软件学报 ISSN 1000-9825, CODEN RUXUEW
2025,36(3):1186−1201 [doi: 10.13328/j.cnki.jos.007158] [CSTR: 32375.14.jos.007158]

E-mail: jos@iscas.ac.cn
http://www.jos.org.cn
Tel: +86-10-62562563

面向动态混合数据的多粒度增量特征选择算法*

王 锋, 姚 珍, 梁吉业

(山西大学 计算机与信息技术学院, 山西 太原 030006)

通信作者: 王锋, E-mail: sxuwangfeng@126.com

摘 要: 在大数据时代, 样本规模以及维数的动态更新和变化极大地增加了计算负担, 在这些动态数据中, 大多的数据样本并不以单一的数据取值形式存在, 而是同时包含符号型数据和数值型数据的混合型数据. 为此, 学者们提出了许多关于混合数据的特征选择算法, 但现有的算法大多只适用静态数据或者小规模的增量数据, 无法处理大规模动态变化的数据, 尤其是数据分布不断变化的大规模增量数据集. 针对这一局限性, 通过分析动态数据中粒空间以及粒结构的变化和更新, 基于信息融合机制, 提出了一种面向动态混合数据的多粒度增量特征选择算法. 该算法重点讨论了动态混合数据中的粒空间构建机制、多数据粒结构的动态更新机制以及面向数据分布变化信息融合机制. 最后, 通过与其他算法在 UCI 数据集上的实验结果进行对比, 进一步验证了所提算法的可行性和高效性.

关键词: 动态混合数据; 数据分布变化; 多粒度计算; 信息融合

中图法分类号: TP18

中文引用格式: 王锋, 姚珍, 梁吉业. 面向动态混合数据的多粒度增量特征选择算法. 软件学报, 2025, 36(3): 1186–1201. http://www.jos.org.cn/1000-9825/7158.htm

英文引用格式: Wang F, Yao Z, Liang JY. Multi-granulation Incremental Feature Selection Algorithm for Dynamic Hybrid Data. Ruan Jian Xue Bao/Journal of Software, 2025, 36(3): 1186–1201 (in Chinese). http://www.jos.org.cn/1000-9825/7158.htm

Multi-granulation Incremental Feature Selection Algorithm for Dynamic Hybrid Data

WANG Feng, YAO Zhen, LIANG Ji-Ye

(School of Computer and Information Technology, Shanxi University, Taiyuan 030006, China)

Abstract: In the era of big data, the sample scale and the dynamic update and variation of dimensionality greatly increase the computational burden. Most of these data sets do not exist in the form of a single data type but are more often hybrid data containing both symbolic and numerical data. For this reason, scholars have proposed many feature selection algorithms for hybrid data. However, most of the existing algorithms are only applicable to static data or small-scale incremental data and cannot handle large-scale dynamic changing data, especially large-scale incremental data sets with changing data distribution. To address this limitation, this paper proposes a multi-granulation incremental feature selection algorithm for dynamic hybrid data based on an information fusion mechanism by analyzing the variations and updates of granularity space and granularity structure in dynamic data. The algorithm focuses on the mechanism of granularity space construction in dynamic hybrid data, the mechanism of dynamic update of multiple data granularity structures, and the mechanism of information fusion for data distribution variations. Finally, the paper verifies the feasibility and efficiency of the proposed algorithm by comparing the experimental results with other algorithms on the UCI dataset.

Key words: dynamic hybrid data; data distribution variation; multi-granulation computing; information fusion

随着信息技术的快速发展, 数据集呈现出规模大、维数高、类型混杂、变化快的特点. 这些特性极大地降低了算法的运行效率并且影响其分类性能. 如何高效地从这种数据中去除冗余以及不相关的属性, 以解决算法效率低的问题, 并改善算法的有效性, 一直是机器学习领域的热点课题之一[1]. 特征选择作为一种数据预处理的有效方

* 基金项目: 国家自然科学基金 (62276158); 山西省回国留学人员科研项目 (2021-007)
收稿时间: 2023-06-16; 修改时间: 2023-08-10, 2023-10-22; 采用时间: 2023-12-07; jos 在线出版时间: 2024-05-08
CNKI 网络首发时间: 2024-05-11

法, 通过去除冗余无关的特征, 有效地降低了数据维度, 从而减低模型学习难度, 减少数据噪声, 同时也可以避免实验的过拟合现象[2,3]. 为此, 针对数据集的规模大、维数高等特点, 众多学者研究并探索了许多有效的特征选择方法和策略. 其中, 面向迅速变化的动态数据集的特征选择技术也取得了可观的研究成果, 尤其针对开放环境下的大数据, 近年来引起了研究者的热切关注和探索[4−10].

传统的动态特征选择求解模型以及算法大多是通过构造传统度量指标的动态更新机制来实现对动态数据的处理. 显然这种处理方式是局部的, 并未及时考虑由数据分布变化带来的信息获取模式的变化. 而开放环境下的动态大数据, 数据规模会不断增加, 其类别信息和特征空间等都会不断地更替和转换, 进而会引起数据内部结构的不断更新, 而由此引发的数据分布的变化为传统的增量式挖掘技术带来了更大的探索空间. 为此, 本文通过讨论动态混合数据中新增数据对原有粒空间以及粒结构的影响, 设计了面向混合数据的动态粒空间构建机制、数据粒随分布变化的更新机制以及多结果的动态信息融合机制, 并在此基础上提出了一种面向混合数据的多粒度增量特征选择算法.

本文第 1、2 节简要介绍了特征选择相关工作及基础知识. 第 3 节阐述了本文相关的算法的思想及具体的步骤. 第 4 节将算法在 10 个 UCI 数据集上实验, 实验结果验证了算法的有效性和高效性. 第 5 节总结了本文算法.

1 特征选择相关工作

现阶段, 众多研究者针对特征选择技术已进行了大量探索, 并取得了许多研究成果[11−20]. 为了度量信息表和决策表的不确定性, 一些学者基于粗糙集理论将信息熵引入启发式特征选择算法中[21−23], 大幅度降低了计算耗时. Xu 等人[24]基于序信息系统的知识粗糙熵, 在系统中引入属性重要性的概念, 利用该测度能度量序信息系统中属性集的不确定性, 提出序信息系统中基于知识粗糙熵的启发式约简算法. Liang 等人[25]提出了多粒度的粗糙特征选择算法, 通过选择不同的小粒度, 然后在每个小粒度上估计原始数据集的约简量, 再将小粒度的所有估计融合在一起, 得到数据表的近似约简. 在实际生活中, 有大量的数据都是由符号型、数值型和缺失数据组成的混合型数据. 如何对混合数据进行特征选择引起了广泛的关注. Hu 等人[26]引入了邻域粗糙集理论来构建混合数据的模糊等价关系和特征选择算法. Wang 等人[27]基于分解融合的思想, 研究了一种面向大规模混合数据集的高效特征选择方法, 可以在更短的时间内获得有效的特征子集. Xiao 等人[28]针对符号型与数值型构成的部分标记混合数据, 提出了一种基于属性依赖度和混合约束条件的半监督特征选择算法. Shu 等人[29]提出一种基于信息增益的混合数据半监督特征选择算法. Chen 等人[30]基于机器学习的决策树算法和多核学习的核融合概念, 提出了基于增益比的特征选择方法和 T-norm 方法来计算属性之间的相似度, 并在混合类型属性的数据集来检测该特征选择方法的适用性. Hu 等人[31]针对混合型分类数据, 提出了一种新的基于三支标签传播的半监督属性约简方法. 但是, 以上算法有一些局限性, 只适用于静态数据集, 现实生活中的数据往往都是在不断地变化, 每次增减数据都重新处理数据显然是十分耗时的. 为了提高算法的效率, 一些增量特征选择算法被提出, 有效地解决了动态数据下的增量特征选择问题. Zhong 等人[32]首先通过矩阵得到知识粒度, 用来衡量系统的不确定性, 然后计算属性重要度, 并利用属性重要度设计静态算法, 然后提出了知识粒度矩阵形式的增量属性约简算法. Liang 等人[33]基于粗糙集理论, 提出了一种基于信息熵的组增量粗糙特征选择算法. Shu 等人[34]采用增量的方式计算新的正区域, 当特征值相对于对象集动态变化时, 基于计算出的正区域, 提出了增量特征选择算法. 为进一步研究特征选择方法, 基于分析特征与类别间的因果关系, 因果特征选择方法作为一类新颖的特征选择策略也引起学者们的深入研究和探索[35,36], 并取得了一系列的成果, 在机器学习和因果关系发现领域均受到广泛关注. 上述研究成果大都适用于静态数据集, 其中少量的适用于处理动态数据集的增量特征选择方法是通过构造传统度量指标的动态更新机制来处理动态数据的, 显然这种处理方式是局部的. 针对开放环境下的动态大数据, 数据分布及其内部结构都在不断地更新, 可有效处理面向数据分布变化的动态数据集的增量特征选择算法仍需要进一步探索.

2 基础知识

为有效处理动态数据的有效特征子集选取问题, 本节基于经典粗糙集理论引入如下的数据集表示方法及其相

关概念.

一个数据表可以表示为一个四元组 $S=(U,C\cup D,V,f)$, 其中 U 称为论域, C 是数据对象的特征集, D 是数据集的类信息. 令 $A=C\cup D$, V_a 表示属性 $a\in A$ 的值域, 并且有 $V=\bigcup_{a\in A}V_a$; 对于任意的 $a\in A$ 和 $x\in U$, $f:U\times A\to V$ 是一个信息函数, 并且有 $f(x,a)\in V_a$, 通常简写为 $S=(U,C\cup D)$.

特征子集 $B\subseteq C$ 可以诱导一个不可区分关系, 表示为: $R_B=\{(x,y)\in U\times U|f(x,a)=f(y,a),\forall a\in B\}$. 关系 R_B 称为等价关系, 可以将论域 U 划分为一组等价类, 表示为

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

其中, $[x]_B=\{y\in U|(x,y)\in R_B\}$, 表示 x 关于 B 的等价关系.

3 面向动态混合数据的多粒度增量特征选择算法

3.1 基于类别信息和特征权重的数据分布度量

为有效跟踪动态数据集的分布变化, 本节中从数据集的类信息和特征权重排序两方面来度量数据分布的变化. 针对数据分布的度量, 一些研究者也已经作了相应的探索, 但都只适合静态数据集, 构造其相应的增量式机制比较困难, 且计算耗时较高. 为此, 本节中分别从类信息和特征权重两个方面具体跟踪数据分布的变化, 进而指导粒空间和粒结构的更新. 类信息主要是数据集中不同类别标签的数据样本的规模的比例, 数据类别的比例可以很直观地检测到类别信息的更新和变化, 而且随着新数据的不断增加, 类别比例的更新也比较简便, 具体定义如下.

定义 1. 令 $S=(U,C\cup D)$ 是一个数据表, 其中 $U/D=\{Y_1,Y_2,\dots,Y_n\}$, 则 U 上的类别比例定义为:

$$\zeta_U:\frac{|Y_1|}{|U|},\frac{|Y_2|}{|U|},\dots,\frac{|Y_n|}{|U|} \tag{1}$$

当动态数据中有新增数据时, 根据新增数据中各类样本的规模, 并结合原来的各类别样本规模, 可直接更新类别比例 ζ. 为进一步跟踪相同类别比例下数据分布的变化, 本节从特征重要性排序的角度来检测数据分布的变化. 由于本文中拟使用拓展的 ReliefF 算法求解每个小的数据粒上的特征选择结果, 而 ReliefF 算法的输出是所有特征的权重值以及依照权重值的特征排序结果. 针对一个动态数据集, 如果数据集中数据样本更新前后对应的特征权重排序结果发生明显变化, 显然该数据集的分布也发生了一定的变化. 为此, 本节中引入数据粒上更新前后的特征排序变化来检测每个数据粒上数据分布的变化, 具体定义如下.

定义 2. 令 $S=(U,C\cup D)$ 是一个数据表, $a_i\in C$ $(i=1,2,\dots,|C|)$, 假设依据特征权重值, U 上的特征排序结果为 $\{s_U(a_1),s_U(a_2),\dots,s_U(a_{|C|})\}$, 当 U 动态更新为 U' 后, U' 上的特征排序结果为 $\{s_{U'}(a_1),s_{U'}(a_2),\dots,s_{U'}(a_{|C|})\}$, 则 $U\to U'$ 的特征排序变化值定义为:

$$\xi_{U'}=\frac{1}{|C|}\sum_{i=1}^{|C|}|s_{U'}(a_i)-s_U(a_i)| \tag{2}$$

当粒空间中多个数据粒都动态更新后, 假设动态更新的数据粒的总数为 N, 则 N 个数据粒上总的特征排序变化值定义见定义 3.

定义 3. 令 $U_1,U_2,\dots,U_N$ 是多个数据子表, 随着数据集的动态更新, $U_j\to U_j'$ $(j=1,2,\dots,N)$, 则 N 个数据子表上的特征变化总值定义为:

$$\xi=\frac{1}{N}\frac{1}{|C|}\sum_{j=1}^{N}\sum_{i=1}^{|C|}|s_{U_j'}(a_i)-s_{U_j}(a_i)| \tag{3}$$

综上所述, 上述两种新度量 ξ 和 ζ 分别从特征由权重引起的排序变化和类别变化两方面跟踪动态数据集中数据分布的变化. 具体的使用策略和规则将在第 3.2 节中结合粒空间的变化和构建做详细介绍.

3.2 面向混合数据的动态粒化机制

本节详细介绍面向动态数据集的粒化机制以及数据粒的动态更新策略. 首先是面向动态数据集的初始化粒化策略. 由于本文拟在每个小的粒上使用拓展的 ReliefF 算法进行特征选择的求解, 而 ReliefF 算法核心思想是随机

抽取多个数据样本来求解所有特征的权重. 因此, 初始化的粒化策略是按照类别比率从原数据集上抽取多个子数据集, 即多个小的数据颗粒, 并使每个小颗粒上的 ζ_{U_j} 与原始数据的 ζ_U 相同, 其中 U_j 是每个小颗粒的数据样本集, 具体的数据集初始化粒化算法详细步骤见算法 1.

算法 1. 面向动态数据集的初始粒化方法.

输入: 数据集 $S=(U,C\cup D)$;
输出: 多个子数据集 $U_1,U_2,\ldots,U_N$.

步骤 1: 依据定义 1 计算 U 上的类别比例 ζ_U;
步骤 2: 从数据集 U 上随机抽取 N 个子数据集 $U_1,U_2,\ldots,U_N$, 并满足下面两个条件:
{ ①依据 ζ_U, 使得 ζ_{U_j} 与 ζ_U 相同, $j=1,2,\ldots,N$;
②对任意 U_j 和 U_i ($i,j\in\{1,2,\ldots,N\}$), 使得 $U_j\cap U_i\approx\varnothing$;
}
步骤 3: 返回 $U_1,U_2,\ldots,U_N$.

关于算法 1 的说明: 首先是子数据集彼此之间尽可能地减少重复的数据样本, 即每次从未被抽取的数据样本中抽取新的子数据集; 其次, 子数据集的规模可随机确定, 也可以抽取不同规模的子数据集, 但由于在求解特征选择过程中, 每个子数据集上的所有样本都要用于更新特征权重, 因此本文建议每个子数据集的规模不必过大.

算法 1 的时间复杂度: 步骤 1 的时间复杂度是 O(|U|); 步骤 2 中, 由于是基于类别信息进行子数据集的抽取, 且每次均从未被抽样的数据中抽取, 则步骤 2 的时间复杂度是 O(|U|); 因此, 算法 1 的时间复杂度为 O(|U|).

在算法 1 的基础上, 下面讨论动态数据集中多颗粒随新增数据的动态更新策略, 该策略主要分为 3 个阶段, 分别是已有颗粒的更新、新颗粒的生成和部分冗余颗粒的删除. 首先, 针对已有的多个颗粒的更新策略是: 当有新数据样本增加后, 计算总数据集的各个类别比率 ζ_U, 并依此来更新每个小颗粒上的数据样本, 如果每个小颗粒上的 ζ_{U_i} 与 ζ_U 不相等, 则从新增数据中抽取样本加入小颗粒中, 使得 $\zeta_{U_i}\approx\zeta_U$. 其次, 如果新数据增加后, 总数据集的类别比率没有改变, 则从新增数据中抽取新的小颗粒, 并使得新增颗粒的类别比率与总数据集的类别比率相同, 即 $\zeta_{U_t}\approx\zeta_U$ (U_t 表示从新增数据中抽取的颗粒的样本集). 另外, 针对新颗粒的生成, 本节中还引入了特征权重排序的变化值 ξ 来进行判断. 当已有的颗粒随新增数据样本更新后, 求解所有颗粒的特征权重排序的变化值 ξ, 如果 ξ 小于给定阈值, 则可视为数据粒更新前后特征权重引起的排序变化比较弱, 即数据的分布未发生明显变化, 如果 ξ 大于给定阈值, 则需要重新抽取多个新的颗粒. 最后, 随着新数据样本的不断增加, 数据颗粒的数目也可能会不断增加, 为有效降低计算耗时, 当数据颗粒数目超过给定阈值时, 则删除一部分已有的颗粒. 综上所述, 动态数据集中, 当新数据增加后, 多颗粒的更新策略需要综合考虑数据粒的更新、生成以及删除等多个方面, 具体的参数和阈值的设定见算法 2.

算法 2. 面向动态数据的多数据粒更新算法.

输入: 数据集 $S=(U,C\cup D)$, U 上的多个数据集 $U_1,U_2,\ldots,U_N$, 新增数据样本集 U_φ;
输出: $U\cup U_\varphi$ 上的 N' 个子数据集 $U'_1,U'_2,\ldots,U'_{N'}$.

步骤 1: 依据定义 1 计算 $U\cup U_\varphi$ 上的类别比例 $\zeta_{U\cup U_\varphi}$;
步骤 2: 如果 $\zeta_{U\cup U_\varphi}\neq\zeta_U$, 则依次从 U_φ 中抽取数据样本并入 U_j 中, U_j 更新为 U'_j, 并使得 $\zeta_{U'_j}$ 与 $\zeta_{U\cup U_\varphi}$ 相同, 其中 $j\in\{1,2,\ldots,N\}$; 如果 $\zeta_{U\cup U_\varphi}=\zeta_U$, 转至步骤 3;
步骤 3: 从 U_φ 上抽取 M 个子数据集 $U'_{N+1},U'_{N+2},\ldots,U'_{N+M}$, 并满足下面两个条件:
{ ①依据 $\zeta_{U\cup U_\varphi}$, 使得 $\zeta_{U'_j}$ 与 $\zeta_{U\cup U_\varphi}$ 相同, $j=N+1,N+2,\ldots,N+M$;
②对任意 U'_j 和 U'_i, ($i,j\in\{N+1,N+2,\ldots,N+M\}$), 使得 $U'_j\cap U'_i\approx\varnothing$;

}

步骤 4: $N+M \to N'$, 返回 $U'_1, U'_2, \ldots, U'_{N'}$. 如果 $M > \eta$, $\eta = N\dfrac{|U_\varphi|}{|U|}$, 则从多个子数据集中随机删除 M' 个 ($M' < M$);

步骤 5: $N'-M' \to N'$, 返回 $U'_1, U'_2, \ldots, U'_{N'}$.

关于算法 2 的说明: 步骤 4 中新增子表的个数 M 与原子表个数 N 关系应满足 $M \approx N\dfrac{|U_\varphi|}{|U|}$, 这样是为了约束总子表的个数, 从而不会使得新增的子表特征排序结果所占权重过大.

算法 2 的时间复杂度: 步骤 1 的时间复杂度为 $O(|U|+|U_\varphi|)$; 步骤 2 和步骤 3 均从新增数据集 U_φ 上抽取数据样本, 则步骤 2 和步骤 3 的时间复杂度均为 $O(|U_\varphi|)$; 步骤 4 的时间复杂度是常量阶的; 因此, 算法 2 的时间复杂度是 $O(|U|+|U_\varphi|)+O(|U_\varphi|)=O(|U|+|U_\varphi|)$.

3.3 面向混合数据的 ReliefF 算法

ReliefF 算法是对经典 Relief 算法的扩展[37], 用于处理多分类的问题. Relief 系列算法的核心思想是同类的数据样本间的相似度应该更大, 而不同类数据样本间的相似度应该相对更小, 一个有用的特征应该让同类的数据样本距离更近, 让不同类数据样本距离更远. 在此基础上, ReliefF 算法是针对多分类问题提出的, 其核心的处理技巧是: 每次从训练集的数据样本中随机抽取一个样本 x, 然后从和 x 同类的样本集中找出 k 个近邻样本 (Near-hit), 从其他不包括 x 的每个类的样本集中找出 k 个近邻样本 (Near-miss), 然后根据权重公式更新每个特征的权重, 以上过程循环 m 次, 得到各个属性的平均权重. 特征的权重越大, 表示该特征的分类能力越强, 反之, 表示该特征分类能力越弱. ReliefF 算法的具体步骤见算法 3.

算法 3. ReliefF 算法.

输入: 数据集 $S=(U, C\cup D)$, 最近邻个数 k, 抽样次数 m, 各个特征的权重初始值 $w(a_i)=0, i=1,2,\ldots,|C|$;
输出: 各个特征的特征权重 $w(a_i)(i=1,2,\ldots,|C|)$.

步骤 1: 计算 $U/D=\{Y_1, Y_2, \ldots, Y_n\}$;
步骤 2: for($r=1; r \leqslant m; r++$)
{
步骤 2.1: 从数据集中 U 中任意抽取一个数据样本 x, 并假设 $x \in Y_q$, 在 Y_q 包含的数据样本中求解 x 的 k 个近邻 $n_t^h, t=1,2,\ldots,k$;
步骤 2.2: for($j=1; j \leqslant n$ and $j \neq q; j++$)
{ 从 Y_j 包含的数据样本中求解 x 的 k 个近邻 $n_t^m(j), t=1,2,\ldots,k$;
}
步骤 2.3: for($i=1; i \leqslant |C|; i++$)
{

$$w(a_i)=w(a_i)-\frac{1}{mk}\sum_{t=1}^{k}Dis(i,x,n_t^h)+\frac{1}{mk}\sum_{j=1,j\neq q}^{n}\frac{|Y_j|}{|U-Y_q|}\sum_{t=1}^{k}Dis(i,x,n_t^m(j));\}$$

}
步骤 3: 输出结果 $w(a_i)(i=1,2,\ldots,|C|)$.

在算法 3 中, m 和 k 均为输入参数, m 为抽取样本的数量, k 表示需要寻找的最近邻的数量; $Dis(i,x,y)$ 表示数据对象 x 和 y 在第 i 个特征 a_i 下的距离; n_t^h 表示与当前样本 x 在同类中距离最近的 k 个最近邻; $n_t^m(j)$ 表示与当前样本 x 在第 j 类中距离最近的 k 个最近邻, 其中第 j 类不包含样本 x.

算法 3 的时间复杂度: 步骤 1 的时间复杂度为 $O(|U|)$; 步骤 2.1 的时间复杂度为 $O(|Y_q|\cdot Dis)$, 其中 Dis 表示数据样本间距离的时间复杂度; 步骤 2.2 的时间复杂度为 $O(|U-Y_q|\cdot Dis)$; 步骤 3 的时间复杂度为 $O(|C|nk\cdot Dis(i))=$

$O(nk\cdot Dis)$, 其中 $Dis(i)$ 表示数据样本在特征 a_i 上距离的时间复杂度; 因此, 步骤 2.1–2.3 的时间复杂度为 $O(|Y_q|\cdot Dis+|U-Y_q|\cdot Dis+nk\cdot Dis)=O(|U|\cdot Dis+nk\cdot Dis)$. 由于 n 为类别数, k 为近邻数, 在静态数据集中 n 和 k 通常是固定不变的, 即 $nk=n\times k$ 是不变的, 因此步骤 2.1–2.3 的时间复杂度主要与 U 的大小相关, 即 $O(|U|\cdot Dis)$. 综上, 算法 3 的时间复杂度为 $O(m|U|\cdot Dis)$. 由此可得, ReliefF 算法的计算耗时与抽样次数、样本规模以及数据维数均有关系, 且随着针对数据样本相似度或距离的深入探索, 样本间相似度或距离的计算也会直接影响到 ReliefF 算法的运行效率.

ReliefF 算法适用于处理数值型数据, 为有效处理混合型数据中的符号型数据, 本节引入一种面向符号型数据相似度度量来求解数据样本的近邻[38], 该相似度度量介绍见定义 4.

定义 4. 令 $S=(U,C\cup D)$ 是一个决策表, 假设 $U/D=\{Y_1,Y_2,\ldots,Y_n\}$. $\forall x,y\in U$, $a_i\in C$, 样本 x 和 y 在特征 a_i 上的内部距离被定义为:

$$D_{in}(i,x,y)=\frac{1}{n}\sum_{j=1}^{n}|[x]_{a_i}\cap Y_j|\cdot|Y_j-[x]_{a_i}|-|[y]_{a_i}\cap Y_j|\cdot|Y_j-[x]_{a_i}| \tag{4}$$

基于上述度量和算法 3 中的 ReliefF 算法, 本节中给出了面向混合型数据的 ReliefF 算法, 见算法 4.

算法 4. 面向混合数据的 ReliefF 算法.

输入: 数据集 $S=(U,C\cup D)$, 最近邻个数 k, 抽样次数 m, 各个特征的权重初始值 $w(a_i)=0,i=1,2,\ldots,|C|$;
输出: 各个特征的特征权重 $w(a_i)(i=1,2,\ldots,|C|)$.

步骤 1: 计算 $U/D=\{Y_1,Y_2,\ldots,Y_n\}$;
步骤 2: for($r=1;r\leqslant m;r++$)
{
步骤 2.1: 从数据集 U 中任意抽取一个数据样本 x, 并假设 $x\in Y_q$, 在 Y_q 包含的数据样本中求解 x 的 k 个近邻 $n_t^h,t=1,2,\ldots,k$;
步骤 2.2: for($j=1;j\leqslant n$ and $j\neq q;j++$)
{ 从 Y_j 包含的数据样本中求解 x 的 k 个近邻 $n_t^m(j),t=1,2,\ldots,k$;
}
步骤 2.3: for($i=1;i\leqslant|C|;i++$)
{

$$w(a_i)=w(a_i)-\frac{1}{mk}\sum_{t=1}^{k}Diff(i,x,n_t^h)+\frac{1}{mk}\sum_{j=1,j\neq q}^{n}\frac{|Y_j|}{|U-Y_q|}\sum_{t=1}^{k}Diff(i,x,n_t^m(j)),$$

其中, $Diff(i,x,y)=\begin{cases}\dfrac{|f(x,a_i)-f(y,a_i)|}{\max(a_i)-\min(a_i)} & \text{如果}a_i\text{是连续的}\\ D_{in}(i,x,y) & \text{如果}a_i\text{是离散的}\end{cases}$
且 $\max(a_i)=\max\{f(x,a_i),x\in U\}$, $\min(a_i)=\min\{f(x,a_i),x\in U\}$;
}
}
步骤 3: 依据特征的权重值 $w(a_i)(i=1,2,\ldots,|C|)$ 对所有特征进行排序, 得到特征重要度排序结果 $\{s(a_1),s(a_2),\ldots,s(a_{|C|})\}$.

基于算法 3 的时间复杂度, 通过分析可得, 算法 4 的时间复杂度为 $O(m|U|\cdot Diff)$, 其中 $Diff$ 为面数据样本距离的时间复杂度.

3.4 多个特征选择结果的融合机制

针对多个动态更新后的多个数据粒, 使用第 3.3 节中的算法 4 可求解到多个按特征权重值排序的特征重要度排序结果, 本节介绍面向多个特征选择排序结果的融合机制. 融合机制的核心思想是: 如果同一个特征在不同特征

排序结果中存在不同的排序位置, 则选择出现频率最高的排序位置为最终结果; 如果频率最高的排序位置有两个或多个, 则将这几个排序位置的均值作为最终结果; 在此基础上, 如果不同特性具有相同的最终排序位置, 则计算特性的平均权重值, 按照权重值排序. 具体的算法步骤见算法 5.

算法 5. 面向多个特征选择排序结果的融合方法.

输入: N' 个特征重要度排序结果: $\{s_j(a_1), s_j(a_2), \ldots, s_j(a_{|C|})\}$, $j = 1,2,\ldots,N'$, 其中 $s_j(a_i)$ 表示特征 a_i 在第 j 个特征选择排序结果上的排序位置;

输出: 融合后的特征排序结果 $\{s'(a_1), s'(a_2), \ldots, s'(a_{|C|})\}$.

步骤 1: 基于 N' 个特征重要度排序结果计算矩阵 $\Omega = [\omega_{it}]$, $i = 1,2,\ldots,|C|$, $t = 1,2,\ldots,|C|$, 其中 $\omega_{it} = |\{s_j(a_i) = t | s_{j'}(a_i) = s_j(a_i), j' \neq j, j' = 1,2,\ldots,N'\}|$ 表示所有特征选择中第 i 个特征在第 t 个位置排序值出现的次数;

步骤 2: 依据 $\Omega = [\omega_{it}]$ 计算 $s'(a_i) = t'$: $\omega_{it'} = \max\{\omega_{it}, t = 1,2,\ldots,|C|\}$, 其中 $s'(a_i)$ 表示 a_i 的所有位置排序值出现最多次数对应的排序值; 如果 t' 不唯一, 最多次数对应的位置排序值有两个或多个, 则将排序值的均值作为最终结果, 即 $s'(a_i) = t' = \dfrac{t'_1 + t'_2 + \ldots}{|\{t'_1, t'_2, \ldots\}|}$ ($t'_1, t'_2, \ldots$ 表示具有相同次数的排序值);

步骤 3: 依据所有特征 a_i 的 $s'(a_i)$ 值排序全部特征, 如果有多个特征 (a_i 和 $a_{i'}$) 的排序值相同, 即 $s'(a_i) = s'(a_{i'})$, 则计算 $\overline{w}(a_i) = \dfrac{1}{N'}\sum_{j=1}^{N'} w_j(a_i)$ 和 $\overline{w}(a_{i'}) = \dfrac{1}{N'}\sum_{j=1}^{N'} w_j(a_{i'})$, 依照 $\overline{w}(a_i)$ 和 $\overline{w}(a_{i'})$ 值更新 $s'(a_i)$ 和 $s'(a_{i'})$;

步骤 4: 返回排序后的特征结果 $\{s'(a_1), s'(a_2), \ldots, s'(a_{|C|})\}$.

关于算法 4 的说明: 步骤 1 中求解了在所有特征选择排序结果中, 每个特征在每个排序位置上出现的次数, 比如矩阵 $\Omega = [\omega_{it}]$ 中第 i 行第 t 列元素表示第 i 个特征在第 t 的位置上共出现了 ω_{it} 次. 步骤 2 在步骤 1 的基础上将矩阵 $\Omega = [\omega_{it}]$ 中每行元素中的最大值作为每个特征的最终位置排序值, 如果出现次数相同的排序值, 则排序值的均值作为最终结果, 假设特征 a_2 在排序第 1, 4 和 5 的位置上的排序次数相同, 且为次数最大值, 即 $\omega_{21} = \omega_{24} = \omega_{25}$, 则有 $t'_1 = 1, t'_2 = 4, t'_3 = 5$ 和 $t' = \dfrac{1+4+5}{3} \approx 3.3$, 所有特征 a_2 的排序值为 3.3, 重要度在第 3 个和第 4 个特征之间. 步骤 3 中介绍了两个特征的权重均值计算公式, 如果有多个特征的计算公式也是类似的处理方式. 步骤 4 输出算法的最终结果, 即所有特征按照其重要度的排序结果.

算法 5 的时间复杂度: 步骤 1 的时间复杂度是 $O(N'|C|)$; 步骤 2 的时间复杂度是 $O(|C|)$; 步骤 3 的时间复杂度为 $O(N'|C|)$; 因此, 算法 5 的时间复杂度为 $O(N'|C|)$.

3.5 基于 ReliefF 的多粒度增量特征选择算法 (MGIFS)

基于以上介绍的动态数据粒更新策略, 面向混合数据的 ReliefF 算法, 以及多个特征排序结果的融合方法, 本节介绍面向混合数据的多粒度增量特征选择算法. 该算法的核心思想是: 动态数据集经过初始化形成粒空间后, 随着新数据的增加, 依据多信息粒更新策略动态更新所有颗粒, 并在每个颗粒上使用面向混合数据的 ReliefF 算法更新特征的权重, 然后计算每个颗粒更新前后的特征权重的变化值以及所有颗粒的总特征变化值, 并依据特征权重的变化值再一次更新粒空间以及每个信息粒上的特征权重值, 最后融合所有的特征权重值, 给出最终的特征权重值以及特征排序. 为方便理解算法 1–算法 5 和本节中新算法的联系, 图 1 给出了算法 1–算法 5 对动态数据集的处理过程. 如图 1 所示, 算法 1、算法 2 重点讨论了面向动态数据集的粒化过程, 算法 4 是在每个小的数据粒上求解特征选择结果, 而算法 5 重点实现了如何有效融合多个特征排序结果. 算法 1–算法 5 的核心思想是将多粒度理论融入到求解动态数据集的特征选择过程中, 每个算法都是在前面算法的基础上求解新的结果, 并不是彼此独立的. 算法 1 实现了面向动态数据集的初始粒化策略, 通过算法 1 可求解到多个小的颗粒; 而随着动态数据集中新数据样本的不断增加, 算法 2 是针对所有数据粒的更新策略, 包括从新增数据样本中抽取新的数据粒、在原有数据粒中增加新的数据样本以及调整每个粒中数据样本的类别比率等操作; 使用算法 4 可在每个数据粒上求解到一个特征选择结果, 多个数据粒上可得到多个特征选择结果; 在此基础上, 算法 5 通过融合上述的多个特征选择结果得到

最终的特征选择结果. 基于上述分析, 将粒化、求解以及融合等过程结合在一起, 设计了面向混合数据的多粒度增量特征选择算法, 具体步骤见算法 6.

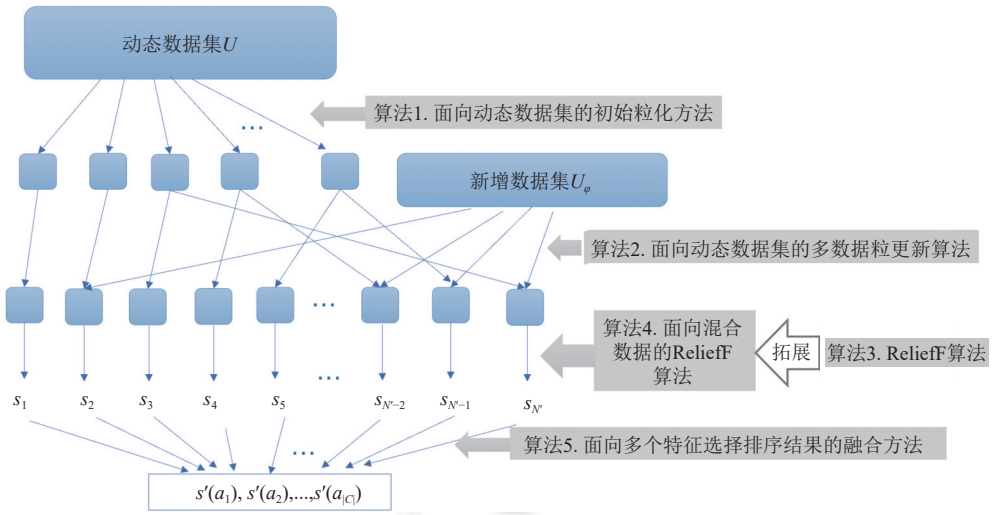

图 1 算法框架图

算法 6. 基于 ReliefF 的多粒度增量特征选择算法 (MGIFS).

输入: 原始数据集 $S=(U,C\cup D)$, U 上的特征权重值 $W(a_i),a_i\in C$, 以及新增数据样本集 U_φ;

输出: $U\cup U_\varphi$ 上的特征排序结果 $\{s'(a_1),s'(a_2),\ldots,s'(a_{|C|})\}$.

步骤 1: 使用算法 1 在 U 上抽取 N 个子数据集 $U_1,U_2,\ldots,U_N$;

步骤 2: 使用算法 4 求解每个 U_j ($j=1,2,\ldots,N$) 上的特征权重值 $w_j(a_i),a_i\in C$;

步骤 3: 使用算法 2 中的步骤 1–步骤 3 更新子数据集, 针对更新后的子数据集, 使用如下方法求解特征权重:

{ ①对更新的子数据集 U'_j ($j\in\{1,2,\ldots,N\}$), 使用每个子数据集上新增数据样本在原特征权重基础上更新权重值 $w_j(a_i)\Rightarrow w'_j(a_i),a_i\in C$;

②对新增加的子数据集 U'_j ($j=N+1,N+2,\ldots,N+M$), 使用算法 4 求解每个 U'_j 上的特征权重值 $w'_j(a_i),a_i\in C$;

}

步骤 4: $N+M\to N'$, 使用定义 2、定义 3 求解 N' 个子数据集上的特征排序变化值 ξ;

步骤 5: 如果 $\xi>\theta$, 则使用算法 2 步骤 3 从 $U\cup U_\varphi$ 上重新抽取多个子数据集, 并使用算法 3 求解特征权重值;

步骤 6: 依照算法 5 融合所有子数据集特征排序结果形成最终排序结果 $\{s'(a_1),s'(a_2),\ldots,s'(a_{|C|})\}$.

关于算法 6 的说明: 步骤 3 详细介绍了针对动态更新数据粒上的特征权重求解策略. 具体的求解方法是对于增加了数据样本的数据粒, 由于原数据粒的特征已经拥有权重值, 则在原特征权重值的基础上使用新增的数据样本进一步更新特征权重值; 而对于新增的数据粒, 则需要基于算法 4 使用该数据粒上的所有样本求解特征的权重值. 因此, 在数据规模动态增加的过程中, 随着数据粒的更新, 为有效降低计算耗时, 并非是对所有数据粒都重新计算其特征权重.

算法 6 的时间复杂度: 步骤 1 的时间复杂度为 $O(|U|)$; 步骤 2 的时间复杂度为 $O\left(\sum_{j=1}^{N}m_j|U_j|\cdot Diff\right)$, 其中 m_j 表示每个子数据集上的抽样次数; 步骤 3 中①的时间复杂度为 $O\left(\sum_{j=1}^{N}m_\varphi|U_j|\cdot Diff\right)$, 其中 m_φ 表示子数据集更新后新增

的数据样本数, ②的时间复杂度为 $O\left(\sum_{j=N+1}^{M} m_j|U'_j|\cdot Diff\right)$, 而更新子数据集的时间复杂度是 $O(|U|+|U_\varphi|)$, 因此步骤 3 总的时间复杂度为 $O\left(\sum_{j=1}^{N} m_\varphi|U_j|\cdot Diff+\sum_{j=N+1}^{M} m_j|U'_j|\cdot Diff\right)$; 步骤 4 的时间复杂度为 $O(N'|C|)$; 步骤 5 的时间复杂度为 $O\left(\sum_{j=1}^{N} m_j|U_j|\cdot Diff\right)$; 步骤 6 的时间复杂度为 $O(N'|C|)$; 由于随着数据规模的增加, $O(N'|C|)$ 是常量阶的, 因此, 算法 6 总的时间复杂度为 $O\left(\sum_{j=1}^{N} m_\varphi|U_j|\cdot Diff+\sum_{j=N+1}^{M} m_j|U'_j|\cdot Diff+\sum_{j=1}^{N} m_j|U_j|\cdot Diff\right)$. 随着新数据的增加以及由数据分布变化引起的数据粒结构变化, 该复杂度中 3 部分的计算耗时显然是不同的, 所以算法 6 的时间复杂度可表示为 $\max\left(O\left(\sum_{j=1}^{N} m_\varphi|U_j|\cdot Diff\right), O\left(\sum_{j=N+1}^{M} m_j|U'_j|\cdot Diff\right), O\left(\sum_{j=1}^{N} m_j|U_j|\cdot Diff\right)\right)$. 显然, 如果新数据增加后数据分布未发生明显变化, 算法 6 的时间复杂度可近似表示为 $O\left(\sum_{j=N+1}^{M} m_j|U'_j|\cdot Diff\right)$, 即只在新生成的数据粒上求解特征权重, 由于增量数据集的规模通常会小于基础数据集, 而在增量数据集上抽取的数据粒也较少, 所以此时算法的计算耗时是最小的; 如果数据分布发生了变化但在设定的阈值内, 算法 6 的时间复杂度可近似表示为 $O\left(\sum_{j=1}^{N} m_\varphi|U_j|\cdot Diff\right)$, 即只在部分数据粒上根据新增数据样本更新特征权重, 由于是基于每个数据粒上原有的特征权重值使用新增数据样本来更新权重, 所以此时算法的计算耗时也相对较小; 如果数据增加后, 数据集的类别比例以及数据粒上的特征排序均发生了明显变化, 超过设定的阈值, 则算法 6 的时间复杂度近似表示为 $O\left(\sum_{j=1}^{N} m_j|U_j|\cdot Diff\right)$, 即需要重新抽取多个数据粒并求解每个数据粒上的特征权重值, 此时算法的计算耗时也会高于前两种情况. 综上分析, 随着新数据样本的不断增加, 算法 6 并不需要每次重新粒化以及在所有数据粒上重新求解特征权重, 而只在检测到数据分布及其粒结构发生了明显变化, 才需要重新粒化, 抽取子数据集, 进而重新求解数据粒上特征权重. ReliefF 算法处理动态数据集需要每次重新抽取数据样本来求解特征权重, 且随着数据规模的不断增加, 抽样次数也会明显增加, 进而增加计算耗时, 因此, 与 ReliefF 算法相比, 算法 6 通过从多粒度视角跟踪数据分布变化, 进而指导数据粒的更新, 避免了每次都在全部数据粒上求解特征权重, 可有效降低计算耗时, 提高计算效率.

4 实验分析

4.1 实验数据和实验设计

为了有效验证本文所提新算法 MGIFS 的有效性, 第 4.2 节、第 4.3 节从 UCI 数据集中选取了 10 个数据集进行实验比较和分析, 数据集的基本信息见表 1. 由于 MGIFS 本质上是给出了一个面向动态混合数据集的多粒度特征选择求解框架, 针对每个数据粒上特征选择结果的求解, 本文中选择了拓展的 ReliefF 算法 (算法 4), 即将面向混合数据的 ReliefF 算法嵌入到了本文提出的多粒度增量特征选择求解框架中. 为此, 第 4.2 节、第 4.3 节的实验分析中将未嵌入多粒度增量特征选择框架的 ReliefF 算法 (算法 4) 作为对比算法, 并分别从两方面来验证算法 MGIFS 的有效性和高效性. 首先, 第 4.2 节中分别使用 MGIFS 算法和面向混合数据的 ReliefF 算法在 10 个数据集上求解特征排序结果, 通过比较特征选择结果的分类性能来验证 MGIFS 算法的有效性. 其次, 为进一步验证算法 MGIFS 的高效性, 第 4.3 节中比较了上述两种算法在数据集持续多次变化中的计算耗时. 在此基础上, 为进一步验证算法 MGIFS 的可行性, 第 4.4 节中又选取了两个近年来提出的增量特征选择算法作为对比算法来进行实验分析[39,40]. 本实验的测试环境是 Intel(R) Core(TM) i7-10700 CPU @ 2.90 GHz, 内存 16.0 GB, 算法编程语言为 Python, 使用的开发工具是 Jet Brains PyCharm Community Edition2020. 另外, 对表 1 中存在缺失值的数据集, 本实验中对其缺失值进行了填补, 填补策略是删除缺失值数量大于样本 1/3 的特征, 对于缺失值小于样本数量 1/3 的特征, 用该列众数填充.

表 1 实验数据集

数据集	样本数量	数值值特征数量	符号值特征数量	特征总数	类别数
zoo	101	16	1	17	7
flags	194	26	3	29	8
audiology	200	0	68	68	24
anneal	798	8	26	34	5
german	1 000	5	15	20	2
cortex	1 080	78	3	81	8
sick	2 800	5	22	27	2
chess	3 196	0	36	36	2
hypothyroid	3 772	28	0	28	4
mushroom	5 645	21	0	21	2

4.2 有效性分析

为了测试算法的有效性, 本节引入了朴素贝叶斯 (NBC) 和支持向量机 (SVM) 两个常用分类器来对 MGIFS 算法和 ReliefF 算法在表 1 中 10 个数据集上特征选择结果的分类性能进行对比验证, 分类精度使用了十折交叉验证方法来确定最终值. 实验中, 基于每个数据集, 随机抽取样本生成一个新的数据集, 该数据集的规模和原数据集规模相同, 特征信息与原数据集相同; 但由于是随机抽取样本生成, 所以新数据集的分布可能与原数据集存在不同. 在此基础上, 将新生成的数据集作为增量数据集加入原数据集中, 并使用上述两种算法求解特征选择结果. 表 2 和表 3 分别列出了上述两个算法在每个数据集上求解到的特征选择结果在分类器 NBC 和 SVM 中的分类精度对比结果. 表 4 中列出了两种算法在 10 个数据集上的依照权重的特征排序结果, 表中仅列出特征权重排序较重要的部分.

表 2 NBC 分类精度对比

数据集	MGIFS	ReliefF
zoo	**0.8578±0.0591**	0.8483±0.1194
flags	0.7955±0.0544	**0.8095±0.1168**
audiology	0.9900±0.0400	0.9900±0.0400
anneal	0.7805±0.0505	0.7805±0.0505
german	**0.7248±0.0848**	0.7188±0.0787
cortex	0.9790±0.0052	0.9790±0.0052
sick	**0.9289±0.0031**	0.9228±0.0234
chess	**0.8714±0.1170**	0.8479±0.1396
hypothyroid	**0.9025±0.0084**	0.9013±0.0045
mushroom	**0.8597±0.0846**	0.8509±0.3642
average	**0.869 0**	**0.864 9**

表 3 SVM 分类精度对比

数据集	MGIFS	ReliefF
zoo	**0.9400±0.0700**	0.9200±0.1200
flags	**0.9662±0.0602**	0.9393±0.0642
audiology	0.9900±0.0290	0.9900±0.0290
anneal	0.7992±0.0118	0.7992±0.0118
german	**0.7238±0.0838**	0.7167±0.0467
cortex	0.9449±0.5000	0.9449±0.5000
sick	**0.9517±0.0161**	0.9389±0.0031
chess	**0.8539±0.1328**	0.7663±0.1163
hypothyroid	**0.9279±0.0013**	0.9228±0.0021
mushroom	0.9216±0.1659	**0.9241±0.3141**
average	**0.901 9**	**0.886 3**

表 4 特征排序结果对比

数据集	MGIFS	ReliefF
zoo	4, 9, 2, 5	4, 5, 9, 11
flags	18, 20, 24, 13	18, 11, 20, 13
audiology	64, 13, 1, 57	64, 13, 1, 57
anneal	3, 12, 5	3, 12, 5
german	14, 3, 1, 20, 10	14, 1, 3, 10, 12
cortex	81, 80, 79, 78, 34	81, 79, 80, 78, 34
sick	13, 24, 18, 10, 12, 20, 28	28, 13, 24, 26, 25, 18, 10
chess	33, 21, 10, 15, 1, 35, 6	33, 21, 10, 15, 1, 34, 6
hypothyroid	21, 18, 20, 14	18, 20, 21, 19
mushroom	8, 10, 4, 7, 17	8, 10, 4, 7, 6

从表 2 和表 3 中的实验比较结果可以看出, 算法 MGIFS 计算精度都要近似或高于 ReliefF 算法, 实验结果表明, 使用 MGIFS 算法可以找到一个与 ReliefF 算法性能接近甚至更优的特征子集. 实验中, 由于增量数据集时随机抽样生成, 因此增量数据集的分布与原数据集的分布之间的差异没有明显的规律. 由于 MGIFS 算法借鉴多粒度思想, 通过跟踪动态数据集类别信息的变化来指导数据粒的更新, 并在求解到每个数据粒上的特征选择结果后, 使用数据粒更新前后特征权重排序结果的变化值来进一步检测数据分布的变化, 进而再一次指导数据粒的更新和特征权重的更新, 因此, 随着数据集规模的动态增加, MGIFS 算法可求解到一个有效的特征选择结果. 而 ReliefF 算法的性能会受到抽样次数以及抽取的数据样本的影响, 随着数据规模的增加, 算法中的抽样次数也会明显增加, 否则会直接影响到权重的更新.

表 4 是上述两种算法在 10 个数据集上的特征选择结果的对比, 表中仅列出特征权重排序较重要的部分. 依据表 4 中的对比结果可发现, 在数据集 audiology、anneal、cortex 上算法 MGIFS 都能找到与 ReliefF 算法相同的特征子集; 在其余数据集中, 也能找到比较接近的特征子集. 结合表 2、表 3 中的分类精度比较可得, 使用 MGIFS 算法不仅可以找到一个性能与 ReliefF 算法接近的特征子集, 甚至在部分数据集上可以找到与 ReliefF 算法相同的特征子集, 进一步验证了算法 MGIFS 的有效性.

4.3 高效性分析

为进一步验证算法 MGIFS 的高效性, 本节中比较了 MGIFS 算法、ReliefF 算法处理多次动态增加数据样本的计算耗时. 实验中, 基于每个数据集, 随机抽取样本生成一个新的数据集, 新数据集的规模和特征信息与原数据集相同, 但数据分布可能会发生变化; 在此基础上, 将每个原数据集最为基础数据集, 新生成的数据集分成 8 份, 每份的规模随机确定, 然后作为增量数据集依次增加到原数据集中, 增加的过程中分别使用上述两种算法求解特征选择结果, 其计算耗时的比较结果见表 5、表 6 和图 2. 其中, 表 5 中的内容是未添加增量数据集前基础数据集上的计算时间, 表 6 中是将 8 份增量数据集依次添加到基础数据集中后总的计算时间, 图 2 中具体给出了每次添加增量数据集后两种算法的计算时间比较结果.

表 5 静态数据计算时间对比

数据集	MGIFS (s)	ReliefF (s)	计算时间提高率 (%)
zoo	1.16	10.56	89.02
flags	6.54	33.70	80.59
audiology	8.24	59.70	86.20
anneal	17.32	58.40	70.34
german	21.37	56.14	61.93
cortex	349.47	791.30	55.84
sick	84.63	526.45	83.92
chess	109.88	450.99	75.63
hypothyroid	193.41	721.46	73.19
mushroom	199.60	910.94	78.08
average	**99.15**	**361.96**	**72.61**

表 6 动态数据计算时间对比

数据集	MGIFS (s)	ReliefF (s)	计算时间提高率 (%)
zoo	18.10	140.32	87.10
flags	123.56	571.77	78.38
audiology	158.56	973.08	83.70
anneal	298.09	1 031.65	71.10
german	260.99	1 027.68	74.60
cortex	3 116.46	12 062.49	74.16
sick	1 243.45	12 151.70	89.76
chess	980.65	8 961.07	89.06
hypothyroid	1 738.56	15 741.89	88.96
mushroom	2 349.71	20 355.73	93.37
average	**1 028.81**	**7 301.74**	**85.91**

根据第 3.5 节中对算法 MGIFS 时间复杂度的分析, 动态数据集的分布变化会直接影响该算法的计算耗时. 为此, 本节实验设计中选择动态多次增加不同规模的增量数据集, 而随着随机规模的增量数据集依次增加到原数据集中后, 每次增加的数据样本数和类别以及类别数都是不确定的, 进而引起的数据分布的变化也是不确定的. 从表 5 和表 6 中算法 MGIFS 分别处理初始化基础数据集和增量数据集的计算时间可得, 每个数据添加的增量数据集的分布变化对计算时间有直接的影响. 如数据集 chess 和 hypothyroid 的规模和维数比较接近, 但是添加增量数据集后, 数据集 chess 的计算时间明显小于 hypothyroid 的计算时间, 侧面反映了 hypothyroid 的增量数据集的数据分布变化比较明显. 另外, 数据维数也会直接影响算法的计算效率, 如数据集 cortex 在 10 个数据集中规模不是最大的, 但维数是最高的, 从表 5 和表 6 中的结果可发现, 使用算法 MGIFS 分别处理 cortex 的基础数据集和增量数据集的计算时间在 10 个数据集中是最多的. 另外, 由于随着增量数据集的添加, ReliefF 算法每次需要重新抽样, 且抽样次数会随着增加, 因此 ReliefF 算法的计算时间会明显增多, 表 5 和表 6 中的实验结果也进一步验证了算法 MGIFS 高效性.

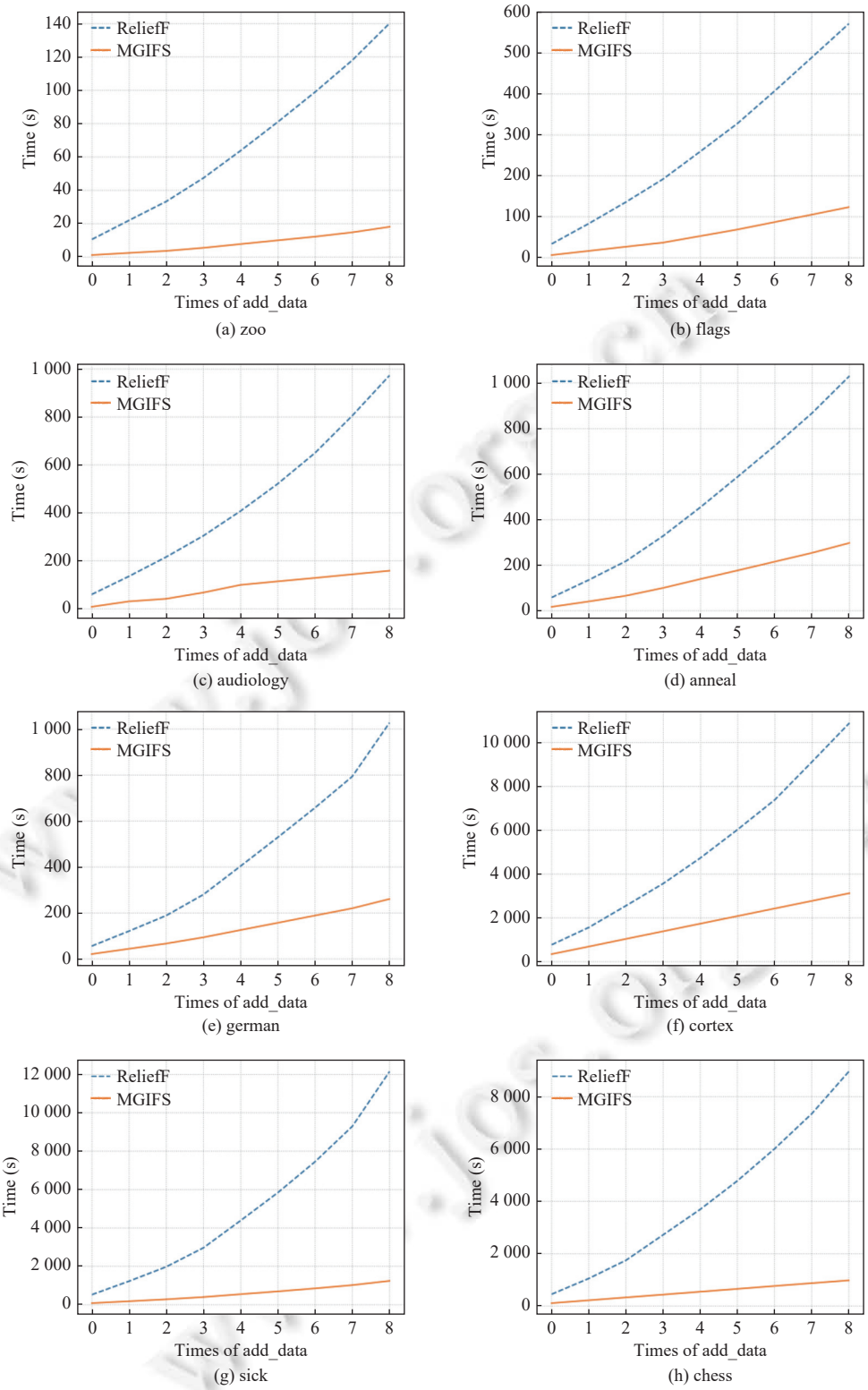

图 2 计算时间的比较结果

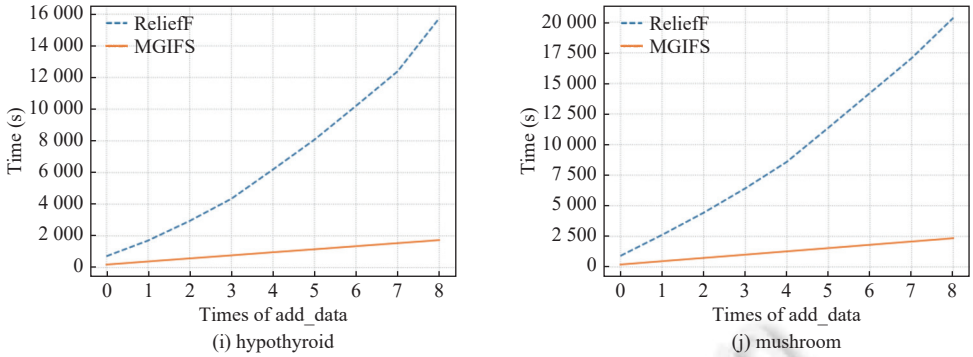

图 2 计算时间的比较结果 (续)

图 2 中的实验结果直观地显示了每次添加增量数据集后 MGIFS、ReliefF 两种算法的计算时间, 其中每个子图的横坐标表示添加数据的次数, 0 代表未添加数据, 纵坐标表示计算时间. 其对比结果可发现, MGIFS 算法的计算时间远小于 ReliefF 算法的计算时间, 并且随着增量数据集的不断添加, 数据规模的不断增大, MGIFS 算法的优势更加明显. 如上所述, 数据集整体规模不断增大, 为保证算法的有效性, ReliefF 算法需要不断增加抽样次数, 即增加更新特征权重值的数据样本集, 进而使得计算时间的增加也会比较明显. 而算法 MGIFS 的计算时间随数据规模的不断增大, 变化比较缓慢, 这也进一步验证了 MGIFS 的计算效率会受数据分布变化的影响, 如果数据分布变化比较缓慢, 该算法的计算耗时并不会明显增加; 如果某个时刻数据分布发生明显变化, 算法 MGIFS 可以迅速跟踪到并且及时调整数据粒的信息, 从而有效为下一次新数据增加后的信息发现提供指导. 综上, 算法 MGIFS 基于 ReliefF 算法, 融入了数据粒化, 多数据粒更新以及信息融合等思想和处理技巧, 能更快地找到一个有效的特征子集.

4.4 实验对比

基于上述的实验分析结果, 为进一步验证本文新算法 MGIFS 的可行性, 本节中选取了两个近年来提出的增量式特征选择算法作为对比算法, 分别为多粒度视角下基于知识粒度的启发式增量特征选择算法 (UARAO)[39]和邻域粒化条件熵的增量式属性约简 (LARNGCE)[40]. 为了保证对比实验的有效性, 本节中选取了与上述文献中相同的仿真实验模式, 即实验数据不再有随机生成数据, 而是将原数据集分成 10 等份, 多次添加来模拟动态数据变化的过程. 实验中选取了上述文献实验分析中使用的 UCI 数据集 (见表 7), 表 8 和表 9 分别给出了上述两个算法与 MGIFS 在 4 个数据集上的特征选择结果在两个分类器上进行十折交叉验证的分类精度值的对比.

表 7 对比实验数据集

数据集	样本数量	数值值特征数量	符号值特征数量	特征总数	类别数
yeast	1 484	8	1	9	13
wall	5 456	4	1	5	4
biodge	1 055	41	0	41	2
magic	19 020	10	0	10	2

表 8 SVM 分类精度对比

数据集	MGIFS	UARAO	IARNGCE
yeast	0.8567±0.0099	0.8567±0.0099	0.8567±0.0099
wall	0.9201±0.0363	0.9192±0.8107	**0.9341±0.3131**
biodge	**0.9266±0.5265**	0.8918±0.3562	0.9029±0.1593
magic	**0.9387±0.7462**	0.9278±0.2342	0.9356±0.0153
average	**0.910 5**	0.898 9	0.907 3

表 9 NBC 分类精度对比

数据集	MGIFS	UARAO	IARNGCE
yeast	0.8743±0.2716	0.8743±0.2716	0.8743±0.2716
wall	**0.9378±0.0395**	0.9356±0.0255	0.9372±0.0502
biodge	**0.9369±0.2828**	0.9027±0.1718	0.9346±0.3364
magic	**0.9603±0.0109**	0.9590±0.0145	0.9586±0.0096
average	**0.927 3**	0.917 9	0.926 1

从表 8 和表 9 的实验结果中可以看出, MGIFS 算法的分类精度在整体上比其他两种算法要高, 这显然是由于本文新算法在进行动态数据集的粒化过程中, 通过跟踪总体数据集上数据分布的变化, 来实时调整每个数据粒上类信息的分布, 进而较好地保证了数据粒上特征选择结果与总体数据集上特征选择结果尽可能的一致. 此外, 由于在现实应用中, 数据通常不是单一类型的, 大都是多种类型混合在一起的, 而 MGIFS 算法能够有效地处理符号取值和连续取值同时存在的混合型数据, 这为有效处理复杂且多元化的动态数据集提供了可以借鉴的新思路, 在实际应用中将会具有更广泛的使用价值.

5 总 结

本文针对动态大数据中面向数据分布变化的增量特征选择面临的挑战, 借鉴多粒度的处理机制, 构建了面向混合数据的动态粒化、数据粒更新、面向数据粒的特征权重求解以及多结果融合机制, 提出了多粒度视角下的面向动态混合数据的增量特征选择算法, 相关分析和实验结果也进一步证实了该算法的有效性和高效性. 研究成果有望丰富基于多粒度思想的动态数据挖掘与算法, 对开放环境下动态大数据的挖掘技术具有重要的理论意义和应用价值. 下一步的研究重点将是继续改进算法, 从而能够快速地跟踪动态数据集的分布变化, 并及时求解到数据集的最优特征子集.

References:

[1] Cai J, Luo JW, Wang SL, Yang S. Feature selection in machine learning: A new perspective. Neurocomputing, 2018, 300: 70–79. [doi: 10.1016/j.neucom.2017.11.077]

[2] Chen DG, Zhang L, Zhao SY, Hu QH, Zhu PF. A novel algorithm for finding reducts with fuzzy rough sets. IEEE Trans. on Fuzzy Systems, 2012, 20(2): 385–389. [doi: 10.1109/TFUZZ.2011.2173695]

[3] Li YH, Hu L, Gao WF. Multi-label feature selection based on sparse coefficient matrix reconstruction. Chinese Journal of Computers, 2022, 45(9): 1827–1841 (in Chinese with English abstract). [doi: 10.11897/SP.J.1016.2022.001827]

[4] Sang BB, Chen HM, Yang L, Li TR, Xu WH. Incremental feature selection using a conditional entropy based on fuzzy dominance neighborhood rough sets. IEEE Trans. on Fuzzy Systems, 2022, 30(6): 1683–1697. [doi: 10.1109/TFUZZ.2021.3064686]

[5] Zhang X, Mei CL, Chen DG, Li JH. Feature selection in mixed data: A method using a novel fuzzy rough set-based information entropy. Pattern Recognition, 2016, 56: 1–15. [doi: 10.1016/j.patcog.2016.02.013]

[6] Shu WH, Qian WB, Xie YH. Incremental feature selection for dynamic hybrid data using neighborhood rough set. Knowledge-based Systems, 2020, 194: 105516. [doi: 10.1016/j.knosys.2020.105516]

[7] Xie MR, She YH. Incremental mechanism of attribute reduction based on maximal-discernibility-pair in fuzzy rough sets. Fuzzy Systems and Mathematics, 2021, 35(6): 111–122 (in Chinese with English abstract).

[8] Wan JH, Chen HM, Yuan Z, Li TR, Yang XL, Sang BB. A novel hybrid feature selection method considering feature interaction in neighborhood rough set. Knowledge-based Systems, 2021, 227: 107167. [doi: 10.1016/j.knosys.2021.107167]

[9] Yang YJ, Wang W, Fu HY, Jay Kuo CC. On supervised feature selection from high dimensional feature spaces. APSIPA Trans. on Signal and Information Processing, 2022, 11(1): e31. [doi: 10.1561/116.00000016]

[10] Liu HW, Sun JG, Liu L, Zhang HJ. Feature selection with dynamic mutual information. Pattern Recognition, 2009, 42: 1330–1339. [doi: 10.1016/j.patcog.2008.10.028]

[11] Li TR, Ruan D, Geert W, Song J, Xu Y. A rough sets based characteristic relation approach for dynamic attribute generalization in data mining. Knowledge-based Systems, 2007, 20(5): 485–494. [doi: 10.1016/j.knosys.2007.01.002]

[12] Sun L, Zhang XY, Qian YH, Xu JC, Zhang SG. Feature selection using neighborhood entropy-based uncertainty measures for gene expression data classification. Information Sciences, 2019, 502: 18–41. [doi: 10.1016/j.ins.2019.05.072]

[13] Miao DQ, Zhao Y, Yao YY, Li HX, Xu FF. Relative reducts in consistent and inconsistent decision tables of the Pawlak rough set model. Information Sciences, 2009, 179(24): 4140–4150. [doi: 10.1016/j.ins.2009.08.020]

[14] Pan WB, Cheng G, Guo XJ, Wang Y. An embedded feature selection using selective ensemble for network traffic. Chinese Journal of Computers, 2014, 37(10): 2128–2138 (in Chinese with English abstract). [doi: 10.3724/SP.J.1016.2014.02128]

[15] Chen Y, Liu KY, Song JJ, Fujita H, Yang XB, Qian YH. Attribute group for attribute reduction. Information Sciences, 2020, 535: 64–80. [doi: 10.1016/j.ins.2020.05.010]

[16] Liang JY, Shi ZZ. The information entropy, rough entropy and knowledge granulation in rough set theory. Int'l Journal of Uncertainty,

Fuzziness and Knowledge-based Systems, 2004, 12(1): 37–46. [doi: 10.1142/S0218488504002631]

[17] Yang YY, Chen DG, Wang H, Wang XZ. Incremental perspective for feature selection based on fuzzy rough sets. IEEE Trans. on Fuzzy Systems, 2018, 26(3): 1257–1273. [doi: 10.1109/TFUZZ.2017.2718492]

[18] Zhang XY, Li JR. Incremental feature selection approach to interval-valued fuzzy decision information systems based on λ-fuzzy similarity self-information. Information Sciences, 2023, 625: 593–619. [doi: 10.1016/j.ins.2023.01.058]

[19] Bai LX, Li H, Gao WF, Xie J, Wang HQ. A joint multiobjective optimization of feature selection and classifier design for high-dimensional data classification. Information Sciences, 2023, 626: 457–473. [doi: 10.1016/j.ins.2023.01.069]

[20] Xu JC, Shen KL, Sun L. Multi-label feature selection based on fuzzy neighborhood rough sets. Complex & Intelligent Systems, 2022, 8(3): 2105–2129. [doi: 10.1007/s40747-021-00636-y]

[21] Wang GY, Yu H, Yang DC. Decision table reduction based on conditional information entropy. Chinese Journal of Computers, 2002, 25(7): 759–766 (in Chinese with English abstract). [doi: 10.3321/j.issn:0254-4164.2002.07.013]

[22] Qian YH, Liang JY. Combination entropy and combination granulation in rough set theory. Int'l Journal of Uncertainty, Fuzziness and Knowledge-based Systems, 2008, 16(2): 179–193. [doi: 10.1142/S0218488508005121]

[23] Qian YH, Liang JY. Combination entropy and combination granulation in incomplete information system. Lecture Notes in Artificial Intelligence, 2006, 4062: 184–190.

[24] Xu WH, Zhang XY, Zhong JM, Zhang WX. Heuristic algorithm for attributes reduction in ordered information systems. Computer Engineering, 2010, 36(17): 69–71 (in Chinese with English abstract). [doi: 10.3969/j.issn.1000-3428.2010.17.024]

[25] Liang JY, Wang F, Dang CY, Qian YH. An efficient rough feature selection algorithm with a multi-granulation view. Int'l Journal of Approximate Reasoning, 2012, 53(6): 912–926. [doi: 10.1016/j.ijar.2012.02.004]

[26] Hu QH, Yu DR, Xie ZX. Information-preserving hybrid data reduction based on fuzzy-rough techniques. Pattern Recognition Letters, 2006, 27(5): 414–423. [doi: 10.1016/j.patrec.2005.09.004]

[27] Wang F, Liang JY. An efficient feature selection algorithm for hybrid data. Neurocomputing, 2016, 193: 33–41. [doi: 10.1016/j.neucom.2016.01.056]

[28] Xiao LS, Wang HJ, Yang Y. Semi-supervised feature selection based on attribute dependency and hybrid constraint. Journal of Computer Applications, 2015, 35(S2): 80–84 (in Chinese with English abstract).

[29] Shu WH, Yan ZC, Yu JH, Qian WB. Information gain-based semi-supervised feature selection for hybrid data. Applied Intelligence, 2023, 53(6): 7310–7325. [doi: 10.1007/s10489-022-03770-3]

[30] Chen JZ, Hu JJ, Zhang GQ. Feature selection based on gain ratio in hybrid incomplete information systems. In: Proc. of the 16th Int'l Conf. on Intelligent Systems and Knowledge Engineering. Chengdu: IEEE, 2021. 728–735. [doi: 10.1109/ISKE54062.2021.9755425]

[31] Hu SD, Miao DQ, Yao YY. Three-way label propagation based semi-supervised attribute reduction. Chinese Journal of Computers, 2021, 44(11): 2332–2343 (in Chinese with English abstract). [doi: 10.11897/SP.J.1016.2021.02332]

[32] Zhong QQ, Wang L, Yang W, Liu C. Incremental attribute reduction based on knowledge granularity under incomplete data. Journal of Physics: Conf. Series, 2021, 2025(1): 012042. [doi: 10.1088/1742-6596/2025/1/012042]

[33] Liang JY, Wang F, Dang CY, Qian YH. A group incremental approach to feature selection applying rough set technique. IEEE Trans. on Knowledge and Data Engineering, 2014, 26(2): 294–308. [doi: 10.1109/TKDE.2012.146]

[34] Shu WH, Shen H. Incremental feature selection based on rough set in dynamic incomplete data. Pattern Recognition, 2014, 47(12): 3890–3906. [doi: 10.1016/j.patcog.2014.06.002]

[35] Yu K, Yang YJ, Ding W. Causal feature selection with missing data. ACM Trans. on Knowledge Discovery from Data, 2022, 16(4): 66. [doi: 10.1145/3488055]

[36] Yu K, Liu L, Li JY, Ding W, Le TD. Multi-source causal feature selection. IEEE Trans. on Pattern Analysis and Machine Intelligence, 2020, 42(9): 2240–2256. [doi: 10.1109/TPAMI.2019.2908373]

[37] Kira K, Rendell LA. The feature selection problem: Traditional methods and a new algorithm. In: Proc. of the 10th National Conf. on Artificial Intelligence. San Jose: AAAI Press, 1992. 129–134.

[38] Wang F, Wei W, Liang JY. A group incremental approach for feature selection on hybrid data. Soft Computing, 2022, 26(8): 3663–3677. [doi: 10.1007/s00500-022-06838-x]

[39] Jing YG, Li TR, Fujita H, Yu Z, Wang B. An incremental attribute reduction approach based on knowledge granularity with a multi-granulation view. Information Sciences, 2017, 411: 23–38. [doi: 10.1016/j.ins.2017.05.003]

[40] Zhao XL, Yang Y. Incremental attribute reduction algorithm based on neighborhood granulation conditional entropy. Control and Decision, 2019, 34(10): 2061–2072 (in Chinese with English abstract). [doi: 10.13195/j.kzyjc.2018.0138]

附中文参考文献:

[3] 李永豪, 胡亮, 高万夫. 基于稀疏系数矩阵重构的多标记特征选择. 计算机学报, 2022, 45(9): 1827–1841. [doi: 10.11897/SP.J.1016.2022.001827]

[7] 谢铭悦, 折延宏. 基于模糊粗糙集极大差别对的增量属性约简. 模糊系统与数学, 2021, 35(6): 111–122.

[14] 潘吴斌, 程光, 郭晓军, 王艳. 基于选择性集成策略的嵌入式网络流特征选择. 计算机学报, 2014, 37(10): 2128–2138. [doi: 10.3724/SP.J.1016.2014.02128]

[21] 王国胤, 于洪, 杨大春. 基于条件信息熵的决策表约简. 计算机学报, 2002, 25(7): 759–766. [doi: 10.3321/j.issn:0254-4164.2002.07.013]

[24] 徐伟华, 张晓燕, 钟坚敏, 张文修. 序信息系统中属性约简的启发式算法. 计算机工程, 2010, 36(17): 69–71. [doi: 10.3969/j.issn.1000-3428.2010.17.024]

[28] 肖丽莎, 王红军, 杨燕. 基于属性依赖的混合约束半监督特征选择. 计算机应用, 2015, 35(S2): 80–84.

[31] 胡声丹, 苗夺谦, 姚一豫. 基于三支标签传播的半监督属性约简. 计算机学报, 2021, 44(11): 2332–2343. [doi: 10.11897/SP.J.1016.2021.02332]

[40] 赵小龙, 杨燕. 基于邻域粒化条件熵的增量式属性约简算法. 控制与决策, 2019, 34(10): 2061–2072. [doi: 10.13195/j.kzyjc.2018.0138]

王锋(1984－), 女, 博士, 副教授, 主要研究领域为特征选择, 粒计算, 机器学习.

梁吉业(1962－), 男, 博士, 教授, CCF 会士, 主要研究领域为粒计算, 机器学习.

姚珍(1999－), 女, 硕士生, 主要研究领域为机器学习, 特征选择.