计算机科学与探索
Journal of Frontiers of Computer Science and Technology
1673-9418/2023/17(12)-2928-14
doi: 10.3778/j.issn.1673-9418.2308062

面向层次结构数据的增量特征选择

折延宏[1+],黄婉丽[2],贺晓丽[1],钱　婷[1]
1. 西安石油大学 理学院,西安 710065
2. 西安石油大学 计算机学院,西安 710065
+ 通信作者 E-mail: yanhongshe@xsyu.edu.cn

摘　要:随着大数据时代的到来,数据样本量越来越多,维度越来越高,同时样本标签存在复杂的层次结构关系。采用包含策略,研究了基于依赖度的分层分类增量特征选择,解决了标签具有树结构且标签分布在任意节点的分层分类问题。首先,利用标签之间的层次结构,采用包含策略来缩小负样本空间。其次,使用模糊粗糙集理论,提出了一个基于包含策略的模糊粗糙集模型,设计了一个基于包含策略的依赖度计算算法和一个非增量特征选择算法。基于此,引入增量机制,提出了基于包含策略的依赖度增量更新方法,设计了两个基于两种策略的增量特征选择算法。最后,将此方法与基于兄弟策略的依赖度进行对比,通过实验验证了所提方法的可行性与高效性。

关键词:模糊粗糙集;依赖度;分层分类;增量特征选择;包含策略
文献标志码:A　　**中图分类号:**TP311

Incremental Feature Selection Oriented for Data with Hierarchical Structure

SHE Yanhong[1+], HUANG Wanli[2], HE Xiaoli[1], QIAN Ting[1]
1. College of Science, Xi'an Shiyou University, Xi'an 710065, China
2. College of Computer, Xi'an Shiyou University, Xi'an 710065, China

Abstract: In the big data era, the sample size is becoming increasingly large, the data dimensionality is also becoming extremely high, moreover, there exists hierarchical structure between different class labels. This paper investigates incremental feature selection for hierarchical classification based on the dependency degree of inclusive strategy and solves the hierarchical classification problem where labels are distributed at arbitrary nodes in tree structure. Firstly, the inclusive strategy is used to reduce the negative sample space by exploiting the hierarchical label structure. Secondly, a new fuzzy rough set model is introduced based on inclusive strategy, and a dependency calculation algorithm based on the inclusive strategy and a non-incremental feature selection algorithm are also proposed. Then, the dependency degree based on the inclusive strategy is proposed by adopting the incremental mechanism. Based on these, two incremental feature selection frameworks based on two strategies are designed. Lastly, a comparative study with the method based on the sibling strategy is performed. The feasibility and efficiency of the proposed algorithms are verified by numerical experiments.

Key words: fuzzy rough sets; dependency degree; hierarchical classification; incremental feature selection; inclusive strategy

基金项目:国家自然科学基金(61976244,12001422,12171388);陕西省自然科学基金(2021JQ-580,2023-JCYB-597,2023-JCYB-027)。
This work was supported by the National Natural Science Foundation of China (61976244, 12001422, 12171388), and the Natural Science Foundation of Shaanxi Province (2021JQ-580, 2023-JCYB-597, 2023-JCYB-027).
收稿日期:2023-08-21　　**修回日期:**2023-11-22

在大数据时代，数据标签类型和数量急剧增加，标签之间往往具有某种特殊的关系，其中层次结构[1-3]最具有代表性，包括树结构和图结构。标签具有层次结构的分类问题是当今的研究热点。用层次结构进行大规模分类学习有很大的优势，对于超多类问题，可以利用层次结构将超多类问题分解为多个子类学习任务，能有效提高建模的效率。

基于粒计算思想的分层分类建模是一种符合人脑认知规律的数据建模方法。Bellmund 等人[4]在 *Science* 上发表的论文认为人脑认知和思维过程依靠多粒度的知识层次结构完成。Aronov 等人[5]在 *Nature* 上发表的论文认为人脑的思考和认知过程所形成的知识呈现出一种低维的几何结构。文献[6]给出了一种在样本标记粒度不够细化的情况下利用层次信息进行建模的方法。文献[7]给出了一种能同时体现共有特征与固有特征的分层特征选择方法。在实际应用中，用户需求的多层次/多粒度也决定了挖掘任务的多层次/多粒度特征。而粒计算是模拟人类思考和解决大规模复杂问题的自然模式[8-10]。模糊粗糙集是粒计算中的一个重要模型，因此利用模糊粗糙集对具有层次结构的数据进行粒化处理可以更充分地学习数据中蕴含的信息。

目前，模糊粗糙集[11-12]在处理平面分类（与分层分类相对应）问题中已有许多应用。许多学者将模糊粗糙集理论应用到特征选择（也称为属性约简）中，采用依赖度[13-14]、条件熵[15-16]、辨识矩阵[17]和相对辨识关系[18-19]等作为特征选择的评价指标。基于依赖函数的启发式算法[11]是模糊决策系统求约简的先驱工作。之后 Bhatt 等人[20]定义了一个紧凑域来降低文献[11]的时间复杂度。Hu 等人[21]提出了基于信息熵的模糊粗糙集的特征选择算法。为了找到合适的约简，Tsang 等人[22]引入了基于辨识矩阵的方法来处理模糊粗糙集。Chen 等人[18]提出了样本对选择方法来搜索可识别矩阵中的所有最小元素，只使用所有的最小元素来寻找模糊决策系统的约简。Wang 等人[14]提出了一种基于模糊粗糙集的特征选择拟合模型，以更好地反映所选择特征子集的分类能力。

模糊粗糙集理论于2019年首次被应用到分层分类特征选择的研究中[23]，文中利用类别之间的层次结构，用排他策略、包含策略和兄弟策略来缩小负样本空间，从而减少求解下近似的计算量，提出基于兄弟策略的依赖度计算算法和特征选择算法。排他策略与平面的分类相同，即如果 A 是正样本，A 以外的其他样本是负样本，在该策略下的负样本搜索空间非常大，因此使用合理的策略非常重要，目前大多关注的是兄弟策略，该策略只把 A 的兄弟节点中样本看作负样本，这种策略考虑的是同层次的横向关系，忽略了不同层次之间样本的关系。包含策略比兄弟策略更复杂，不仅考虑同层的横向关系，也考虑上下层之间的父子关系。然而，已有的研究工作大多关注的是兄弟策略，相比而言包含策略考虑的层次范围更广，可以更好地弥补兄弟策略未能考虑上下层关系的缺点，这也是本文使用包含策略的一个研究动机。

此外，已有的分层分类的特征选择研究大多利用标签的层次关系构建正则项、最小化损失函数和正则项，基于此建立优化模型[24-27]。然而，上述的方法都是针对静态数据集。现实场景中数据是不断动态增加的，相应地，一些基于动态数据信息的增量学习方法[28-32]被提出。然而，这些方法大多局限于平面分类中，分层分类中涉及的较少。Fan 等人[33]介绍了一种基于多核模糊粗糙集的增量层次分类方法，但它们更侧重于目标概念的粗糙近似的更新，而不是特征选择。Luo 等人[34]提出了一种迭代的增量粗糙集方法。而在分层分类下考虑使用包含策略的模糊粗糙集的增量研究也非常必要。

综上，本文研究的动机如下：(1)在分层分类问题中，考虑模糊粗糙集的增量可以更好地模拟现实数据。(2)已有的研究大多针对的是数据标签只分布在叶子节点，现实世界中标签具有任意性，研究标签分布在任意节点更具有现实意义。(3)包含策略能够更好地学习标签之间的层次信息，更适合应用标签分布在任意节点的场景中。

因此，将针对标签具有树结构，且标签分布在叶子节点和内部节点情形的动态数据集，将包含策略应用到模糊粗糙集模型中，基于此研究分层分类的增量特征选择算法。基于文献[23]提出一个基于包含策略的模糊粗糙集模型，设计一个非增量特征选择算法，并引入增量机制，即当有新样本加入时，研究下近似、正域和依赖度的增量更新策略。由此，本文设计一个增量特征选择算法，并提出基于两种不同策略的增量特征选择框架。最后，通过数值实验验证所提算法是有效的。

本文主要贡献如下：

(1)提出基于包含策略的模糊粗糙集模型，并设计基于该模型的非增量特征选择算法；

(2)在该模型中引入增量机制，提出增量更新方

法,以及基于包含策略的依赖度更新算法和增量特征选择算法,两个版本的增量特征选择框架;

(3)研究动态数据集,且标签分布在内部节点和叶子节点,使得所提算法适用范围更广。

1 预备知识

本文所使用的符号含义如表1所示。

表1 符号描述

Table 1 Symbol description

符号	含义
R_B	由 B 导出的模糊 T -相似关系
D_{tree}	关于 D 的树结构
$des(d_i)$	d_i 的子孙节点
$inc(d_i)$	d_i 的包含节点
$X_{inc}^{d_i}$	d_i 的包含正样本
$\overline{inc}(d_i)$	d_i 的包含负节点
$\overline{X}_{inc}^{d_i}$	d_i 的包含负样本
$\Delta\overline{X}_{inc}^{d_i'}$	$\overline{X}_{inc}^{d_i}$ 中新加入的样本
$\boldsymbol{D}_B^2$	关于属性子集 B 的距离平方矩阵
rem	剩余属性
red	属性约简
D_{inc}^U	U 中下近似可能会发生变化的节点

在模糊决策表 $\langle U,C,D\rangle$ 中,U 是非空对象集,C 是非空实值型条件属性集,D 是非空符号型决策属性集。本文只考虑 D 中只有一个决策属性,也就是单标签问题。令 $B\subseteq C$,如果二元关系 R_B 满足以下条件:$\forall x,y,z\in U$,(1)自反性 $R_B(x,x)=1$;(2)对称性 $R_B(x,y)=R_B(y,x)$;(3) T -传递性 $T(R_B(x,y),R_B(y,z))\leqslant R_B(x,z)$,那么 R_B 为模糊 T -相似关系[35],其中,T 为三角模[21]。

使用高斯函数[36]来计算模糊 T -相似关系:

$$R_B(x,y)=\exp\left(-\frac{(D_B(x,y))^2}{2\sigma^2}\right)$$

其中,σ 为参数,$D_B(x,y)=\sqrt{\sum_{a\in B}(a(x)-a(y))^2}$ 为 x 与 y 之间的距离,$a(x)$ 为 x 在属性 a 下的值。

序对 $(D_{\text{tree}},\prec)$[23]用来描述决策类的层次结构,$D_{\text{tree}}=\{d_0,d_1,\cdots,d_l\}$,其中 d_0 为根节点,不是真实的类,l 是类的个数。"$\prec$"代表"子类"关系且满足以下条件:(1)反对称性 $\forall d_i,d_j\in D_{\text{tree}}$,如果 $d_i\prec d_j$,那么 $d_j\nprec d_i$;(2)反自反性 $\forall d_i\in D_{\text{tree}}$,$d_i\nprec d_i$;(3)传递性 $\forall d_i,d_j,d_k\in D_{\text{tree}}$,如果 $d_i\prec d_j$,$d_j\prec d_k$,那么 $d_i\prec d_k$。

为了便于分析,重新叙述包含策略[23]的定义。称 $des(d_i)\cup\{d_i\}$ 为 d_i 的包含节点,记为 $inc(d_i)$;称 $inc(d_i)$ 中的样本为 d_i 的包含正样本,记为 $X_{inc}^{d_i}=\{x:x\in d,d\in(des(d_i)\cup\{d_i\})\}$;称 $D_{\text{tree}}(des(d_i)\cup\{d_i\})$ 为 d_i 的包含负节点,记为 $\overline{inc}(d_i)$;称 $\overline{inc}(d_i)$ 中的样本为 d_i 的包含负样本,记为 $\overline{X}_{inc}^{d_i}=\{x:x\in d,d\in D_{\text{tree}}\backslash(des(d_i)\cup\{d_i\})\}$。显然,$X_{inc}^{d_i}\cup\overline{X}_{inc}^{d_i}=U$。

例1 图1是一个 D_{tree} 的示例,$D_{\text{tree}}=\{d_0,d_1,\cdots,d_6\}$,其中 d_0 是树结构的根节点,不是真实的标签,不参与求子孙节点和包含节点的运算。以 d_1 为例,d_1 的子孙节点为 $des(d_1)=\{d_3,d_4\}$,d_1 的包含节点 $inc(d_1)=\{d_1,d_3,d_4\}$,d_1 的包含负节点 $\overline{inc}(d_1)=\{d_2,d_5,d_6\}$。

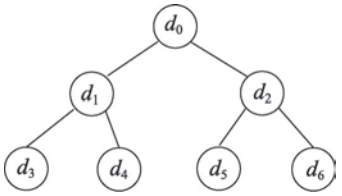

图1 标签的树结构示例

Fig.1 Example of tree structure for labels

当模糊决策表具有层次结构时,称其为分层决策表,记作 $\langle U,C,D_{\text{tree}}\rangle$。在分层决策表 $\langle U,C,D_{\text{tree}}\rangle$ 中,R_B 是由 $B\subseteq C$ 导出的模糊 T -相似关系。U 被决策属性划分为 $\{d_1,d_2,\cdots,d_l\}$,l 为类的个数。基于包含策略的上下近似算子[23]被定义为:$\forall x\in U$

$$\overline{R_{B\,inc}^U}\,d_i(x)=\sup_{y\in X_{inc}^{d_i}}R_B(x,y)$$

$$\underline{R_{B\,inc}^U}\,d_i(x)=\inf_{y\in\overline{X}_{inc}^{d_i}}(1-R_B(x,y))$$

2 基于包含策略的非增量特征选择

本章先研究基于包含策略的模糊粗糙集模型,基于此,设计非增量依赖度计算算法和非增量特征选择算法。

2.1 基于包含策略的模糊粗糙集模型

在分层分类问题中,负样本只是选取了 U 中除 d_i 其余样本的一部分,因此当 $x\notin d_i$ 时,不一定满足 $\underline{R_{B\,inc}^U}\,d_i(x)=0$。为了下近似具有更好的性质,将文献[23]中基于包含策略的下近似修正如下。

定义1 在分层决策表 $\langle U,C,D_{\text{tree}}\rangle$ 中,R_B 是由 $B\subseteq C$ 导出的模糊 T -相似关系,d_i 的下近似可以定义为:

$$\underline{R_{B\,inc}^U}\,d_i(x)=\begin{cases}\inf_{y\in\overline{X}_{inc}^{d_i}}(1-R_B(x,y)), & x\in d_i\\ 0, & x\notin d_i\end{cases}$$

定义2 在分层决策表 $\langle U,C,D_{\text{tree}}\rangle$ 中，R_B 是由 $B\subseteq C$ 导出的模糊 T -相似关系，关于属性集 B 的正域可以定义为：

$$POS^U_{B\,inc}(D_{\text{tree}})=\bigcup_{i=1}^{l}\underline{R^U_{B\,inc}}d_i$$

通过定义2可以得出以下结论。

定理1 在分层决策表 $\langle U,C,D_{\text{tree}}\rangle$ 中，R_B 是一个由 $B\subseteq C$ 导出的模糊 T -相似关系，则 $POS^U_{B\,inc}(D_{\text{tree}})(x)=\underline{R^U_{B\,inc}}d_i(x),\ x\in d_i$。

证明 由定义2可得，$POS^U_{B\,inc}(D_{\text{tree}})(x)=(\bigcup_{i=1}^{l}\underline{R^U_{B\,inc}}d_i)(x)=\sup_{i=1}^{l}\underline{R^U_{B\,inc}}d_i(x)$，由定义1可得当 $x\notin d_i$ 时，$\underline{R^U_{B_{sinc}}}d_i(x)=0$，故 $\sup_{i=1}^{l}\underline{R^U_{B\,inc}}d_i(x)=\underline{R^U_{B\,inc}}d_i(x),\ x\in d_i$。

定义3 在分层决策表 $\langle U,C,D_{\text{tree}}\rangle$ 中，关于属性集 B 的依赖度可以定义为：

$$\gamma^U_{B\,inc}(D_{\text{tree}})=\frac{|POS^U_{B\,inc}(D_{\text{tree}})|}{|U|}$$

性质1 在分层决策表 $\langle U,C,D_{\text{tree}}\rangle$ 中，有：

$$\gamma^U_{B\,inc}(D_{\text{tree}})\leqslant\gamma^U_{C\,inc}(D_{\text{tree}})$$

证明 因为 $B\subseteq C$，所以 $\forall x,y\in U$，那么，$R_B(x,y)=\exp\left(-\frac{D_B(x,y)^2}{2\sigma^2}\right)\geqslant R_C(x,y)=\exp\left(-\frac{D_C(x,y)^2}{2\sigma^2}\right)$。因此，$\underline{R^U_{B\,inc}}d_i(x)\leqslant\underline{R^U_{C\,inc}}d_i(x)$。由定义2可得 $POS^U_{B\,inc}d_i(x)\leqslant POS^U_{C\,inc}d_i(x)$，再由定义3可得出上述结论。

由性质1可知依赖度关于属性集的变化是单调的，这是依赖度可以作为特征选择的评价指标的依据。下面基于定义3提出一个适用于分层分类的属性约简定义。

定义4 对于分层决策表 $\langle U,C,D_{\text{tree}}\rangle$，$B\subseteq C$，如果 B 满足条件：(1) $\gamma^U_{C\,inc}(D_{\text{tree}})-\gamma^U_{B\,inc}(D_{\text{tree}})\leqslant\varepsilon$；(2)对 $\forall a\in B$，$\gamma^U_{C\,inc}(D_{\text{tree}})-\gamma^U_{B-\{a\}inc}(D_{\text{tree}})>\varepsilon$，则称 B 为一个属性约简。其中，ε 为参数且 $\varepsilon\in[0,1]$。

在定义4中 ε 用来衡量 $\gamma^U_{C\,inc}(D_{\text{tree}})$ 和 $\gamma^U_{B\,inc}(D_{\text{tree}})$ 的接近程度。第一个条件使 $\gamma^U_{C\,inc}(D_{\text{tree}})$ 和 $\gamma^U_{B\,inc}(D_{\text{tree}})$ 的差值在 $[0,\varepsilon]$ 之间，第二个条件使 $\gamma^U_{C\,inc}(D_{\text{tree}})$ 和 $\gamma^U_{B-\{a\}inc}(D_{\text{tree}})$ 的差值在 $[\varepsilon,1]$。

2.2 算法

基于包含策略的依赖度的定义，本文提出非增量依赖度计算算法和非增量特征选择算法。

记 $R_B(x,y)$ 中的 $(D_B(x,y))^2$ 为 $D^2_B(x,y)$。称 $\boldsymbol{D}^2_B=(D^2_B(x,y))_{|U|\times|U|}$ 为距离平方矩阵。将 $\gamma^U_{C\,inc}(D_{\text{tree}})$ 简记为 $\gamma^U_C(D_{\text{tree}})$。

算法1 基于包含策略的非增量依赖度计算算法(Inc-NIDC)

输入：分层决策表 $\langle U,C,D_{\text{tree}}\rangle$，属性集 B，包含负样本 $\overline{X}^{d_i}_{inc}(\forall d_i\in U/D_{\text{tree}})$ 和 U 的划分 U/D_{tree}。

输出：依赖度 $\gamma^U_B(D_{\text{tree}})$。

1. 初始化下近似 $\underline{R^U_{B\,inc}}=2$，$sum=0$
2. For $d_i\in U/D_{\text{tree}}$ do
3. For $x\in d_i$ do
4. For $y\in\overline{X}^{d_i}_{inc}$ do
5. 计算距离平方 $D^2_B(x,y)$，并存入距离平方矩阵 $\boldsymbol{D}^2_B$
6. 计算 $1-R_B(x,y)$
7. If $1-R_B(x,y)<\underline{R^U_{B\,inc}}$ then
8. $\underline{R^U_{B\,inc}}=1-R_B(x,y)$
9. End if
10. End for
11. $sum=sum+\underline{R^U_{B\,inc}}$
12. End for
13. End for
14. $\gamma^U_B(D_{\text{tree}})=\frac{sum}{|U|}$
15. Return $\gamma^U_B(D_{\text{tree}})$

算法1是基于包含策略的非增量依赖度计算算法。第1步初始化 $\underline{R^U_{B\,inc}}$ 为大于1的数，第5步计算距离平方矩阵，时间复杂度为 $O(|C|)$；第4~10步是计算 $\underline{R^U_{B\,inc}}d_i(x)$，由于第2步和第3步是个双循环，第4~10步被计算了 $|d_1|+|d_2|+\cdots+|d_l|=|U|$ 次，因此第2~13步的时间复杂度为 $O(|C||U|^2)$；第14步是计算关于 B 的依赖度。综上，算法1的时间复杂度为 $O(|C||U|^2)$。

算法2 基于包含策略的非增量特征选择算法(Inc-NIFS)

输入：分层决策表 $\langle U,C,D_{\text{tree}}\rangle$。

输出：属性约简 red。

1. 初始化 $B=\varnothing$，$rem=C$，$\gamma^U_B(D_{\text{tree}})=0$
2. 计算 $U/D_{\text{tree}}=\{d_1,d_2,\cdots,d_l\}$
3. 寻找每个 $d_i\in U/D_{\text{tree}}$ 的包含负节点 $\overline{inc}(d_i)$ 和包含负样本 $\overline{X}^{d_i}_{inc}$
4. 通过Inc-NIDC计算 $\gamma^U_C(D_{\text{tree}})$
5. While $\gamma^U_C(D_{\text{tree}})-\gamma^U_B(D_{\text{tree}})>\varepsilon$ do
6. For $a_0\in rem$ do
7. 通过Inc-NIDC计算 $\gamma^U_{B\cup\{a_0\}}(D_{\text{tree}})$
8. End for

9. 找出一个属性 $a=\arg(\max_{a_0\in rem}\gamma^U_{B\cup\{a_0\}}(D_{\text{tree}}))$

10. $rem=rem-\{a\}$

11. $B=B\cup\{a\}$

12. End while

13. 令 $Q=B$

14. For $a\in Q$ do

15. 通过 Inc-NIDC 计算 $\gamma^U_{B-\{a\}}(D_{\text{tree}})$

16. If $\gamma^U_C(D_{\text{tree}})-\gamma^U_{B-\{a\}}(D_{\text{tree}})\leqslant\varepsilon$ then

17. $B=B-\{a\}$

18. End if

19. End for

20. Return $red=B$

算法 2 是基于包含策略的非增量特征选择算法。“*rem*”表示剩余属性,“*red*”表示属性约简。第 2 步是求 U 的划分,时间复杂度为 $O(l|U|)$。第 3 步计算每个 $d_i\in U/D_{\text{tree}}$ 的包含负节点和包含负样本,时间复杂度为 $O(l^2)$。第 4 步通过 Inc-NIDC 计算 $\gamma^U_C(D_{\text{tree}})$,时间复杂度为 $O(|C||U|^2)$。第 5~12 步通过 Inc-NIDC 计算依赖度,采用启发式思想添加属性,时间复杂度为 $O(|C|^3|U|^2)$。第 13~19 步删除 B 中的冗余属性直到满足定义 4 中的第二个条件为止,时间复杂度为 $O(|C|^2|U|^2)$。综上,算法 2 的时间复杂度为 $O(|C|^3|U|^2)$。

3 基于包含策略的增量特征选择

本章首先研究基于包含策略的模糊粗糙集的增量更新方法,然后基于此研究其增量算法,并基于两种特征选择策略提出两个版本的增量特征框架。

3.1 基于包含策略的模糊粗糙集的增量更新方法

本节首先研究当分层决策表加入一些新样本时,下近似的增量更新方法,进而探究正域以及依赖度的更新方法。

3.1.1 下近似的增量更新

令 $\langle U,C,D_{\text{tree}}\rangle$ 为原分层决策表,其中 U 被决策属性划分为 $\{d_1,d_2,\cdots,d_l\}$,l 为决策类个数。ΔU 为新加入样本集,ΔU 被决策属性划分为 $\{\Delta d_1,\Delta d_2,\cdots,\Delta d_k\}$,$k$ 为新加入样本的决策类个数。加入样本后的分层决策表记为 $\langle U',C,D_{\text{tree}}\rangle$,其中 U' 被决策属性划分为 $\{d_1',d_2',\cdots,d_l'\}$。$\forall i\in\{1,2,\cdots,k\}$, $d_i'=d_i\cup\Delta d_i$; $\forall i\in\{k+1,k+2,\cdots,l\}$, $d_i'=d_i$; $\forall i\neq j\in\{1,2,\cdots,l\}$, $d_i\cap d_j=\varnothing$, $d_i'\cap d_j'=\varnothing$ 。

将样本分为 $x\in U$ 和 $x\in\Delta U$ 两种情况来研究下近似的变化。

(1)情况 1:$x\in U$ 。

首先寻找 U 中的下近似可能发生变化的节点。当使用包含策略时,下近似可能发生改变的节点与新加入样本所属的节点有关(这里的节点也是决策表中的类)。从定义 1 中,可以看到 y 的范围影响下近似的变化。现假设新加入样本在同一个类 d_t 中,如果 $d_i(x\in d_i)$ 的包含负样本中含有 d_t 的样本,即 $d_t\in D_{\text{tree}}\backslash(des(d_i)\cup\{d_i\})$,则 d_i 的下近似可能改变。由于这些节点关系较复杂,将问题转化为寻找下近似不受加入样本影响的节点。当 $d_t\in des(d_i)\cup\{d_i\}$ 时,d_i 的下近似不会改变。等价于存在新加入样本的类为 d_t,在 $anc(d_t)\cup\{d_t\}$ 中的节点下近似不会发生改变。也就是 $D_{\text{tree}}\backslash(anc(d_t)\cup\{d_t\})$ 中节点的下近似可能会发生变化。将所有下近似可能发生变化的节点集合记为 D^U_{inc}。$D^U_{inc}=\bigcup_{t=1}^{k}(D_{\text{tree}}\backslash(anc(d_t)\cup\{d_t\}))=D_{\text{tree}}\backslash\bigcap_{t=1}^{k}(anc(d_t)\cup\{d_t\})$。$D^U_{inc}$ 中样本之外的其他样本在所属类的下近似中的隶属度一定不会发生变化。

(2)情况 2:$x\in\Delta U$ 。

ΔU 是新加入的样本,在原分层决策表中没有计算过 x 在所属类的下近似中的隶属度,因此需要额外的计算。

下面通过例 2 展示直观判断与使用情况 1 的计算方法求下近似可能会发生变化的节点。

例 2 表 2 为原分层决策表,将表 3 中的样本加入表 2,原标签树结构的变化如图 2 所示。由于 $y\in\overline{inc}(d_i)$,

表 2 原分层决策表的示例数据

Table 2 Example data of original hierarchical decision table

U	C	D
x_1	0	d_1
x_2	0.12	d_1
x_3	0.19	d_2
x_4	0.37	d_2
x_5	0.45	d_3
x_6	0.49	d_3
x_7	0.31	d_4
x_8	0.62	d_4
x_9	0.35	d_5
x_{10}	0.81	d_5
x_{11}	0.89	d_6
x_{12}	0.92	d_6

表3 添加样本的示例数据

Table 3 Example data of incoming samples

ΔU	C	D
x_{13}	0.15	d_1
x_{14}	0.20	d_1
x_{15}	0.31	d_2
x_{16}	0.35	d_2
x_{17}	0.50	d_3
x_{18}	0.52	d_3

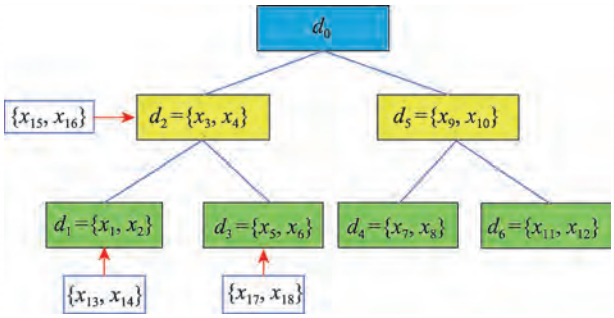

图2 添加样本时树结构变化

Fig.2 Change of tree structure while incoming samples

而下近似的变化只与 y 有关，只判断加入样本后每个节点的包含负节点是否有新加入样本。

（1）基于包含策略的下近似的定义来直观判断下近似可能发生变化的节点。由于 D_{tree} 中 d_0 不是真实的类，故不参与运算。$\overline{inc}(d_1)=\{d_2,d_3,d_4,d_5,d_6\}$，$\overline{inc}(d_2)=(d_4,d_5,d_6)$，$\overline{inc}(d_3)=(d_1,d_2,d_4,d_5,d_6)$，$\overline{inc}(d_4)=(d_1,d_2,d_3,d_5,d_6)$，$\overline{inc}(d_5)=(d_1,d_2,d_3)$，$\overline{inc}(d_6)=(d_1,d_2,d_3,d_4,d_5)$，由表2可知，只有 d_1、d_2 和 d_3 有新样本加入。因此，$\overline{inc}(d_i)$ 中只要包含 d_1、d_2 和 d_3 其中任何一个，d_i 的下近似都可能会发生变化。因此，下近似可能会发生变化的节点为 $\{d_1,d_3,d_4,d_5,d_6\}$。

（2）通过 D_{inc}^U 的计算方法判断下近似可能发生变化的节点。$anc(d_1)\cup\{d_1\}=\{d_1,d_2\}$，$anc(d_2)\cup\{d_2\}=\{d_2\}$，$anc(d_3)\cup\{d_3\}=\{d_2,d_3\}$，则 $\bigcap_{t=1}^{k}(anc(d_t)\cup\{d_t\})=\{d_2\}$。因此，$D_{inc}^U=D_{\text{tree}}\setminus\bigcap_{t=1}^{k}(anc(d_t)\cup\{d_t\})=\{d_1,d_3,d_4,d_5,d_6\}$。

使用 D_{inc}^U 的计算所得结果与基于包含策略的下近似定义的计算结果一致。通过例2可以进一步验证 D_{inc}^U 理论分析是正确的，D_{inc}^U 的计算更易于设计算法实现。由情况1和情况2可以得到以下定理。

定理2 对于原分层决策表 $\langle U,C,D_{\text{tree}}\rangle$ 和新分层决策表 $\langle U',C,D_{\text{tree}}\rangle$，$U'=U\cup\Delta U$，$R_B$ 是由 $B\subseteq C$ 导出的模糊 T-相似关系，则：

$$\underline{R_{B\,inc}^{U'}}\ d_i'(x)=\min\{\underline{R_{B\,inc}^{U}}\ d_i(x),\inf_{y\in\Delta\overline{X}_{inc}^{d_i'}}(1-R_B(x,y))\},$$

$x\in d_i\in D_{inc}^U$

其中，$\Delta\overline{X}_{inc}^{d_i'}=\{x:x\in\overline{X}_{inc}^{d_i'}\cap\Delta U\}$。

定理3 对于原分层决策表 $\langle U,C,D_{\text{tree}}\rangle$ 和新的分层决策表 $\langle U',C,D_{\text{tree}}\rangle$，$U'=U\cup\Delta U$，$R_B$ 是由 $B\subseteq C$ 导出的模糊 T-相似关系，则：

$$\underline{R_{B\,inc}^{U'}}\ d_i'(x)=\inf_{y\in\overline{X}_{inc}^{d_i'}}(1-R_B(x,y)),\ x\in\Delta d_i,\ x\in\Delta U$$

3.1.2 正域的增量更新

由前面分析可知，加入样本后 D_{inc}^U 中节点的样本的下近似值可能变化，把下近似的隶属度值一定会变化的样本放入集合 S 中，即 $S=\{x:\inf_{y\in\Delta\overline{X}_{inc}^{d_i'}}(1-R_B(x,y))<\underline{R_{B\,inc}^{U}}\ d_i(x),\ x\in D_{inc}^U\}$。则 $U\setminus S$ 中的样本下近似值一定不会变化，可得到以下定理。

定理4 对于原分层决策表 $\langle U,C,D_{\text{tree}}\rangle$ 和新分层决策表 $\langle U',C,D_{\text{tree}}\rangle$，$U'=U\cup\Delta U$，$R_B$ 是由 $B\subseteq C$ 导出的模糊 T-相似关系，则：

$$POS_{B\,inc}^{U'}(D_{\text{tree}})(x)=\begin{cases}\inf_{y\in\Delta\overline{X}_{inc}^{d_i'}}(1-R_B(x,y)), & x\in S\\ \underline{R_{B\,inc}^{U}}\ d_i(x), & x\in U\setminus S\\ \underline{R_{B\,inc}^{U'}}\ d_i'(x), & x\in\Delta U\end{cases}$$

3.1.3 依赖度的增量更新

通过定理3和定义3可得到以下依赖度的增量更新定理。

定理5 对于原分层决策表 $\langle U,C,D_{\text{tree}}\rangle$ 和新分层决策表 $\langle U',C,D_{\text{tree}}\rangle$，$U'=U\cup\Delta U$，$R_B$ 是一个由 $B\subseteq C$ 导出的模糊 T-相似关系，则：

$$\gamma_{B\,inc}^{U'}(D_{\text{tree}})=(\sum_{x\in S}POS_{B\,inc}^{U'}(D_{\text{tree}})(x)+\sum_{x\in U\setminus S}POS_{B\,inc}^{U}(D_{\text{tree}})(x)+\sum_{x\in\Delta U}POS_{B\,inc}^{U'}(D_{\text{tree}})(x))/|U'|$$

3.2 算法

基于模糊粗糙集的增量更新方法，设计一个基于包含策略的依赖度更新算法、增量特征选择算法和两个版本的增量特征选择框架。

算法3 基于包含策略的依赖度更新算法（Inc-IDU）

输入：分层决策表 $\langle U,C,D_{\text{tree}}\rangle$，新进样本 ΔU，正域 $POS_{B\,inc}^{U}(D_{\text{tree}})$，属性约简 B，距离平方矩阵 $\boldsymbol{D}_B^2$，D_{inc}^U 和 ΔU 的划分 $\Delta U/D_{\text{tree}}$。

输出：依赖度 $\gamma_B^{U'}(D_{\text{tree}})$。

1. 初始化 $S=\varnothing$，$sum1=0$，$sum2=0$，$sum3=0$，$R_{inc}^{\Delta}=0$，$U'=U\cup\Delta U$

2. For $d_i \in D_{inc}^{U}$ do

3. For $x \in d_i$ do

4. 计算 $D_B^2(x,y)$，$y \in \Delta\overline{X}_{inc}^{d_i'}$，并存入 $\boldsymbol{D}_B^2$ 中，计算 $R_{inc}^{\Delta} = \inf_{y \in \Delta\overline{X}_{inc}^{d_i'}}(1 - R_B(x,y))$

5. If $R_{inc}^{\Delta} < POS_{B\,inc}^{U}(D_{\text{tree}})(x)$ then

6. $POS_{B\,inc}^{U'}(D_{\text{tree}})(x) = R_{inc}^{\Delta}$

7. $sum1 = sum1 + POS_{B\,inc}^{U'}(D_{\text{tree}})(x)$

8. End if

9. End for

10. End for

11. For $x \in U \backslash S$ do

12. $sum2 = sum2 + POS_{B\,inc}^{U}(D_{\text{tree}})(x)$；

13. End for

14. For $\Delta d_i \in \Delta U/D_{\text{tree}}$ do

15. For $x \in \Delta d_i$ do

16. 计算 $D_B^2(x,y)$，$y \in \overline{X}_{inc}^{d_i'}$，并存入 $\boldsymbol{D}_B^2$ 中，计算 $POS_{B\,inc}^{U'}(D_{\text{tree}})(x) = \inf_{y \in \overline{X}_{inc}^{d_i'}}(1 - R_B(x,y))$

17. $sum3 = sum3 + POS_{B\,inc}^{U'}(D_{\text{tree}})(x)$

18. End for

19. End for

20. Return $\gamma_B^{U'}(D_{\text{tree}}) = \dfrac{sum1 + sum2 + sum3}{|U'|}$

算法3是基于包含策略的依赖度更新算法。第2~10步是增量更新 S 中样本正域的隶属度，对这部分样本的正域隶属度进行求和，时间复杂度为 $O(|D_{inc}^{U}||d_i||\Delta\overline{X}_{inc}^{d_i'}||C|)$；第11~13步是计算 $U-S$ 中样本的正域隶属度之和，时间复杂度为 $O(|U-S|)$；第14~19步是计算 ΔU 中样本的正域隶属度之和，时间复杂度为 $O(|\Delta U||\overline{X}_{inc}^{d_i'}||C|)$。综上，算法3的时间复杂度为 $\max\{O(|D_{inc}^{U}||d_i||\Delta\overline{X}_{inc}^{d_i'}||C|), O(|\Delta U||\overline{X}_{inc}^{d_i'}||C|)\}$。

算法4 基于包含策略的增量特征选择算法（Inc-IFS）

输入：分层决策表 $\langle U, C, D_{\text{tree}}\rangle$，新进样本 ΔU，原属性约简 B，正域 $POS_{B\,inc}^{U}(D_{\text{tree}})$，距离平方矩阵 $\boldsymbol{D}_B^2$。

输出：新的属性约简 B。

1. 初始化 $rem = C - B$，$U' = U \cup \Delta U$

2. 计算 $\Delta U/D_{\text{tree}} = \{\Delta d_1, \Delta d_2, \cdots, \Delta d_k\}$

3. 计算 $D_{inc}^{U} = D_{\text{tree}} \backslash \bigcap_{t=1}^{k}((anc(d_t) \cup \{d_t\}))$

4. 寻找每个 $d_i \in D_{inc}^{U}$ 的新增加的包含负样本 $\Delta\overline{X}_{inc}^{d_i'}$

5. 通过 $d_i' = d_i \cup \Delta d_i$ 计算 U'/D_{tree}

6. 通过Inc-IDU计算 $\gamma_B^{U'}(D_{\text{tree}})$ 和 $\gamma_C^{U'}(D_{\text{tree}})$

7. While $\gamma_C^{U'}(D_{\text{tree}}) - \gamma_B^{U'}(D_{\text{tree}}) > \varepsilon$ do //添加属性策略（第7~15步）

8. For $a_0 \in rem$ do

9. 用 $D_{B\cup\{a_0\}}^2(x,y) = D_B^2(x,y) + D_{a_0}^2(x,y)$ 去替换Inc-NIDC的第5步，记为“Inc-NIDC +”

10. 通过“Inc-NIDC +”来计算 $\gamma_{B\cup\{a_0\}}^{U'}(D_{\text{tree}})$

11. End for

12. 找出一个属性 $a = \arg(\max \gamma_{B\cup\{a_0\}}^{U'}(D_{\text{tree}}))$

13. $rem = rem - \{a\}$

14. $B = B \cup \{a\}$

15. End while

16. 令 $Q = B$ //删除冗余属性策略（第16~23步）

17. For $a \in Q$ do

18. 用 $D_{B-\{a\}}^2(x,y) = D_B^2(x,y) - D_a^2(x,y)$ 去替换Inc-NIDC的第5步，记为“Inc-NIDC -”

19. 通过“Inc-NIDC -”来计算 $\gamma_{B-\{a\}}^{U'}(D_{\text{tree}})$

20. If $\gamma_C^{U'}(D_{\text{tree}}) - \gamma_{B-\{a\}}^{U'}(D_{\text{tree}}) \leqslant \varepsilon$ then

21. $B = B - \{a\}$

22. End if

23. End for

24. Return 属性集 B

算法4是基于包含策略的增量特征选择算法。第2步求 ΔU 的类划分，时间复杂度为 $O(l|U|)$；第3步计算 D_{inc}^{U}，时间复杂度为 $O(k)$；第6步通过Inc-IDU更新依赖度，时间复杂度为 $\max\{O(|D_{inc}^{U}||d_i||\Delta\overline{X}_{inc}^{d_i'}||C|), O(|\Delta U||\overline{X}_{inc}^{d_i'}||C|)\}$；“Inc-NIDC +”和“Inc-NIDC -”的时间复杂度为 $O(|U|^2)$，第7~15步为添加属性策略，从剩余属性中一直添加属性，直到满足定义4中的条件（1）为止，此时间复杂度为 $O(|C|^2|U|^2)$；第16~23步为删除冗余属性策略，删除 B 中的元素，直到满足定义4的条件（2）为止，此时间复杂度为 $O(|C||U|^2)$。综上，算法4的时间复杂度为 $O(|C|^2|U|^2)$。

综上易得，算法4的时间复杂度小于算法2的时间复杂度。另外，由于 $|D_{inc}^{U}||d_i| < |U|$，$|\Delta\overline{X}_{inc}^{d_i'}| < |U|$，$|\overline{X}_{inc}^{d_i'}| < |U|$，算法3的时间复杂度也小于算法1的时间复杂度。

接下来，基于两种策略提出两个增量特征选择框架，用以解决批处理大规模数据集的分层分类问题。先将训练集划分为 N 份子数据集，当不同的子数据集加入当前数据集 T 时采用两种不同的策略寻找属性约简。策略1（对应算法5）：在每次子数据集加入时只执行添加属性策略，当第 N 个子数据集都完成上述策略后再执行删除冗余属性策略；策略2（对应算法6）：在每次子数据集加入时执行添加属性策略和删除冗余属性策略。

算法5 增量算法的框架1(Inc-IFS-v1)

输入：分层决策表$\langle U,C,D_{\text{tree}}\rangle$。

输出：属性约简B。

1. 将数据集的训练集按类均匀地划分为N份$\{U_1,U_2,\cdots,U_N\}$，且使U_1包含所有的类

2. 初始化当前数据集$T=\varnothing$，$B=\varnothing$

3. For $i=1:N$ do //策略1

4. $T=T\cup U_i$

5. If $i=1$ then //求第一块子数据集的属性集B

6. 通过Inc-NIDC求$\gamma_B^T(D_{\text{tree}})$和$\gamma_C^T(D_{\text{tree}})$

7. 通过Inc-IFS的添加属性策略(第7~15步)得到B

8. Else //求第2到N块子数据集到达后的B

9. 通过Inc-IFS的添加属性策略(第7~15步)更新B

10. End if

11. End for

12. 执行Inc-IFS的删除冗余属性策略(第16~23步)

13. Return属性约简B

算法6 增量算法的框架2(Inc-IFS-v2)

输入：分层决策表$\langle U,C,D_{\text{tree}}\rangle$。

输出：属性约简B。

1. 将数据集的训练集按类均匀地划分为N份$\{U_1,U_2,\cdots,U_N\}$，且使U_1包含所有的类

2. 初始化当前数据集$T=\varnothing$，$U=\varnothing$

3. For $i=1:N$ do //策略2

4. $T=T\cup U_i$

5. If $i=1$ then //求第一块子数据集的约简

6. 通过Inc-IFS寻找属性约简B

7. Else //求第2到N块子数据集到达后的属性约简B

8. 通过Inc-IFS更新属性约简B

9. End if

10. End for

11. Return属性约简B

由于策略1只进行一次删除冗余属性策略，从理论上看，Inc-IFS-v1的运行时间小于Inc-IFS-v2。

4 实验分析

本章先从运行时间、所选择特征个数、F_H测度(基于F_1的分层分类准确率度量)[37-38]和平均TIE四个指标将FFS-HC[23]与本文所提的Inc-NIDC、Inc-IFS-v1和Inc-IFS-v2进行对比。然后对Inc-IFS-v1和Inc-IFS-v2进行参数ε敏感度分析。最后通过实验结果对所提的三个特征选择算法进行评价。

TIE(tree induced error)为树诱导误差[39]，TIE值会随着样本量的增多而变大。平均TIE不受测试样本量影响，可以更好地度量算法性能，因此这里采用平均TIE来表示算法性能$\left(\text{平均}TIE=\dfrac{TIE\text{值}}{\text{测试样本个数}}\right)$。

4.1 实验设计

实验环境：Intel® Core™ i5-7200U CPU@2.50 GHz 2.71 GHz 12.0 GB，MATLAB R2016a。

数据集：表4是在分层分类问题中经常使用的数据集，这些数据集的真实标签在叶子节点，由于树结构中父子节点存在语义关系，子节点样本隐含在父节点中，为了构造出本文所研究类型的数据集，把叶子节点中部分样本的标签提升为其祖先节点，并使每个节点中的样本尽可能地均匀分布，此时标签的个数记为d'，如表4所示。

为了使上层节点具有子节点样本尽可能多的信息，将每个叶子节点中样本按一定比例$\dfrac{num_{\text{samples}}}{num_{\text{nodes}}\times(d_{\text{layer}}+1)}$随机划分份数为$d_{\text{layer}}$，向上层节点划分。$num_{\text{samples}}$表示该节点中样本个数，$num_{\text{nodes}}$表示该节点的父节点的子节点个数，$d_{\text{layer}}$表示需向上泛化的层数。该比例大小也会随着向上泛化的迭代过程而动态变化。这样既使叶子节点均匀，又使上层节点的样本分布均匀，且尽可能均匀地包含下层节点样本，可以避免样本分布的不平衡。

分类器：支持向量机(support vector machine, SVM[40])、K近邻(k-nearest neighbors, KNN[41])(k设为常用值3)、随机森林(random forest, RF[42])。

数据处理：(1)对数据集进行最大最小归一化处

表4 数据集

Table 4 Datasets

数据集	类型	样本	训练集	测试集	特征个数	$\lvert d\rvert$	$\lvert d'\rvert$	节点个数
Bridges	数值型	108	65	43	11	6	7	8
DD	数值型	3 625	3 020	605	473	27	31	32
Protein194	数值型	8 525	5 121	3 404	473	194	201	202
CLEF	数值型	9 307	6 405	3 202	252	10	16	17
SAIAPR5000	图像型	5 000	3 009	1 991	512	196	205	256
VOC	图像型	12 283	7 178	5 105	1 000	20	29	30

理，$\bar{a}(x_i)=\dfrac{a(x_i)-\min_j a(x_j)}{\max_j a(x_j)-\min_j a(x_j)}$，$x_i\in U$，显然 $\bar{a}(x_i)\in[0,1]$。

(2)用训练集寻找约简，然后采用5折交叉验证方法在测试集对所选择特征进行测试，求出 F_H 值和平均TIE值。

4.2 四个算法对比

这部分，通过对比FFS-HC[23]、Inc-NIFS、Inc-IFS-v1和Inc-IFS-v2算法的运行时间、所选特征个数、F_H 测度和平均TIE来评价所提的三个算法，结果如表5和图3、图4所示。

参数设置：Inc-NIFS、Inc-IFS-v1和Inc-IFS-v2中令 $\varepsilon=0.01$，$\sigma=0.2$，$N=10$。

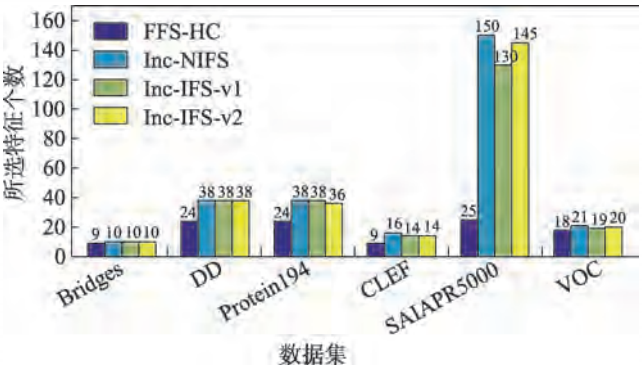

图3 4个算法所选特征个数对比

Fig.3 Comparison of the number of selected features for 4 algorithms

表5 算法FFS-HC、Inc-NIFS、Inc-IFS-v1和Inc-IFS-v2的运行时间对比

Table 5 Comparison of running time for FFS-HC, Inc-NIFS, Inc-IFS-v1 and Inc-IFS-v2 algorithms **单位：s**

数据集	FFS-HC	Inc-NIFS	Inc-IFS-v1	Inc-IFS-v2
Bridges	0.07	**0.04**	0.07	0.09
DD	1 167.77	11 276.98	**203.35**	274.44
Protein194	2 939.14	63 616.80	**1 270.35**	3 472.62
CLEF	144.11	843.20	**66.29**	152.33
SAIAPR5000	**359.85**	246 672.55	791.09	1 170.97
VOC	4 809.81	45 387.63	**662.42**	1 087.62

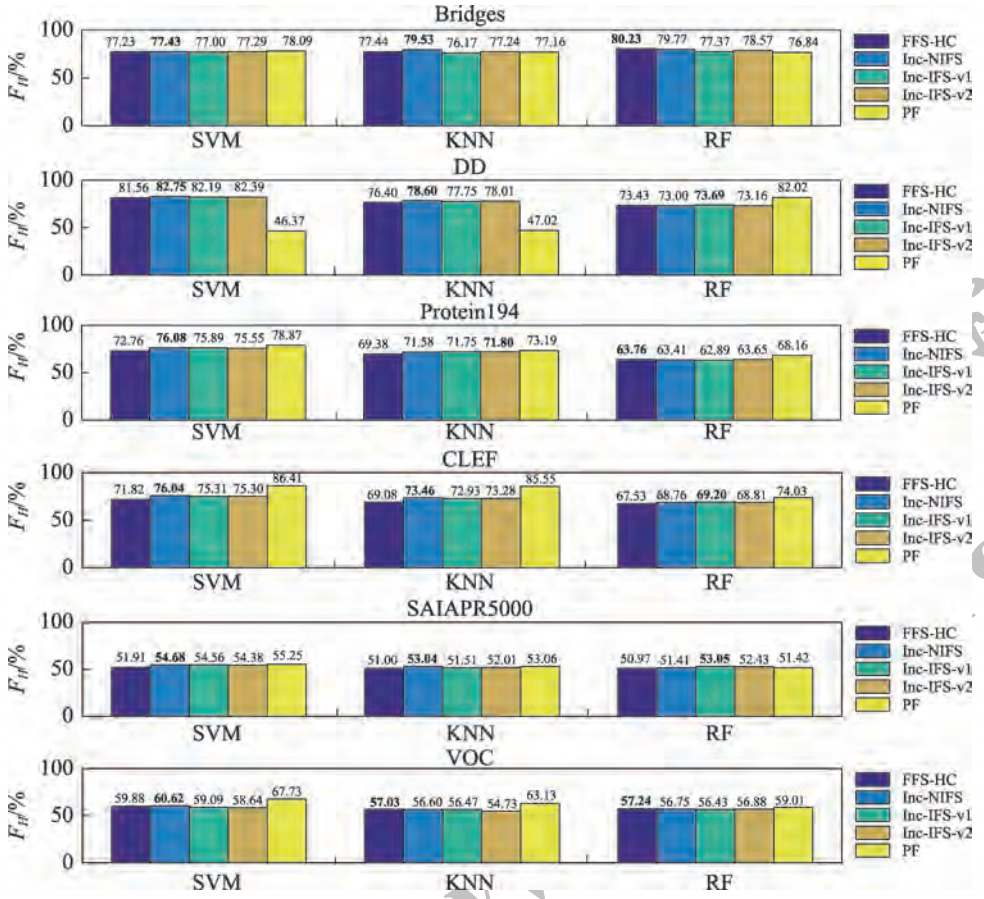

图4 4个算法的 F_H 值对比

Fig.4 Comparison of F_H for 4 algorithms

表5是FFS-HC、Inc-NIFS、Inc-IFS-v1和Inc-IFS-v2的运行时间。加粗表示运行时间最短。在Bridges数据集上Inc-NIFS的运行时间最短，这是因为这个数据集太小，而样本又分多次到达，导致增量计算时间较长；而在其他大规模数据集上除了SAIAPR外，Inc-IFS-v1的运行时间最短，且与FFS-HC和Inc-NIFS的运行时间差距非常大。尤其在VOC数据集上，FFS-HC约是Inc-IFS-v1的7.3倍，Inc-NIFS约是Inc-IFS-v1的68.5倍。在SAIAPR5000数据集上，Inc-NIFS约是Inc-IFS-v1的311.8倍。从实验中也可以看出在所有数据集上Inc-IFS-v1比Inc-IFS-v2的时间效率更高。

图3是FFS-HC、Inc-NIFS、Inc-IFS-v1和Inc-IFS-v2的所选特征个数。在所有数据集上FFS-HC所选特征个数小于其他3个算法；Inc-NIFS、Inc-IFS-v1和Inc-IFS-v2所选特征个数基本上区别不大；FFS-HC在SAIAPR5000数据集上所选特征个数远远小于其他3个算法。

图4是FFS-HC、Inc-NIFS、Inc-IFS-v1和Inc-IFS-v2的F_H测度，其中PF表示全部特征，加粗表示4个算法中F_H值最大的。本文分别在分类器SVM、KNN和RF上对比不同算法的效果。在数据集DD上，算法之间的F_H测度受分类器的影响，4个算法的F_H值在分类器SVM和KNN上都大于PF，而在分类器RF上却均小于PF；在VOC数据集上，FFS-HC的F_H值在KNN和RF上最大，其余数据集上Inc-NIFS的F_H值最大的情况居多，有些情况下，Inc-IFS-v1最大。在大部分数据集上，Inc-NIFS、Inc-IFS-v1和Inc-IFS-v2之间的F_H值基本一致，相差不超过1个百分点。因此，可以充分说明Inc-NIFS、Inc-IFS-v1和Inc-IFS-v2可行并且有效。

表6是FFS-HC、Inc-NIFS、Inc-IFS-v1和Inc-IFS-v2的平均TIE值对比。加粗表示在对应分类器下平均TIE值最小。平均TIE值越小，表示算法越好。从全部数据集上看，SAIAPR5000数据集上的算法的平均TIE值大于其他数据集，说明在SAIAPR5000这个数据集上分类误差都大于其他数据集；从算法角度看，在分类器SVM上Inc-NIFS更占优势，但是这些算法差别不大，相差不超过0.3。整体来看，FFS-HC、Inc-NIFS、Inc-IFS-v1和Inc-IFS-v2的平均TIE几乎没有差别。

综合表5、表6和图3、图4，从运行时间上看，Inc-IFS-v1的时间效果最好，Inc-IFS-v2次之；从分类效果上看，Inc-NIFS、Inc-IFS-v1和Inc-IFS-v2都不低于FFS-HC，且3个算法之间差别不大。综合时间和分类效果，Inc-IFS-v1能在最短时间内完成特征选择，且最大程度地保证分类精度。

4.3 Inc-IFS-v1和Inc-IFS-v2的参数敏感性分析

本节从运行时间、所选择特征个数、F_H测度方面分析Inc-IFS-v1和Inc-IFS-v2的ε在{0.05,0.01,0.005,0.001,0.000 5,0.000 1}上的敏感度，参数$\sigma=0.2$，$N=10$，结果如图5~图7所示。

图5是Inc-IFS-v1和Inc-IFS-v2随着阈值ε的减小运行时间的变化。阈值ε越小，而添加属性的条件就越严格，对应的运行时间就越长。在VOC数据集上，Inc-IFS-v2的运行时间远远大于其他参数，可能因为在这个参数下每次到达子数据集都会进行添加属性和删除冗余属性策略，而再次增加样本后原属性约简总不满足$\gamma^U_{C\,inc}(D_{\text{tree}})-\gamma^U_{B\,inc}(D_{\text{tree}})<\varepsilon$，需要再次添加属性和删除冗余属性，VOC有1 000个属性，会使运行时间差距更明显。综合来看，Inc-IFS-v1和Inc-IFS-v2的运行时间随着阈值减小变长，并且Inc-IFS-v1的运行时间总小于Inc-IFS-v2。

图6是Inc-IFS-v1和Inc-IFS-v2所选择特征的个数比较。从图中可以看到，阈值ε越小，属性约简的

表6 4个算法的平均TIE对比

Table 6 Comparison of average TIE for 4 algorithms

数据集	SVM				KNN				RF			
	FFS-HC	Inc-NIFS	Inc-IFS-v1	Inc-IFS-v2	FFS-HC	Inc-NIFS	Inc-IFS-v1	Inc-IFS-v2	FFS-HC	Inc-NIFS	Inc-IFS-v1	Inc-IFS-v2
Bridges	1.14	1.14	**1.07**	1.09	1.14	**1.02**	1.21	1.12	**1.00**	1.02	1.12	1.12
DD	1.09	**1.01**	1.04	1.03	1.40	**1.27**	1.32	1.30	1.57	1.60	**1.56**	1.59
Protein194	1.63	**1.43**	1.44	1.47	1.84	1.70	**1.69**	**1.69**	**2.17**	2.19	2.22	2.18
CLEF	**2.02**	1.69	1.74	1.74	2.23	**1.89**	1.92	1.90	2.31	2.21	**2.18**	2.20
SAIAPR5000	3.67	**3.43**	3.44	3.46	3.73	**3.58**	3.71	3.67	3.83	3.82	**3.69**	3.75
VOC	2.25	**2.22**	2.29	2.31	2.70	2.69	**2.64**	2.82	**2.62**	2.66	2.68	2.64

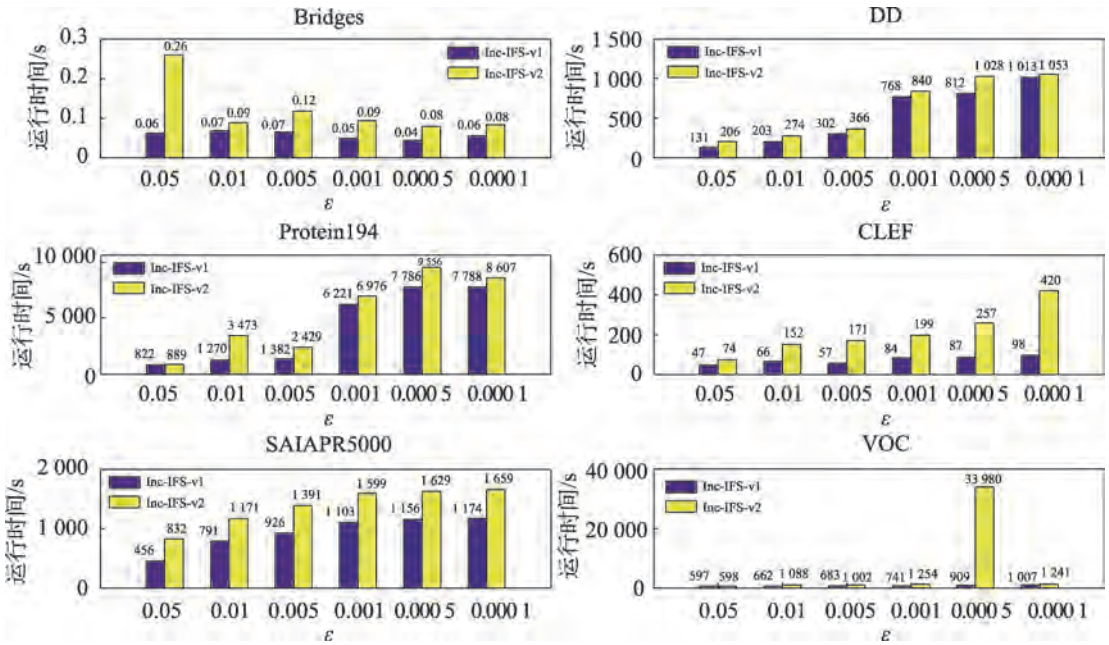

图5 Inc-IFS-v1和Inc-IFS-v2的运行时间对比

Fig.5 Comparison of running time for Inc-IFS-v1 and Inc-IFS-v2

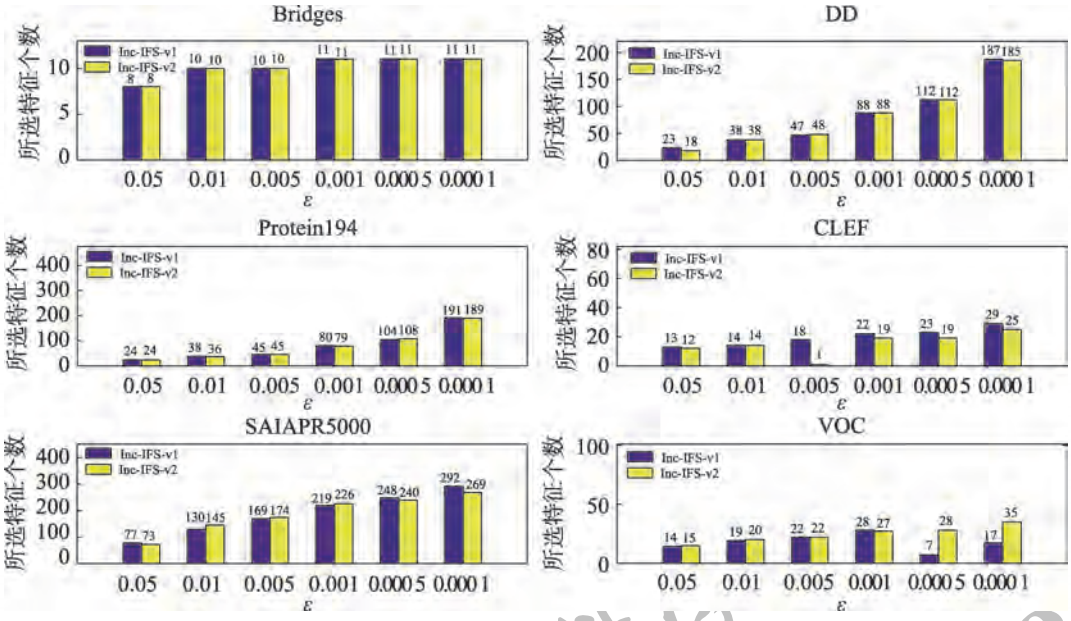

图6 Inc-IFS-v1和Inc-IFS-v2的所选特征个数对比

Fig.6 Comparison of the number of selected features for Inc-IFS-v1 and Inc-IFS-v2

条件越严格,Inc-IFS-v1和Inc-IFS-v2所选择特征的个数越多,且Inc-IFS-v1和Inc-IFS-v2所选择特征的个数基本差别不大。在VOC数据集上,当$\varepsilon=0.000\,5$时,Inc-IFS-v1只挑选了7个特征,在CLEF数据集上,当$\varepsilon=0.005$时,Inc-IFS-v2只挑选了1个特征。这可能因为这种情况下许多特征作为冗余属性被删除,而这些少量特征仍然符合约简定义。

图7为Inc-IFS-v1和Inc-IFS-v2的F_H值。在DD数据集上,阈值$\varepsilon<0.005$时,Inc-IFS-v1和Inc-IFS-v2的F_H值在分类器SVM和KNN上减小。可能是因为随阈值ε减少,所选特征个数变多,多了一些干扰性特征,使得分类精度下降。整体上看,随着阈值ε的减少,Inc-IFS-v1和Inc-IFS-v2的F_H值呈不明显的上升趋势。综上,Inc-IFS-v1和Inc-IFS-v2的分类精度

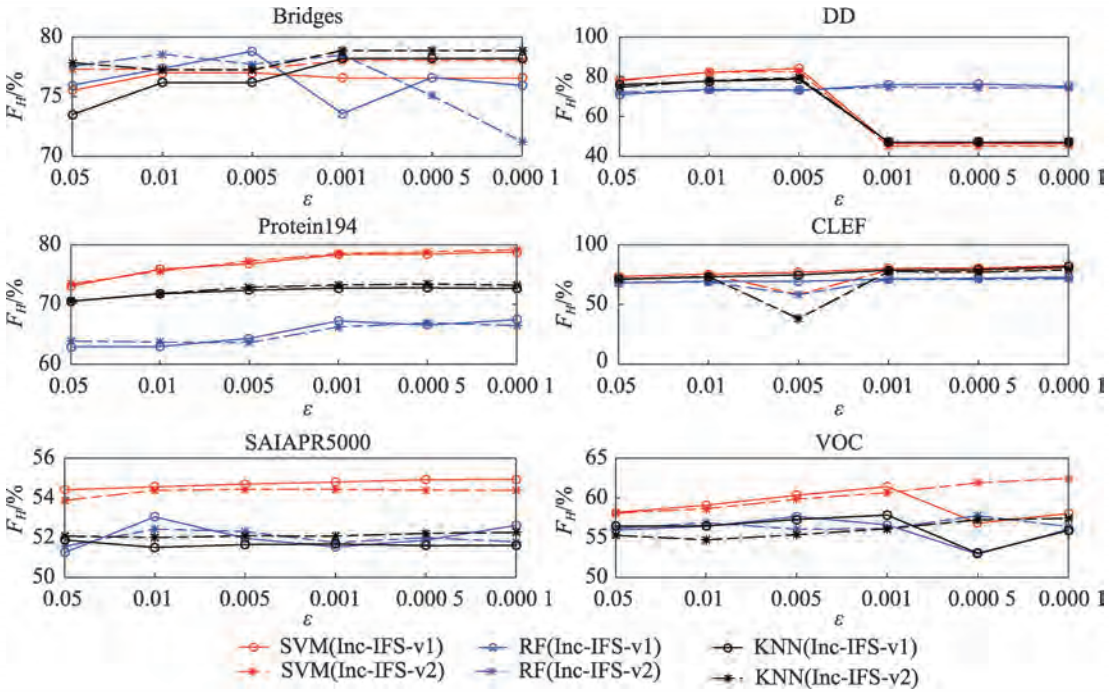

图7 Inc-IFS-v1和Inc-IFS-v2的 F_H 值对比

Fig.7 Comparison of F_H values for Inc-IFS-v1 and Inc-IFS-v2

随着阈值 ε 的变化，敏感性较弱。

综合图5~图7，从时间上看，随着阈值 ε 变小，Inc-IFS-v1和Inc-IFS-v2的运行时间越长，即运行时间的敏感性较大；从所选特征个数和分类精度看，随着阈值 ε 变小，在有些数据集上分类精度有上升趋势，在有些数据集上没有明显变化。

5 总结与展望

本文给出了包含策略和基于包含策略的模糊粗糙集新的形式化定义，提出了基于包含策略的模糊粗糙集模型，并设计了一个非增量特征选择算法Inc-NIFS。然后引入依赖度的增量机制，设计距离平方矩阵来缩短添加属性过程的时间。由此，提出了增量特征选择算法Inc-IFS，以及两种增量特征选择框架Inc-IFS-v1和Inc-IFS-v2，两者效率均高于Inc-NIFS，且Inc-IFS-v1的效率最高。

除了样本增加外，还包括特征添加和特征值动态变化的情况。分层分类学习中，可以选择的策略也多种多样，除了包含策略，还有兄弟策略、排他策略、排他兄弟策略、排他包含策略等。这些均可作为未来的研究工作。在未来的研究中，将考虑随着样本到达，基于兄弟策略的模糊粗糙集增量，与本文所提的算法进行对比分析；将研究在特征动态增加的情况下基于包含策略、兄弟策略以及其他策略的增量机制。

参考文献：

[1] PARMEZAN A R S, SOUZA V M A, BATISTA G E. Towards hierarchical classification of data streams[C]// Proceedings of the 23rd Iberoamerican Congress on Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications, Madrid, Nov 19-22, 2018: 314-322.

[2] SILLA C N, FREITAS A A. A survey of hierarchical classification across different application domains[J]. Data Mining and Knowledge Discovery, 2011, 22: 31-72.

[3] OSMANI A, HAMIDI M, ALIZADEH P. Clustering approach to solve hierarchical classification problem complexity[C]// Proceedings of the 36th AAAI Conference on Artificial Intelligence, the 34th Conference on Innovative Applications of Artificial Intelligence, the 12th Symposium on Educational Advances in Artificial Intelligence, Feb 22-Mar 1, 2022: 7904-7912.

[4] BELLMUND J L S, GÄRDENFORS P, MOSER E I, et al. Navigating cognition: spatial codes for human thinking[J]. Science, 2018, 362(6415): eaat6766.

[5] ARONOV D, NEVERS R, TANK D. Mapping of a non-spatial dimension by the hippocampal-entorhinal circuit[J]. Nature, 2017, 543: 719-722.

[6] 刘浩阳, 林耀进, 刘景华, 等. 由粗到细的分层特征选择[J]. 电子学报, 2022, 50(11): 2778-2789.
LIU H Y, LIN Y J, LIU J H, et al. Hierarchical feature selection from coarse to fine[J]. Acta Electronica Sinica, 2022, 50(11): 2778-2789.

[7] 林耀进, 白盛兴, 赵红, 等. 基于标签关联性的分层分类共有与固有特征选择[J]. 软件学报, 2022, 33(7): 2667-2682.
LIN Y J, BAI S X, ZHAO H, et al. A label correlation based common and specific feature selection for hierarchical classification[J]. Journal of Software, 2022, 33(7): 2667-2682.

[8] 梁吉业, 钱宇华, 李德玉, 等. 大数据挖掘的粒计算理论与方法[J].中国科学: 信息科学, 2015, 45(11): 1355-1369.
LIANG J Y, QIAN Y H, LI D Y, et al. Theory and method of granular computing for big data mining[J]. Scientia Sinica: Informationis, 2015, 45(11): 1355-1369.

[9] 王国胤, 傅顺, 杨洁, 等. 基于多粒度认知的智能计算研究[J]. 计算机学报, 2022, 45(6): 1161-1175.
WANG G Y, FU S, YANG J, et al. A review of research on multi-granularity cognition based intelligent computing[J]. Chinese Journal of Computers, 2022, 45(6): 1161-1175.

[10] 王国胤,于洪. 多粒度认知计算——一种大数据智能计算的新模型[J]. 数据与计算发展前沿, 2019, 1(6): 75-85.
WANG G Y, YU H. Multi-granularity cognitive computing—a new model for big data intelligent computing[J]. Frontiers of Data & Computing, 2019, 1(6): 75-85.

[11] MORSI N N, YAKOUT M M. Axiomatics for fuzzy rough sets[J]. Fuzzy Sets and Systems, 1998, 100: 327-342.

[12] DUBOIS D, PRADE H. Rough fuzzy sets and fuzzy rough sets[J]. International Journal of General System, 1990, 17(2/3): 191-209.

[13] JENSEN R, SHEN Q. Fuzzy-rough attribute reduction with application to web categorization[J]. Fuzzy sets and Systems, 2004, 141(3): 469-485.

[14] WANG C, QI Y, SHAO M, et al. A fitting model for feature selection with fuzzy rough sets[J]. IEEE Transactions on Fuzzy Systems, 2016, 25(4): 741-753.

[15] ROSENBERG A, HIRSCHBERG J. V-measure: a conditional entropy-based external cluster evaluation measure[C]// Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning, Prague, Jun 28-30, 2007: 410-420.

[16] ZHANG X, MEI C, CHEN D, et al. Active incremental feature selection using a fuzzy-rough-set-based information entropy[J]. IEEE Transactions on Fuzzy Systems, 2019, 28(5): 901-915.

[17] JENSEN R, SHEN Q. New approaches to fuzzy-rough feature selection[J]. IEEE Transactions on Fuzzy Systems, 2008, 17(4): 824-838.

[18] CHEN D, ZHANG L, ZHAO S, et al. A novel algorithm for finding reducts with fuzzy rough sets[J]. IEEE Transactions on Fuzzy Systems, 2011, 20(2): 385-389.

[19] YANG Y, CHEN D, WANG H, et al. Incremental perspective for feature selection based on fuzzy rough sets[J]. IEEE Transactions on Fuzzy Systems, 2017, 26(3): 1257-1273.

[20] BHATT R B, GOPAL M. On fuzzy-rough sets approach to feature selection[J]. Pattern Recognition Letters, 2005, 26(7): 965-975.

[21] HU Q, YU D, XIE Z. Information-preserving hybrid data reduction based on fuzzy-rough techniques[J]. Pattern Recognition Letters, 2006, 27(5): 414-423.

[22] TSANG E C C, CHEN D, YEUNG D S, et al. Attributes reduction using fuzzy rough sets[J]. IEEE Transactions on Fuzzy Systems, 2008, 16(5): 1130-1141.

[23] ZHAO H, WANG P, HU Q, et al. Fuzzy rough set based feature selection for large-scale hierarchical classification [J]. IEEE Transactions on Fuzzy Systems, 2019, 27(10): 1891-1903.

[24] ZHAO H, HU Q, ZHU P, et al. A recursive regularization based feature selection framework for hierarchical classification [J]. IEEE Transactions on Knowledge and Data Engineering, 2019, 33(7): 2833-2846.

[25] TUO Q, ZHAO H, HU Q. Hierarchical feature selection with subtree based graph regularization[J]. Knowledge-Based Systems, 2019, 163: 996-1008.

[26] ZHENG J, LUO C, LI T, et al. A novel hierarchical feature selection method based on large margin nearest neighbor learning[J]. Neurocomputing, 2022, 497: 1-12.

[27] LIN Y, LIU H, ZHAO H, et al. Hierarchical feature selection based on label distribution learning[J]. IEEE Transactions on Knowledge and Data Engineering, 2023, 35(6): 5964-5976.

[28] CHEN H, LI T, RUAN D. Maintenance of approximations in incomplete ordered decision systems while attribute values coarsening or refining[J]. Knowledge-Based Systems, 2012, 31: 140-161.

[29] LUO C, LI T, CHEN H, et al. Fast algorithms for computing rough approximations in set-valued decision systems while updating criteria values[J]. Information Sciences, 2015, 299: 221-242.

[30] LUO C, LI T, CHEN H. Dynamic maintenance of approximations in set-valued ordered decision systems under the attribute generalization[J]. Information Sciences, 2014, 257: 210-228.

[31] YANG X, QI Y, YU H, et al. Updating multigranulation rough approximations with increasing of granular structures

[J]. Knowledge-Based Systems, 2014, 64: 59-69.
[32] LIU D, LI T, ZHANG J. Incremental updating approximations in probabilistic rough sets under the variation of attributes [J]. Knowledge-Based Systems, 2015, 73: 81-96.
[33] FAN W, HE C, ZENG A, et al. An incremental approach based on hierarchical classification in multikernel fuzzy rough sets under the variation of object set[C]//Proceedings of the 18th International Conference on Intelligent Computing Methodologies, Xi'an, Aug 7-11, 2022. Cham: Springer International Publishing, 2022: 3-17.
[34] LUO C, LI T, CHEN H, et al. Incremental rough set approach for hierarchical multicriteria classification[J]. Information Sciences, 2018, 429: 72-87.
[35] ZADEH L A. Similarity relations and fuzzy orderings[J]. Information Sciences, 1971, 3(2): 177-200.
[36] CHEN D, HU Q, YANG Y. Parameterized attribute reduction with Gaussian kernel based fuzzy rough sets[J]. Information Sciences, 2011, 181(23): 5169-5179.
[37] KOSMOPOULOS A, PARTALAS I, GAUSSIER E, et al. Evaluation measures for hierarchical classification: a unified view and novel approaches[J]. Data Mining and Knowledge Discovery, 2015, 29: 820-865.
[38] AHO A V, HOPCROFT J E, ULLMAN J D. On finding lowest common ancestors in trees[C]//Proceedings of the 5th Annual ACM Symposium on Theory of Computing, Apr 30-May 2, 1973: 253-265.
[39] DEKEL O, KESHET J, SINGER Y. Large margin hierarchical classification[C]//Proceedings of the 21st International Conference on Machine Learning, Banff, Jul 4-8, 2004: 27.
[40] JOACHIMS T. Making large-scale SVM learning practical [R]. 1998.
[41] GUO G, WANG H, BELL D, et al. KNN model-based approach in classification[C]//On the Move to Meaningful Internet Systems 2003: CoopIS, DOA, and ODBASE-OTM Confederated International Conferences, Catania, Nov 3-7, 2003: 986-996.
[42] RIGATTI S J. Random forest[J]. Journal of Insurance Medicine, 2017, 47(1): 31-39.

折延宏（1983—），男，陕西延安人，博士，教授，CCF会员，主要研究方向为机器学习、不确定性数据建模等。
SHE Yanhong, born in 1983, Ph.D., professor, member of CCF. His research interests include machine learning, uncertainty data modelling, etc.

黄婉丽（1998—），女，河南洛阳人，硕士研究生，主要研究方向为粒计算、分层分类等。
HUANG Wanli, born in 1998, M.S. candidate. Her research interests include granular computing, hierarchical classification, etc.

贺晓丽（1982—），女，山西朔州人，博士，副教授，主要研究方向为不确定性推理、粒度计算等。
HE Xiaoli, born in 1982, Ph.D., associate professor. Her research interests include uncertainty reasoning, granular computing, etc.

钱婷（1985—），女，山东聊城人，博士，副教授，主要研究方向为粗糙集、概念格、不确定性推理等。
QIAN Ting, born in 1985, Ph.D., associate professor. Her research interests include rough sets, concept lattices, uncertainty reasoning, etc.