基于邻域粗糙集的局部特征选择

摘要

特征子集的选择被认为是模式识别、机器学习和数据挖掘的一个重要的预处理步骤。大多数研究都集中在处理同质特征选择，即数值特征或分类特征。《基于邻域粗糙集的异构特征子集选择》这篇文章引入了一个邻域粗糙集模型来处理异构特征子集的选择问题，该邻域模型通过为不同类型的属性分配不同的阈值来减少数值特征和分类特征。在该模型中，决策的邻域上下近似值的大小反映了特征子集的识别能力，较低近似的大小被计算为决策和条件属性之间的依赖性，作者利用邻域依赖性来评估异构特征子集的重要性，构造前向特征子集选择算法，并将该算法与一些经典算法进行了比较。实验结果表明，基于邻域模型的方法对异构数据的处理更加灵活。同时，我通过 Matlab 软件对这篇论文的模型和算法都进行了复现，得到了较为理想的结果。最后在此基础上做出了创新，将邻域粗糙集和局部特征选择继续了有效结合，不仅解决了异构特征子集选择和属性缩减的问题，还避免了单纯依赖全局特征子集的选择可能会影响模型的识别精度等问题。

Keywords: 特征子集选择，邻域粗糙集，局部特征选择

1 引言

随着信息获取和存储技术的飞速发展，现代数据集的特征维度呈现出爆炸式增长，从数十个特征到数百、数千甚至更多的特征，数据的复杂性也在不断增加，这些海量的特征不仅带来了丰富的信息，也带来了更大的计算和存储挑战。首先，特征数量的增加会导致数据冗余和噪声的增加，从而使得数据分析变得更加复杂。冗余特征可能重复传递相似的信息，而噪声特征则可能包含不可靠或无效的数据，这些都会干扰模型的准确性和鲁棒性。其次，随着特征数量的增加，数据的维度也随之上升，进而引发了所谓的“维度灾难”。维度灾难是指随着维度的扩展，数据空间变得极度稀疏，许多机器学习算法因此难以有效运作。这不仅增加了模型训练的时间，还可能导致过拟合问题，进而降低模型的解释性和预测精度。

特征选择是一项关键的数据预处理技术，旨在从原始特征集合中筛选出最具预测能力的特征，以提升机器学习模型的性能和效率。通过有效的特征选择，不仅能够显著降低计算和存储成本，还能去除冗余或无关特征，从而增强模型的可解释性和稳定性。此外，特征选择还能够减少特征空间的维度，优化数据处理流程，提升学习算法的训练速度和预测准确性。同时，特征选择有助于缓解过拟合问题，增强模型的泛化能力，确保其在面对未见数据时依然能够保持良好的性能。

近年来，特征选择技术取得了显著进展，通常可分为符号法和数值法两大类。符号法将所有特征视为分类变量，典型代表是基于粗糙集理论的属性约简方法，该方法通过分析特征间的依赖关系来筛选出关键特征。而数值法则将特征视为实值变量，特别适用于连续数据，主

要通过在实数空间中操作变量来进行特征选择。当数据中同时存在分类特征和实值等异构特征时，符号法通常会引入离散化算法，将实值变量的值域划分为多个区间，并将这些区间视为符号特征。数值方法则通过隐式或显式地将分类特征编码为一系列整数，并将其视为数值变量。

显然，数值属性的离散化可能会导致信息丢失，因为它未考虑数值与离散值之间的隶属度。在离散化过程中，至少有两类结构信息会丧失：一是邻域结构，二是真实空间中的有序结构。另一方面，在数值方法中，使用欧氏距离衡量分类属性之间的相似性或差异性是不合适的。为了解决混合特征集的问题，研究者们提出了一些异构距离函数。然而，从异构数据中选择特征的方法尚未得到充分研究，仍需进一步探索。

Pawlak 提出的粗糙集模型，通过等价关系将集合划分为互斥的等价类，从而形成元素概念，这种方法特别适用于处理具有标称属性的数据集。然而，在数值数据的空间中，邻域的概念极其关键。通过邻域关系，可以从具有数值特征的宇宙中生成邻域颗粒族，并利用这些邻域颗粒来近似决策类。因此，基于邻域关系的粗糙集模型为处理具有异构属性的数据提供了一个有效的工具。基于这一理论，胡清华教授等人在论文《基于邻域粗糙集的异构特征子集选择》中提出了一种新颖的模型，该模型基于邻域粗糙集理论，定义了异构特征与决策之间的依赖关系，并构建了异构数据的属性显著性度量，还提出了一系列属性约简算法，旨在解决异构特征子集选择和属性缩减的问题。

特征选择的核心目标是在保证精度不下降的前提下，尽可能减少特征的数量。首先，特征选择通常将不同特征组合所提供的判别信息视为一个组合优化问题，通过解决这一优化问题来评估不同特征组合对模型的影响。其次，传统的特征选择方法通常寻求一个全局最优的特征子集，这一方法适用于整个样本空间。然而，不同样本区域可能存在局部特征子集，即每个局部区域可能与其独特的优化特征集相关联。因此，单纯依赖全局特征子集的选择可能会影响模型的识别精度、计算效率，并限制后续研究成果的应用。基于以上考虑，本研究分析了不同局部区域内特征之间的关系，以及特征与分类之间的关联，旨在设计高效的局部区域特征子集搜索策略和合理的局部特征子集评估函数，进而构建一种新型的局部特征选择算法，为从大数据中挖掘有价值的信息提供了合理且可行的途径。

2 相关工作

2.1 特征选择

特征选择技术近年来得到了迅速发展，并且已经涵盖了多种方法，主要包括过滤方法、包装方法、嵌入式方法，以及无监督和监督学习方法。过滤方法的核心思想是通过评估单个特征与目标变量之间的相关性来选择特征，而不考虑特征之间的相互作用。这种方法通常计算简单，效率较高，但忽视了特征间的潜在依赖关系，可能导致信息丢失。包装方法则将特征选择与学习算法紧密结合，旨在通过搜索最优特征子集来提升模型的整体性能。该方法能够充分利用特征间的相互关系，但在特征空间非常广阔的情况下计算开销较大。嵌入式方法则是在模型训练过程中自动进行特征选择，常借助正则化技术（如 L1、L2 正则化）来实现。此方法能够有效地进行特征筛选，并且通过训练过程同时优化模型和特征子集。无监督特征选择侧重于挖掘数据的内在结构，通过聚类、降维等方法发现数据中的潜在模式，而监督特征选择则通过标签信息引导特征的选择，通常能获得更高的分类性能。

2.2 粗糙集

粗糙集理论是由 Pawlak 提出的，这一理论已经被证明在分类数据中的特征选择、规则提取和知识发现方面具有重要的应用价值。为了更好地处理模糊信息，Shen 和 Jensen 提出了一种模糊-粗糙快速简化算法，将经典粗糙集模型中的依赖函数推广到模糊环境中。这一算法有效地适应了数据中的模糊性，提高了粗糙集在模糊数据中的应用性能。然而，尽管这一算法能够较好地处理模糊数据，但传统的粗糙集理论和简化算法在面对大规模和高维复杂数据集时仍然面临计算效率低下和收敛性差的问题。Bhatt 和 Gopal 在这一背景下发现，传统的快速还原算法在许多实际数据集上存在不收敛的情况，因此他们提出了“紧凑计算域”上模糊粗糙集的概念。这一方法通过引入紧凑的计算域来减少冗余信息，从而有效地减少计算量，显著提高了算法在复杂数据集上的计算效率和稳定性。紧凑计算域的引入不仅优化了粗糙集理论的应用范围，也为处理大规模数据集提供了一种有效的解决方案。进一步的研究中，胡等人对香农熵在模糊集中的应用进行了扩展，提出了一种基于熵度量的模糊信息量计算方法，用于评估模糊近似空间的不确定性。该方法通过数值属性诱导模糊关系，结合符号特征生成脆关系，并利用广义信息熵来衡量引入的信息量。基于这一方法，研究者能够在特征选择和特征子集评估中更精确地捕捉到数据中的潜在规律。该研究不仅增强了模糊集理论的应用深度，还通过信息熵度量实现了更加细致的特征选择，在处理异构数据集时能够有效减少不确定性和噪声的影响。尽管如此，从数值属性中生成模糊等价关系仍然是一个计算量巨大的过程，尤其是在大规模数据集上，这一过程往往会带来显著的计算开销。此外，如何在不同分类任务中生成有效的模糊关系，尤其是在面对高维度和复杂数据集时，仍然是一个亟待解决的开放问题。在实际应用中，优化模糊等价关系的生成过程，减少计算复杂度，并确保其在不同分类任务中的有效性，仍然是未来研究的关键挑战之一。

3 本文方法

3.1 本文方法概述

本文旨在实现并优化论文《基于邻域粗糙集的异构特征子集选择》中提出的邻域粗糙集模型及其特征选择算法，以提高数据分析和分类任务中的特征选择效率。首先，针对数据集中的每个样本，构建其邻域，采用欧几里得距离度量并通过设置适当的阈值来实现邻域的划分。接着，计算特征子集在邻域内的上下近似，并通过比较下近似与全集的大小比值来评估该特征子集的依赖度，从而衡量特征在描述数据集中的样本类属关系方面的有效性。在特征选择过程中，定义了一种基于依赖度的评估函数，用于衡量每个特征的相对重要性。算法通过迭代优化的方式，逐步选择能够最大化数据依赖度的特征，直至满足停止条件为止，从而获得最具代表性的特征子集。此外，为了使该方法适应不同学习算法的需求，引入了 Delta 值作为控制参数，用以调节邻域粗糙集模型的粒度。通过调整 Delta 值，可以生成多个不同粒度的特征子集，进而在特征选择的过程中更好地平衡计算效率与分类性能。在实验部分，通过对 CART 和线性 SVM 两种经典学习算法进行评估，考察了不同特征子集对分类性能的影响。最后，本文提出了一种创新策略，将邻域粗糙集理论融入局部特征选择框架。首先将整个样本空间划分为多个局部区域，在每个区域内应用上述基于邻域粗糙集的特征选择算法，选择出每个局部区域的最优特征子集，然后使用现有 LFSDC 中应用的分类方法来进行评估最

终的分类性能。该方法不仅能够提高特征选择的精度，还能适应不同类型的数据和任务需求，从而进一步优化分类效果。

3.2 邻域粗糙集

经典粗糙集理论在处理复杂数据时存在一定局限性，需要将连续数据离散化后才能进行分析和推理，但这一过程中可能会丢失部分信息，无法全面反映数据的特征。为解决这一问题，胡清华教授在经典粗糙集理论基础上，引入了邻域概念，并提出了邻域粗糙集理论模型。该模型通过计算条件属性间的距离，并构造邻域关系，能够同时处理离散数据和连续数据，避免了信息丢失问题，更加准确地表达数据的内在规律。以下是邻域粗糙集理论基础知识的简要介绍。

定义 1 (度量) 给定一个 N 维实数空间 Ω, $\Delta = R^N \times R^N \to R$, 若 Δ 满足以下条件:

1. $\Delta(x_1, x_2) \geq 0$，当 $x_1 = x_2$ 时，Δ 值为 0，$\forall x_1, x_2 \in R^N$;
2. $\Delta(x_1, x_2) = \Delta(x_2, x_1), \forall x_1, x_2 \in R^N$;
3. $\Delta(x_1, x_3) \leq \Delta(x_1, x_2) + \Delta(x_2, x_3), \forall x_1, x_2, x_3 \in R^N$;

则称 Δ 为 R 上的一个度量。其中，(Ω, Δ) 称为度量空间，$\Delta(x_i, x_j)$ 为一个距离函数，表示样本 x_i 和样本 x_j 之间的距离。

定义 2 (δ-邻域) 假定在 N 维实数空间 Ω 上论域为 $U = \{x_1, x_2, \ldots, x_n\}$, n 表示论域中对象的个数，则对于任意对象 $x_i \in U$ 的 δ-邻域定义为:

$$\delta(x_i) = \{x \mid x \in U, \Delta(x, x_i) \leq \delta\} \tag{3-1}$$

其中，$\delta > 0$, $\delta(x_i)$ 表示由对象 x_i 生成的邻域粒子。

定义 3 (邻域关系) 假定在 N 维实数空间 Ω 上论域为 $U = \{x_1, x_2, \ldots, x_n\}$，$n$ 表示论域中对象的个数，$B \subseteq C$，对于样本 $x_i \in U$，由属性集合 B 生成在 U 上的邻域关系 N_B 定义为:

$$N_B = \{(x_i, x_j) \in U \times U \mid x_j \in \delta_B(x_i)\} \tag{3-2}$$

定义 4 (邻域决策信息系统) 一个邻域决策信息系统 NDS 可由 $NDS = \langle U, A = C \cup D, V, f, N\rangle$ 表示，其中 $U = \{x_1, x_2, \ldots, x_n\}$ 是由对象组成的非空有限集合，称为论域，n 为对象的个数；A 表示所有属性组成的集合，C 表示条件属性集合，D 表示决策属性集合，且 $C \cap D = \emptyset$；V 表示 A 中所有属性的值域；f 表示 $U \times A \to V$ 的映射；N 表示属性集合 C 生成在论域 U 上的邻域关系。

定义 5 (自适应邻域) 给定一个邻域决策信息系统 $NDS = \langle U, A = C \cup D, V, f, N\rangle$，$U = \{x_1, x_2, \ldots, x_n\}$，$B \subseteq C$。对于目标对象 $x_i \in U$，假定 $N_B(x_i) = \{x_{i,1}, x_{i,2}, \ldots, x_{i,j}, \ldots, x_{i,n-1}\}$ 为 x_i 与论域 U 中其余对象之间的距离排序集合，且对象之间的距离需满足 $\Delta(x_i, x_{i,1}) \leq \Delta(x_i, x_{i,2}) \leq \ldots \leq \Delta(x_i, x_{i,j}) \leq \ldots \leq \Delta(x_i, x_{i,n-1})$，那么对象 x_i 与任意对象 $x_{i,k} \in U$ 之间的密度定义为：$Density(x_i, x_{i,k}) = \Delta(x_i, x_{i,k})/k$。从对象 x_i 到 $x_{i,n-1}$，假定 x_i 与其余对象之间的密度值第一次呈现变化趋势的对象 $x_{i,k}$，称为拐点对象。那么，在属性集合 B 下，对象 x_i 的自适应邻域 $IP_B(x_i)$ 定义为:

$$IP_B(x_i) = \{x_{i,1}, x_{i,2}, \ldots, x_{i,k-1}\} \tag{3-3}$$

其中，$IP_B(x_i)$ 内的对象为在集合 $N_B(x_i)$ 下，x_i 与拐点对象 $x_{i,k}$ 之间的对象集合。

定义 6（上、下近似集） 给定一个邻域决策信息系统 $NDS = \langle U, A = C \cup D, V, f, N\rangle$，$B \subseteq C$，$\forall X \subseteq U$，$\delta_B(x)$ 表示对象 x 在属性集合 B 上的 δ-邻域，N_B 表示属性集合 B 生成在 U 的邻域关系，那么 X 关于 B 的上近似集 $(\overline{N_BX})$、下近似集 $(\underline{N_BX})$ 分别定义为：

$$\overline{N_BX} = \{x_i \mid \delta_B(x_i) \cap X \neq \emptyset, x_i \in U\} \tag{3-4}$$

$$\underline{N_BX} = \{x_i \mid \delta_B(x_i) \subseteq X, x_i \in U\} \tag{3-5}$$

与经典粗糙集理论的定义类似，同样定义 X 关于 B 的正域 $POS_B(X)$、负域 $NEG_B(X)$ 及边界域 $BND_B(X)$ 分别定义为：

$$POS_B(X) = \underline{N_BX} \tag{3-6}$$

$$BND_B(X) = \overline{N_BX} - \underline{N_BX} \tag{3-7}$$

$$NEG_B(X) = U - \overline{N_BX} \tag{3-8}$$

邻域粗糙集可由图 1 展示。

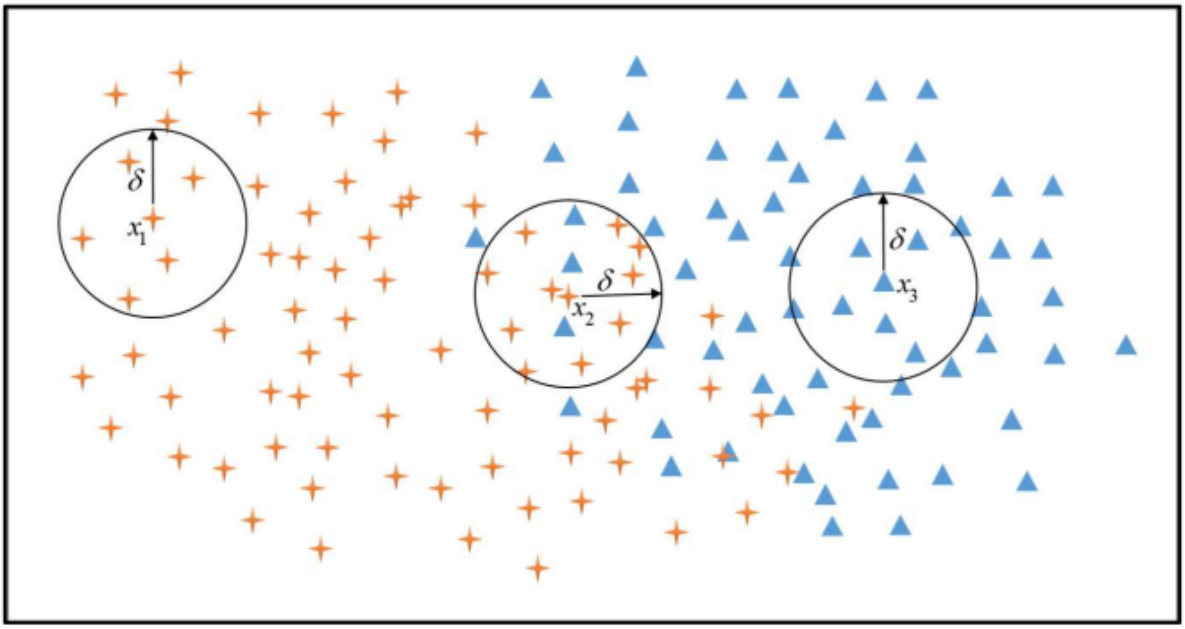

图 1. 邻域粗糙集示意图

定义 7（依赖度） 给定一个邻域决策信息系统 $NDS = \langle U, A = C \cup D, V, f, N\rangle$，$B \subseteq C$，则决策属性 D 相对于条件属性集合 B 的依赖度定义为：

$$\gamma_B(D) = \frac{|POS_B(D)|}{|U|}, \quad 0 \leq \gamma_B(D) \leq 1 \tag{3-9}$$

定义 8 给定一个邻域决策系统 $NDS = \langle U, A = C \cup D, V, f, N\rangle$，$B \subseteq C$，$b \in B$，若 $\gamma_B(D) = \gamma_{B-\{b\}}(D)$，则表示 b 是属性集合 B 中不必要的属性；若 $\gamma_B(D) > \gamma_{B-\{b\}}(D)$，则表示 b 是属性集合 B 中的必要属性。如果 $\forall b \in B$，对集合 D 都是必要的，那么则认为条件属性集合 B 相对于决策属性 D 独立。

定义 9 给定一个邻域决策系统 $NDS = \langle U, A = C \cup D, V, f, N\rangle$，$B \subseteq C$，若属性集合 B 是条件属性集合 C 相对于决策属性集合 D 的一个约简，则 B 需满足如下条件：

1. $\gamma_B(D) = \gamma_C(D)$;

2. 对 $\forall b \in B$，有 $\gamma_B(D) > \gamma_{B-\{b\}}(D)$。

3.3 特征选择算法

定义 10 (属性重要度) 给定一个邻域决策系统 $NDS = \langle U, A = C \cup D, V, f, N\rangle$, $B \subseteq C$, $a \in B$，将属性 a 在属性集合 B 中相对于决策属性 D 的属性重要度定义为：

$$\mathrm{Sig}(a, B, D) = \gamma_B(D) - \gamma_{B-\{a\}}(D), \quad 0 \leq \mathrm{Sig}(a, B, D) \leq 1 \tag{3-10}$$

如果 $\mathrm{Sig}_1(a, B, D) = 0$，那么特征 a 在 B 中相对于 D 是多余的；否则，a 在 B 中是不可或缺的。

基于粗糙集的特征选择算法旨在找到一个特征子集，该子集不仅能够保持与原始数据相同的区分能力，还能剔除所有冗余特征。借助所提出的度量方法，可以设计一个前向贪心搜索算法来进行特征约简。该算法包含四个关键步骤：子集生成、子集评价、停止准则和结果验证。在算法中，从一个空的属性集合开始，每轮通过添加一个特征，使该特征能够最大化红色集中的依赖性增量，这是子集生成的核心策略。接着，通过最大化依赖性增量对每个子集进行评价，并将这一评价嵌入特征选择过程中。最后，算法会继续添加特征，直到依赖性增量小于预设的阈值时才会停止。

3.4 局部特征选择

对于一个局部特征选择问题，该算法会采用区域划分策略将所有样本划分成 N 个区域 (其中 N 为样本个数)。然后对于每个样本，将其视为中心样本，根据一个不纯度计算得出相应的半径，建立起一个超球体作为其局部区域。这个过程主要有两步：首先，根据公式 (3-11) 确定超球 $Q^{(i)}$ 所代表的是样本 i 的局部区域。

$$\begin{cases} \min \delta(Q^{(l)}) \\ s.t.\ v(Q^{(l)}) \geq \gamma \end{cases} \tag{3-11}$$

其中，$\delta(\cdot)$ 表示超球中的样本数，$v(\cdot)$ 用于计算不纯度，其定义为超球体中具有不同类别标签的样本数量与具有相同标签的样本数量之比。γ 是一个常数，代表不纯度水平。其次，将中心样本与超球中其他样本的最大距离设定为局部区域 i 的半径，由公式 (3-12) 表示。

$$r^{(i)} = \max_{j \in Q^{(i)}, j \neq i} \left(D(x^{(i)}, x^{(j)}) \right) \tag{3-12}$$

其中 $r^{(i)}$ 是局部区域 i 的半径，$x^{(i)}$ 代表局部区域 i 的中心样本，j 表示样本在超球 $Q^{(i)}$ 内的索引。其中，$D(x^{(i)}, x^{(j)})$ 表示两个样本 $x^{(i)}$ 和 $x^{(j)}$ 之间的欧氏距离。

然后使用上述基于邻域粗糙集的特征子集算法去选择每一个局部区域的最优特征子集。由于局部特征选择会产生了多个局部特征子集，直接使用与传统特征选择方案一致的分类方法是不合适的。为了测试新样本 x^p 的标签，将计算样本的相似度 $S(x^p, c_k)$，新样本 x^p 与标签 c_k 的相似度可以由超球体 $Q^{(i)}$ 类标签为 c_k 的包含 x^p 的个数表示。具体表达式如公式 (3-13) 所示：

$$S(x^p, c_k) = \frac{\sum_{i \in Y_{c_k}} \psi_i(x^p, r^{(i)})}{\sum_{Y_{c_k}}} \tag{3-13}$$

其中 Y_{c_k} 是属于 c_k 的训练样本集，$\psi(\cdot)$ 用于检查新样本 x^p 是否落入 $Q^{(i)}$ 的局部区域，其类别标签 c_k 和最近的邻居样本都属于类别 c_k。如果是这样，那么 $\psi(\cdot) = 1$，否则 $\psi(\cdot) = 0$。在计算完 x^p 与所有类别的相似度后，新样本 x^p 的类别为具有最大相似度的类别。

4 复现细节

4.1 与已有开源代码对比

由于论文《基于邻域粗糙集的异构特征子集选择》未提供开源代码，本文根据论文中的详细描述和伪代码，完成了算法的完整复现。在复现过程中，深入学习了论文中提出的邻域粗糙集模型、特征选择方法及相关评估函数，并基于这些理论构建了相应的代码框架。随后，按照实验设计逐步实现了算法的各个功能模块，确保复现结果与论文中的方法一致。

4.2 实验环境搭建

在此次实验中，准备了一台配备 Intel Core i5 处理器、8GB RAM 及充足硬盘空间的计算机，在计算机上安装了 Windows 10 操作系统，并下载并安装了 Matlab 2020b 版本，从 UCI 机器学习库获取了所需的数据集，最终在本地计算机上进行实验。

4.3 界面分析与使用说明

Matlab 提供了一个直观且功能强大的图形用户界面（GUI），极大地方便了用户进行实验操作、调整参数和查看结果，其界面包含多个关键组成部分。首先，命令窗口是用户与 Matlab 交互的核心区域，允许用户直接输入和执行 Matlab 命令，实现实时计算和测试，能够即时反馈结果，便于快速调试；其次，编辑器用于编写、编辑和运行 Matlab 脚本与函数，具备语法高亮、自动补全等智能功能，提升了代码编写效率，还支持调试模式，帮助用户逐步执行代码并检查变量状态，确保代码的正确性；工作区显示当前会话中的所有变量及其值，用户可以实时监控和修改数据，方便对实验过程和结果进行深入分析；最后，图形窗口则用于展示绘图结果和交互式图表，支持多种数据可视化形式，如 2D 和 3D 图形、散点图、曲线图等，帮助用户直观地分析数据趋势并进行交互式探索。通过将这些功能模块有效结合，Matlab 的图形用户界面为用户提供了一个高效、便捷的实验平台，显著提升了数据分析、算法开发和结果展示的整体效率，点击运行键即可启动代码，具体如图 2 所示。

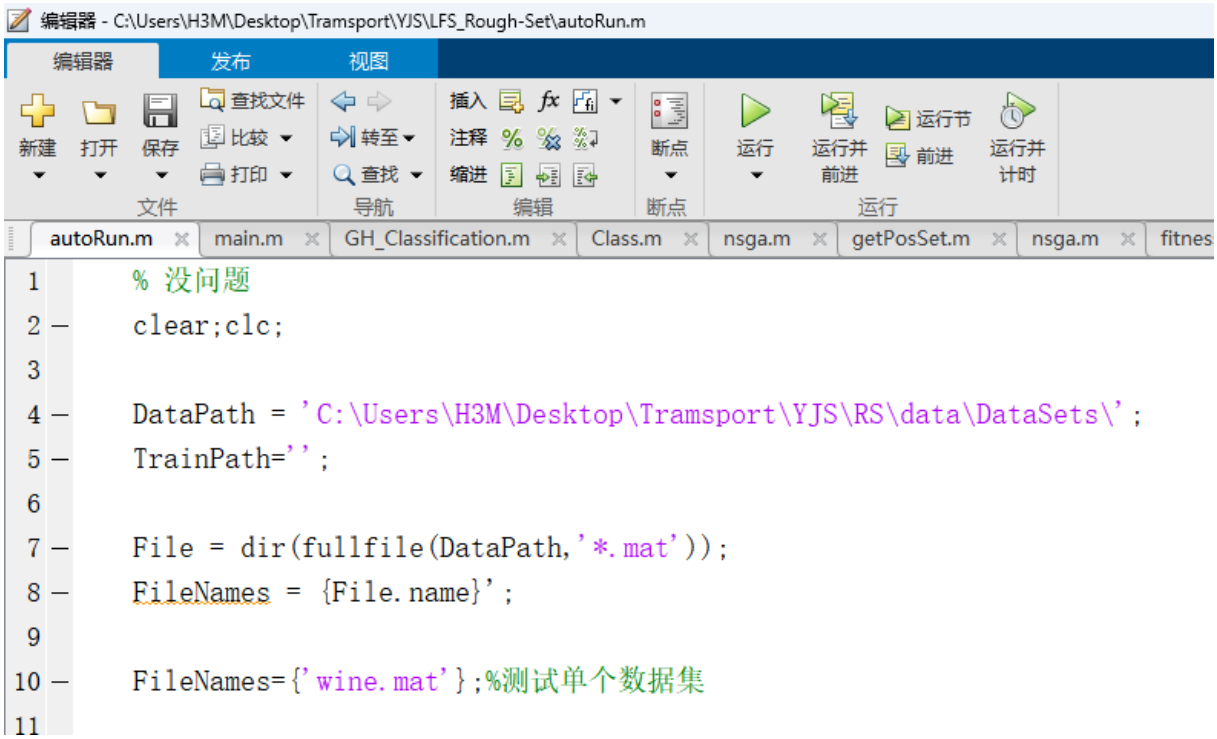

图 2. Matlab 界面分析

4.4 创新点

本研究的创新之处在于提出了一种基于邻域粗糙集的局部特征选择方法，打破了传统特征选择方法依赖全局优化的局限。传统方法通常侧重于寻找适用于整个样本空间的最优特征子集，但往往忽视了不同样本区域之间可能存在的差异，这导致无法充分挖掘局部特征的独特性，从而影响模型的精度和效率。相比之下，局部特征选择算法能够针对不同样本区域的特征分布和分类需求，动态调整并选择最具判别力的特征，从而提升分类模型的准确性和运算效率。此外，结合邻域粗糙集的局部特征选择方法，还能够有效处理异构特征的问题，进一步增强了模型的适应性和鲁棒性。

5 实验结果分析

通过上述模型和方法的介绍，我使用 Matlab 软件对代码进行了复现。根据论文中提供的数据集链接，我下载了六个数据集：Anneal、Credit、Iono、Sonar、Wdbc 和 Wine，并统计了它们的样本数量、特征数量和类别数量，在未进行特征选择的情况下，使用 CART 和 SVM 算法在十折交叉验证的情况得到分类准确度，具体如表 1 所示。

由于不同的学习算法以不同方式利用可用特征，换句话说，不同的学习算法可能需要不同的特征子集才能实现最佳的分类性能，因此在本研究中调整了邻域粗糙集模型中的 Delta 值，使 Delta 值在 0.02 到 0.4 之间，以 0.02 为步长变化，从而获得不同的特征子集。随后，使用基于 10 倍交叉验证的 CART 和线性 SVM 对选定的特征子集进行评估。表 2 展示了在不同数据集上，使用两种算法时，最大 Delta 值及其对应求出的特征子集。从表 2 可以看出，CART 和 SVM 在某些情况下选择的最佳特征子集相同，但在大多数情况下，它们是不同的。这表明，对于不同类型的学习任务和分类算法，并没有单一算法始终优于其他算法。在

表 1. CART 和 SVM 在不同数据集上的性能比较

Data	Sample	Feature	Class	CART	SVM
Anneal	798	6	5	99.89 ± 0.35	99.89 ± 0.35
Credit	690	6	2	82.73 ± 14.86	81.44 ± 7.18
Iono	351	34	2	87.55 ± 6.93	93.79 ± 5.08
Sonar	208	60	2	72.07 ± 13.94	85.10 ± 9.49
Wdbc	569	31	2	90.50 ± 4.55	98.08 ± 2.25
Wine	178	13	3	89.86 ± 6.35	98.89 ± 2.34

此，Delta 值作为控制分析粒度的参数，能够使算法在不同粒度层次上获得不同的属性，从而确定最适合描述不同学习算法识别问题的最佳特征子集。

表 2. 不同数据集的 CART 和 SVM 与 Delta 值的比较

Data	CART	Delta	SVM	Delta
Anneal	27,3,1	0.10	27,3,1	0.10
Credit	11,2,6,14,3,9	0.02	11,2,6,14,3,9	0.02
Iono	1,5,3,28,19,24,31,8,34,21,2,5, 30,32,4,11,7,23,9,18,22,26,10	0.40	1,5,19,4,30,34,25,8,3,7,14,12	0.20
Sonar	55,1,48,12,21,26,42,17,6	0.24	55,1,48,12,21,26,42,17,6	0.24
Wdbc	23,22,28,26,25,8	0.08	8,21,22,19,28,12,25,2,10,2,9, 27,23,7,16,1,11,15,6,26,30,18	0.26
Wine	10,7,1	0.02	13,10,11,1,12,5,2,7,3,4,8,9	0.32

在 Wine 数据集上，我分别使用 CART 算法和 SVM 算法，利用十折交叉验证在不同的 Delta 值下计算了最高分类精度，并绘制了相应的折线图，具体见图 3 和图 4。表 3 展示了在相同数据集下，CART 和 SVM 算法的分类准确度对比，分别列出了论文中报告的结果与通过复现代码得到的实验结果。

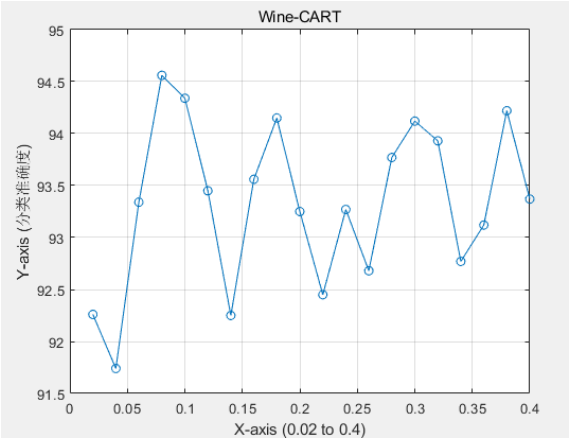

图 3. CART 算法

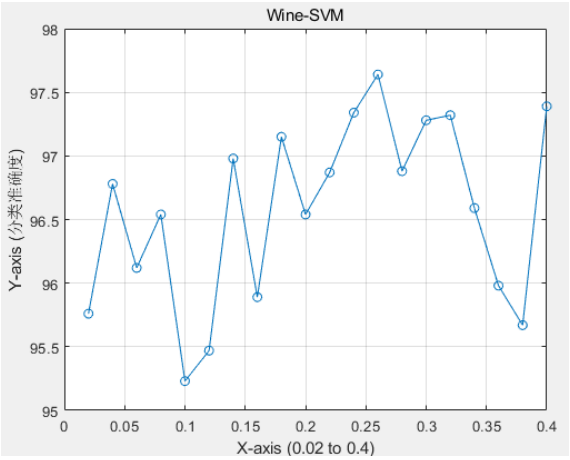

图 4. SVM 算法

最后，我将邻域粗糙集与局部特征选择相结合，并在相同数据集上采用十折交叉验证进

表 3. 论文和使用复现代码在 CART 和 SVM 的准确性比较

Data	CART 准确度 (论文)	CART 准确度 (复现)	SVM 准确度 (论文)	SVM 准确度 (复现)
Anneal	100 ± 0.0	100 ± 0.0	100 ± 0.0	100 ± 0.0
Credit	83.03 ± 18.51	82.96 ± 18.67	85.48 ± 18.51	85.39 ± 17.92
Iono	70.60 ± 5.23	70.31 ± 5.71	89.82 ± 5.62	89.48 ± 5.95
Sonar	77.83 ± 8.71	77.49 ± 8.62	93.89 ± 0.11	93.21 ± 0.37
Wdbc	94.72 ± 2.23	93.98 ± 2.17	97.73 ± 2.19	97.81 ± 2.52
Wine	93.26 ± 4.37	93.30 ± 4.48	98.89 ± 2.34	98.73 ± 2.29

行分类实验。尽管所提出的方法在分类性能上有所提升，但与改进前相比，性能提升幅度较小。具体结果见表 4。

表 4. 全局特征选择与局部特征选择的分类准确度比较

Data	全局特征选择的分类准确度	局部特征选择的分类准确度
Anneal	100 ± 0.0	100 ± 0.0
Credit	82.96 ± 18.67	85.24 ± 9.37
Iono	70.31 ± 5.71	77.14 ± 4.28
Sonar	77.49 ± 8.62	78.12 ± 7.34
Wdbc	93.98 ± 2.17	93.71 ± 3.01
Wine	93.30 ± 4.48	95.16 ± 3.92

6 总结与展望

在大多数情况下，减少冗余或不相关的特征有助于提高分类性能并降低分类成本。经典的粗糙集模型在特征选择和属性缩减方面得到了广泛的应用和讨论，但该模型主要适用于处理名义数据。为了解决这一局限性，论文《基于邻域粗糙集的异构特征子集选择》中提出了一种新的基于邻域粗糙集的异构特征选择方法。该方法设计了一个特征评估函数——邻域依赖关系，该函数反映了样本在决策阳性区域中的百分比。理论分析表明，特征的显著性随着特征子集的增加而单调递增，这一特性对于将评估函数集成到搜索策略中具有重要意义。在此基础上，作者还提出了一种贪婪特征选择算法，并通过实验验证了该方法能够有效处理分类属性和数值变量的非离散性问题，从而找到一个较小且有效的特征子集。实验结果还表明，邻域大小对特征选择的效果有显著影响。

在复现该实验过程中，我使用 Matlab 软件对论文中的模型和算法进行了实现，并取得了较为理想的效果。同时，我发现单纯依赖全局特征子集的选择可能会影响模型的识别精度、计算效率，并限制后续研究成果的应用，所以我进行了部分创新，将邻域粗糙集与局部特征选择进行了有效结合，但性能提升幅度较小。未来，我将继续深入研究这一方向，探索如何在邻域框架下进一步优化局部特征选择方法，为基于邻域的学习系统奠定坚实基础，并推动异构数据处理和特征选择技术的进一步发展。

参考文献

[1] Qinghua Hu, Daren Yu, Jinfu Liu, Congxin Wu. Neighborhood rough set based heterogeneous feature subset selection. *Information Sciences*, 178(18): 3577-3594, 2008.