

基于改进徒步优化算法的变精度邻域粗糙集特征选择研究

朱永佳

烟台大学计算机与控制工程学院, 山东 烟台

收稿日期: 2025年3月25日; 录用日期: 2025年4月23日; 发布日期: 2025年4月30日

摘要

特征选择是数据降维的有效方法, 粗糙集理论是处理不确定性信息的有效工具, 在粗糙集理论中, 变精度邻域粗糙集是处理连续型数值信息的有效工具, 在分类任务中, 变精度邻域粗糙集可以为属性集提供重要性度量。徒步优化算法是群智能优化算法中的一种, 可以寻找组合最优解, 本文针对徒步优化算法无法动态平衡局部与全局寻优能力, 容易陷入局部最优的问题, 提出改进徒步优化算法, 并提出基于改进徒步优化算法的特征选择算法。为了验证所提算法的有效性, 在8个UCI数据集上进行了基于改进徒步优化算法的特征选择算法的对比实验, 通过实验验证了所提出算法的有效性和可行性。

关键词

徒步优化算法, 优化算法改进, 变精度邻域粗糙集, 特征选择

Research on Variable Precision Neighborhood Rough Set Feature Selection Based on Improved Hiking Optimization Algorithm

Yongjia Zhu

School of Computer and Control Engineering, Yantai University, Yantai Shandong

Received: Mar. 25th, 2025; accepted: Apr. 23rd, 2025; published: Apr. 30th, 2025

Abstract

Feature selection is an effective method for data dimensionality reduction, and rough set theory is an effective tool for dealing with uncertainty information. In rough set theory, variable

文章引用: 朱永佳. 基于改进徒步优化算法的变精度邻域粗糙集特征选择研究[J]. 计算机科学与应用, 2025, 15(4): 478-486. DOI: 10.12677/csa.2025.154119

precision neighborhood rough set is an effective tool for dealing with continuous numerical information systems. Hiking optimization algorithm is one of the swarm intelligence optimization algorithms, which can find the combined optimal solution. In this paper, aiming at the problem that the hiking optimization algorithm cannot dynamically balance the local and global optimization ability, and is easy to fall into the local optimum, an improved hiking optimization algorithm is proposed. A feature selection algorithm based on improved hiking optimization algorithm was proposed. In order to verify the effectiveness of the proposed algorithm, comparative experiments of feature selection algorithm based on improved hiking optimization algorithm are carried out on 8 UCI data sets, and the effectiveness and feasibility of the proposed algorithm are verified by experiments.

Keywords

Hiking Optimization Algorithm, Optimization Algorithm Improvement, Variable Precision Neighborhood Rough Set, Feature Selection

Copyright © 2025 by author(s) and Hans Publishers Inc.

This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

1. 引言

数据降维作为人工智能数据预处理的关键步骤，在机器学习领域具有重要的理论和实践意义。通过减少特征数量，数据降维不仅能够提高算法效率、降低存储需求，还能有效去除冗余信息，揭示数据潜在结构，从而增强模型的泛化能力。

粗糙集理论[1]是处理不确定性信息的有效工具，变精度邻域粗糙集[2]是基于邻域粗糙集的扩展模型，在邻域粗糙集的基础上引入了变精度参数 β ，允许在分类过程中存在一定的误差，增加了模型对于噪声样本的处理能力，在连续型数值信息系统的分类任务中，变精度邻域粗糙集可以有效地提供属性集的重要度量。徒步优化算法[3]是群智能优化算法的一种，徒步优化算法的思想源于徒步旅行者登山的过程，徒步者根据目前领队的位置进行移动，并且受到地形的影响发生速度改变。

针对徒步优化算法无法动态平衡局部和全局寻优能力、易陷入局部最优解的问题，本文结合多策略改进了徒步优化算法，结合变精度邻域粗糙集，提出特征选择算法。在8个UCI数据集上进行对比实验，实验结果验证了本文所提算法在分类精度上具有一定优势，验证了所提算法的可行性和有效性。

2. 基本概念

2.1. 变精度邻域粗糙集模型

定义 1 [2]: 邻域决策系统定义为 $RDIS = (U, C \cup D, V, f)$ ，其中 $U = \{x_1, x_2, \dots, x_n\}$ 表示论域，是非空有限对象的集合， $C = \{a_1, a_2, \dots, a_m\}$ 表示非空有限的条件属性集， $D = \{d\}$ 表示决策属性， $V = V_C \cup V_D$ 表示值域，其中 V_C 表示元素在条件属性下的属性值的集合， V_D 表示元素在决策属性下的属性值的集合。 f 表示元素到值域的映射， $f_C = U \times C \rightarrow V_C$ 表示条件属性到值的映射， $f(x, a)$ 表示元素 x 在条件属性 a 下的值， $f_D = U \times D \rightarrow V_D$ 表示元素对于决策属性的映射， $f(x, d)$ 表示元素 x 的决策值。

定义 2 [2]: 在邻域粗糙集模型中，邻域决策系统 $RDIS = (U, C \cup D, V, f)$ ， (U, Δ) 定义为度量空间，其中 Δ 定义为度量，对于元素 $x, y, z \in U$ ，满足以下性质：

- 1) $\Delta(x, y) \geq 0$ ，当且仅当 $x = y$ 时， $\Delta(x, y) = 0$ ；
- 2) $\Delta(x, y) = \Delta(y, x)$ ；
- 3) $\Delta(x, y) + \Delta(y, z) \geq \Delta(x, z)$ 。

在邻域粗糙集模型中，采用欧几里得距离作为距离度量，对于任意属性集 $B \subseteq C$ ，在 B 上的距离度量 Δ 表示为：

$$\Delta_B(x, y) = \sqrt{\sum_{a \in B} (f(x, a) - f(y, a))^2} \quad (1)$$

定义 3 [2]：在邻域决策系统 $RDIS = (U, C \cup D, V, f)$ 中，属性集 $B \subseteq C$ ，对于元素 $\forall x \in U$ ，关于 B 的 δ 邻域定义为：

$$\delta_B(x) = \{y \in U \mid \Delta_B(x, y) \leq \delta\} \quad (2)$$

定义 4 [2]：在变精度邻域粗糙集模型中，邻域决策系统 $RDIS = (U, C \cup D, V, f)$ ，属性集 $B \subseteq C$ ，给定阈值 $\beta \in (0.5, 1]$ ，对于 $\forall X \subseteq U$ ， X 关于 B 的下近似和上近似定义为：

$$\underline{N}_B^\beta(X) = \left\{ x \in U \mid \frac{|\delta_B(x) \cap X|}{|\delta_B(x)|} \geq \beta \right\} \quad (3)$$

$$\overline{N}_B^\beta(X) = \left\{ x \in U \mid \frac{|\delta_B(x) \cap X|}{|\delta_B(x)|} \geq 1 - \beta \right\} \quad (4)$$

定义 5 [2]：在变精度邻域粗糙集模型中，邻域决策系统 $RDIS = (U, C \cup D, V, f)$ ，论域 U 以决策属性 $D = \{d\}$ 划分为 $U/D = \{X_1, X_2, \dots, X_{|U/IND(D)|}\}$ ，属性集 $B \subseteq C$ ，论域 U 关于决策属性 D 的下近似和上近似定义为：

$$\underline{N}_B^\beta(D) = \bigcup_{i=1}^{|U/IND(D)|} \underline{N}_B^\beta(X_i) \quad (5)$$

$$\overline{N}_B^\beta(D) = \bigcup_{i=1}^{|U/IND(D)|} \overline{N}_B^\beta(X_i) \quad (6)$$

定义 6 [2]：在变精度邻域粗糙集模型中，邻域决策系统 $RDIS = (U, C \cup D, V, f)$ ，属性集 $B \subseteq C$ ，关于决策属性 D 的正域、边界域定义为：

$$POS_B^\delta(D) = \underline{N}_B^\delta(D) \quad (7)$$

$$BND_B^\delta(D) = \overline{N}_B^\delta(D) \quad (8)$$

定义 7 [2]：在变精度邻域粗糙集模型中，邻域决策系统 $RDIS = (U, C \cup D, V, f)$ ，属性集 $B \subseteq C$ ，决策属性 D 关于属性集 B 的依赖度定义为：

$$\gamma_B^\delta(D) = \frac{|POS_B^\delta(D)|}{|U|} \quad (9)$$

2.2. 徒步优化算法

定义 8 [3]：徒步优化算法中，徒步者的速度由当前地形坡度决定，其速度定义为：

$$w_{i,t} = 6e^{-3.5|s_{i,t}+0.05|} \quad (10)$$

其中, $w_{i,t}$ 表示第 i 个徒步者 h_i 在第 t 次迭代中的速度, $s_{i,t}$ 表示徒步者 h_i 在第 t 次迭代中所处位置的坡度, 定义为:

$$s_{i,t} = \frac{dh}{dx} = \tan \theta_{i,t} \quad (11)$$

其中, dh 表示地形的高度差异, dx 表示徒步者的行走距离, $\theta_{i,t}$ 表示地形的倾斜角度, $\theta_{i,t} \in [0^\circ, 50^\circ]$ 。

定义 9 [3]: 徒步优化算法中, 徒步者的实际当前速度计算方式定义为:

$$w_{i,t} = w_{i,t-1} + \gamma_{i,t} (X_{best} - \alpha_{i,t} X_{i,t}) \quad (12)$$

其中, $\gamma_{i,t}$ 是 $[0,1]$ 内均匀分布的随机数, $w_{i,t}$ 和 $w_{i,t-1}$ 表示徒步者的当前速度和上一时刻速度, X_{best} 表示领队 h_{best} 的当前位置, $X_{i,t}$ 表示徒步者 h_i 在第 t 时刻的位置, $\alpha_{i,t}$ 表示一个扫描因子 SF , $SF \in [1,3]$ 。徒步者的位置更新函数表示为:

$$X_{i,t} = X_{i,t-1} + w_{i,t} \quad (13)$$

定义 10 [3]: 徒步优化算法中, 徒步者的位置初始化定义为:

$$X_{i,t} = lb + \delta (ub - lb) \quad (14)$$

其中, lb 和 ub 分别表示解空间维度的下界和上界, δ 是 $[0,1]$ 范围内均匀分布的随机数向量, 经过初始化可以得到一个徒步者位置矩阵:

$$R = \begin{bmatrix} X_{1,t} \\ X_{2,t} \\ \vdots \\ X_{pop,t} \end{bmatrix} = \begin{bmatrix} x_{1,t}^1 & x_{1,t}^2 & \cdots & x_{1,t}^d \\ x_{2,t}^1 & x_{2,t}^2 & \cdots & x_{2,t}^d \\ \vdots & \vdots & \vdots & \vdots \\ x_{pop,t}^1 & x_{pop,t}^2 & \cdots & x_{pop,t}^d \end{bmatrix} \quad (15)$$

其中, pop 表示徒步者数量, m 表示解空间维度, $x_{i,t}^j$ 表示徒步者 h_i 在第 t 时刻上, 在第 j 个维度上的值, 徒步者群体以集合 $H = \{h_1, h_2, \dots, h_{pop}\}$ 表示。

定义 11 [3]: 给定优化目标函数 f , 在 t 时刻, 徒步者 h_i 的适应度定义为:

$$fit_{i,t} = f(X_{i,t}) \quad (16)$$

徒步优化算法的算法流程如文献[3]所示。

3. 基于改进徒步优化算法的特征选择算法

改进徒步优化算法使用 Tent-Logistic-Cosine 混沌映射[4]来进行初始化。

定义 12: 改进徒步优化算法中, 使用 Tent-Logistic-Cosine 混沌映射进行初始化的表达式定义为:

$$x_i^{j+1} = \begin{cases} \cos\left(\pi\left(2rx_i^j + 4(1-r)x_i^j(1-x_i^j) - 0.5\right)\right), & \text{if } x_i^j < 0.5 \\ \cos\left(\pi\left(2r(1-x_i^j) + 4(1-r)x_i^j(1-x_i^j) - 0.5\right)\right), & \text{else, } r \in [0,1] \end{cases} \quad (17)$$

改进徒步优化算法在地形倾斜角度 $\theta_{i,t}$ 的生成上使用了正态分布[5]。

定义 13: 改进徒步优化算法中, 通过正态分布生成地形倾斜角度 $\theta_{i,t}$ 表达式定义为:

$$\theta_{i,t} = \frac{e^{-\frac{(x-\mu)^2}{2\sigma^2}}}{\sqrt{2\pi}\sigma} \quad (18)$$

对扫描因子的生成使用了 Beta 分布[6], Beta 分布是一种参数化的概率分布。通过调整参数,可以控制分布的偏向程度,本文设置 $\alpha=1$, $\beta=\frac{5t}{T}$ 。

定义 14: 改进徒步优化算法中,通过 Beta 分布生成扫描因子 SF 的表达式定义为:

$$SF = 1 + (3-1) \times \frac{x^{\alpha-1}(1-x)^{\beta-1}}{\int_0^1 u^{\alpha-1}(1-u)^{\beta-1} du} = \frac{2x^{\alpha-1}(1-x)^{\beta-1}}{B(\alpha,\beta)} + 1 \quad (19)$$

改进徒步优化算法在地形坡度上应用逃逸能量理论[7],可以动态平衡全局搜索和局部开发的能力。

定义 15: 改进徒步优化算法中,徒步者的速度 $w_{i,t}$ 的更新策略定义为:

$$w_{i,t} = \begin{cases} w_{i,t-1} + \gamma_{i,t}(\beta_{best} - \alpha_{i,t}\beta_{i,t}), & \text{if } rand < r_E \\ \gamma_{i,t}(\beta_{best} - \alpha_{i,t}\beta_{i,t}), & \text{else} \end{cases} \quad (20)$$

其中, r_E 表示平缓概率,计算方式为:

$$r_E = 0.1E + 0.6 \quad (21)$$

$$E = 2E_0 \left(1 - \frac{t}{T}\right) \quad (22)$$

$$E_0 = 2rand - 1 \quad (23)$$

为了使算法在算法后期不易陷入局部最优解,本文应用动态反向学习策略[8]。

定义 16: 改进徒步优化算法中,生成反向解的公式定义如下:

$$X_i^{do} = X_i + w \cdot rand \cdot (rand \cdot (up + lb - X_i) - X_i) \quad (24)$$

此处权重 w 取值为 1。在获取动态反向解后,根据反向解和当前解的优劣进行选择:

$$X_{i,t} = best(f(X_{i,t}), f(X_{i,t}^{do})) \quad (25)$$

定义 17: 在改进徒步优化算法的特征选择算法中,徒步者位置的二元化转换公式定义为:

$$x_{i,t}^j = \begin{cases} 1, & \text{if } Z(x_{i,t}^j) > 0.5 \\ 0, & \text{else} \end{cases} \quad (26)$$

其中 Z 表示双曲正切函数映射,其计算方式为:

$$Z(x_{i,t}^j) = \tanh(x_{i,t}^j) = \frac{e^{x_{i,t}^j} - e^{-x_{i,t}^j}}{e^{x_{i,t}^j} + e^{-x_{i,t}^j}} \quad (27)$$

在特征选择算法中,同时兼顾分类能力和特征子集长度两个指标,所以优化算法同时优化两个目标函数,采用分层优化法,依赖度为主函数,特征子集长度为次函数,两个函数的表达式为:

$$fit^1 = f^1(X_{i,t}) = \gamma_B^{\delta}(D) \quad (28)$$

$$fit^2 = f^2(X_{i,t}) = \frac{|C-B|}{|C|} \quad (29)$$

在算法迭代时，徒步者当前位置的更新机制为：

$$X_{i,t} = \begin{cases} X_{i,t}^{new}, & \text{if } fit_{new}^1 > fit^1 \vee (fit_{new}^1 = fit^1 \wedge fit_{new}^2 > fit^2) \\ X_{i,t}, & \text{else} \end{cases} \quad (30)$$

根据上述定义，本文提出基于改进徒步优化算法的特征选择算法(Improved Hiking Optimization Algorithm for Feature Selection Algorithm, IHOAFSA)，算法流程如表 1 所示。

Table 1. Improved Hiking Optimization Algorithm for Feature Selection Algorithm (IHOAFSA)

表 1. 基于改进徒步优化算法的特征选择算法

输入：邻域决策系统 $RDIS = (U, C \cup D, V, f)$ ，种群数 pop ，迭代数 T ；

输出：特征子集 B ；

1. 初始化：解空间维度 $d = |C|$ ；
2. 根据定义 12 生成徒步者群体 $H = \{h_1, h_2, \dots, h_{pop}\}$ ；
3. 如果 $t > T$ ，转步骤 8；
4. 对于 $\forall h_i \in H$ ，计算其适应度 $fit_{i,t}^1$ 和 $fit_{i,t}^2$ ；
5. 设在 $fit_{i,t}^1$ 最大的徒步者中， $fit_{i,t}^2$ 最大的徒步者为领队 h_{best} ，其位置为 X_{best} ；
6. 对于 $\forall h_i \in H$ ：
 - 6.1. 获取徒步者 h_i 上一时刻位置 $X_{i,t-1}$ ；
 - 6.2. 根据定义 13 生成地形倾斜角度 $\theta_{i,t}$ ；
 - 6.3. 根据定义 14 生成扫描因子 SF ；
 - 6.4. 根据定义 8 计算受地形影响速度 $w_{i,t-1}$ ；
 - 6.5. 根据定义 15 计算当前速度 $w_{i,t}$ ；
 - 6.6. 计算新的当前位置 $X_{i,t}$ ，限制在上下界 lb 和 ub 之间；
 - 6.7. 计算适应度 fit^1 和 fit^2 ；
 - 6.8. 根据定义 16 计算动态反向解 $X_{i,t}^{do}$ ，限制在上下界 lb 和 ub 之间；
 - 6.9. 计算动态反向解的适应度 fit^1 和 fit^2 ；
 - 6.10. 对徒步者当前位置进行更新；
7. $t = t + 1$ ，转步骤 3；
8. 对最优个体位置 X_{best} 进行二分化得到特征子集 B ；
9. 返回特征子集 B 。

4. 实验分析

为了验证所提出算法 IHOAFSA 的有效性，本节与算法 PSORSFS [9]、FSARSR [10]、FSRSWOA [11] 进行对比实验，实验统一设置邻域半径 $\delta = 0.1$ 、迭代次数 $T = 50$ ，种群数 $pop = 20$ ，设置算法 IHOAFSA 的阈值 $\beta = 0.8$ ，本节实验的数据集来源为 UCI 数据集(<https://archive.ics.uci.edu/datasets/>)，包括 8 个数据集，数据集信息如表 2 所示，表中分别标注了数据集的名称、缩写，后续以缩写表示数据集，样本数即元素数，属性数即特征数，类别数为论域以决策类的划分种类数，实验前预先将数据集进行最小最大归一化到 $[0,1]$ 之间。由于优化算法的结果具有随机性，本实验将每个数据集在各个算法上分别独立运行 5 次，在 SVM 分类器和 KNN 分类器上以十倍交叉验证的方式记录每一次特征子集的分类精度，求 5 次分类精度的平均分类精度以及分类精度的标准差，实验结果如表 3、表 4 以及图 1、图 2 所示。

实验采用 Inter Core i5-12490F 4.6GHz 处理器、16GB 内存和 window11 操作系统，算法的实现代码的编程语言为 Python 3.11，在开发工具 PyCharm 2024.3 上编译运行。

Table 2. UCI datasets information
表 2. UCI 数据集信息

序号	数据集	缩写	样本数	属性数	类别数
1	Breast Cancer	Breast	286	9	2
2	Cervical	Cervical	88	17	2
3	Fertility	Fertility	72	19	2
4	Forty Soybean Cultivars	Forty	100	9	2
5	Image Segmentation	Image	210	19	7
6	Iris	Iris	150	4	3
7	Wine	Wine	178	13	3
8	Zoo	Zoo	101	16	7

Table 3. Classification accuracy under SVM classifier (%)
表 3. SVM 分类器下的分类精度(%)

数据集	PSORSFS	FSARSR	FSRSWOA	IHOAFSA
Breast	72.37 ± 0.52	73.23 ± 1.54	72.16 ± 0.17	72.30 ± 0.00
Cervical	74.82 ± 1.98	81.43 ± 1.34	83.79 ± 3.86	83.61 ± 6.55
Fertility	88.00 ± 0.00	88.00 ± 0.00	88.00 ± 0.00	88.00 ± 0.00
Forty	79.75 ± 0.64	79.88 ± 0.70	78.62 ± 1.68	80.25 ± 0.85
Image	84.38 ± 2.78	85.71 ± 3.61	84.86 ± 3.05	88.86 ± 0.23
Iris	96.67 ± 0.00	96.67 ± 0.00	96.67 ± 0.00	96.67 ± 0.00
Wine	90.14 ± 3.40	90.57 ± 4.77	91.35 ± 3.52	91.31 ± 2.59
Zoo	84.93 ± 2.88	90.89 ± 1.47	86.91 ± 5.26	91.85 ± 1.68
Average	83.88	85.80	85.30	86.61

Table 4. Classification accuracy under KNN classifier (%)
表 4. KNN 分类器下的分类精度(%)

数据集	PSORSFS	FSARSR	FSRSWOA	IHOAFSA
Breast	71.71 ± 1.84	70.95 ± 1.28	72.25 ± 1.76	72.94 ± 0.00
Cervical	76.39 ± 2.44	83.32 ± 4.90	81.57 ± 5.41	84.96 ± 6.41
Fertility	85.20 ± 0.75	85.20 ± 1.72	84.80 ± 1.17	85.60 ± 0.80
Forty	71.81 ± 4.53	69.94 ± 3.58	66.69 ± 0.15	71.88 ± 1.49
Image	81.90 ± 1.51	83.43 ± 2.44	82.38 ± 2.02	84.76 ± 0.67
Iris	95.33 ± 0.00	95.33 ± 0.00	95.33 ± 0.00	95.33 ± 0.00
Wine	91.29 ± 2.56	90.67 ± 3.77	91.12 ± 3.48	92.07 ± 2.50
Zoo	88.73 ± 2.01	90.07 ± 1.65	87.56 ± 1.72	91.24 ± 1.44
Average	82.80	83.61	82.71	84.85

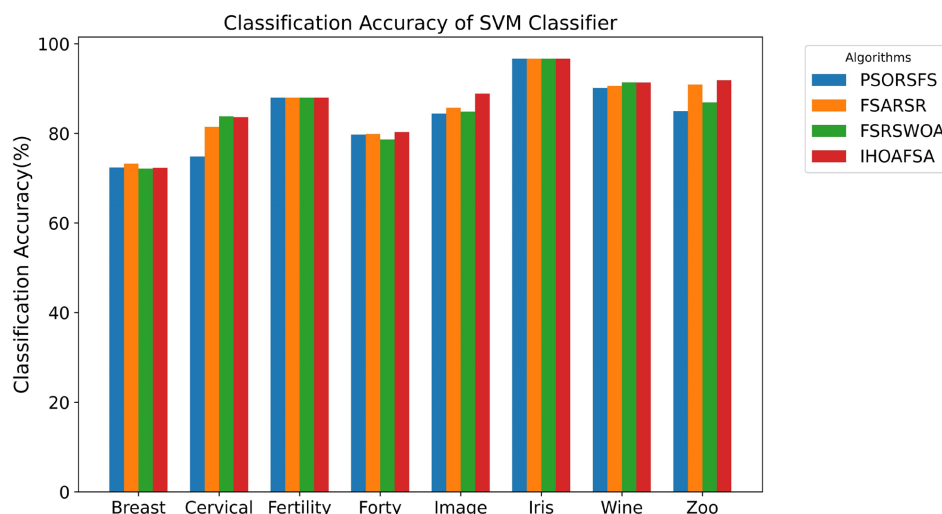


Figure 1. Comparison of classification accuracy under SVM classifier

图 1. SVM 分类器下的分类精度对比图

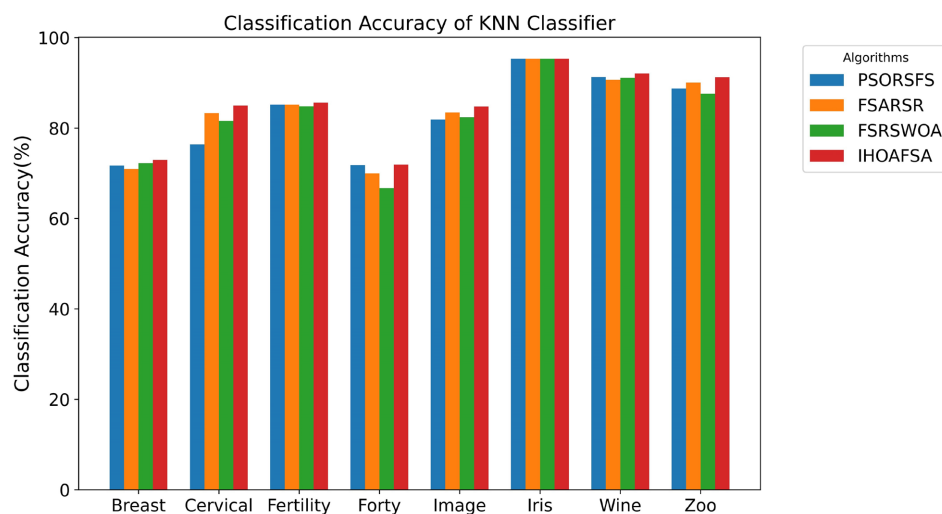


Figure 2. Comparison of classification accuracy under KNN classifier

图 2. KNN 分类器下的分类精度对比图

根据实验结果所示, 本文所提算法在 SVM 分类器和 KNN 分类器的大部分数据集上具有最高的平均分类精度, 并且算法 IHOAFSA 具有最高的平均分类精度的平均值, 实验结果表明了本文所提算法在分类精度上对比其他算法具有一定优势, 表明了算法的可行性和有效性。

5. 结论

本文介绍了变精度邻域粗糙集和徒步优化算法, 针对徒步优化算法无法动态调整局部和全局寻优能力的问题, 采用多策略改进徒步优化, 结合变精度邻域粗糙集提供的属性集重要度量, 提出基于改进徒步优化算法的特征选择算法。通过在 UCI 数据集上的对比实验验证了本文所提算法的可行性和有效性。

基金项目

本文受烟台市科技计划项目(编号: 2022XDRH016)的资助。

参考文献

- [1] Pawlak, Z. (1982) Rough Sets. *International Journal of Computer & Information Sciences*, **11**, 341-356. <https://doi.org/10.1007/bf01001956>
- [2] Hu, Q., Yu, D., Liu, J. and Wu, C. (2008) Neighborhood Rough Set Based Heterogeneous Feature Subset Selection. *Information Sciences*, **178**, 3577-3594. <https://doi.org/10.1016/j.ins.2008.05.024>
- [3] Oladejo, S.O., Ekwe, S.O. and Mirjalili, S. (2024) The Hiking Optimization Algorithm: A Novel Human-Based Metaheuristic Approach. *Knowledge-Based Systems*, **296**, Article ID: 111880. <https://doi.org/10.1016/j.knosys.2024.111880>
- [4] Hua, Z., Zhou, Y. and Huang, H. (2019) Cosine-Transform-Based Chaotic System for Image Encryption. *Information Sciences*, **480**, 403-419. <https://doi.org/10.1016/j.ins.2018.12.048>
- [5] Krithikadatta, J. (2014) Normal Distribution. *Journal of Conservative Dentistry*, **17**, 96-97. <https://doi.org/10.4103/0972-0707.124171>
- [6] McDonald, J.B. and Xu, Y.J. (1995) A Generalization of the β Distribution with Applications. *Journal of Econometrics*, **66**, 133-152. [https://doi.org/10.1016/0304-4076\(94\)01612-4](https://doi.org/10.1016/0304-4076(94)01612-4)
- [7] Heidari, A.A., Mirjalili, S., Faris, H., Aljarah, I., Mafarja, M. and Chen, H. (2019) Harris Hawks Optimization: Algorithm and Applications. *Future Generation Computer Systems*, **97**, 849-872. <https://doi.org/10.1016/j.future.2019.02.028>
- [8] Xu, Y., Yang, Z., Li, X., Kang, H. and Yang, X. (2020) Dynamic Opposite Learning Enhanced Teaching-Learning-Based Optimization. *Knowledge-Based Systems*, **188**, Article ID: 104966. <https://doi.org/10.1016/j.knosys.2019.104966>
- [9] Wang, X., Yang, J., Teng, X., Xia, W. and Jensen, R. (2007) Feature Selection Based on Rough Sets and Particle Swarm Optimization. *Pattern Recognition Letters*, **28**, 459-471. <https://doi.org/10.1016/j.patrec.2006.09.003>
- [10] Chen, Y., Zhu, Q. and Xu, H. (2015) Finding Rough Set Reducts with Fish Swarm Algorithm. *Knowledge-Based Systems*, **81**, 22-29. <https://doi.org/10.1016/j.knosys.2015.02.002>
- [11] 王生武. 粗糙集中基于鲸鱼算法和模糊决策的特征选择方法研究[D]: [硕士学位论文]. 成都: 西南交通大学, 2020. <https://doi.org/10.27414/d.cnki.gxjnu.2020.000780>