子集,用于后续的聚类处理,而大多数特征选择技术通过训练仅输出唯一的特征子集,如果测试集存在噪声或者不完整则会导致聚类的性能下降[4]. 采用聚类技术能够发现信息量最大的多个特征类簇[5],有助于兼容不同的测试集,聚类技术和特征选择结合的技术称为混合特征聚类算法,这些技术能够最大化聚类准确率,同时保持较低的特征冗余度[6]. 通常混合分类器处理高维数据集的能力强于单一的特征选择技术[7],目前主流的混合聚类算法主要分为两种策略:基于软件、硬件的并行计算策略和基于快速学习算法的策略. 第一种策略采用 GPU、云计算等并行计算架构. 第二种策略则包括动态森林技术(dynamic clustering forest)[8]、特征分组技术[9]、粒子群混合聚类算法

收稿日期:2020-07-08.
基金项目:东莞市科技局项目(2020507156694)、东莞职业技术学院横向课题(202021189).
通讯作者:吉珊珊,讲师,研究方向:计算机智能信息处理与控制、计算机仿真、计算机教育. E-mail:jss060@163.com

等. 综合不同的实验结果,并行计算架构对于加速聚类速度具有显著的效果,但对于聚类的准确率并不具备改进效果,甚至以牺牲聚类准确率为代价以加快数据处理的速度[10]. 基于学习算法的混合聚类算法不仅提高了高维数据的聚类准确率,并且通过降维处理加快了聚类的速度.

综上所述,本文设计了基于学习方法和预测方法的高维数据聚类算法. 通过二元人工蜂群优化算法选择每个簇的最优特征子集,将每个特征子集作为节点构建径向基函数网络神经树,通过该机制能够有效地提高聚类算法的聚类准确率,并且加快算法的处理速度. 以神经网络构建决策树,通过不同的神经树预测各个类簇,神经树不仅具备决策树生成规则的优势,而且具备径向基函数网络的泛化能力. 通过门网络机制聚集森林所有神经树的最优结果,最终决定最优的类标签.

1　人工蜂群优化算法

1.1　基础人工蜂群算法

目前存在多个基于种群的优化算法,如差分进化算法、粒子群算法和进化算法等,在高维数值问题的效果上人工蜂群算法好于其他的同类型算法,同时能够高效地用于解决多维工程问题.

人工蜂群(artificial bee colony,ABC)共有雇佣蜂、观察蜂和侦察蜂 3 种成员,雇佣蜂负责搜索和定位食物源,雇佣蜂和观察蜂分享食物源的位置信息,观察蜂对食物源的邻域进行开发,寻找更加优质的食物源. 如果在 T 次迭代之后无法提高食物源的质量,此时启动侦察蜂阶段,在搜索空间中随机选择一个新的食物源. 随机选择食物源的方法定义为:

$$x_{i,d}=x_{d\min}+\mathrm{rand}(0,1)(x_{d\max}-x_{d\min}), \tag{1}$$

式中,x_i 为食物源 i 的位置,rand(0,1)函数表示输出区间[0,1]服从均匀分布的一个随机数,$x_{d\max}$ 和 $x_{d\min}$ 分别为搜索空间维度 d 的上界和下界.

蜂群根据下式产生候选食物源的位置:

$$v_{i,d}=x_{i,d}+\mathrm{rand}(-1,1)(x_{i,d}-x_{j,d}), \tag{2}$$

式中,v 为产生的候选食物源,i 和 j 为两个食物源,其范围为$\{1,2,\cdots,SN\}$,SN 为食物源的数量,d 的范围为$\{1,2,\cdots,D\}$,D 为最大的维度,x 为当前食物源的位置,rand(-1,1)为生成区间[-1,1]随机数的函数.

如果式(2)产生的候选食物源优于当前的食物源,那么雇佣蜂和观察蜂按食物源质量更新位置. 食物源的质量评价方法为:

$$fit_i=\begin{cases}1/(1+f_i), & \text{if } f_i\geqslant 0,\\ 1+\mathrm{abs}(f_i), & \text{其他情况},\end{cases} \tag{3}$$

式中,f_i 为目标函数的结果,x_i 为食物源,abs 为取绝对值的运算符.

根据式(4)计算一个概率值,观察蜂根据该概率值选择食物源.

$$p_i=\frac{fit_i}{\sum_{j=1}^{SN}fit_j}, \tag{4}$$

式中,fit 为食物源 x 的适应度,SN 为食物源的数量. 式(4)为高频率、高质量的食物源分配了更高的被选择可能性.

ABC 算法全部流程可参考文献[11].

1.2　改进的二元人工蜂群算法

为了使 ABC 算法适用于本文的特征选择问题,采用连续-二元映射机制将 ABC 的解转化为二元形式. 采用式(5)将连续解 x_i 映射至二元空间(z_i):

$$z_{i,d}=\mathrm{mod}\ 2(\mathrm{round}(\mathrm{abs}(\mathrm{mod}\ 2(x_{i,d})))), \tag{5}$$

式中,z_i 为一个二元候选解,d 为维度,其取值范围为$\{0,1,\cdots,D\}$,mod 2(·)函数将输入值除以 2,abs(·)为取绝对值的函数,round(·)为取整数的函数. 式(5)将解约束到区间[-a,a]内,a 是一个正实数.

在每个维度生成一个[0,1]区间的随机数,如果随机数大于或等于 0.5,将该维度的值改为 1,否则,改为 0. 二元的食物源位置定义为下式:

$$v_{i,d}=x_{i,d}\otimes[\varphi(x_{i,d}\otimes x_{j,d})], \tag{6}$$

式中,“⊗”表示 XOR 运算,φ 为一个逻辑“非”运算.

另一个二元 ABC 版本采用比特运算代替连续 ABC 算法的实数运算,二元人工蜂群(Binary Artificial Bee Colony,BABC)的食物源位置更新方法为:

$$v_{i,d}=x_{i,d}\otimes[\phi\odot(x_{i,d}\otimes x_{j,d})], \tag{7}$$

式中,ϕ 为以相等概率生成 0 或 1 的函数,⊗为 XOR 运算,⊙为逻辑“与”运算,⊕为逻辑“或”运算.

本文对连续 ABC 做修改,实现 BABC 算法. 首先,修改式(1)的初始化程序,以相等的概率产生 0 和 1 值:

$$x_{i,d}=\begin{cases}1, & \text{如果 rand}(\ )\geqslant 0.5,\\ 0, & \text{其他},\end{cases} \tag{8}$$

式中,$x_{i,d}$为食物源 i 在维度 d 的位置,rand(·)是生成[0,1]区间随机数的函数.

对雇佣蜂阶段和观察蜂阶段(式(2))做修改,采用算法 1 选择、更新食物源. 算法中 $v_{i,d}$是式(7)生成的位置,D 为维度的总数量,ceil( )返回给定数的最小整数. max_flip 是 0~1 之间的值,表示支持的最高维度,max_flip 控制种群的收敛速度,ceil( )保证至少选择 1 个维度.

算法 1　雇佣蜂和观察蜂阶段的子程序

输入:该阶段选择的食物源 j,食物源 i 在维度 d 的位置 $x_{i,d}$　//雇佣蜂阶段或者观察蜂阶段
输出:更新的位置 $v_{i,d}$
1. dim=ceil(max_flip×D);//ceil( )函数返回大于或者等于指定表达式的最小整数
2. 从食物源 j 选择 dim 个随机维度;
3. $d=0$;
4. while $d<$dim {
5.　if $x_{i,d}\neq x_{j,d}$ {
6.　$v_{i,d}=x_{j,d}$;
7.　} else {
8.　$v_{i,d}=x_{i,d}$;
9.　}10.　d++;
11. }

本文对 ABC 的修改能够提高雇佣蜂阶段和观察蜂阶段的多样性,原因是这两个阶段的蜂群随机选择食物源和维度,该机制有助于提高种群的总体多样性. 在雇佣蜂阶段和观察蜂阶段,维度可能不等于 dim,如果从食物源选择一个随机维度,该维度可能与当前蜜蜂的位置不同,但随着种群收敛,位置的差异会逐渐降低.

2　基于神经树的特征混合聚类算法

本文提出的高维数据混合聚类算法的流程框图如图 1 所示. 首先,对输入数据做基于聚类的特征选择处理,初步过滤部分冗余度高的特征;然后,基于分类的特征子集和样本集建立神经树,为神经树的叶节点分配类标签;最终,采用门网络分割类簇,区分类簇之间的交集.

在特征选择和特征聚类两类方法中,如果一些目标类相关的特征与其他类的冗余度也较高,那么这两种方法均会删除此类特征. 混合特征聚类算法能够有效地解决该问题,其基本思想是根据分类准确率将特征集分类,选出性能最好的特征子集. 本算法引入 ABC 算法筛选出与目标类最相关的特征子集.

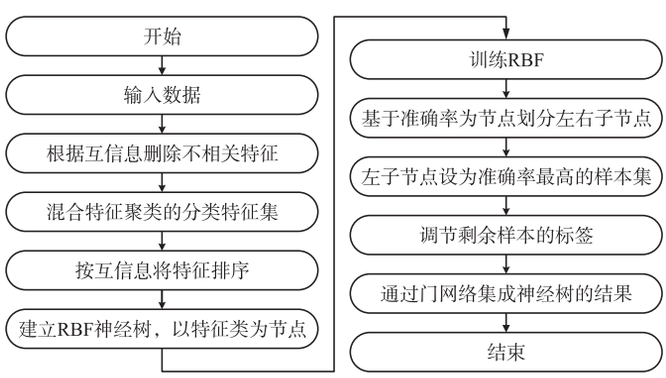

图 1　高维数据的混合聚类算法流程框图

Fig. 1　Flow chart of hybrid clustering algorithm for high dimensional data

2.1　基于互信息初步筛选特征集

采用互信息(Mutual Information,MI)评估不相关特征和冗余特征,一种互信息评估方法为 $MI(x,y)$,比较特征 $x\in X$ 和目标类 y 之间的相关性,如果 $MI(x,y)$ 较低,那么特征 x 是类 y 的不相关特征. 另一种互信息的评估方法为 $MI(x',x'')$,计算每对特征 $\{x',x''\}X$ 的 MI,寻找冗余的特征. 离散数据的 MI 计算为下式:

$$MI(x,y)=\sum_{i=1}^{n}\sum_{j=1}^{n}p(x(i),y(j))\times\log\left(\frac{p(x(i),y(j))}{p(x(i))\times p(y(j))}\right),\tag{9}$$

连续数据的 MI 计算为下式:

$$MI(x,y)=\int_{x(i)}^{X}\int_{x(j)}^{Y}P(x(i),y(j))\log_2\frac{P(x(i),y(j))}{P(x(i))P(y(j))}\mathrm{d}x\mathrm{d}y,\tag{10}$$

式中,$P(x(i))$ 为特征 x 的出现概率,$P(x(i),y(j))$ 为 $(x(i),y(j))$ 同时出现的联合概率. 因为连续 MI 的计算复杂度高,所以采用式(9)的离散 MI.

设计了基于 MI 的过滤式特征选择程序,其步骤为:

Step1.计算所有特征的互信息 MI;

Step2.计算全部 MI 的平均值和标准偏差;

Step3.如果某个特征的 MI 值小于 diff(diff=平均值−标准偏差),那么从特征空间删除该特征.

2.2　基于 BABC 的特征处理

聚集初步筛选后剩余的特征集,为每个类簇构建一个神经树. 每个神经树的时间复杂度为 $O(hNM)$,M 为特征数量,N 为样本数量,h 为神经树的深度. 为了减少每个神经树的计算时间,定义参数 δ 使复杂度满足 $O(hNM)<\delta$,根据这个不等式能够确定每个类的特征数量. 然后通过 ABC 搜索每个类的最优特征子集,ABC 的目标函数是最大化径向基函数(Radical Basis Function,RBF)网络的精度、最小化特征的冗余度. 特征聚类算法的程序如算法 2 所示.

算法 2　特征聚类算法

输入:特征集 $F_s=\{F_m,m=1,2,\cdots,M\}$,$c$ 个类,学习样本 $S=\{(x,y)\mid x=(x_1,x_2,\cdots,x_M),y\in\{1,\cdots,C\}\}$,样本数量 N,特征数量 M.

输出:特征分类结果 $F_c=[F^{(1)},\cdots,F^{(K)}]$,每个类 $F^{(i)}$ 的特征数量为 FN.

1.$m=0$;

2.while $m<M$ do

3.式(1)计算 $MI(x_m,y)$;

4.计算互信息 $MI(x_m,y)$ 的均值 avg 和标准偏差 θ;　/* 初步筛选特征集 */

5.$F_{\text{filter}}=\{F_m',m'=1,\cdots,M'\}$,其中 $MI(x_m',y)\geqslant(avg-\theta)$;　/* 确定特征类的数量 */

6. $CC=\text{round}(M'/FN)$;//round( )返回四舍五入的结果

7.随机生成 CC 个蜜源,搜索空间的维度为 FN,FN 也是神经树支持的最多特征.

8.应用 BABC 选出最优的特征分类;

9.将每个 Fc 中的特征按互信息降序排列;

10.m++;

11.return $Fc=[F^{(1)},\cdots,F^{(K)}]$,其中 $K=CN$,$F^{(i)}=\{f_t,\cdots,f_d\}$,其中 $F^{(i)}$ 共有 FN 个特征,$MI(f_t,y)\geqslant\cdots\geqslant MI(f_d,y)$.

2.3　结合 BABC 和特征选择的实现方式

BABC 和特征选择的结合方式有如下两点:

(1)解表示方法:BABC 的蜜源为二元值(0 或 1),如果某个蜜源为 1,表示选择该蜜源对应的特征,每个空间维度为一个特征分类,值为 1 的蜜源数量约束为满足 $O(hNM)<\delta$ 的最多特征数量.

(2)适应度函数:采用式(3)的适应度函数最大化径向基函数网络的准确率、最小化特征的冗余度,以封装式方法最大化径向基函数网络的准确率,以过滤式方法最小化特征的冗余度:

$$\text{fit}(k)=\frac{\text{acc}(k)}{\text{red}(k)},\tag{11}$$

式中,$\text{acc}(k)$ 为训练径向基函数网络的准确率,$\text{red}(k)$ 为特征的内部冗余度. $\text{red}(k)$ 的计算方法为:

$$\mathrm{red}(k)=\frac{1}{|c_k|^2}\sum_{x',x''\in c_k} MI(x',x''), \tag{12}$$

式中,$|c_k|$为类簇 k 的特征数量. 直接将式(9)的(x,y)替换为(x',x''),计算类簇 k 中所有特征的 $MI(x',x'')$.

3 构建神经树

3.1 径向基函数网络的决策树

每个神经树按特征分类将数据样本分类,其处理步骤为:

Step 1 在每个神经树的根节点中,将该类簇的特征子集按 MI 值降序排列,在神经树的每个深度将 MI 值最高的新特征作为分支.

Step 2 神经树的每个节点为一个径向基函数网络,径向基函数网络训练当前特征子集的样本. 径向基函数网络是一种单隐层前馈神经网络,使用径向基函数作为隐层神经元激活函数,输出层则是对隐层神经元输出的线性组合. 采用 k-means 聚类方法选择隐层节点的中心,网络输入层和隐层的节点数量均等于特征的数量,输出层的节点数量等于类标签的数量.

Step 3 对于父节点中的每个 RBF 网络 n_l,评估每个类的计算误差. 采用反向传播方式和上文选出的特征对 RBF 进行训练,选出误差最小的一个类,将该类的所有样本放入左子节点,剩余样本放入右子节点. 图 2 所示是构建神经树的流程图,图 3 所示是构建神经树的一个实例. 因为左子节点的样本同质,所以左子节点为叶节点,而右子节点继续分支处理.

Step 4 右子节点采用父节点的先验知识,从而加快 RBF 网络的训练速度. 首先,将父节点的特征集输入右子节点,然后,加入一个新特征. 父节点径向基函数网络的最终权重 $w(n_l)$也输入右子节点,设计了算法 3 确定右子节点径向基函数网络的最优权重,算法 3 中 $w(n_{l+1})$是一个均匀随机数. 如果重新计算右子神经树的权重,需要计算矩阵(M 行×N 列)的伪逆矩阵,其复杂度为 $O(MN^3)$. 因此,本文利用父节点的先验知识加速右子节点的分支速度.

Step 5.如果类簇的所有特征输入分类器或者右子节点达到同质,那么跳至步骤 6;否则更新神经树的结构,增加深度 $l=l+1$,返回步骤 3.

Step 6.该步骤将比例最高的类标签分配至右子节点的所有样本.

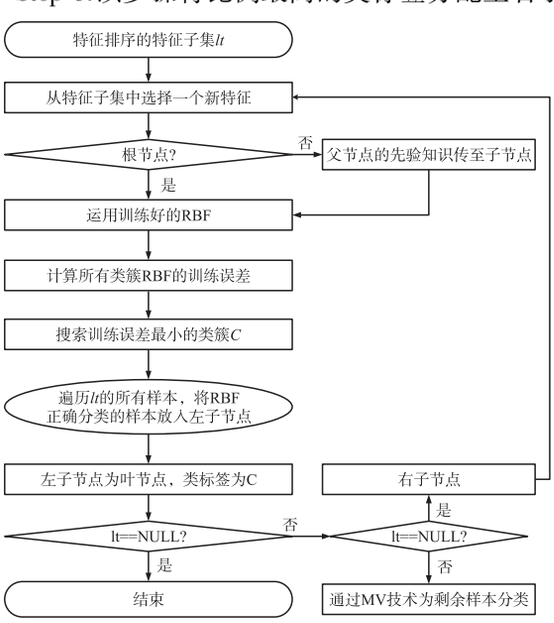

图 2 构建神经树的流程图

Fig. 2 Flow chart for building a neural tree

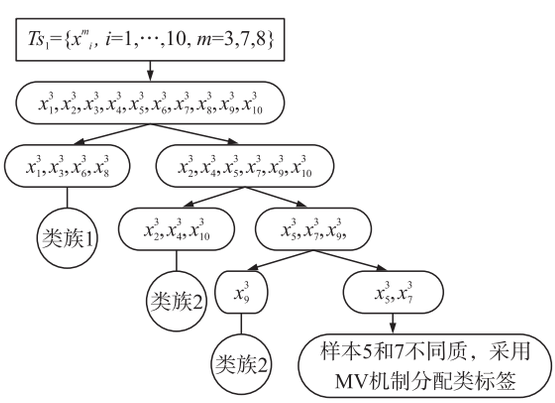

图 3 构建神经树的一个实例

Fig. 3 Building an instance of a neural tree

算法 3　右子节点的权重更新算法

输入:最大迭代次数 $\max_{\text{iter}}$,输出:最优权重 $w(n_{l+1})$

1. $i=0$;
1. while($w(n_{l+1})$收敛 && $i<\max_{\text{iter}}$){
2. $w(n_{l+1})=w(n_{l+1})-\alpha\nabla E(n_{l+1})$;
3. 更新步长 α 和梯度值 $E(n_{l+1})$;
4. $i=i+1$;
5. }

3.2　基于门网络的类簇分割

特征聚类阶段并未保证特征类簇间的分离性,所以不同类簇之间可能存在交集. 将新样本的特征集按特征分类做分解,样本加入对应的神经树,神经树可能为样本分配不同的类标签,采用门网络集成所有的结果. 门网络的设计目标主要有两点:(1)结合所有神经树的结果,最终为每个样本产生一个唯一的类标签;(2)如果一个样本存在多个类标签,那么采用多数投票(Majority Voting,MV)技术根据可靠性选出唯一的类标签.

具体过程为:检查是否所有叶节点分配了唯一的类标签,如果存在叶节点分配了多个类标签的情况,那么采用 MV 技术根据可靠性选出唯一的类标签. 最终所有的叶节点均分配了唯一的类标签.

3.3　算法的时间复杂度分析

神经树的计算复杂度依赖径向基函数网络节点的数量和每个径向基函数网络的复杂度.

径向基函数网络负责搜索最优的权重向量 $\boldsymbol{W}'$,并为每个样本分配正确的类标签. 假设样本的数量为 N,特征的数量为 M,隐层神经元的数量为 n_{hidd},类的数量为 C. 一个全连接 RBF 网络输入层-隐层的时间复杂度为 $O(Mn_{\text{hidd}})$,隐层-输出层的时间复杂度为 $O(n_{\text{hidd}}C)$,所有训练样本的时间复杂度为 $O(N(Mn_{\text{hidd}}+n_{\text{hidd}}C))$.

本算法采用一个迭代方法确定 RBF 网络最终的权重向量,该迭代程序的复杂度等于 $O(Cn_{\text{hidd}}^3)$,小于伪逆矩阵的复杂度. 最终,RBF 网络的总体时间复杂度为 $O(Nn_{\text{hidd}}(M+C)+(Cn_{\text{hidd}}^3))$. 神经树的时间复杂度为:

$$O(|node_{\text{RBF}}|\times(Nn_{\text{hidd}}(M+C))+(Cn_{\text{hidd}}^3)), \tag{13}$$

式中,$|node_{\text{RBF}}|$为神经树中径向基函数网络节点的数量.

4　结论与讨论

通常集成分类器处理高维数据集的能力较强,本算法采用 BABC 优化特征的分类,其次采用 RBF 神经树增强算法的分类性能. 在提高聚类性能的工作中,本算法主要设计了 3 个机制:(1)在 RBF 网络的最后一层,RBF 网络仅识别一个类,这种一对多分类器(one-versus-rest,1-v-r)具有较高的分类准确率. (2)在最后一层中,神经树多个 RBF 网络积累的知识能够纠正一些错误分类的样本. (3)神经树最终的分类结果是统计了若干个 RBF 网络的结果所总结的结果,其准确率应当较高.

首先,通过 1.2 小节的算法最大化径向基函数网络的准确率. 然后,通过第 2 小节的方法选出每个特征子集的样本集,利用选择的样本集合第 3 小节的方法训练以径向基函数网络为节点的神经树. 最终,采用门网络将连接的类簇分离,获得最终的聚类结果.

4.1　实验数据集和实验方法

将本算法与其他 5 个混合特征聚类算法比较,分别为:AdaBoost ANN[12]、MOGPEF[13]、EFS-MI[14]、EFSHDD[15]、MCDCV[16]. AdaBoost ANN 是一种基于人工神经网络的特征聚类算法,与本文的神经树策略相似;MOGPEF 是一种基于多目标遗传算法的特征聚类算法与本文的 BABC 策略相似;EFS-MI、EFSHDD、MCDCV 则是近期对于高维数据聚类准确率和计算效率均较好的 3 个混合聚类算法.

采用低维数据集和高维数据集评估本算法的综合性能,表 1 所示是低维数据集的一般属性,其特征数量的范围为 4~60,低维数据集主要来自于 UCI 数据集. 表 2 所示是高维数据集的一般属性,其特征数量范围为 10 000 以上,其中 Arcene、Twin gas 和 URL 也来自于 UCI 数据集,Gas 2 来自于文献[17].

表 1　低维数据集的基本属性

Table 1　Basic property of low dimensional datasets

数据集名称	样本数量	分类数量	特征数量
IRIS	150	3	4
Glass	214	6	9
Breast	799	2	9
Wine	178	3	13
Heart	270	2	13
WDBC	569	2	30

表 2　高维数据集的基本属性

Table 2　Basic property of high dimensional datasets

数据集名称	样本数量	分类数量	特征数量
Arcene	900	2	10 000
Gas 2	180	10	150 000
Twin gas	640	10	480 000
URL	2 396 130	2	3 231 961

4.2　处理低维数据集的性能

首先评估本算法对于一般低维数据集的处理效果,为了保证实验结果的置信度,所有算法的分类准确率结果采用十折交叉验证机制统计算法的准确性. 图 4 是 5 个集成分类算法对低维数据集的分类准确率结果,图中结果显示本算法对于 6 个不同的数据集均实现了较高的分类准确率,Adaboost ANN 算法和 MCDCV 算法对于 Glass 数据集的分类准确率低于 50%,而 MOGPEF 算法、EFS-MI 算法和 EFSHDD 算法对于 Glass 数据集的分类准确率也远低于本算法,综合图中可看出,其他 5 个集成分类算法对于不同数据集的稳定性较差,而本算法对于 6 个数据集的平均分类准确率均高于 90%.

分类算法的处理时间是一种重要的性能,比较了本算法与其他算法的时间效率. 图 5 是 6 个集成分类算法对低维数据集的平均处理时间结果(10 次独立时间的平均值). 图中结果显示本算法对于 IRIS 数据集的时间高于其他两个算法,对于 Glass 数据集和 Breast 数据集的时间则高于 EFS-MI 算法,对于 Wine 数据集、Heart 数据集和 WDBC 数据集的时间则低于其他两个算法. 总体而言,EFSHDD 算法的处理时间较长,该算法需要针对每个特征计算其预测精度,该过程消耗大量的时间. 本算法通过高效的初步筛选机制过滤了大量的冗余特征,在构架神经树的过程,神经树的每个深度对应一个类簇,因此预测的效率较高.

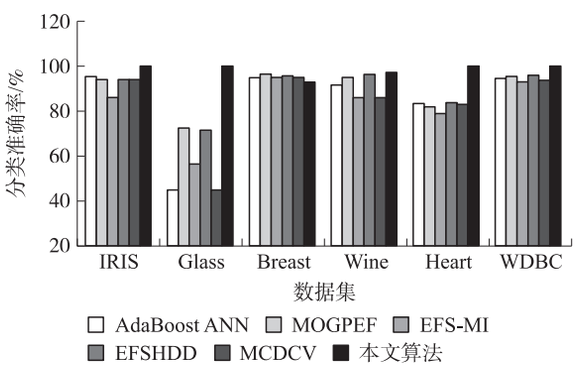

图 4　低维数据集的分类准确率结果

Fig. 4　Classification accuracy results for low-dimensional data sets

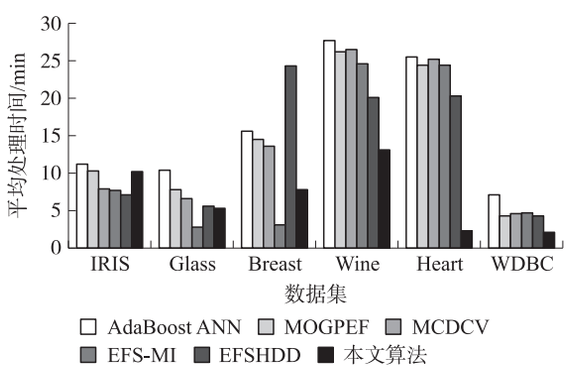

图 5　低维数据集的平均处理时间

Fig. 5　Average processing time for low-dimensional data sets

4.3　处理高维数据集的性能

本部分将重点评估本算法对于高维数据集的处理效果. 为了保证实验结果的置信度,所有算法的分类准确率结果采用十折交叉验证机制统计算法的准确性. 图 6 是 5 个集成分类算法对高维数据集的分类准确率结果,图中结果显示本算法对于 4 个不同的数据集均实现了较高的分类准确率,4 个数据集的特征数量均高于 10 000,Arcene 的特征数最低,6 个分类算法均实现了可接受的准确率结果. 比较 Arcene、Gas 2、Twin gas 3 个数据集的结果,其特征量从 10 000 大幅度增至 480 000,本算法的分类准确率从 97.26%降至 94.69%,依然保持较高的分类准确率,URL 数据集的特征量高达 3 231 961,而本算法对于 URL 的准确率降低至约 80%,依然明显高于其他 5 个分类算法. 观察高维数据的实验结果,本算法对于特征数量大于样本数量的数据集性能更好,原因是神经树森林中多个树可能包含重复的样本和非冗余的特征,而通过训练神经树中不同特征类簇的小样本集合,综合不同角度的神经树能够有效地提高最终的分类准确率.

高维数据分类处理的时间复杂度较高,所以时间是高维数据分类处理的关键性能,比较了本算法与其他算法的时间效率. 图 7 是 6 个混合分类算法对高维数据集的平均处理时间结果(10 次独立时间的平均

值). 图中显示 6 个算法对于 Arcene 数据集的处理时间较为接近,而随着特征量的大幅度增加,EFS-MI 算法和 EFSHDD 算法均表现出明显地增长,而本算法时间的增长幅度较小.

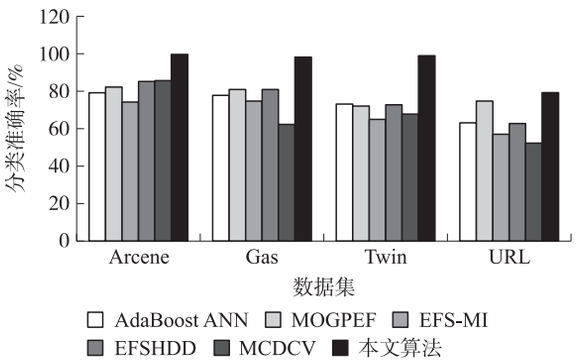

图 6　高维数据集的分类准确率结果

Fig. 6　Classification accuracy results for high-dimensional data sets

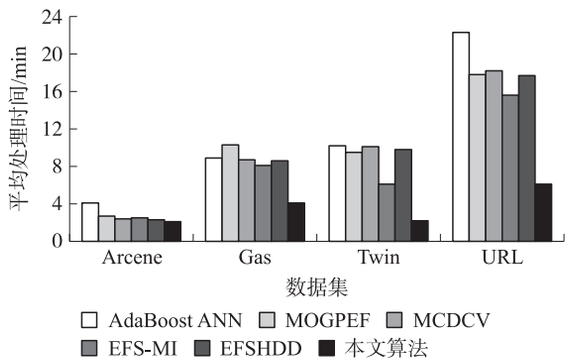

图 7　高维数据集的平均处理时间

Fig. 7　Average processing time for high dimensional data sets

4.4　噪声鲁棒性的性能

测试聚类算法对于噪声的鲁棒性,为数据集增加 15% 的高斯噪声处理,本算法对每个数据集独立地运行 20 次,统计 20 次的平均结果作为最终的实验结果,结果如图 8 所示. 结果显示,本算法对于高斯噪声的分类性能略低低于无噪声的数据,但是性能的衰减极小,属于可接受的范围.

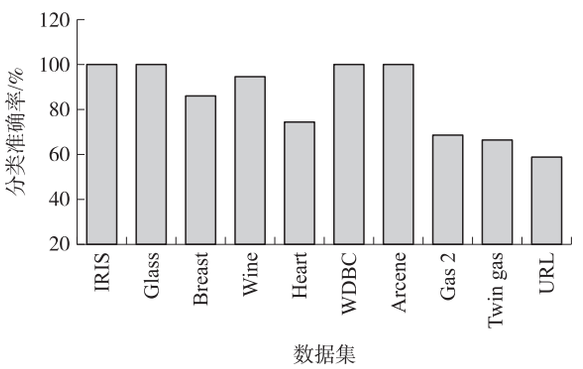

图 8　噪声数据集的分类准确率结果

Fig. 8　Classification accuracy results for noise data sets

5　结论

本文设计了混合特征聚类算法,算法的混合特征聚类算法支持多特征子集的聚类处理,能够有效地增强对高维大数据的性能,对于噪声也具有较好的鲁棒性. 本算法通过高效的初步筛选机制过滤了大量的冗余特征,在构架神经树的过程中,神经树的每个深度对应一个类簇,因此预测的效率较高. 随着数据特征量的大幅度增加,算法时间的增长幅度较小,能够高效地处理高维数据. 未来将关注于将本算法应用于 GPU 或者 MapReduce 等并行计算架构,提高本算法的处理效率.

[参考文献]

[1] 刘娜,毛晓菊,吴敏. 集群分类映射的文本多标签模糊关联降维聚类[J]. 计算机工程与设计,2017,38(6):1657-1663.

[2] 王翔,胡学钢. 高维小样本分类问题中特征选择研究综述[J]. 计算机应用,2017,37(9):2433-2438.

[3] GARCíA T M,GÓMEZ V F,MELIÁN B B,et al. High-dimensional feature selection via feature grouping:a variable neighborhood search approach[J]. Information sciences,2016,326(C):102-118.

[4] BOLÓN G V,SÁNCHEZ M N,ALONSO B A. Feature selection for high-dimensional data[J]. Computational management science,2016,5(2):65-75.

[5] 金利英,赵升吨. 混合测量子空间聚类算法的研究[J]. 西安交通大学学报,2018(3):139-144.

[6] CHEN C,DONG D,QI B,et al. Quantum ensemble classification:a sampling-based learning control approach[J]. IEEE Transactions on neural networks & learning systems,2017,28(6):1345-1359.

[7] LI Y,GUO H,XIAO L,et al. Adapted ensemble classification algorithm based on multiple classifier system and feature selection for classifying multi-class imbalanced data[J]. Knowledge-based systems,2016,94:88-104.

[8] SONG G,YE Y,ZHANG H,et al. Dynamic clustering forest:an ensemble framework to efficiently classify textual data stream

with concept drift[J]. Information sciences,2016,357:125-143.

[9] FARID D M,NOWE A,MANDERICK B. A feature grouping method for ensemble clustering of high-dimensional genomic big data[C]//2016 Future Technologies Conference(FTC). San Francisco,USA,IEEE,2016:260-268.

[10] DAGDIA Z C,ZARGES C,GAËL B,et al. A distributed rough set theory based algorithm for an efficient big data pre-processing under the spark framework[C]//IEEE International Conference on Big Data. Seattle,USA,IEEE,2018:911-916.

[11] LIN K C,ZHANG K Y,HUANG Y H,et al. Feature selection based on an improved cat swarm optimization algorithm for big data classification[J]. Journal of supercomputing,2016,72(8):3210-3221.

[12] BAIG M M,AWAIS M M,EL-ALFY E S M. AdaBoost-based artificial neural network learning[J]. Neurocomputing,2017,248(26):120-126.

[13] NAG K,PAL N R. A multiobjective genetic programming-based ensemble for simultaneous feature selection and classification[J]. IEEE transactions on cybernetics,2017,46(2):499-510.

[14] HOQUE N,SINGH M,BHATTACHARYYA D K. EFS-MI:an ensemble feature selection method for classification[J]. Complex & intelligent systems,2018,4(2):105-118.

[15] BRAHIM A B,LIMAM M. Ensemble feature selection for high dimensional data:a new method and a comparative study[J]. Advances in data analysis & classification,2018,12(4):937-952.

[16] GüNEY H,ÖZTOPRAK H. Microarray-based cancer diagnosis:repeated cross-validation-based ensemble feature selection[J]. Electronics letters,2018,54(5):272-274.

[17] FONOLLOSA J,RODRíGUEZLUJÁN I,TRINCAVELLI M,et al. Data set from chemical sensor array exposed to turbulent gas mixtures[J]. Data in brief,2015,3(C):216-220.

[责任编辑:顾晓天]