Applied Intelligence (2021) 51:1602–1615
https://doi.org/10.1007/s10489-020-01863-5

Kernelized fuzzy rough sets based online streaming feature selection for large-scale hierarchical classification

Shengxing Bai[1,2] · Yaojin Lin[1,2] · Yan Lv[1,2] · Jinkun Chen[3] · Chenxi Wang[1]

Published online: 30 September 2020

Abstract
In recent years, many online streaming feature selection approaches focus on flat data, which means that all data are taken as a whole. However, in the era of big data, not only the feature space of data has unknown and evolutionary characteristics, but also the label space of data exists hierarchical structure. To address this problem, an online streaming feature selection framework for large-scale hierarchical classification task is proposed. The framework consists of three parts: (1) a new hierarchical data-oriented kernelized fuzzy rough model with sibling strategy is constructed, (2) the online important feature is selected based on feature correlation analysis, and (3) the online redundant feature is deleted based on feature redundancy. Finally, an empirical study using several hierarchical classification data sets manifests that the proposed method outperforms other state-of-the-art online streaming feature selection methods.

Keywords Online feature selection · Hierarchical classification · Kernelized fuzzy rough sets · Sibling strategy

1 Introduction

Hierarchies Taxonomies are popular for organizing large volume data sets in various application domains [9, 15]. For example, ImageNet is an image database organized refer to the WordNet hierarchy (currently only the nouns), in which hundreds and thousands of images are used to depict each node of the hierarchy. It also has been used in many areas including biology data [9], Wikipedia [24], geographical data [39], and text data [3, 6, 44]. Therefore, large-scale hierarchical classification learning is an important and popular learning paradigm in machine learning and data mining communities [9, 15].

From the viewpoint of biologists, the discovery of new species is attributed to the new features detected. Furthermore, these new features are now available in the existed species [50]. Therefore, the challenge of hierarchical classification learning is that the full feature space is unknown before learning begins. As we know, the full feature space determines the final label category of the samples. For example, in the diagnosis of lung cancer, through clinical testing in a period, doctors can gradually obtain clinical signs of lung cancer patients. Further, these patients may need to be diagnosed with small cell lung cancer, which is the subcategory of lung cancer. This phenomenon suggests that it is infeasible to collect all features of disease before diagnosis beginning. Therefore, the dynamic characteristic of feature might make the feature space of training data become high dimensional and uncertain. In order to explore online knowledge discovery with a dynamic feature space, some streaming feature selection algorithms are proposed.

Contrary to traditional feature selection methods, streaming feature assumes that all features are precomputed and presented to a learner before feature selection takes place, and streaming feature selection is defined as features that flow in one by one over time whereas the number of training examples is fixed [5, 28, 48, 55–57]. For example, hot topics are continuously changing in the social network platforms such as Twitter, and Facebook. When a popular topic appears, it accompanies with a set of new keywords. These new keywords may act as key features to distinguish the popular topic. At present, a number of

✉ Yaojin Lin
zzlinyaojin@163.com

[1] School of Computer Science, Minnan Normal University, Zhangzhou, 363000, People's Republic of China

[2] Laboratory of Data Science, Intelligence Application, Minnan Normal University, Zhangzhou, 363000, People's Republic of China

[3] School of Mathematics and Statistics, Minnan Normal University, Zhangzhou, 363000, People's Republic of China

existing online streaming feature selection algorithms have been proposed. Wu et al. [48] presented an online streaming feature selection framework based on Markov blanket. Lin et al. [35] proposed a multi-label online streaming feature selection algorithm based on fuzzy mutual information. Currently, existing online streaming feature selection algorithms assume that classes are independent of each other, and often ignore the hierarchical structure between classes in hierarchical classification data.

Motivated by the above discussion, a new algorithm named KFOHFS, i.e., Kernelized Fuzzy rough sets based Online Hierarchical streaming Feature Selection, is proposed in this paper, due to the kernelized fuzzy rough sets model can effectively measure the fuzzy relation between samples under the hierarchical label space. More specifically, KFOHFS conducts online streaming feature selection for large-scale hierarchical classification through three intuitive steps. Firstly, we define a new kernelized fuzzy rough sets model for large-scale hierarchical classification learning, and design a novel dependency function to determine whether the candidate feature on the flow is important to the label space relative to the selected features. Secondly, we present two steps for online streaming feature selection, i.e, online important feature selection, and online redundant feature recognition, which can be used to obtain discriminative features and discard redundant and useless features, respectively. Finally, an online heuristic streaming feature selection algorithm is proposed. Extensive experiments show the competitive performance of KFOHFS against some state-of-the-art online streaming feature selection algorithms.

The remainder of this paper is organized as follows. Section II discuses related work. Section III introduces kernelized fuzzy rough sets. In Section IV, we present a kernelized fuzzy rough sets based online streaming feature selection algorithm for large-scale hierarchical classification. Our experiments on several hierarchical classification data sets are demonstrated in Section V. Section VI summarizes this paper and outlines the future directions for this work.

2 Related work

In the feature selection process of large-scale hierarchical classification learning, hierarchical class information are helpful for selecting a feature subset [2, 7, 51]. There are many proposed feature selection algorithms that leverage the hierarchical class in a tree. For instance, Freeman et al. [22] proposed a method using genetic algorithms for combining feature selection and hierarchical classifier. Song et al. [42] developed a feature selection algorithm for hierarchical text classification. Zhao et al. [53] presented a feature selection framework with recursive regularization for hierarchical classification.

However, the above mentioned feature selection algorithms assume that global features are precomputed and presented to a learner before feature selection takes place [4]. In many real-life applications, features may exist in a streaming format and arrive one feature at a time. At present, a number of existing online streaming feature selection algorithms have been proposed. Roughly speaking, according to the number of labels associated with the instances, online streaming feature selection algorithms can be grouped into streaming feature selection for traditional single-label learning and multi-label learning [46], respectively.

For traditional single-label learning, Yu et al. [49] proposed a scalable and accurate online feature selection approach(SAOLA) for high dimensional data. The proposed algorithm employs an online pairwise comparison to maintain a parsimonious model over time. Nevertheless, SAOLA ignores the hierarchical structure of the classes. Javidi and Eskandari [29] proposed a method(SFS-RS) from the rough set perspective via considering the problem of streamwise feature selection, in which, Rough Set Theory is used to control the unknown feature space in SFS-RS. Eskandari and Javidi [14] proposed a new rough set model(OS-NRRSARA-SA) for online streaming feature selection. However, SFS-RS and OS-NRRSARA-SA cannot deal with numerical features and ignore the hierarchical structure of the classes. Rahmaninia and Moradi [40] proposed two online stream feature selection methods based on mutual information(OSFSMI and OSFSMI-k, respectively). However, these methods ignore the hierarchical structure of the classes and need domain knowledge before learning. Zhou et al. [57] proposed an online streaming feature selection algorithm(OFS-Density) based on a new neighborhood relation which using the density information of the surrounding instances. OFS-Density uses a fuzzy equal constraint for redundant analysis to make the selected feature subset with low redundancy but ignores the hierarchical structure of the classes. For multi-label learning, Lin et al. [35] proposed a multi-label online streaming feature selection algorithm based on fuzzy mutual information. Liu et al. [36] proposed an online multi-label streaming feature selection algorithm based on neighborhood rough sets.

Nevertheless, all aforementioned online streaming feature selection methods assume that classes are independent of each other and often ignore the hierarchical structure between classes in hierarchical classification data. Motivated by these factors, we utilize the hierarchical class structure and present an online streaming feature selection framework. Under this framework, we propose a kernelized fuzzy rough sets based online streaming feature selection algorithm for large-scale hierarchical classification learning.

3 Preliminary on kernelized fuzzy rough sets

In this section, we will review the notations and definitions of kernelized fuzzy rough sets. Fuzzy rough sets is a feasible method used to deal with numerical and fuzzy data [1, 17, 25, 26, 34, 37, 45]. However, how to effectively generate fuzzy similarity relations from data is still an important problem. Therefore, Hu et al. [27] proposed a kernelized fuzzy rough sets (KFRS) model, which used kernel function to measure the relation between samples.

Formally, a kernel fuzzy approximation space can be written as $< U, A, D, k >$, where U called a universe, A is the set of condition attributes, D is the set of decision attributes, and k is a kernel function satisfying reflexive, symmetric, and T_{cos}-transitive. All samples can be divided into subset $\{d_1, d_2, ..., d_m\}$ according to D, where m is the number of classes. For $\forall x \in U$,

$$d_i(x) = \begin{cases} 1, & x \notin d_i; \\ 0, & x \in d_i. \end{cases}$$

Definition 1 [27] Given a kernel fuzzy approximation space $< U, A, D, k >$, $x \in U$, k is a kernel function satisfying reflexive, symmetric, and T_{cos}-transitive, the fuzzy lower and upper approximation operators are defined as

$$\begin{aligned} \underline{k_S} d_i(x) &= \inf_{y \notin d_i} (1 - k(x, y)); \\ \underline{k_\theta} d_i(x) &= \inf_{y \notin d_i} (\sqrt{1 - k^2(x, y)}); \\ \overline{k_T} d_i(x) &= \sup_{y \in d_i} k(x, y); \\ \overline{k_\sigma} d_i(x) &= \sup_{y \in d_i} (1 - \sqrt{1 - k^2(x, y)}). \end{aligned} \tag{1}$$

where T, S, θ, and σ stand for fuzzy triangular norm, fuzzy triangular conorm, $T-$rediduated implication and its dual, respectively.

For simplicity, we only use select fuzzy triangular conorm in the rest of paper.

Definition 2 [27] Given a kernel fuzzy approximation space $< U, A, D, k >$, let $B \subseteq A$ be a subset of attributes. The kernel fuzzy positive region of D in term of B is defined as

$$POS_B^S(D) = \bigcup_{i=1}^{m} \underline{k_S} d_i. \tag{2}$$

Definition 3 [27] Given a kernel fuzzy approximation space $< U, A, D, k >$, let $B \subseteq A$ be a subset of attributes. The kernel fuzzy dependency function of D in term of B is defined as

$$\gamma_B^S(D) = \frac{|\bigcup_{i=1}^{m} \underline{k_S} d_i|}{|U|}. \tag{3}$$

Definition 4 [27] Given a kernel fuzzy approximation space $< U, A, D, k >$, let $B \subseteq A$ be a subset of attributes. The significance of a feature $f \in A - B$ relative to D under B is defined as

$$SIG(f, B, D) = \gamma_{B \cup f}^S(D) - \gamma_B^S(D). \tag{4}$$

The significance reflects the approximation ability of kernel fuzzy equivalence class induced by conditional attributes with respect to the decision attribute.

4 The proposed algorithms

4.1 Kernelized fuzzy rough sets for hierarchical classification

There exist different categories of hierarchical classification learning, such as graph-based and tree-based. In this paper, we propose a kernelized fuzzy rough sets for tree-based hierarchical classification learning. For simplicity, Table 1 describes the symbols most commonly used in this paper. Given a tree-based hierarchical class structure kernel fuzzy approximation space $< U, C, D_{tree}, k >$, U is a non-empty set of samples, C is a set of condition attributes, D_{tree} is the decision attribute which divides the samples into subset $\{d_1, d_2, ..., d_m\}$ (m is the number of the classes), and a kernel function k satisfying reflexive, symmetric, and T_{cos}-transitive. In these symbols, D_{tree} satisfies a pair $(D_{tree}, \prec)$, where "$\prec$" represents the "IS-A" relationship, which is the subclass-of relationship with the following properties [31]:

(1) Asymmetry: if $d_i \prec d_j$ then $d_j \nprec d_i$ for every $d_i, d_j \in D_{tree}$;
(2) Anti-reflexivity: $d_i \nprec d_i$ for every $d_i \in D_{tree}$;

Table 1 Description of symbols

Symbol	Meaning
D, $\hat{D}$	Sets of predicted and true classes
D_{aug}, $\hat{D}_{aug}$	Augmented Sets of predicted and true classes
$anc(d_i)$	The set of ancestor categories of class d_i
$des(d_i)$	The set of descendant categories of class d_i
$sib(d_i)$	The set of sibling categories of class d_i
$LCA(d_i, \hat{d}_j)$	Lowest common ancestor of classes d_i and d_j

Table 2 Three strategies of positive and negative samples' definitions

Method	Positive sample	Negative samples
Exclusive strategy [19]	A	$Not\ A$
Inclusive strategy [19]	$A + des(A)$	$Not\ [A + des(A)]$
Sibling strategy [12]	A	$sib(A)$

(3) Transitivity: if $d_i \prec d_j$ and $d_j \prec d_k$, then $d_i \prec d_k$ for every $d_i, d_j, d_k \in D_{tree}$.

Given the hierarchical class structure, there are several methods used to define the set of positive (same) and negative (different) classes for a target sample, as shown in Table 2. Compared with other strategies, sibling strategy based hierarchical class can reduce the search scope of the negative samples via using the pre-defined class hierarchy [52].

In this paper, we adopt sibling strategy as the final strategy. For $\forall x \in U$, we have

$$d_i(x) = \begin{cases} 0 & x \in sib(d_i); \\ 1 & x \in \{d_i\}. \end{cases} \tag{5}$$

Definition 5 Given $< U, C, D_{tree}, k >$, $\forall x \in U$, let d_i be a class of samples labeled with i, the fuzzy lower and upper approximation operators with sibling strategy are respectively defined as

$$\begin{aligned} \underline{k_{S}}_{sib} d_i(x) &= \inf_{y \in sib(d_i)} (1 - k(x, y)); \\ \underline{k_{\theta}}_{sib} d_i(x) &= \inf_{y \in sib(d_i)} \left(\sqrt{1 - k^2(x, y)}\right); \\ \overline{k_{T}}_{sib} d_i(x) &= \sup_{\substack{y \in \{d_i\} \\ k}} (x, y); \\ \overline{k_{\sigma}}_{sib} d_i(x) &= \sup_{y \in \{d_i\}} (1 - \sqrt{1 - k^2(x, y)}). \end{aligned} \tag{6}$$

Example 1 Considering the example data in Table 3, we have 12 samples and each sample is characterized by a condition attribute C. D_{tree} is the decision attribute which divides the samples into subset $\{d_1, d_2, d_3, d_4, d_5, d_6\}$. The tree structure of example data is shown in Fig. 1. Assume Gaussian kernel $k(x, y) = exp(-\frac{||x-y||^2}{\sigma})$ is used to compute the lower approximation with sibling strategy, and the parameter σ is set as 0.2. For x_3 with class d_2, we have $sib(d_2) = \{d_3, d_4\}$. Then, we can compute the lower approximation with the sibling strategy as follow:

$$\begin{aligned} \underline{k_{S}}_{sib} d_2(x_3) &= \inf_{y \in sib(d_2)} (1 - k(x_3, y)) = \inf_{y \in \{d_3, d_4\}} (1 - k(x_3, y)) \\ &= 1 - exp(-\frac{-||x_3 - x_7||}{0.2}) = 0.0695 \end{aligned}$$

Several properties of the kernelized fuzzy rough sets for hierarchical classification are discussed as follows.

Proposition 1 *Given $< U, C, D_{tree}, k >$, let d_i be a class of samples labeled with i, $\forall x \in U$, we have*

$$\begin{aligned} \underline{k_{S}}_{sib} d_i(x) &\geq \underline{k_S} d_i(x), \\ \underline{k_{\theta}}_{sib} d_i(x) &\geq \underline{k_\theta} d_i(x). \end{aligned} \tag{7}$$

Proof Suppose y_i is the sample with class $y_i \in sib(d_i)$ such that $\underline{k_{S}}_{sib} d_i(x) = 1 - k(x, y_i)$. Suppose y_j is the sample with class $y_j \in D_{tree} \backslash d_i$ such that $\underline{k_S} d_i(x) = 1 - k(x, y_j)$. Since $sib(d_i) \subseteq D_{tree} \backslash d_i$, we have $k(x, y_i) \leq k(x, y_j)$. Therefore, $\underline{k_{S}}_{sib} d_i(x) \geq \underline{k_S} d_i(x)$. Analogically, we can also obtain $\underline{k_{\theta}}_{sib} d_i(x) \geq \underline{k_\theta} d_i(x)$. □

Proposition 2 *Given $< U, C, D_{tree}, k >$, $x \in U$. If d_i is a class of samples labeled with i and $\forall x \in U$, we have*

$$\begin{aligned} \overline{k_{T}}_{sib} d_i(x) &= \overline{k_T} d_i(x), \\ \overline{k_{\sigma}}_{sib} d_i(x) &= \overline{k_\sigma} d_i(x). \end{aligned} \tag{8}$$

Proof Since $\overline{k_T} d_i(x) = \sup_{y \in d_i} k(x, y)$ and $\overline{k_{T}}_{sib} d_i(x) = \sup_{y \in d_i} k(x, y)$. Therefore, $\overline{k_{T}}_{sib} d_i(x) = \overline{k_T} d_i(x)$. Analogically, $\overline{k_{\sigma}}_{sib} d_i(x) = \overline{k_\sigma} d_i(x)$. □

4.2 Kernelized fuzzy rough sets using sibling strategy based feature evaluation

As we know, the kernelized fuzzy rough sets theory is an effective tool for selective discriminative features, and feature evaluation is a main step in the process of feature selection.

Definition 6 Given $< U, C, D_{tree}, k >$, let $B \subseteq C$ be a subset of attributes. $D_{tree} = \{d_0, d_1, d_2, ..., d_m\}$, where d_0 is the root of the tree and is not the real class, and U is divided into $\{d_1, d_2, ..., d_m\}$ by the decision attribute, where m is the

Table 3 Example data

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

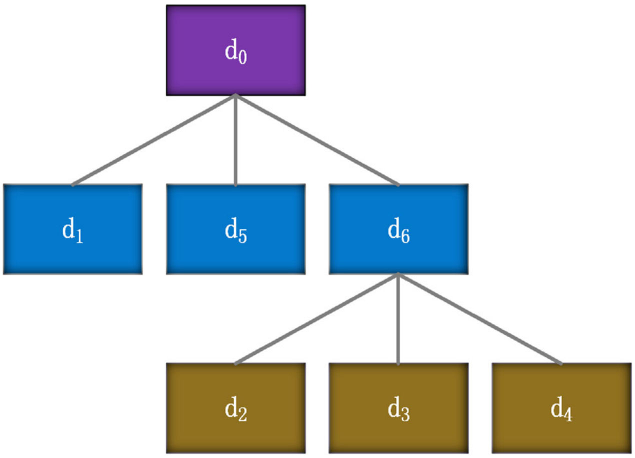

Fig. 1 Tree structure of example data

number of classes. The kernel fuzzy positive region of D_{tree} in term of B is defined as

$$POS^S_{B\,sib}(D_{tree}) = \bigcup_{i=1}^{m} \underline{k_S}_{sib} d_i. \tag{9}$$

Definition 7 Given $< U, C, D_{tree}, k >$, let $B \subseteq C$ be a subset of attributes, and U is divided into $\{d_1, d_2, ..., d_m\}$ by the decision attribute, where m is the number of classes. The quality of classification approximation is defined as

$$\gamma^S_{B\,sib}(D_{tree}) = \frac{|\cup_{i=1}^{m} \underline{k_S}_{sib} d_i|}{|U|}. \tag{10}$$

As $\underline{k_S}_{sib} d_i(x) = \inf_{y \in sib(d_i)} (1 - k(x, y))$, we can get

$$|\cup_{i=1}^{m} \underline{k_S}_{sib} d_i| = \sum_{j=1}^{|U|} \sum_{i=1}^{m} \underline{k_S}_{sib} d_i(x_j). \tag{11}$$

Let $x_j \notin \{d_i\}$, we have $\underline{k_S} d_i(x_j) = 0$. We also have $\underline{k_S}_{sib} d_i(x_j) = 0$ according to Proposition 1. Thus, we have

$$\begin{aligned} \textstyle\sum_{j=1}^{|U|} \sum_{i=1}^{m} \underline{k_S}_{sib} d_i(x_j) &= \textstyle\sum_{j=1}^{|U|} \underline{k_S}_{sib} d(x_j) \\ &= \textstyle\sum_{j=1}^{|U|} \inf_{x_j \in \{d\}, y \in sib(d)} (1 - k(x_j, y)), \end{aligned} \tag{12}$$

where d is the class label of x_j.

The coefficient of classification quality manifests the approximation ability of the approximation space, or the ability that the decision attribution is defined by the granulated space, contained in feature subset [27]. The coefficient named the dependency between decision attribute and condition attribute is able to evaluate the condition attributes with degree $\gamma^S_{B\,sib}(D_{tree})$.

Algorithm 1 Sibling Strategy based Feature Evaluation (SSFE).

Input: $< U, C, D_{tree}, k >$, **reg** $= 0$, and B
Output: **reg**

```
for i = 1 to |U| do
    Compute decision of sample x_i as d_i;
    Select sample with class sib(d_i) as X_sib;
    if |X_sib|==0 then
        Random select samples as X_sib out of d_i;
    end if
    for each y ∈ X_sib do
        Compute 1 − k(x_i, y);
    end for
    Select ŷ such that k_S_sib d_i(x_i) = 1 − k(x_i, ŷ);
    reg = reg + 1 − k(x_i, ŷ);
end for
reg = reg/|U|;
return reg.
```

4.3 Online streaming feature selection for large-scale hierarchical classification via kernelized fuzzy rough sets

In this section, we propose a framework of online streaming feature selection for large-scale hierarchical classification learning. This framework consists of two-phase: online important feature selection and online redundant feature update. The details of the proposed method is shown in the following sections.

4.3.1 Online important feature selection

In order to measure the significance of feature relative to the decision attribute under the selected features, the kernel fuzzy dependency with respect to D_{tree} can be employed. Because the dependency reflects the discernibility of feature, and the greater the dependency is, the greater the recognition power of feature has. The significance of feature in a tree-based hierarchical class structure using kernel fuzzy approximation space $< U, C, D_{tree}, k >$, can be defined as follow.

Definition 8 Given the decision attribute D_{tree} , S_{t-1} is the selected feature subset at time $t-1$, and F_t is a new arrived feature at time t. Therefore, the significance degree of feature F_t can be defined as

$$SD(F_t, S_{t-1}, D_{tree}) = \frac{|\gamma^S_{S_{t-1} \cup F_t\, sib}(D_{tree}) - \gamma^S_{S_{t-1}\, sib}(D_{tree})|}{|\gamma^S_{S_{t-1}\, sib}(D_{tree})|}. \tag{13}$$

As $\gamma^S_{S_{t-1}sib}(D_{tree}) \in [0, 1]$, and $\gamma^S_{S_{t-1}\cup F_t sib}(D_{tree}) \geq \gamma^S_{S_{t-1}sib}(D_{tree})$, we have $SD(F_t, S_{t-1}, D_{tree}) \in [0, 1]$. We say that feature F_t is superfluous relative to the currently selected features if $SD(F_t, S_{t-1}, D_{tree}) = 0$; Otherwise, F_t has a positive impact on the selected features S_{t-1}.

Definition 9 Given the decision attribute D_{tree}, S_t is the selected feature subset at time t. For each feature $F_i \in S_t$, we can calculate the dependency between F_i and D_{tree}, and the mean value of all dependency values between each feature F_i and the decision attribute D_{tree} can be defined as

$$\Re(S, D_{tree}) = \frac{\sum_{F_i \in S_t} \gamma^S_{S_t sib}(D_{tree})}{|S|}. \tag{14}$$

Definition 10 Assume S_{t-1} is the selected feature subset at time $t-1$, F_t is a new feature at time t. If $\gamma^S_{F_t sib}(D_{tree}) \geq \Re(S_{t-1}, D_{tree})$, F_t is identified as an important feature with respect to the decision attribute D_{tree}; Otherwise, F_t is abandoned as a nonsignificant feature.

From Definitions 8 and 10, the local optimum is an important analysis process, i.e., it is meaningful for the arrival sequence of features to choose the new feature. What is more, it is hard to get a satisfied condition in the following features if there is a high discriminative capacity in the former arrived features. In addition, F_t is redundant but links with the currently selected features. Besides, F_t can not be identified worthless since it would be much more precious compared with its corresponding superfluous features. Accordingly, a further online redundancy updation is necessary.

4.3.2 Online redundant feature updation

In this section, online redundant feature updation can get an optimal feature subset by reevaluating the newly arrived feature F_t. F_t is considered as an superfluous feature in the online important feature selection period. Reevaluating feature can be completed in two steps: (1) selecting the redundant feature within new features, i.e., redundancy recognition, and (2)ensuring the preserved features, i.e., redundancy updation. In order to clearly filter out superfluous features, pairwise comparisons are used to online calculate the correlations between features and the decision attribute.

Definition 11 (Redundancy recognition) Assume S_{t-1} is the selected feature subset at time $t-1$, and an important threshold δ is given. If $\exists F_k \in S_{t-1}$ such that $SD(F_i, F_k, D_{tree}) \leq \delta (0 \leq \delta \leq 1)$, it proves that adding F_i alone to F_k does not enhance the predictive capability of F_k. That is, F_k is redundant with F_i.

Definition 12 (Redundancy updation) Assume S_{t-1} is the selected feature subset at time $t-1$, $F_k \in S_{t-1}$, F_t is a new feature at time t, and an important threshold δ is given. If $SD(F_t, F_k, D_{tree}) \leq \delta (0 \leq \delta \leq 1)$ holds, then F_t should be added into S_{t-1} if $\gamma^S_{F_t sib}(D_{tree}) \geq \gamma^S_{F_k sib}(D_{tree})$; Otherwise, F_k should be preserved if $\gamma^S_{F_t sib}(D_{tree}) \leq \gamma^S_{F_k sib}(D_{tree})$.

4.4 Kernelized Fuzzy rough sets based online hierarchical streaming feature selection(KFOHFS)

To illustrate the process of online hierarchical streaming feature selection, a flowchart of online streaming feature selection framework is given in Fig. 2. Under this framework, we propose the KFOHFS algorithm in detail, as shown in Algorithm 2.

Algorithm 2 Kernelized Fuzzy rough sets based Online Hierarchical streaming Feature Selection(KFOHFS).

Input: F_t: predictive features; D_{tree}: decision attribute; S_{t-1}: the selected feature set at time $t-1$; δ: a redundance threshold$(0 \leq \delta \leq 1)$.
Output: S_t: the selected features at time t.

1: $depMean = 0$;
2: **repeat**
3:   %online importance selection%;
4:   Get a new feature F_t at time t;
5:   Compute the dependency of $SD(F_t, S_{t-1}, D_{tree})$ by combining algorithm 1 and (13) ;
6:   **if** $SD(F_t, S_{t-1}, D_{tree}) > \delta$ **then**
7:     $S_t = S_{t-1} \bigcup F_t, deapMean = \Re(S_t, D_{tree})$ and go to Step 24;
8:   **else**
9:     **if** $\gamma^S_{F_t sib}(D_{tree}) < depMean$ **then**
10:       Discard F_t and go to Step 24;
11:     **else**
12:       %online redundancy updation%
13:       **while** $\exists F_k \in S_{t-1}$ **do**
14:         Compute the dependency of $SD(F_t, F_k, D_{tree})$;
15:         **if** $SD(F_t, F_k, D_{tree}) \leq \delta$ and $\gamma^S_{F_t sib}(D_{tree}) \leq \gamma^S_{F_k sib}(D_{tree})$ **then**
16:           Discard F_t and go to Step 24;
17:         **end if**
18:         **if** $SD(F_t, F_k, D_{tree}) \leq \delta$ and $\gamma^S_{F_t sib}(D_{tree}) \geq \gamma^S_{F_k sib}(D_{tree})$ **then**
19:           $S_{t-1} = S_{t-1} - F_k$, $S_t = S_{t-1} \bigcup F_t$, $deapMean = \Re(S_t)$;
20:         **end if**
21:       **end while**
22:     **end if**
23:   **end if**
24: **until** no features are available
25: return S_t.

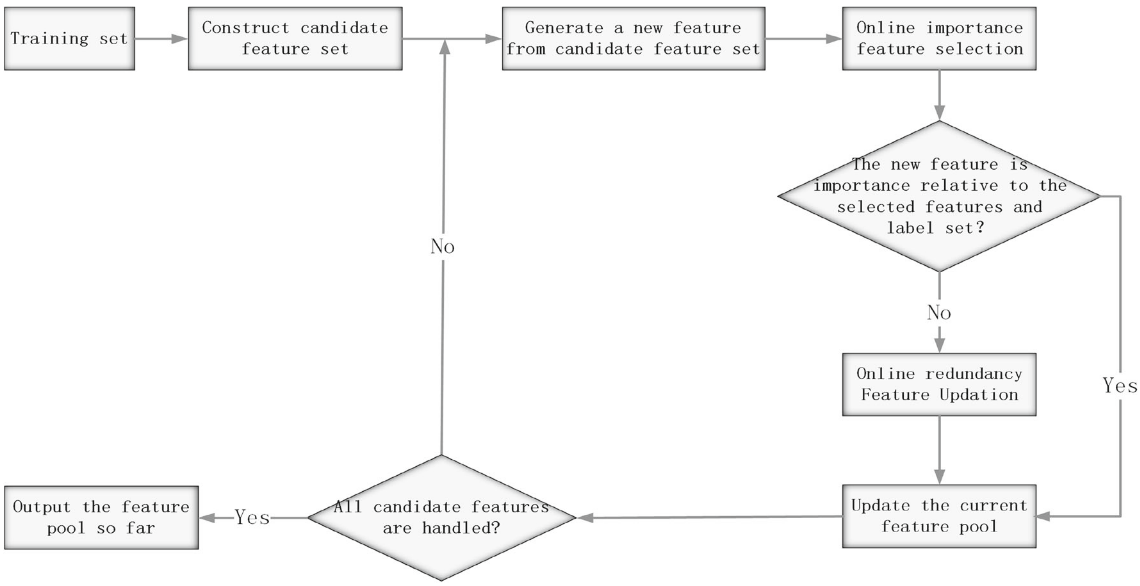

Fig. 2 The process of online hierarchical streaming feature selection

5 Experimental analysis

In this section, we first describe the information of data sets, evaluation measures, and comparative methods respectively. Then, the influence of parameter δ is reported. Moreover, we compare the performance of four evaluation metrics among 6 algorithms to verify the effectiveness of the proposed method. Finally, statistical analysis and time complexity analysis are adopted to further explore the performance analysis.

5.1 Data sets and experimental settings

5.1.1 Data sets

There are six data sets in the experiments, and their basic information is listed in Table 4. For these data sets, AWAphog [33] has 10 classes and 9,607 samples, which is collected from Animals. Bridges [10] is from the University of Colifornia-Irvine (UCI) library. Cifar [32] is labeled subsets of the 80 million tiny image data sets. VOC [20] provides the vision and machine learning communities with a standard data set of images and annotation as well as standard evaluation procedures. DD [16] is a protein data set, which has 27 real classes and four major structural classes. F194 [47] is also a protein data set, which has 194 classes, which are all leaf nodes.

5.1.2 Hierarchical classification evaluation measures

To evaluate the performance of the proposed algorithm, three additional hierarchical classification evaluation measures, i.e., Tree Induced Error (TIE) [13], Hierarchical-F_1 [11] and Lowest Common Ancestor-F_1 ($LCA - F_1$) [43], are introduced to describe the degree of misclassification in hierarchical structure, respectively.

Let D and $\hat{D}$ denote true classes and predicted classes of instances respectively. Then, the augmentation of D and $\hat{D}$ is defined as

$$D_{aug} = D \cup anc(D),\ \hat{D}_{aug} = \hat{D} \cup anc(\hat{D}), \tag{15}$$

and the lowest common ancestor augmentation of D and $\hat{D}$ is defined as

$$D_{aug}^{LCA} = D \cup LCA(D, \hat{D}),\ \hat{D}_{aug}^{LCA} = \hat{D} \cup LCA(D, \hat{D}). \tag{16}$$

Hierarchical Precision and Hierarchical Recall are defined as

$$P_H = \frac{|\hat{D}_{aug} \cap D_{aug}|}{|\hat{D}_{aug}|},\ R_H = \frac{|\hat{D}_{aug} \cap D_{aug}|}{|D_{aug}|}. \tag{17}$$

Table 4 Dataset description

Data set	Type	Instance	Features	Class	Node	Height
AWAphog	Image	6405	252	10	17	4
Bridges	Num&Sym	108	12	6	8	3
Cifar	Image	50000	512	100	121	3
VOC	Image	7178	1000	20	30	5
DD	Protein	3625	473	27	32	3
F194	Protein	8525	473	194	202	3

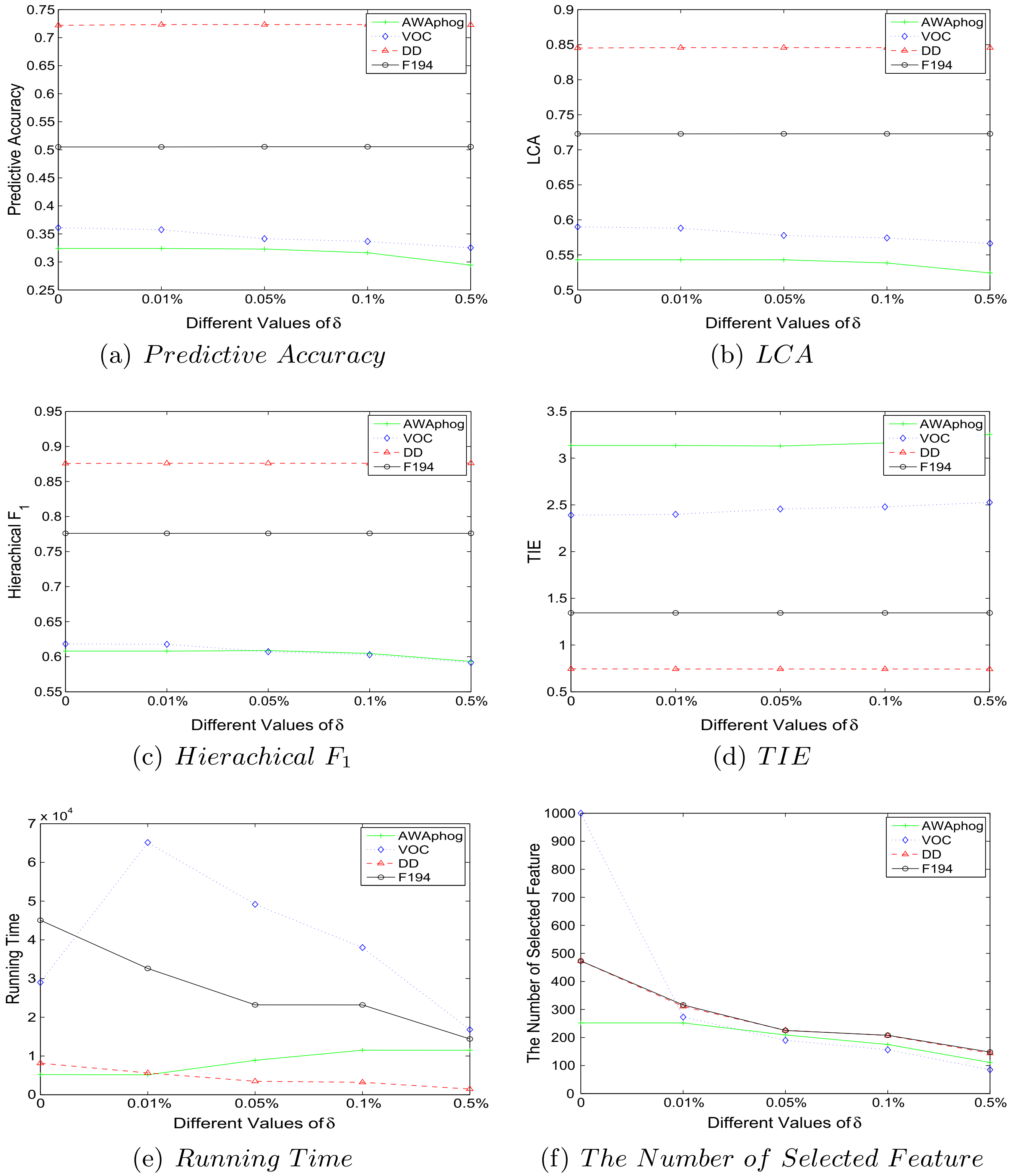

Fig. 3 Comparison of KFOHFS (control algorithm) with different values δ

Table 5 Predictive accuracy using the LSVM classifier

Data set	OFS-Density	OFS-A3M	Fast-OSFS	OSFS	SAOLA	KFOHFS
AWAphog	0.1867	0.2512	0.2195	0.2080	0.1781	**0.3240**
Bridges	0.5644	0.6256	0.6300	0.6300	0.6300	**0.6311**
VOC	0.2898	0.3165	0.2671	0.2594	0.2568	**0.3575**
DD	0.3079	0.7054	0.3707	0.3704	0.2929	**0.7233**
F194	0.1010	0.0879	0.2537	0.2197	0.2252	**0.5051**
Cifar	0.1289	0.2710	0.0747	0.0674	0.0201	**0.2768**
Average	*0.2631*	*0.3763*	*0.3026*	*0.2925*	*0.2672*	***0.4696***

Lowest Common Ancestor Hierarchical Precision and Lowest Common Ancestor Hierarchical Recall are defined as

$$P_{LCAH} = \frac{|\hat{D}_{aug}^{LCA} \cap D_{aug}^{LCA}|}{|\hat{D}_{aug}^{LCA}|}, R_{LCAH} = \frac{|\hat{D}_{aug}^{LCA} \cap D_{aug}^{LCA}|}{|D_{aug}^{LCA}|}. \tag{18}$$

where $|\cdot|$ denotes the count of elements.

The TIE is computed by predicting class $\hat{D}_i$ when the true classes is D_i

$$TIE(D, \hat{D}) = \frac{1}{|D|}\sum_{i=1}^{|D|} |E_H(D_i, \hat{D}_i)|, \tag{19}$$

where $E_H(D_i, \hat{D}_i)$ is the set of edges along the path from D_i to $\hat{D}_i$ in the hierarchy, and $|\cdot|$ denotes the count of elements.

The $Hierachical - F_1$ is the F_1-measure of hierarchical precision and recall, and defined as

$$Hierachical - F_1 = \frac{2 \cdot P_H \cdot R_H}{P_H + R_H}. \tag{20}$$

The $LCA - F_1$ is the F_1-measure of lowest common ancestor hierarchical precision and recall, and defined as

$$LCA - F_1 = \frac{2 \cdot P_{LCAH} \cdot R_{LCAH}}{P_{LCAH} + R_{LCAH}}. \tag{21}$$

5.1.3 Experimental settings

To explain the effectiveness of the proposed algorithm, five state-of-the-art online streaming feature selection methods, including OFS-Density [57], OFS-A3M [54], Fast-OSFS [48], OSFS [48], and SAOLA [49] are selected as baselines. For OSFS, Fast-OSFS, and SAOLA, the significance level α is set as 0.01, as suggested in the literature. For KFOHFS, the parameter δ is acquiescently set to 0.01% and more details refer to Section 5.2. Besides, the basic classifier LSVM is used to evaluate the classification performance of all feature selection algorithms. Ultimately, $Predictive\ Accuracy$, $LCA - F_1$, $Hierachical - F_1$, and TIE are selected as criteria to evaluate the performance of feature selection. Since the four criteria come from different evaluate viewpoints, and few algorithms are superior to the algorithms based on the above four criteria.

5.2 The influence of δ

In this section, we will analyze the influence of δ in KFOHFS. Four values (0.01%, 0.05%, 0.1% and 0.5%) of δ and the absolutely equivalent constraint ($\delta = 0$) are selected as compared objects. Fig. 3 demonstrates the experimental results of five different δ values (0, 0.01%, 0.05%, 0.1% and 0.5%) on these data sets (AWAphog, DD, VOC, F194), in which, Fig. 3(e) and Fig. 3(f) represent the running time and the mean of selected features on these data sets, respectively.

Table 6 $LCA - F_1$ score using the LSVM classifier (↑)

Data set	OFS-Density	OFS-A3M	Fast-OSFS	OSFS	SAOLA	KFOHFS
AWAphog	0.4482	0.4948	0.4712	0.4643	0.4484	**0.5431**
Bridges	0.7528	0.7861	0.7852	0.7852	0.7852	**0.7917**
VOC	0.5411	0.5605	0.5236	0.5184	0.5165	**0.5882**
DD	0.5743	0.8370	0.6339	0.6338	0.5786	**0.8458**
F194	0.4526	0.4517	0.5809	0.5553	0.5635	**0.7227**
Cifar	0.4352	0.5328	0.3967	0.3908	0.3559	**0.5365**
Average	*0.534*	*0.6105*	*0.5652*	*0.558*	*0.5414*	***0.6713***

Table 7 $Hierachical - F_1$ score using the LSVM classifier (↑)

Data set	OFS-Density	OFS-A3M	Fast-OSFS	OSFS	SAOLA	KFOHFS
AWAphog	0.5231	0.5672	0.5446	0.5391	0.5297	**0.6080**
Bridges	0.7852	0.8031	0.8068	0.8068	0.8068	**0.8117**
VOC	0.5571	0.5870	0.5300	0.5230	0.5204	**0.6177**
DD	0.6100	0.8703	0.6874	0.6874	0.6286	**0.8760**
F194	0.5048	0.5117	0.6597	0.6311	0.6438	**0.7760**
Cifar	0.4511	0.5515	0.4102	0.4033	0.3651	**0.5551**
Average	*0.5719*	*0.6485*	*0.6065*	*0.5985*	*0.5824*	***0.7074***

From Fig. 3, we have the following observations: (1) There is no significant difference between different values of δ. In which, $\delta = 0$ and $\delta = 0.01\%$ get the best performance in three data sets (AWAphog, DD, F194), and $\delta = 0.01\%$ gets the best performance in all data sets; (2) With the augment of values of δ, the corresponding running time fleetly increases in two data sets (DD, F194); (3) For the number of selected features, $\delta = 0$ selects more features than others, which denotes some redundant features are caused by the exactly equal constraint.

In summary, the exactly equal restriction is able to eliminate redundant features and improve predictive accuracy. Therefore, in the following experiments, we set $\delta = 0.01\%$.

5.3 Performance analysis on evaluation measures

In this section, we group experiments into two parts: (1) We make comparison on the performance of four evaluation metrics ($Predictive\ Accuracy$, $LCA - F_1$, $Hierachical - F_1$, and TIE) among OFS-Density, OFS-A3M, Fast-OSFS, OSFS, SAOLA, and KFOHFS; (2) Based on the statistical analysis in the comparison algorithms, we analyze performance in a systematic way.

Tables 5-8 show the performance of OFS-Density, OFS-A3M, Fast-OSFS, OSFS, SAOLA, and KFOHFS with respect to four evaluation metrics. Among these tables, bold font embodies the optimum performance of each data set, italics shows the average performance of each algorithm on all data sets, ↑ manifests the larger the better, and ↓ demonstrates the smaller the better, respectively. The experiments show that, in all four evaluation measures, KFOHFS dramatically outperforms other online streaming feature selection algorithms for all datasets.

The Friedman test [21] and Bonferroni-Dunn test [18] are adopted to further explore the performance analysis over the six feature selection algorithms. Given k comparing algorithms and N data sets, the average rank of algorithm j on all data sets is $R_j = \frac{1}{N}\sum_{i=1}^{N} r_i^j$, where r_i^j is the rank of the j-th algorithm on the i-th data set. Under the null-hypothesis, the Friedman statistic following a Fisher distribution with $(k-1)$ and $(k-1)(N-1)$ degrees of freedom can be defined as

$$F_F = \frac{(N-1)\chi_F^2}{N(k-1) - \chi_F^2},$$
$$where \quad \chi_F^2 = \frac{12N}{k(k+1)}\left(\sum_{i=1}^{k} R_i^2 - \frac{k(k+1)^2}{4}\right) \tag{22}$$

Table 9 presents the Friedman statistic F_F on different evaluation metrics and the corresponding critical values. In accordance with Table 9, the null hypothesis of "equal" performance among all algorithms is obviously rejected on all different evaluation measures at significance level $\alpha = 0.10$. Afterwards, we select given post-hoc tests, such

Table 8 TIE score using the LSVM classifier (↓)

Data set	OFS-Density	OFS-A3M	Fast-OSFS	OSFS	SAOLA	KFOHFS
AWAphog	3.8151	3.4626	3.6434	3.6868	3.7627	**3.1363**
Bridges	1.1389	1.0000	1.0093	1.0093	1.0093	**0.9722**
VOC	2.6800	2.5632	2.7877	2.8175	2.8268	**2.3980**
DD	2.3399	0.7779	1.8759	1.8759	2.2284	**0.7437**
F194	2.9713	2.9300	2.0418	2.2133	2.1370	**1.3440**
Cifar	3.2936	2.6907	3.5390	3.5803	3.8094	**2.6691**
Average	*2.7065*	*2.2374*	*2.4829*	*2.5305*	*2.6289*	***1.8772***

Table 9 Summary of the Friedman statistics $F_F(k = 6, N = 6)$ and the critical value on different evaluation measures(k : comparing algorithms; N : data sets)

Evaluation measure	F_F	critical value ($\alpha = 0.1000$)
Predictive Accuracy	6.9318	2.0800
$LCA - F_1$	7.5249	
$Hierachical - F_1$	8.9381	
TIE	10.4791	

as the Bonferroni-Dunn test, to further analyze the related performance among the comparing algorithms. Here, the difference between the average ranks of KFOHFS and one baseline is compared with the following *critical difference* (CD):

$$CD_\alpha = q_\alpha \sqrt{\frac{k(k+1)}{6N}}. \tag{23}$$

Hence, we have $q_\alpha = 2.3260$ at significance level $\alpha = 0.10$, and thus CD=2.5124 ($k = 6$, $N = 6$).

To visually display the relative performance of KFOHFS and other algorithms, Fig. 4 clarifies the CD diagram on different evaluation metrics, where the average ranks of each comparing algorithm are signed along the axis. From Fig. 4, we can observe that KFOHFS performs obviously better than OFS-Density, SAOLA, and OSFS on all evaluation measures. In conclusion, KFOHFS is not statistically better than OFS-A3M and Fast-OSFS, but it outperforms all competing algorithms on all data sets, due to KFOHFS utilizes the hierarchical class information.

5.4 Time complexity analysis

To illustrate the efficiency of each algorithm, we compare the time complexity of each algorithm (OFS-Density, OFS-A3M, Fast-OSFS, OSFS, SAOLA) in this section. As we know, the dependency between features is taken as the main time complexity of KFOHFS. According to Section 4.2, the time complexity of SSFE is $O(|U|^2 \cdot log|U|)$. $|C|$ is the total number of features. Thus, the time complexity of KFOHFS is $O(|C|^2 \cdot |U|^2 \cdot log|U|)$. The time complexity of both OFS-Density and OFS-A3M is $O(|C|^2 \cdot |U|^2 \cdot \log|U|)$, where $|C|$ is the total number of features and $|U|$ is the number of samples. The time complexity of OSFS is $O(|C|^2 \cdot k^{|C|})$, where $k^{|C|}$ is all subsets of size and is less than or equal to $k(1 \leq k \leq |C|)$. The worst time complexity of Fast-OSFS is $O(|CR| \cdot |C| \cdot k^{|C|})$, where $|CR|$ is the number of all relevant features in $|C|$. The time complexity of SAOLA is $O(|C|^2)$.

From the above theoretical analysis of time complexity, it can be observed that the time complexity of OFS-Density, OFS-A3M, and KFOHFS is equal. Compared with other comparison algorithms, the influence of the total number of samples should be considered in the calculation process, which need more calculation time. Therefore, the time complexity of Fast-OSFS, OSFS and SAOLA is relatively optimal.

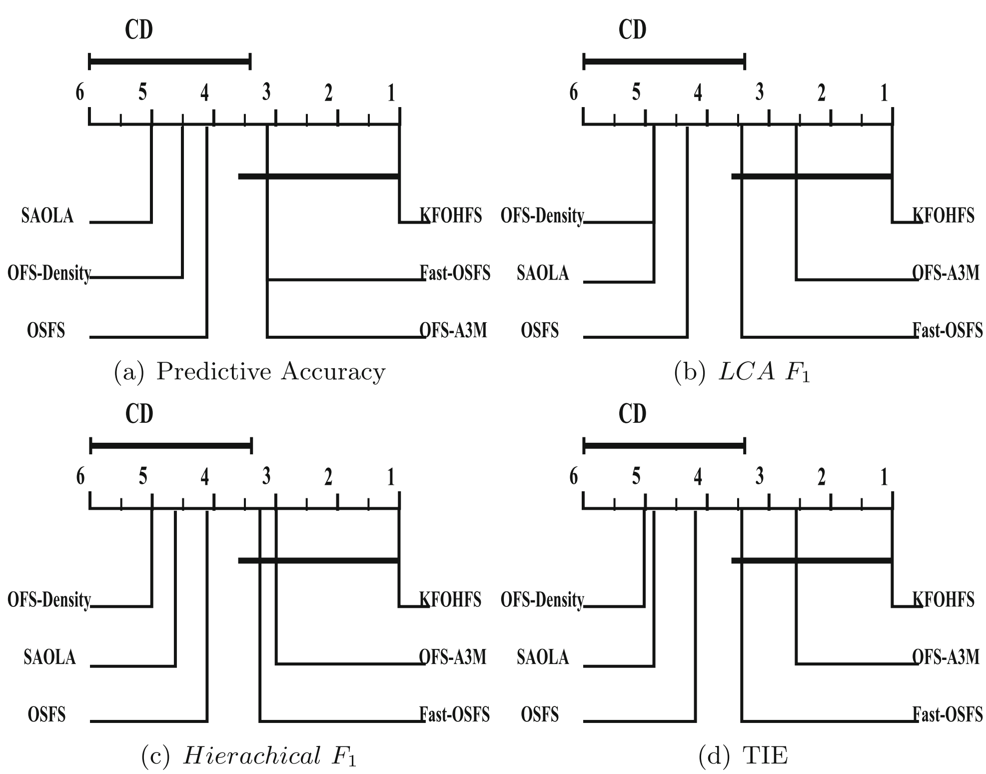

Fig. 4 Comparison of KFOHFS (control algorithm) against other comparing algorithms with the Bonferroni-Dunn test

6 Conclusions

In this paper, we presented a kernelized fuzzy rough sets based online streaming feature selection for large-scale hierarchical classification learning. We first used sibling nodes as the nearest samples from different classes to granulate all instances, and define a new dependency function to evaluate the features. Then, two phases were divided in the proposed online hierarchical streaming feature selection, i.e., online important feature selection and online redundant feature updation. Specially, KFOHFS did not need the domain knowledge before learning, and measured the fuzzy relation between samples effectively. In addition, KFOHFS took advantage of hierarchical class structure for classification learning. Compared with the other five state-of-the-art online streaming feature selection algorithms, KFOHFS achieves competitive performance against all competitors in all flat and hierarchical evaluations. However, the current implementation of the algorithm is limited to a tree structure of class labels. In the future, we will design online streaming feature selection algorithms for general graph structures.

Acknowledgments We are very grateful to the anonymous reviewers for their valuable comments and suggestions. This work is supported by Grants from the National Natural Science Foundation of China (No. 61672272),the Natural Science Foundation of Fujian Province (Nos. 2018J01547 and 2018J01548) and the Department of Education of Fujian Province (No. JAT180318).

References

1. Abualigah L, Hanandeh E (2015) Applying genetic algorithms to information retrieval using vector space model. International Journal of Computer Science, Engineering and Applications 5:19–28
2. Abualigah L, Khader A (2017) Unsupervised text feature selection technique based on hybrid particle swarm optimization algorithm with genetic operators for the text clustering. J Supercomput 73:4773–4795
3. Abualigah L, Khader A, Hanandeh E (2017) A new feature selection method to improve the document clustering using particle swarm optimization algorithm. J Computational Sci 25:456–466
4. Abualigah L, Khader A, Hanandeh E, Gandomi A (2017) A novel hybridization strategy for krill herd algorithm applied to clustering techniques. Appl Soft Comput 60:423–435
5. Abualigah L, Khader A, Hanandeh E (2018) Hybrid clustering analysis using improved krill herd algorithm. Appl Intell 48: 4047–4071
6. Abualigah L, Khader A, Hanandeh E (2018) A combination of objective functions and hybrid krill herd algorithm for text document clustering analysis. Eng Appl Artif Intel 73:111–125
7. Abualigah L (2019) Feature Selection and Enhanced Krill Herd Algorithm for Text Document Clustering. Studies in Computational Intelligence
8. Aho A, Hopcroft J, Ullman J (1976) On finding lowest common ancestors in trees. SIAM J Comput 5:115–132
9. Ashburner M, Ball C, Blake J, Botstein D, Butler H, Cherry J, Sherlock G (2000) Gene ontology: tool for the unification of biology. The gene ontology consortium. Nat Genet 25:25–29
10. Blake C, Merz C (2000) UCI repository of machine learning databases. http://www.ics.uci.edu/~mlearn/MLRepository.html
11. Cai L, Hofmann T (2007) Exploiting known taxonomies in learning overlapping concepts. International Joint Conference on Artificial Intelligence, Hyderabad, pp 714–719
12. Ceci M, Malerba D (2007) Classifying web documents in a hierarchy of categories: a comprehensive study. Intell Info Sys 28:37–38
13. Dekel O, Keshet J, Singer Y (2004) Large margin hierarchical classification. International Conference on Machine Learning, Alberta, pp 1–8
14. Eskandari S, Javidi M (2016) Online streaming feature selection using rough sets. Int J Approx Reason 69:35–57
15. Deng J, Dong W, Socher R, Li L, Li K, Fei L (2009) ImageNet: A large-scale hierarchical image database. Computer Vision and Pattern Recognition, Florida, 248–255
16. Ding C, Dubchak I (2001) Multi-class protein fold recognition using support vector machines and neural networks. Bioinformatics 17:349–358
17. Dubois D, Prade H (1990) Rough fuzzy sets and fuzzy rough sets. Int J Gen Syst 17:191–209
18. Dunn O (1961) Multiple comparisons among means. J Am Stat Assoc 56:52–64
19. Eisner R, Poulin B, Szafron D, Lu P, Greiner R (2005) Improving protein function prediction using the hierarchical structure of the gene ontology. Computational Intelligence in Bioinformatics and Computational Biology, La Jolla, pp 1–10
20. Everingham M, Van G, Williams C, Win J, Zisserman A (2010) The Pascal Visual Object Classes (VOC) challenge. Int J Comput Vis 88:303–338
21. Friedman M (1940) A comparison of alternative tests of significance for the problem of *m* rankings. Ann Math Stat 11: 86–92
22. Freeman C, Kulic D, Basir O (2011) Joint feature selection and hierarchical classifier design. Systems, Man and Cybernetics, Arizona, 1728–1734
23. Genton M (2002) Classes of kernels for machine learning: a statistics perspective. J Mach Learn Res 2:299–312
24. Gopal S, Yang Y (2015) Hierarchical bayesian inference and recursive regularization for large-scale classification. ACM Transactions on Knowledge Discovery From Data 9:18–29
25. Hu Q, Yu D, Xie Z (2006) Information-preserving hybrid data reduction based on fuzzy-rough techniques. Pattern Recogn Lett 27:414–423
26. Hu Q, Xie Z, Yu D (2007) Hybrid attribute reduction based on a novel fuzzy-rough model and information granulation. Pattern Recogn 40:3509–3521
27. Hu Q, Yu D, Pedrycz W, Chen D (2011) Kernelized fuzzy rough sets and their applications. IEEE Trans Knowl Data Eng 23:1649–1667
28. Hu X, Zhou P, Li P, Wang J, Wu X (2018) A survey on online feature selection with streaming features. Frontiers of Computer Science in China 12:479–493
29. Javidi M, Eskandari S (2016) Streamwise feature selection: a rough set method. Int J Mach Learning Cybern 9:667–676
30. Jensen R, Shen Q (2009) New approaches to fuzzy-rough feature selection. IEEE Trans Fuzzy Syst 17:824–838
31. Kosmopoulos A, Partalas I, Gaussier E, Paliouras G, Androutsopoulos I (2015) Evaluation measures for hierarchical classification: a unified view and novel approaches. Data Min Knowl Disc 29:820–865
32. Krizhevsky A, Hinton G (2009) Learning multiple layers of features from tiny images. Handbook of Systemic Autoimmune Diseases 1:1124–1232

33. Lampert C, Nickisch H, Harmeling S (2009) Learning to detect unseen object classes by between-class attribute transfer. Computer Vision and Pattern Recognition, Florida, 951–958
34. Li Y, Wu S, Lin Y, Liu J (2017) Different classes' ratio fuzzy rough set based robust feature selection. Knowl Based Sys 120:74–86
35. Lin Y, Hu Q, Liu J, Li J, Wu X (2017) Streaming feature selection for multilabel learning based on fuzzy mutual information. IEEE Trans Fuzzy Syst 25:1491–1507
36. Liu J, Lin Y, Li Y, Weng W, Wu S (2018) Online multi-label streaming feature selection based on neighborhood rough set. Pattern Recogn 84:273–287
37. Mi J, Zhang W (2004) An axiomatic characterization of a fuzzy generalization of rough sets. Inform Sci 160:235–249
38. Moser B (2006) On representing and generating kernels by fuzzy equivalence relations. J Mach Learn Res 7:2603–2620
39. Nouranivatani N, Lopezsastre R, Williams S (2015) Structured output prediction with hierarchical loss functions for seafloor imagery taxonomic categorization. Iberian Conference on Pattern Recognition and Image Analysis, Santiago de Compostela, 173–183
40. Rahmaninia M, Moradi P (2017) OSFSMI: Online stream feature selection method based on mutual information. Appl Soft Comput 68:733–746
41. Silla C, Freitas A (2011) A survey of hierarchical classification across different application domains. Data Mining Knowledge Discovery 22:31–72
42. Song J, Zhang P, Qin S, Gong J (2015) A method of the feature selection in hierarchical text classification based on the category discrimination and position information. IEEE Trans Eng Manag 53:555–569
43. Struyf J, Deroski S, Blockeel H, Clare A (2005) Hierarchical multi-classification with predictive clustering trees in functional genomics. Portuguese Conference on Artificial Intelligence, Covilha, 272–283
44. Sun A, Lim E (2001) Hierarchical text classification and evaluation. International Conference on Data Mining, California, 521–528
45. Wang C, Shao M, He Q, Qian Y, Qi Y (2016) Feature subset selection based on fuzzy neighborhood rough sets. Knowl Based Sys 111:173–179
46. Wang C, Lin Y, Liu J (2019) Feature selection for multi-label learning with missing labels. Appl Intell 49:3027–3042
47. Wei L, Liao M, Gao X, Zou Q (2015) An improved protein structural classes prediction method by incorporating both sequence and structure information. IEEE Trans Nanobioscience 14:339–349
48. Wu X, Yu K, Ding W, Wang H, Zhu X (2013) Online feature selection with streaming features. IEEE Trans Pattern Anal Mach Intell 35:1178–1192
49. Yu K, Wu X, Ding W, Pei J (2016) Scalable and accurate online feature selection for big data. ACM Transactions on Knowledge Discovery From Data 11:16–37
50. Zhang J, Li C, Lin Y, Shao Y, Li S (2017) Computational drug repositioning using collaborative filtering via multi-source fusion. Expert Systems With Applications 84:281–289
51. Zhao H, Zhu P, Wang P, Hu Q (2017) Hierarchical feature selection with recursive regularization. International Joint Conference on Artificial Intelligence, Melbourne, 3483–3489
52. Zhao H, Wang P, Hu Q, Zhu P (2019) Fuzzy rough set based feature selection for large-scale hierarchical classification. IEEE Trans Fuzzy Syst 27:1891–1903
53. Zhao H, Hu Q, Zhu P, Wang Y, Wang P (2019) A recursive regularization based feature selection framework for hierarchical classification. IEEE Trans Knowl Data Eng 27:1–13
54. Zhou P, Hu X, Li P (2017) A New online feature selection method using neighborhood rough set. IEEE International Conference on Big Knowledge. Hefei, 135–142
55. Zhou P, Hu X, Li P, Wu X (2017) Online feature selection for high-dimensional class-imbalanced data. Knowl Based Sys 136:187–199
56. Zhou P, Hu X, Li P, Wu X (2019) Online streaming feature selection using adapted Neighborhood Rough Set. Inform Sci 481:258–279
57. Zhou P, Hu X, Li P, Wu X (2019) OFS-Density: A novel online streaming feature selection method. Pattern Recogn 86:48–61

Shengxing Bai is currently a M.S. student with the School of Computer Science, Minnan Normal University, Zhangzhou, China. His current research interests on machine learning and data mining for hierarchical classification.

Yaojin Lin received the B.S. degree from Northeast Agricultural University, Harbin, China; the M.S. degree from Xiamen University, Xiamen, China; and the Ph.D. degree from the Hefei University of Technology, Hefei, China. He is the Dean and Professor of School of Computer Science, Minnan Normal University, Zhangzhou, China. He has authored nearly 100 papers in journals and international conferences. His research interests include artificial intelligence, machine learning, and data mining.

Yan Lv received the B.S. degree from the School of Computer and Information, Anqing Normal University, Anqing, China, in 2019. She is currently working towards the M.S. degree from School of Computer Science, Minnan Normal University, Zhangzhou, China. Her research interests include artificial intelligence, machine learning, and data mining.

Jinkun Chen received the B.S. and M.S. degrees from Southwest Jiaotong University, Chengdu, China and the Ph.D. degree from Hebei Normal University, Shijiazhuang, China. He is the associate professor of School of Mathematics and Statistics, Minnan Normal University, Zhangzhou, China. He has authored nearly 30 papers in journals. His research interests include machine learning and data mining.

Chenxi Wang received the B.S. degree from Northeast Agricultural University, Harbin, China; the M.S. degree from Huazhong University of Science and Technology, Wuhan, China. She is the associate professor of School of Computer Science, Minnan Normal University, Zhangzhou, China. Her research interests include machine learning, and data mining.