基于粗糙集与密度峰值聚类的特征选择算法

曹栋涛, 舒文豪, 钱进

计算机科学 COMPUTER SCIENCE
http://www.jsjkx.com
DOI:10.11896/jsjkx.230600038

基于粗糙集与密度峰值聚类的特征选择算法

曹栋涛[1]　舒文豪[1]　钱　进[2]
1 华东交通大学信息工程学院　南昌 330013
2 华东交通大学软件学院　南昌 330013
(1767831966@qq.com)

摘　要　特征选择可以有效地去除高维数据中的冗余和不相关的特征,保留重要的特征,从而降低模型计算的复杂性,提高模型精度。在特征选择过程中,针对数据中存在的离群点和边界点等可能影响分类效果的噪声数据,提出了基于粗糙集与密度峰值聚类的特征选择方法。首先,通过密度峰值聚类方法去除噪声数据,并挑出簇类中心;然后,结合粗糙集理论的思想,按簇类中心划分数据,并根据同一簇类的点应具有相同标签的假设,定义特征重要性评价指标;最后,设计了一种启发式特征选择算法,用于挑选出使簇类结构纯度更高的特征子集。在6个UCI数据集上,与其他算法进行了分类精度、特征选择个数和运行时间的对比实验,实验结果验证了所提算法的有效性和高效性。

关键词:特征选择;高维数据;噪声数据;粗糙集;密度峰值聚类
中图法分类号　TP391

Feature Selection Algorithm Based on Rough Set and Density Peak Clustering

CAO Dongtao[1],SHU Wenhao[1] and QIAN Jin[2]
1 School of Information Engineering,East China Jiaotong University,Nanchang 330013,China
2 School of software,East China Jiaotong University,Nanchang 330013,China

Abstract　Feature selection can effectively remove redundant and irrelevant features from high-dimensional data and retain important features,thus reducing the complexity of model computation and improving model accuracy. While in feature selection process,to deal with these noisy data that may affect the classification effect,such as outlier points and boundary points,a feature selection method based on rough set and density peak clustering is proposed. At first,noisy data are removed by density peak clustering method and cluster class centers are picked out. Then,the data are divided by cluster class centers by combining the idea of rough set theory,and the feature importance evaluation measure is defined according to the assumption that the data points of same cluster have same label. Finally,a heuristic feature selection algorithm is designed to pick up the feature subset that can makes for a purer homogeneous cluster structure. Experimental comparisons of classification accuracy,number of selected features and running time are conducted with other algorithms on six UCI datasets,and the experimental results verify the effectiveness and efficiency of the proposed algorithm.

Keywords　Feature selection,High-dimensional data,Noisy data,Rough sets,Density peak clustering

1　引言

随着人工智能、云计算、物联网等信息技术的高速发展,现实应用中的大量数据都呈现出高维的特点,直接使用高维的数据可能造成“维度灾难”[1],这不仅影响现实应用任务的运行效率,而且会降低其学习性能。特征选择[2-4]是一种有效的数据预处理方法,通过搜寻相对于学习任务最优的特征子集,剔除不相关和冗余的特征,可以有效降低数据维度,提高任务分类精度和运算效率,现已被广泛应用在数据挖掘、知识发现等领域。

然而,在高维数据的特征选择任务过程中,数据的分布对学习过程有一定的影响。离群点的存在可能会导致学习器性能变差。同时,对于数据边界点,极容易出现被误分类的情况,容易影响最终的分类精度。例如,市场营销需要根据重要特征将消费者分成不同的群体,以便制定不同的营销策略和推广方案。在对市场的细分中,边界点通常被视为难以划分

到稿日期:2023-06-04　返修日期:2023-07-28
基金项目:国家自然科学基金(62266018,61966016);江西省自然科学基金(20202BABL202037,20232ACB202013,20232BAB202052);江西省研究生创新基金项目(YC2022-s547)
This work was supported by the National Natural Science Foundation of China(62266018,61966016),Jiangxi Province Natural Science Foundation(20202BABL202037,20232ACB202013,20232BAB202052) and Jiangxi Postgraduate Innovation Fund Project(YC2022-s547).
通信作者:舒文豪(shuwenhao@126.com)

到任何一个群体中的消费者，而那些不符合市场细分目标的对象被视为离群点，剔除边界点和离群点并进一步分析消费者特征，可以提高市场分析和营销策略的准确性和有效性。图 1 给出了一个直观的数据分布示意图。如图 1 所示，两个簇类中心的分界线周围分布了多个边界点，这些边界点极容易被错误分类。同时，图 1 中明显地偏离了各个簇类的数据对象可视为离群点，这些离群点同样会对学习任务产生干扰，直接影响学习模型的拟合精度。这些边界点和离群点通常有一个共同的特点：较为散乱地分布在离簇中心较远的位置。如何处理边界点和离群点，提高学习性能，对于各种特征选择方法来说依旧是一个挑战。

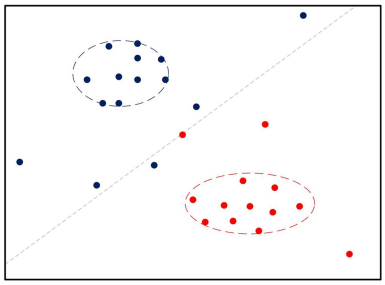


图 1 数据分布示意图

Fig. 1 Schematic diagram of data distribution

粗糙集理论[5]由波兰科学家 Pawlak 于 1982 年提出，是一种处理不精确、不确定与不完备数据的重要数学方法。该理论最大的优势是不需要提供数据本身以外的任何先验知识对数据中的不确定信息进行分析和推理，并从中挖掘潜在的有用知识，为数据挖掘、粒计算等领域提供了丰富的理论基础。基于粗糙集的特征选择方法是粗糙集理论研究的核心内容之一，其在保证特征对数据区分能力不变的前提下，从中挑选出相对于学习任务最优（少）的特征子集，以达到数据降维的目的，从而提高学习效率并改善分类效果。这种方法对处理高维数据且含有大量冗余特征的大数据研究具有重要的意义。基于粗糙集的特征选择方法对离散型数据有着良好的分类效果，但在处理连续型数据时，须将数据进行离散化，这必将导致部分原始信息的丢失。邻域粗糙集[6-7]在处理连续型数据方面有着明显的优势，同时也能处理多种类型的混合数据，这得到了众多学者的关注和研究。Yang 等[8]提出了基于距离度量学习的邻域粗糙集特征选择算法，该算法提高了邻域关系的判别能力；Mariello 等[9]通过减少特征中的不确定性信息，提出了基于邻域熵的特征选择算法；Wang 等[10]根据差别矩阵的特征评价方法，提出了基于邻域差别矩阵的特征选择算法；Hu 等[11]利用正域与属性集的单调关系，提出了基于邻域属性依赖度的特征选择算法；Sheng 等[12]在基于邻域区分度的特征选择算法上进一步提出了增量式的特征选择算法。然而，在根据邻域粗糙集理论对数据进行特征选择的过程中，需要为每个数据对象都构建邻域。同时，在评估特征重要性的过程中，需要评估特征对所有数据样本的邻域精度的影响，这样的评估策略使得特征子集在每次变化时都需要重构所有对象邻域进行评估，资源消耗过高。此外，离群点和边界点等可能影响学习性能的数据对象也参与了邻域的构建。边界点的错误分类会导致其所在邻域的精度降低，而离群点在邻域的构建中会直接影响模型对邻域参数的拟合，从而影响分类精度。

密度峰值聚类（Density Peaks Clustering，DPC）算法由 Rodriguez 等[13]于 2014 年提出，并发表在 *Science* 上，是一种基于密度峰值进行核聚类分析的数据挖掘技术。该算法考虑局部密度和相对距离，并绘制决策图，以进一步快速识别簇类中心和噪声点，最后根据簇类中心将输出分裂成多个相互独立的子空间，以此完成聚类。DPC 算法能够自动发现簇类中心，实现任意形状数据的高效聚类，其因无需事先指定聚类簇数、无需先验知识，也无需迭代确定簇类中心等特点引起了学者们的兴趣。Zhou 等[14]结合密度峰值聚类算法，提出了一种密度峰值聚类的彩色图像分割方法。Huang 等[15]在复杂网络中节点相似度度量以及密度峰值聚类算法的基础上，提出了一种基于点距离和密度峰值聚类的社区发现方法。Du 等[16]针对 DPC 算法不能有效地对任意形状或多流形结构的数据进行分组的缺点，提出了一种基于测地距离的密度峰聚类方法。Bian 等[17]为了在处理聚类中的模糊性和不确定性方面提供灵活的适应性，提出了模糊峰的新概念，并设计了一种新的基于模糊算子的模糊密度峰值聚类方法。Liu 等[18]针对 DPC 算法中截断距离难设定的问题，采用 k 近邻算法[19]重新定义局部密度，使得结合了 k 近邻算法的 DPC 算法具有更好的鲁棒性。为消除在特征选择过程中，模型容易受到离群点、边界点等容易影响分类精度的噪声点的影响，本文采用 DPC 算法对噪声数据进行剔除，以此提高数据的整体质量，从而改善聚类和分类任务的精度。同时，本文利用 DPC 算法能快速识别聚类中心并高效聚类的优势，结合粗糙集理论思想，提出了以少量核为中心对数据进行等价类划分并进行特征评估的搜索策略，避免了基于邻域粗糙集的特征选择算法对每个对象构建邻域并减少了评估过程中的资源消耗。

本文提出了基于粗糙集和密度峰值聚类的特征选择算法。首先，通过密度峰值聚类方法剔除影响分类性能的数据点并标记簇类中心，提高了数据的整体质量。然后，结合粗糙集理论方法，根据簇类中心划分数据，并定义特征重要性评估度量，以选取使簇类纯度较高的特征。最后，构建一种启发式的特征选择方法，通过评估以少量簇类中心所构建的簇类，弥补了基于邻域粗糙集的特征选择算法对所有邻域进行评估时资源消耗高的缺点，提高了运行效率。同时，实验结果验证了所提算法的有效性和高效性。

本文的主要贡献包括 3 个方面：

1）针对离群点和边界点等可能影响分类精度的噪声数据，采用密度峰值聚类算法将其剔除，从而提高数据的整体质量。

2）利用密度峰值聚类算法标记簇类中心，结合粗糙集理论思想，根据簇类中心对数据进行划分，并定义了特征重要性评估度量来选取出使簇类纯度更高的特征子集，从而提高分类性能。

3）构建了一种启发式的特征选择算法。实验结果表明，所提算法可以有效地对数据进行特征选择。同时，在保证分类精度的情况下，相比其他几种算法，其特征选择个数相对较少，运行时间相对更短。

2 基本理论

2.1 粗糙集

在粗糙集理论中，假设数据可以由一个四元组的决策系统来表示，即 $DS=\langle U,A=C\cup D,V,f\rangle$，其中：

1）$U=\{x_1,x_2,\cdots,x_n\}$为全体数据对象的集合；

2）A 是数据对象的全体特征的集合，其中 $C=\{a_1,a_2,\cdots,a_m\}$为条件特征集，$D=\{d\}$为决策特征集；

3）V 是所有数据对象的特征值的值域，即 $V=\bigcup_{a\in A}V_a$，其中V_a表示所有对象在特征 a 下的值域；

4）$f:U\times A\rightarrow V$ 是信息函数，表示为 $\forall a\in A,\forall x\in U$，$f(x,a)\in V_a$，即对象 x 在特征 a 下的值。

定义 1 给定一个决策系统 $DS=\langle U,A=C\cup D,V,f\rangle$，对于 $\forall B\subseteq C$，则 U 在 B 下的不可区分的关系表示为：

$$R_B=\{(x,y)\in U\times U\mid f(x,a)=f(y,a),\forall a\in B\}$$

且通过不可区分的关系R_B，可以诱导出 U 在 B 下的划分，即：

$$U/R_B=\{[x]_B\mid x\in U\}$$

其中$[x]_B$为等价类，表示为：

$$[x]_B=\{y\in U\mid (x,y)\in R_B\}$$

定义 2 给定一个决策系统 $DS=\langle U,A=C\cup D,V,f\rangle$ 和一个不可区分关系 R，对于 $\forall X\subseteq U$，则 X 关于 R 的下近似集和上近似集分别表示为：

$$\underline{RX}=\bigcup\{x\in U\mid [x]_R\subseteq X\}$$

$$\overline{RX}=\bigcup\{x\in U\mid [x]_R\cap X\neq\varnothing\}$$

其中，下近似集合也可称为正区域，即$POS_R(X)=\underline{RX}$。

2.2 邻域粗糙集

定义 3 给定一个决策系统 $DS=\langle U,A=C\cup D,V,f\rangle$，对于 $\forall B\subseteq C$，$\forall x$ 在 B 下的邻域集表示为：

$$\delta_B(x)=\{y\in U\mid DIS_B(x,y)\leqslant\varepsilon\}$$

其中，ε 为邻域半径，$DIS_B(x,y)$用于度量对象 x 和 y 之间相对于 B 的距离，可以采用欧氏距离来度量，即：

$$DIS_B(x,y)=\sqrt{\sum^{a\in B}\|f(x,a)-f(y,a)\|^2}$$

通过对 U 中的每个对象都构建邻域集，可以得到 U 在 B 下的划分，为$\{\delta_B(x_1),\delta_B(x_2),\cdots,\delta_B(x_n)\}$。

根据定义 3 可知，邻域粗糙集理论在每次特征子集变化后，需要对数据中的所有对象逐个重新构建邻域，这会导致极大的运算消耗。同时，由于邻域粗糙集并没有处理离群点和边界点等，边界点极容易被错误分类，离群点大多偏离簇类中心较远，影响邻域半径的拟合，这些可能影响精度的数据同样参与了构建邻域的过程，会降低最终的分类效果。

2.3 密度峰值聚类

DPC 算法基于两个基本假设：1）簇中心被簇中其他密度较低的数据点包围；2）不同簇中心之间的距离相对较远。基于这两个基本假设，可以定义出两个基本判别参数：局部密度和相对距离。图 2 是一个简单的数据样本分布示意图[13]，其中红、蓝色样本分别表示两个簇类样本，而黑色样本因分布明显脱离簇类，可以认为其可能会影响模型的分类性能，因此可以视其为噪声点。根据 DPC 算法计算各个数据点的局部密度 ρ 和相对距离 δ，可以绘制一个直观的决策图辅助分析。例如，对图 2 中各个数据点计算局部密度和相对距离后，绘制的决策图如图 3 所示[13]。由图 3 可知，簇类中心的局部密度和相对距离都较高，而噪声点的局部密度较低但相对距离较高。因此，可以通过 DPC 算法的决策图分析获取簇类中心，并剔除噪声点。最后，将剩余的数据点进行分配，完成聚类。

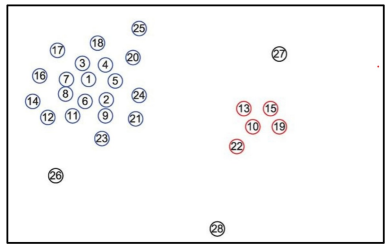


图 2 基于 DPC 算法的数据分布示意图（电子版为彩图）

Fig. 2 Schematic diagram of data distribution based on DPC algorithm

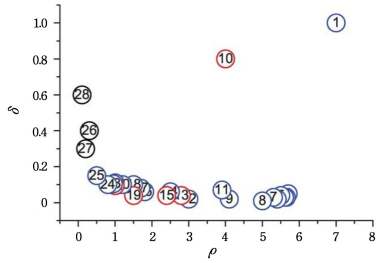


图 3 基于 DPC 算法的决策示意图（电子版为彩图）

Fig. 3 Schematic diagram of decision-making based on DPC algorithm

3 基于粗糙集与密度峰值聚类的特征选择方法

上文简单分析了邻域粗糙集理论处理混合数据的优势，对基于邻域粗糙集的特征选择过程中存在的运算消耗高、没有考虑处理影响分类性能的噪声数据等缺点进行了简单的阐述。本节提出了一种基于粗糙集与密度峰值聚类的特征选择算法，通过 DPC 算法剔除噪声数据，并结合粗糙集理论设计了一种启发式的特征选择算法，可以有效地进行特征选择，提高运算效率。

上文对 DPC 算法的整个过程进行了简单的描述，具体的 DPC 算法通常可以分成以下 5 个步骤。

1）给定一个数据集 U，对于 $\forall x_i\in U$，其局部密度ρ_i的计算式为：

$$\rho_i=\sum_{i\neq j}\chi(d_{ij}-d_c)$$

其中，d_{ij}表示任意两个对象x_i和x_j的距离。d_c是一个超参数，表示截断距离，且对于 $\chi(\cdot)$，表示为：

$$\chi(d)=\begin{cases}0, & d>0\\ 1, & d\leqslant 0\end{cases}$$

局部密度ρ_i相当于计算分布在数据对象x_i周围，且由其截断距离d_c划分所构成区域中的样本数量。因此，局部密度的计算会直接受到截断距离的影响。

2）对于 $\forall x_i\in U$，其相对距离δ_i的计算式为：

$$\delta_i=\min_{j:\rho_j>\rho_i}(d_{ij})$$

计算数据对象x_i的相对距离δ_i是指，从x_i周围的数据对象

中搜寻到比其密度更大但又与其相距最近的数据对象x_j后，再计算对象x_i和x_j的距离；但如果x_i是数据集U中的局部密度全局最大的数据对象，则其相对距离表示为：

$$\delta_i = \max_j(d_{ij})$$

3)获取所有数据对象的局部密度和相对距离后，以局部密度ρ为横轴，以相对距离δ为纵轴，绘制决策图。

4)利用决策图，将局部密度ρ和相对距离δ都相对较高的数据对象标记为簇类中心点；将局部密度ρ相对较低但相对距离δ相对较高的数据对象标记为噪声点。

5)将噪声点剔除，把剩余的数据对象根据距离就近分配给所标记的簇类中心点进行聚类。

DPC算法在聚类过程中可以快速确定簇类中心，无需先验知识，也无需事先指定簇类个数，更重要的是不需要不断迭代来确定簇类中心，大大减少了运行的时间消耗，但其也存在缺点。

1)在计算局部密度的过程中，截断距离的设定会直接影响最终的聚类效果，但最优的截断距离通常需要穷举搜索，较为耗时；

2)在计算相对距离的过程中，每次都需要全局搜索密度更高但相距最近的数据点，运算消耗高；

3)在绘制决策图后，需要根据自身的主观判断来获取簇类中心和噪声点，具有较大的主观不确定性。

针对以上问题，本文参考文献[18]，采用k近邻思想来重新定义局部密度和相对距离的计算，以去除截断距离的设定，避免每次都进行全局搜寻，提高运行效率。

定义4 给定一个数据集U，对于$\forall x_i \in U$，其局部密度ρ_i定义为：

$$\rho_i = \frac{1}{\sum_{x_j \in kNN(x_i)} d_{ij}}$$

其中，$kNN(x_i)$表示数据对象x_i的k个最近邻对象所构成的集合，d_{ij}表示对象x_i和x_j之间的距离，这里的距离度量指标采用欧氏距离。

定义5 给定一个数据集U，对于$\forall x_i \in U$，如果x_i的局部密度ρ_i比其k个最近邻对象的局部密度都更大，则其相对距离δ_i定义为：

$$\delta_i = \max_{x_j \in kNN(x_i)} d_{ij}$$

否则，其相对距离δ_i定义为：

$$\delta_i = \min_{x_j \in kNN(x_i):\rho_j > \rho_i} d_{ij}$$

在根据定义4和定义5计算局部密度和相对距离的过程中，将搜寻范围限定在k个最近邻对象中，无需全局搜索，这样可以大大缩短运行时间。同时，去除截断距离参数的设定，k值的设定令DPC算法具有更好的鲁棒性。

基于定义4和定义5计算局部密度和相对距离，并绘制决策图，以进一步分析判断簇类中心和噪声点。在剔除噪声点后，根据簇类中心，结合粗糙集理论思想重新定义不可区分关系并对数据进行划分，同时根据同簇同标记假设提出特征重要性度量，最后设计了基于粗糙集和密度峰值聚类的特征选择算法。

定义6 给定一个决策系统$DS=\langle U, A=C\cup D, V, f\rangle$，$U$的簇类中心集合$V=\{v_1, v_2, \cdots, v_q\}$和噪声数据对象集合$W=\{w_1, w_2, \cdots, w_p\}$，对于$\forall B\subseteq C$，剩余对象$U'=\{x \mid x\in U, x\notin W\}$在$B$下，根据$V$中的各个簇类中心进行就近分配。最后，对剩余对象$U'$聚类并将其划分成数个簇类，即：

$$U'/R_B^{v_i} = \{[v_1]_B, [v_2]_B, \cdots, [v_q]_B\}$$

其中，$R_B^{v_i}=\{(x,y)\in U\times U \mid (x,y)\rightarrow v_i, \forall v_i\in V\}$表示同簇不可区分关系，表示在特征子集$B$的条件下，对于任意两个数据对象$x$和$y$，如果被划分给同一个簇类中心，那么对象$x$和$y$就存在$B$下的同簇不可区分关系。$[v_i]_B$则表示以$v_i$为簇类中心的同簇类，表示为：

$$[v_i]_B=\{x\in U' \mid (v_i, x)\rightarrow v_i\}$$

在定义6中，在剔除噪声数据后，根据簇类中心将剩余数据划分成数个以簇类中心为核心的同簇类，同簇类也可视为粗糙集理论中的等价类。同时，假设同簇类对象应具有同样的决策，就如同粗糙集理论中的正区域对象拥有同样的决策，我们定义了特征重要度评估度量，以同簇类的纯度反映特征的重要度。

定义7 给定一个决策系统$DS=\langle U, A=C\cup D, V, f\rangle$，对于$\forall B\subseteq C$，$U$的簇类中心集合为$V=\{v_1, v_2, \cdots, v_q\}$和噪声对象集合为$W=\{w_1, w_2, \cdots, w_p\}$，那么特征子集$B$的重要度评估可以表示为：

$$Sig(B)=\frac{\sum_{i=1}^{q} \left| \bigcup \{x\in [v_i]_B \mid f(x,d)=f(v_i,d)\} \right|}{|U'|}$$

其中，$U'=\{x \mid x\in U, x\notin W\}$，$v_i\in V$为簇类中心，$[v_i]_B$表示以$v_i$为簇类中心的同簇类，$d\in D$是决策特征，$f(x,d)$表示对象$x$在决策特征$d$的特征值。

根据定义7评估特征重要性，并设计了一种启发式的特征选择算法，具体的算法描述如算法1所示。

算法1 基于粗糙集与密度峰值聚类的特征选择算法(RS-DPC-FS)

输入：DS=〈U, A=C∪D, V, f〉, k

输出：特征选择结果 Red

1. Red ←Ø；
2. 计算任意两个数据对象的欧氏距离；
3. 根据定义4计算所有数据对象的局部密度；
4. 根据定义5计算所有数据对象的相对距离；
5. 绘制决策图，获取簇类中心集V和噪声点集W；
6. 剔除噪声点，保留剩余对象U′=U−W；
7. $\forall a_i\in C-Red$，根据定义6，在特征子集$Red\cup a_i$的条件下，将剩余对象U′就近分配给V中的各个簇类中心并划分出多个同簇类；
8. 根据定义7，计算特征重要度 Sig(Red ∪a_i)；
9. 选择 $a_b = \arg\max_{a_i\in C-Red} Sig(Red\cup a_i)$，若 $Sig(Red\cup a_b) > Sig(Red)$，则令 $Red=Red\cup a_b$，并转至步骤7；否则执行步骤10；
10. 返回并输出 Red。

算法RS-DPC-FS的时间复杂度分析如下：步骤2计算任意两个对象的欧氏距离，其时间复杂度为$O(|U|^2|C|)$；步骤3和步骤4分别计算局部密度和相对距离，其时间复杂度为$O(|U|^2)$；步骤7—步骤9是搜索获取特征子集的过程，其时间复杂度为$O(|U|^2|C| + |U|) + O(|U|^2|C-1| + |U|) + \cdots + O(|U|^2 + |U|) =$

$O\left(\frac{|C||C+1|}{2}|U|^2+|U||C|\right)$。因此，整个算法的时间复杂度为 $O(|U|^2|C|^2)$。

4 实验结果与分析

在本节中，为了验证本文算法的效率和有效性，从 UCI 数据库中选取了 6 个公开数据集进行测试，数据集的具体信息如表 1 所列。所有的测试实验的具体环境为：Windows 10 系统，Intel(R) Core(TM) i5-7300HQ CPU 2.5 GHz 处理器，8.00 GB 运行内存，算法编程语言为 Python，程序开发工具为 Pycharm 2017。本文中，针对数据集存在的连续型数据，使用 Rossta[20] 对其进行归一化处理。

表 1 6 个 UCI 数据集的描述

Table 1 Description of six UCI datasets

序号	数据集	样本数	特征数	类别数
1	Wine	178	18	3
2	Credit	690	15	2
3	Heart	270	13	2
4	Seeds	210	7	3
5	Parkinson	1 040	27	2
6	BreatTissue	106	9	6

4.1 参数设置

在基于粗糙集与密度峰值聚类的特征选择算法（以下简写为 RS-DPC-FS 算法）的特征选择过程中，首先需要设定 k 值以选定近邻对象并计算局部密度和相对距离，而后绘制决策图。不同的 k 值参数会得到不同的决策图，从而影响特征子集的结果，对最终的分类精度起着重要的作用。因此，本文将 k 值以步长为 2 从 3 增加到 9 进行实验，分析参数对决策图绘制的影响，从而进一步分析参数对特征选择最终结果的影响，为每个数据集选取最佳的参数设定。

图 4－图 9 是各个数据集在不同 k 值参数下所绘制的决策图，其中每个子图的横轴为局部密度，纵轴为相对距离。从图 4－图 9 可以看出，随着 k 值的增大，整个决策图的样本分布相对更为分散。在 Heart 数据集下，在 k 值从 3 增加到 9 的决策图中，局部密度分布逐渐变得更加分散，这是因为局部密度根据 k 个近邻样本的距离计算，由于 k 值增加，距离增大，其局部密度减小，这就导致样本局部密度逐渐趋近于零时，局部密度相差减小，使得样本差异性更小，分布更为密集，因此放大决策图后观察，数据分布就相对更加分散，直接影响数据的决策图分布。

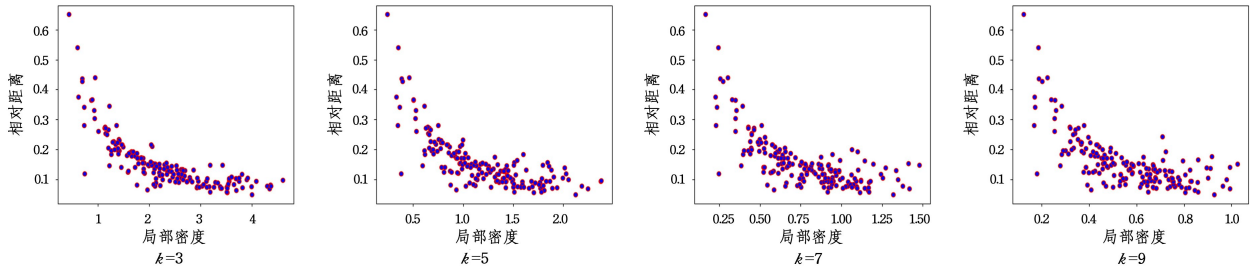


图 4 在 Wine 数据集下的 k 值为[3,9]的决策图

Fig. 4 Decision diagrams with k value in [3,9] on Wine dataset

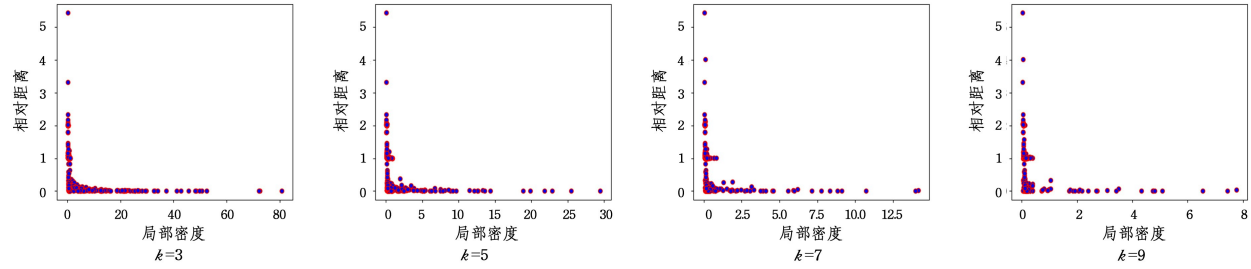


图 5 在 Credit 数据集下的 k 值为[3,9]的决策图

Fig. 5 Decision diagrams with k value in [3,9] on Credit dataset

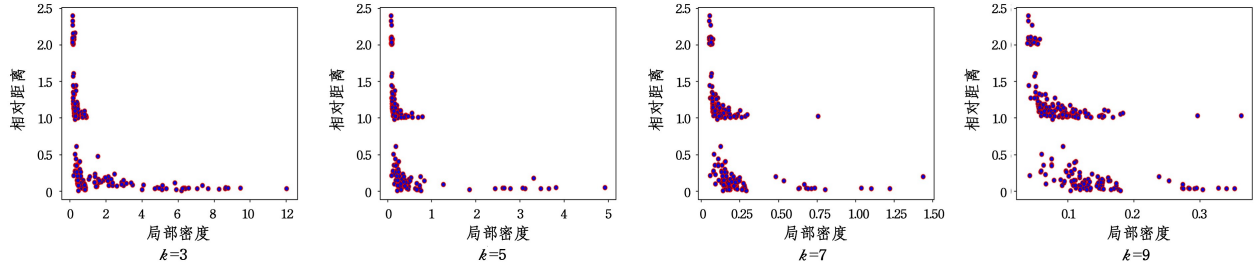


图 6 在 Heart 数据集下的 k 值为[3,9]的决策图

Fig. 6 Decision diagrams with k value in [3,9] on Heart dataset

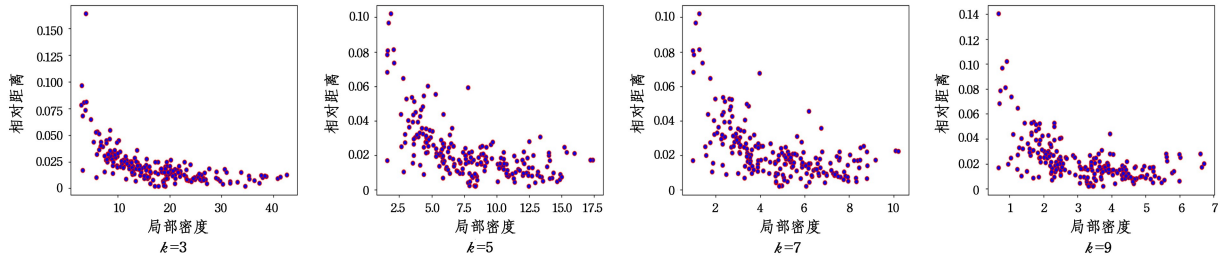


图 7 在 Seeds 数据集下的 k 值为[3,9]的决策图

Fig. 7 Decision diagrams with k value in [3,9] on Seeds dataset

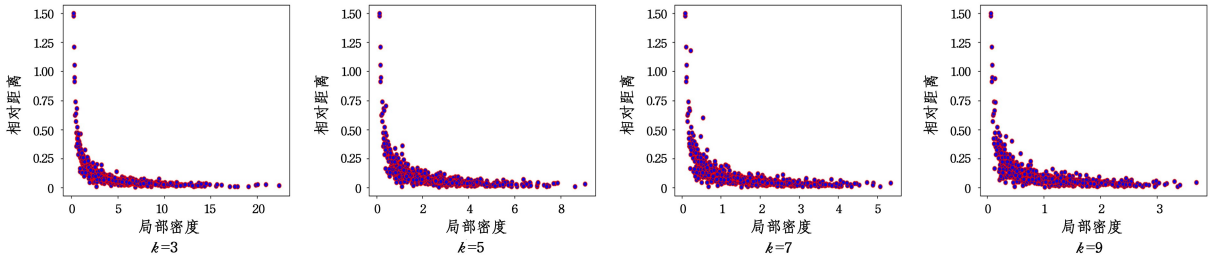


图 8 在 Parkinson 数据集下的 k 值为[3,9]的决策图

Fig. 8 Decision diagrams with k value in [3,9] on Parkinson dataset

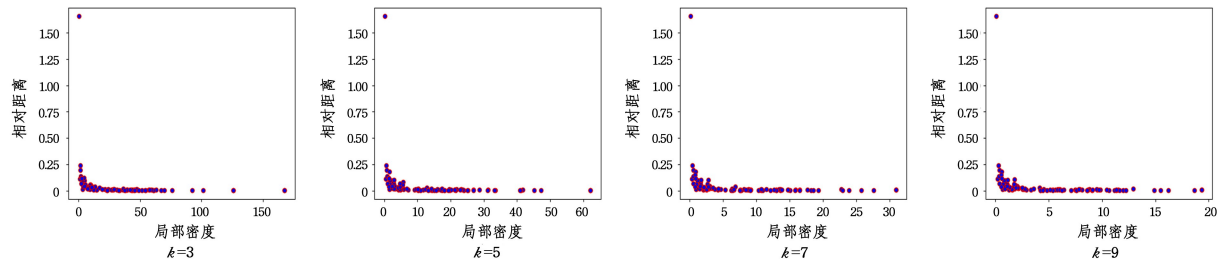


图 9 在 BreatTissue 数据集下的 k 值为[3,9]的决策图

Fig. 9 Decision diagrams with k value in [3,9] on BreatTissue dataset

在根据决策图获取簇类中心和噪声点的过程中，为了避免主观判断簇类中心点和噪声点可能导致的误差，本文中的簇类中心点选取数据对象局部密度最大的前 10%，噪声点选取数据对象相对距离最大的前 10%。各个数据集在不同 k 值下的决策树(C4.5)分类器下的分类精度和特征选择数量如图 10 所示。

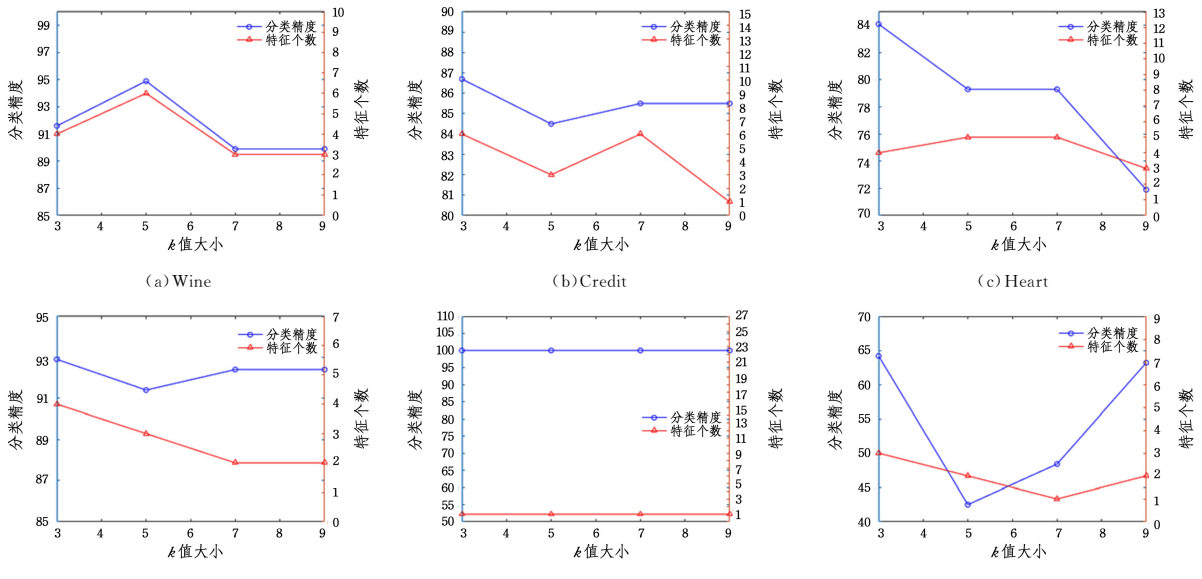


图 10 6 个数据集在不同 k 值参数下的分类精度和特征个数

Fig. 10 Classification accuracy and feature number of six data sets with different value parameters

由图 10 可知，随着 k 值的增加，分类精度和特征选择个数都可能会随之变化。在 Wine 数据集中，当 $k=5$ 时，其分类精度为 94.9%，个数为 6，此时特征选择结果在此参数下能获取最佳的精度，因此 RS-DPC-FS 算法在 Wine 数据集的最佳 k 值参数设为 5。Parkinson 数据集中，由于在不同 k 值的分类精度都为 100，且特征个数都为 1，因此将最佳 k 值设为最小的 3。针对剩余的 4 个数据集 Credit，Heart，Seeds，Breat-Tissue，当 k 参数分别设为 3，3，3，3 时，数据集能获得最优的分类性能。

4.2　不同算法的性能比较

为了验证所提算法的可行性和有效性，本文将 RS-DPC-FS 算法与其他 4 种基于邻域粗糙集的特征选择方法进行对比实验，即基于邻域差别矩阵的特征选择算法（NDMFS）[11]、基于邻域区分度的特征选择算法（NDFS）[13]、基于邻域熵的特征选择算法（NEFS）[10]、基于邻域属性依赖度的特征选择算法（NADFS）[12]。在基于邻域粗糙集的特征选择过程中，计算信息粒度需要设置邻域半径 ε，该参数决定邻域粒度的大小，影响最终的特征选择结果。因此，为了保持实验对比的有效性，本文使用文献[21]的方法，以步长 0.02 从 0.02 增加到 0.4 进行实验，分析各种算法的邻域半径 ε 对各个数据集进行特征选择的最终结果的影响，并为每种算法针对每个数据集选取最佳的邻域半径 ε 参数，最后将获取的最佳结果与本文算法进行特征选择的最佳结果进行对比。图 11－图 16 给出了 4 种算法在 6 个数据集上在不同邻域参数时，在决策树（C4.5）分类器下的分类精度和特征个数。

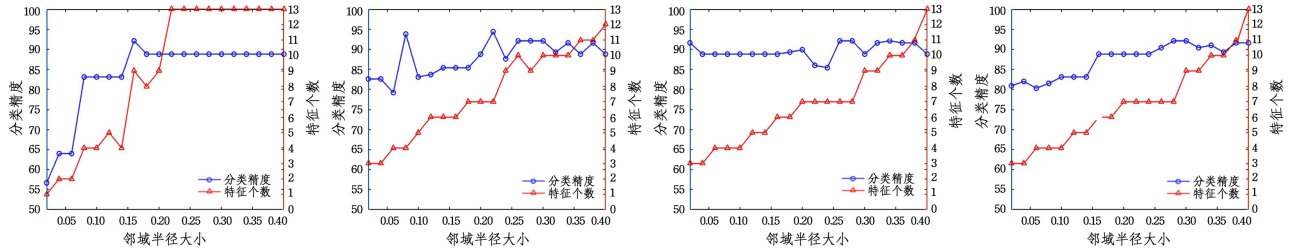


图 11　算法 NDMFS，NDFS，NEFS，NADFS 在 Wine 数据集上对于不同邻域半径的分类精度和特征个数

Fig. 11　Classification accuracy and feature number of algorithms NDMFS，NDFS，NEFS and NADFS for different neighborhood radii on Wine dataset

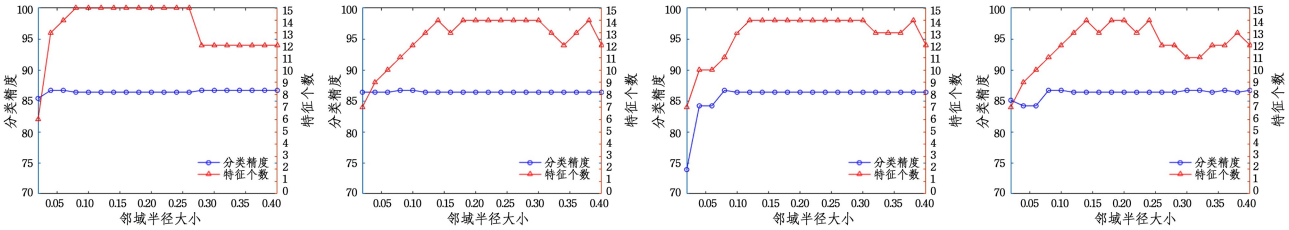


图 12　算法 NDMFS，NDFS，NEFS，NADFS 在 Credit 数据集上对于不同邻域半径的分类精度和特征个数

Fig. 12　Classification accuracy and feature number of algorithms NDMFS，NDFS，NEFS and NADFS for different neighborhood radii on Credit dataset

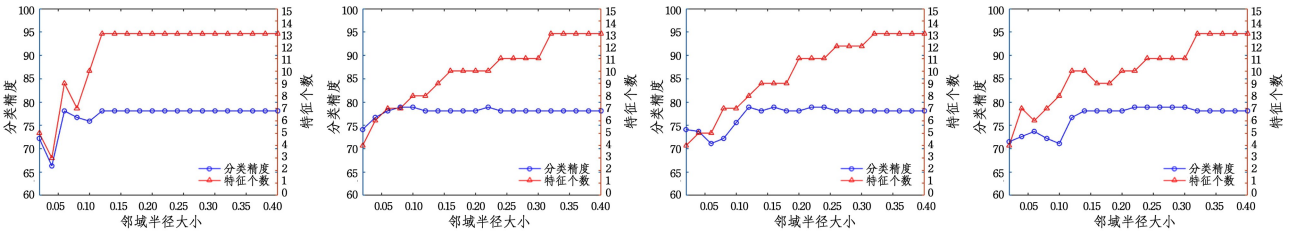


图 13　算法 NDMFS，NDFS，NEFS，NADFS 在 Heart 数据集上对于不同邻域半径的分类精度和特征个数

Fig. 13　Classification accuracy and feature number of algorithms NDMFS，NDFS，NEFS and NADFS for different neighborhood radii on Heart dataset

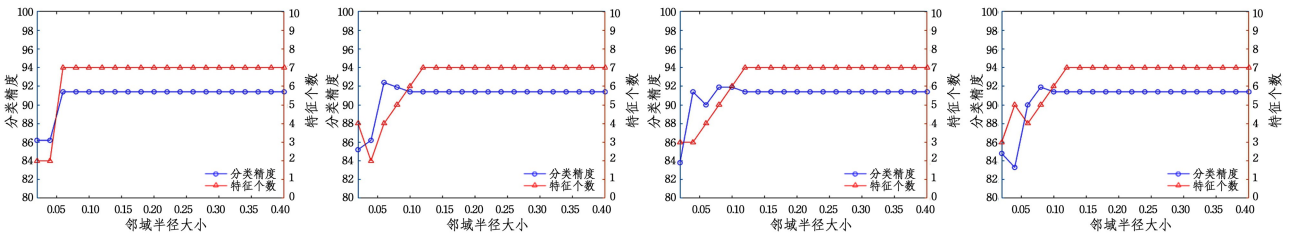


图 14　算法 NDMFS，NDFS，NEFS，NADFS 在 Seeds 数据集上对于不同邻域半径的分类精度和特征个数

Fig. 14　Classification accuracy and feature number of algorithms NDMFS，NDFS，NEFS and NADFS for different neighborhood radii on Seeds dataset

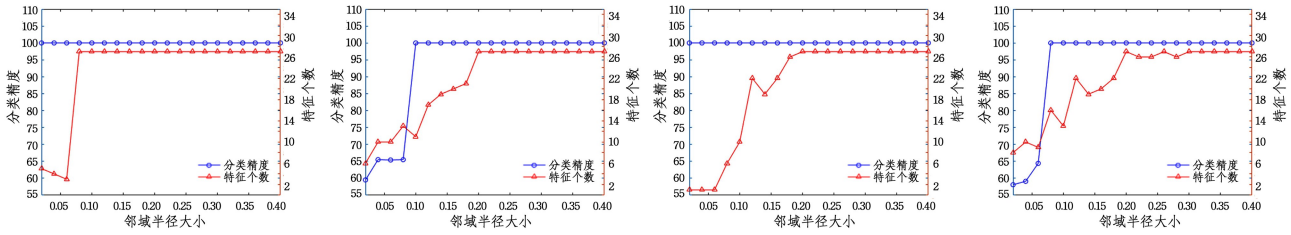


图 15 算法 NDMFS, NDFS, NEFS, NADFS 在 Parkinson 数据集上对于不同邻域半径的分类精度和特征个数

Fig. 15 Classification accuracy and feature number of algorithms NDMFS, NDFS, NEFS and NADFS for different neighborhood radii on Parkinson dataset

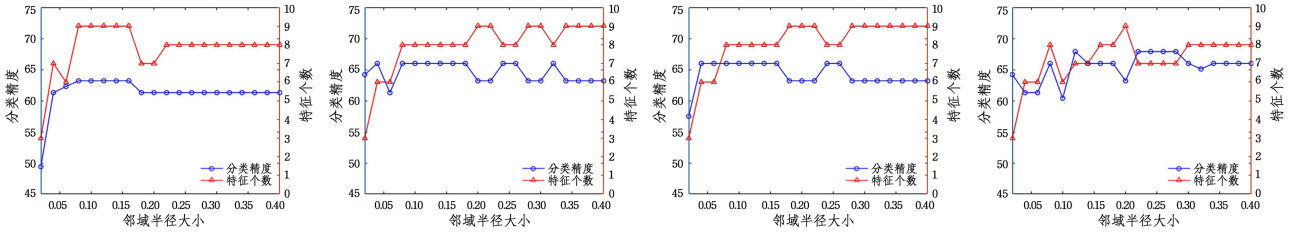


图 16 算法 NDMFS, NDFS, NEFS, NADFS 在 BreatTissue 数据集上对于不同邻域半径的分类精度和特征个数

Fig. 16 Classification accuracy and feature number of algorithms NDMFS, NDFS, NEFS and NADFS for different neighborhood radii on BreatTissue dataset

根据图 11－图 16，对于每个数据集的子图，从左到右分别排列的是 NDMFS 算法、NDFS 算法、NEFS 算法、NADFS 算法在该数据集中的实验结果。从图 11－图 16 可以看出，随着邻域半径从 0.02 逐渐增加到 0.4，各种算法在不同数据集上的实验结果都会不断变化。

在 Wine 数据集中，NDMFS 算法在邻域半径为 0.16 时，可得到的最佳分类精度为92.1%；NDFS 算法在邻域半径为 0.22 时，能获取的最佳分类精度为 94.4%；NEFS 算法在邻域半径为 0.26 时能获得的最佳分类精度为 92.1%；NADFS 算法在邻域半径为 0.28 时能获得的最佳分类精度为92.1%。为获得各个数据集下的最佳分类性能，针对剩下的 5 个数据集 Credit, Heart, Seeds, Parkinson, BreatTissue，邻域半径 ε 参数在 NDMFS 算法中分别设为 0.04, 0.06, 0.06, 0.06, 0.08；在 NDFS 算法中分别设为 0.08, 0.08, 0.06, 0.10, 0.04；在 NEFS 算法中分别设为 0.08, 0.12, 0.08, 0.02, 0.04；在 NADFS 算法中分别设为 0.08, 0.22, 0.08, 0.10, 0.12。

4.2.1 分类精度上的对比

本文采用决策树(C4.5)、支持向量机(SVM)和 KNN 分类器 3 种分类器，将本文算法与其他 4 种不同特征选择算法的分类性能进行比较。采用十字交叉验证法，对于每个数据集，我们都将数据集均分成 10 份，并轮流将其中的 9 份作为训练集，将剩余的 1 份作为测试集进行验证。表 2－表 4 分别列出了 RS-DPC-FS 算法与 NDMFS 算法、NDFS 算法、NEFS 算法和 NADFS 算法在 3 种分类器下的最佳分类精度对比结果。其中“Raw”表示特征全集下的分类精度，“Average”表示不同算法下的平均分类精度，数据右上角的“+ = −”分别标记了本文提出的 RS-DPC-FS 算法相比其他算法的分类精度拥有更好\一样\更差的效果，“Win\Tie\Lose”用于统计 RS-DPC-FS 算法在所有数据集中，相比其他算法拥有“更好\一样\更差”的分类性能的数据集个数。同时，表 5 列出了 5 种算法在 6 个数据集上的特征选择个数。

表 2 算法 RS-DPC-FS, NDMFS, NDFS, NEFS 和 NADFS 在 C4.5 分类器下的分类精度

Table 2 Classification accuracy ofalgorithms RS-DPC-FS, NDMFS, NDFS, NEFS and NADFS under C4.5 classifier

数据集	Raw	RS-DPC-FS	NDMFS	NDFS	NEFS	NADFS
Wine	88.8	94.9	92.1^{+}	94.4^{+}	92.1^{+}	92.1^{+}
Credit	86.4	86.7	$86.7^{=}$	$86.7^{=}$	$86.7^{=}$	$86.7^{=}$
Heart	78.1	84.1	78.1^{+}	78.9^{+}	78.9^{+}	78.9^{+}
Seeds	91.4	92.9	91.4^{+}	92.4^{+}	91.9^{+}	91.9^{+}
Parkinson	100.0	100.0	$100.0^{=}$	$100.0^{=}$	$100.0^{=}$	$100.0^{=}$
BreatTissue	63.2	64.2	63.2^{+}	66.0^{-}	66.0^{-}	67.9^{-}
Average	84.65	87.13	85.25	86.40	85.93	86.25
Win\Tie\Lose	—	—	4/2/0	3/2/1	3/2/1	3/2/1

表 3 算法 RS-DPC-FS, NDMFS, NDFS, NEFS 和 NADFS 在 SVM 分类器下的分类精度

Table 3 Classification accuracy of algorithms RS-DPC-FS, NDMFS, NDFS, NEFS and NADFS under SVM classifier

数据集	Raw	RS-DPC-FS	NDMFS	NDFS	NEFS	NADFS
Wine	98.9	98.9	98.3^{+}	$98.9^{=}$	99.4^{-}	96.6^{+}
Credit	85.9	85.8	86.2^{-}	85.5^{+}	85.4^{+}	85.5^{+}
Heart	84.1	84.8	84.4^{+}	83.7^{+}	83.7^{+}	84.4^{+}
Seeds	92.9	93.3	92.9^{+}	94.3^{-}	$93.3^{=}$	$93.3^{=}$
Parkinson	98.3	100.0	$100.0^{=}$	$100.0^{=}$	$100.0^{=}$	$100.0^{=}$
BreatTissue	69.8	67.9	$67.9^{=}$	69.8^{-}	69.8^{-}	69.8^{-}
Average	88.31	88.45	88.28	88.70	88.60	88.26
Win\Tie\Lose	—	—	3/2/1	2/2/2	2/2/2	3/2/1

表 4 算法 RS-DPC-FS，NDMFS，NDFS，NEFS 和 NADFS 在 KNN 分类器下的分类精度

Table 4 Classification accuracy of algorithms RS-DPC-FS，NDMFS，NDFS，NEFS and NADFS under KNN classifier

数据集	Raw	RS-DPC-FS	NDMFS	NDFS	NEFS	NADFS
Wine	97.2	97.8	96.1^{+}	96.1^{+}	98.3^{-}	96.1^{+}
Credit	—	—	—	—	—	—
Heart	—	—	—	—	—	—
Seeds	93.3	94.3	93.3^{+}	93.8^{+}	92.9^{+}	92.9^{+}
Parkinson	81.8	99.9	99.1^{+}	90.4^{+}	$99.9^{=}$	90.1^{+}
BreatTissue	67.9	68.9	67.9^{+}	$68.9^{=}$	$68.9^{=}$	67.9^{+}
Average	85.05	90.225	89.1	87.3	90	86.75
Win\Tie\Lose	—	—	4/0/0	3/1/0	1/2/1	4/0/0

表 5 算法 RS-DPC-FS，NDMFS，NDFS，NEFS 和 NADFS 的特征选择个数

Table 5 Feature selection numbers of algorithms RS-DPC-FS NDMFS，NDFS，NEFS and NADFS

数据集	特征选择个数				
	RS-DPC-FS	NDMFS	NDFS	NEFS	NADFS
Wine	**6**	9	7	7	9
Credit	**6**	13	11	11	11
Heart	**4**	9	7	8	10
Seeds	**4**	7	**4**	5	5
Parkinson	**1**	3	11	**1**	13
BreatTissue	**3**	9	6	6	7

从分类精度表中可以看出，在 C4.5 分类器下，RS-DPC-FS 算法得到的平均分类精度为 87.13%，相比 NDMFS 算法的 85.25%，NDFS 算法的 86.4%，NEFS 算法的 85.93%和 NADFS 算法的 86.25%，RS-DPC-FS 算法的分类性能明显更优；在 SVM 分类器下，RS-DPC-FS 算法得到的平均分类精度为 88.45%，优于 NDMFS 算法的 88.28%和 NADFS 算法的 86.26%。此外，通过 Win\Tie\Lose 的统计结果可知，相比其他 4 种算法，RS-DPC-FS 算法在大部分数据集下都可以得到“更好”或“一样”的分类结果。

同时，结合表 5 所列的特征选择结果数量可知，算法 RS-DPC-FS 在 6 个数据集中都获得了最少的特征选择个数。在 Parkinson 数据集中，尽管 5 种特征选择算法的结果在 C4.5 和 SVM 分类器下都为 100%，但 RS-DPC-FS 算法的特征选择结果数量为 1，而 NDMFS 算法的特征选择结果为 3，NDFS 算法的特征选择结果为 11，NADFS 算法的特征选择结果为 13，这表明了 RS-DPC-FS 算法的有效性。

实验结果表明，本文算法能有效地进行特征选择，保证较好的分类效果和更少的特征选择个数。

4.2.2 运行时间上的对比

为了验证所提算法在运行过程中的效率，下文将 RS-DPC-FS 算法与其他 4 种算法进行运行时间的对比，这 4 种算法包括：NDMFS 算法、NDFS 算法、NEFS 算法、NADFS 算法。首先在参数设定方面，实验沿用上文的参数设定对 RS-DPC-FS 算法中的 k 和其他 4 种算法中的邻域半径 ε 进行设置。在每种算法对于每个数据集获取最佳的分类效果后，记录其运行时间，每次实验运行 5 次，最后取 5 次的平均运行时间。图 17 为 5 种算法在 6 个数据集上的平均运行时间的对比图。

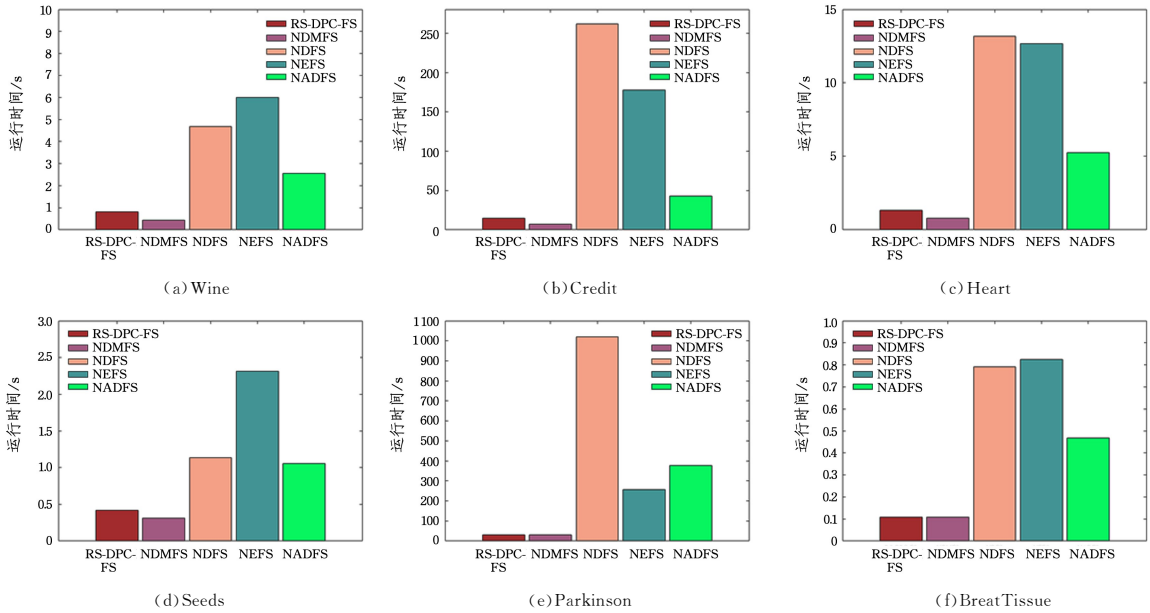


图 17 5 种算法在 6 个数据集上的运行时间

Fig. 17 Running time of five algorithms on six data sets

如图 7 所示，其中每个子图的横轴代表各种算法，从左到右分别是 RS-DPC-FS 算法、NDMFS 算法、NDFS 算法、NEFS 算法和 NADFS 算法，纵轴代表平均运行时间。由图可知，RS-DPC-FS 算法与 NDMFS 算法在运行时间上相差不大，而相比 NDFS 算法、NEFS 算法和 NADFS 算法，RS-DPC-FS 算法的运行时间明显更短。在 Parkinson 数据集中，RS-DPC-FS 算法的平均运行时间为 31.015s，相比 NDMFS 算法的 30.1175 s，两种算法在运行时间上相差不大。同时，算法 NDFS，NEFS 和 NADFS 在该数据集中的平均运行时间分别为 1020.2694 s，257.7392 s 和 376.7008 s，而 RS-DPC-FS 算法的平均运行时间为 31.015 s，相比算法 NDFS，NEFS 和 NADFS 分别缩短了 96.96%，87.96%和 91.76%。

综上所述，本文算法有着较高的运行效率，在保证较好的分类性能的前提下，可以用较短的运行时间进行特征选择。

4.3 算法的统计检验分析

为了进一步验证所提算法的有效性，采用 Friedman Test 及 Nemenyi Test 两种统计检验方法对上述不同算法的实验结果进行统计分析。

Friedman Test 是一种非参数统计检验方法，其表达式为：

$$F_F=\frac{(T-1)\chi_F^2}{T(s-1)-\chi_F^2}$$

其中，

$$\chi_F^2=\frac{12T}{s(s+1)}\left(\sum_{i=1}^{s}R_i^2-\frac{s(s+1)^2}{4}\right)$$

T 和 s 分别为需验证的实验数据集数量和实验算法数量，R_i 代表第 i 种算法在不同分类器上分类精度结果的平均排名值。

表 6 列出了算法 RS-DPC-FS，NDMFS，NDFS，NEFS 和 NADF 在分类器 C4.5，SVM 和 KNN 上的分类精度结果的平均排名。这里需要注意的是，对于 C4.5 分类器，在 Parkinson 和 Credit 数据集上，尽管各种算法的特征结果各不相同（具体可以参考表 5 中的特征选择个数表），但所有算法的精度结果都相同。为了更显著地表现各种算法在精度上的性能差异性，在统计过程中对 Parkinson 和 Credit 数据集上的精度结果进行剔除。同样地，将 SVM 分类器上的 Parkinson 数据集的精度结果也剔除。而对于 KNN 分类器，Credit 和 Heart 数据集无法在 KNN 分类器上进行精度测试，因此在统计过程中将其忽略。

表 6 算法 RS-DPC-FS，NDMFS，NDFS，NEFS 和 NADFS 分类精度结果的平均排名

Table 6 Average ranking of classification accuracies of algorithms RS-DPC-FS, NDMFS, NDFS, NEFS and NADFS

性能指标	RS-DPC-FS	NDMFS	NDFS	NEFS	NADFS
C4.5 分类器	1.750	4.750	2.375	3.250	2.875
SVM 分类器	2.6	3.4	2.7	3.1	3.2
KNN 分类器	1.625	3.625	3.000	2.250	4.500

根据表 6 的平均排名结果可以计算出算法分类性能的 Friedman 值，如表 7 所列。同时，本次统计检验中的一次性检验值为 $\alpha=0.1$，可得最终的置信度值为 2.480。由表 7 可知，在 C4.5 和 KNN 分类器上的分类性能的 Friedman 值大于置信度值 2.480，因此可以拒绝零假设，这表明各种算法在 C4.5 和 KNN 分类器上表现的精度性能显著不同。为进一步区分各种算法，采用 Nemenyi 检验进行后续检验，其中 Nemenyi 检验的临界差表达式为：

$$CD=q_\alpha\sqrt{\frac{s(s+1)}{6T}}$$

表 7 算法 RS-DPC-FS，NDMFS，NDFS，NEFS 和 NADFS 分类精度结果的 Friedman 值

Table 7 Friedman statistics of classification accuracies of algorithms RS-DPC-FS, NDMFS, NDFS, NEFS and NADFS

性能指标	Friedman 值	置信度值（$\alpha=0.1$）
C4.5 分类器	3.114	
SVM 分类器	0.192	2.480
KNN 分类器	3.114	

根据临界表值可得 $q_{(0.1)}=2.459$，进而根据临界差公式计算可得 $CD_{(0.1)}=2.749$。最后根据表 6 中的平均排名结果及计算所得的 $CD_{(0.1)}=2.749$ 绘制各种算法在 C4.5 和 KNN 分类器上表现的 Nemenyi 检验结果图，如图 18 所示。

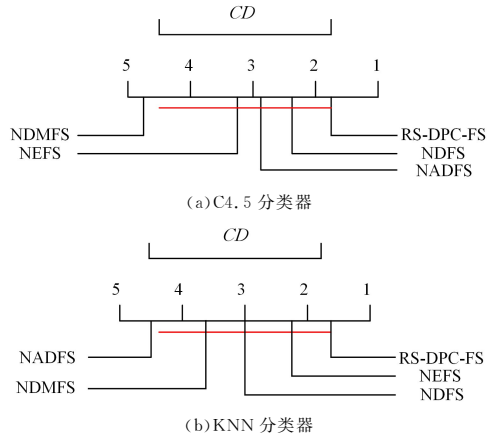


图 18 算法 RS-DPC-FS，NDMFS，NDFS，NEFS 和 NADFS 的 Nemenyi 检验结果图

Fig. 18 Nemenyi test results of algorithms RS-DPC-FS, NDMFS, NDFS, NEFS and NADFS

由图 18 可知，在 C4.5 和 KNN 分类器下，算法 RS-DPC-FS 的分类精度都最优。在 C4.5 分类器中，算法 RS-DPC-FS 的分类精度明显比算法 NDMFS 好，而与算法 NDFS，NADFS 和 NEFS 不存在显著的差异。同时，在 KNN 分类器中，算法 RS-DPC-FS 的分类精度明显比算法 NADFS 好，而与算法 NEFS，NDFS 和 NDMFS 不存在显著的差异。

综上所述，统计结果表明，本文算法的精度相比其他几种算法存在一定程度的优势。

结束语 本文提出了基于粗糙集与密度峰值聚类的特征选择算法。首先，采用密度峰值聚类算法标记簇类中心并剔除可能影响分类效果的噪声数据，提高数据的整体质量。其次，结合粗糙集理论思想，根据簇类中心对剩余数据进行划分，并提出特征重要性评估函数。最后，构建启发式特征选择算法，选取出使得簇类纯度更高的特征子集。将本文算法在 6 个数据集中，与其他 4 种算法在 3 种分类器下进行分类效果的对比实验，在运行时间上进行效率的对比实验分析，实验结果充分验证了本文所提出的 RS-DPC-FS 算法有效性和高效性。之后，将探索 RS-DPC-FS 算法在多标记数据上的可行性。

参 考 文 献

[1] JING Y G, JING L X, WANG B L, et al. Incremental attribute reduction algorithm for attribute values and attribute changes [J]. Journal of Shandong University: Science Edition, 2020, 55(1): 62-68.

[2] WANG C Z, HUANG Y, SHAO M W, et al. Feature Selection Based on Neighborhood Self-Information[J]. IEEE Transactions on Cybernetics, 2020, 50(9): 4031-4042.

[3] WANG Q, QIAN Y H, LIANG X Y, et al. Local neighborhood rough set[J]. Knowledge-Based Systems, 2018, 153: 53-64.

[4] WANG D, CHEN H M, LI T R, et al. A novel quantum grasshopper optimization algorithm for feature selection[J]. International Journal of Approximate Reasoning, 2020, 127: 33-53.

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

[6] LIU Y,CHENG L,SUN L. Feature selection method based on K-S test and neighborhood rough set[J]. Journal of Henan Normal University:Natural Science Edition,2019,47(2):21-28.

[7] XUE Z A,PANG W L,YAO S Q,et al. Intuitionistic fuzzy three-branch decision-making model based on prospect theory [J]. Journal of Henan Normal University:Natural Science Edition,2020,48(5):31-36,79.

[8] YANG X L,CHEN H M,LI T R,et al. Neighborhood rough sets with distance metric learning for feature selection[J]. Knowledge-Based Systems,2021,224:107076.

[9] MARIELLO A,BATTITI R. Feature Selection Based on the Neighborhood Entropy[J]. IEEE Transactions on Neural Networks and Learning Systems,2018,29(12):6313-6322.

[10] WANG C Z,HE Q,SHAO M W,et al. Feature selection based on maximal neighborhood discernibility[J]. International Journal of Machine Learning & Cybernetics,2019,9(11):1929-1940.

[11] HU Q H,ZHAO H,YU D R. Fast reduction algorithm of symbolic and numerical attributes based on neighborhood rough sets [J]. Pattern Recognition and Artificial Intelligence,2008,21(6):730-738.

[12] SHENG K,WANG W,BIAN X F,et al. Neighborhood discrimination incremental attribute reduction algorithm for mixed data [J]. Acta Electronica,2020,48(4):682-696.

[13] RODRIGUEZ A,LAIO A. Clustering by fast searchand find of density peaks[J]. Science,2014,344(6191):1492-1496.

[14] ZOU X H,YE X D,TAN Z Y. A color image segmentation method based on density peak clustering[J]. Microcomputer System,2017,38(4):868-871.

[15] HUANG L,LI Y,WANG G S,et al. Community discovery method based on point distance and density peak clustering[J]. Journal of Jilin University:Engineering Edition,2016,46(6):2042-2051.

[16] DU M,DING S,XU X,et al. Density peaks clustering using geodesic distances[J]. International Journal of Machine Learning & Cybernetics,2018,9(8):1355-1349.

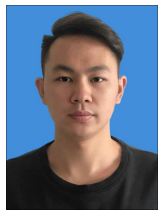
[17] BIAN Z K,CHUNG F L,WANG S T. Fuzzy Density Peaks Clustering[J]. IEEE Transactions on Fuzzy Systems,2021,29(7):1725-1738.

[18] LIU R,HUANG W,FEI Z,et al. Constraint-based clustering by fast search and find of density peaks[J]. Neurocomputing,2019,330:223-237.

[19] XUE X N,GAO S P,PENG H M,et al. Density peak clustering algorithm based on K nearest neighbor and multi-class merging [J]. Journal of Jilin University:Science Edition,2019,57(1):111-120.

[20] Rosetta:A rough set toolkit for analysis of data [OL]. http://www.lcb.uu.se/tools/rosetta/index.php.

[21] HU Q H,YU D R,LIU J F,et al. Neighborhood rough set based heterogeneous feature subset selection[J]. Information Sciences,2008,178(18):3577-3594.



CAO Dongtao,born in 1997,master. His main research interests include machine learning,data mining,rough set,etc.

SHU Wenhao,born in 1985,Ph.D,associate professor,master supervisor. Her main research interests include data mining, knowledge discovery, rough set,etc.

(责任编辑:喻藜)