Fuzzy Rough Set Based Feature Selection for Large-Scale Hierarchical Classification

Hong Zhao, Ping Wang, Qinghua Hu, *Senior Member, IEEE*, and Pengfei Zhu

***Abstract*—The classification of high-dimensional tasks remains a significant challenge for machine learning algorithms. Feature selection is considered to be an indispensable preprocessing step in high-dimensional data classification. In the era of big data, there may be hundreds of class labels, and the hierarchical structure of the classes is often available. This structure is helpful in feature selection and classifier training. However, most current techniques do not consider the hierarchical structure. In this paper, we design a feature selection strategy for hierarchical classification based on fuzzy rough sets. First, a fuzzy rough set model for hierarchical structures is developed to compute the lower and upper approximations of classes organized with a class hierarchy. This model is distinguished from existing techniques by the hierarchical class structure. A hierarchical feature selection problem is then defined based on the model. The new model is more practical than existing feature selection approaches, as many real-world tasks are naturally cast in terms of hierarchical classification. A feature selection algorithm based on sibling nodes is proposed, and this is shown to be more efficient and more versatile than flat feature selection. Compared with the flat feature selection algorithm, the computational load of the proposed algorithm is reduced from 98.0% to 6.5%, while the classification performance is improved on the *SAIAPR* dataset. The related experiments also demonstrate the effectiveness of the hierarchical algorithm.**

***Index Terms*—Feature selection, fuzzy rough sets, granular computing, hierarchical classification.**

I. Introduction

In THE era of big data, we can observe the following new trends in classification learning.

1) The number of samples continues to increase. We now have abundant datasets for model training.
2) The number of features used to describe the samples has increased from tens to hundreds of thousands, resulting in high-dimensional tasks.
3) The number of class labels is also becoming larger and larger. There are several hundred class labels in some classification tasks, and the class labels form a hierarchical structure, e.g., large-scale web categorization [1], image recognition [2], and gene classification [3].

The number of features is a crucial factor affecting the performance of a classifier. Feature selection aims to select a subset of features to decrease the time complexity, reduce the storage burden, and improve the generalization ability of classification [4]–[6]. This has a significant impact on both the running time and accuracy of the subsequent processing steps. Thus, it is highly desirable to develop effective algorithms that can select informative features from the raw data [7].

Various feature selection algorithms have been developed to select features for binary classification or multiclass tasks. However, there are complex classification structures in real-world applications, where the class labels to be predicted are hierarchically related [8]. Many real-world knowledge systems use a hierarchical scheme to organize their data, particularly ImageNet, Wikipedia [9], Internet web content, biological data [10], geographical data [11], and text data [12]. Hierarchical classification is an increasingly popular method that addresses the problem of classifying items into a hierarchy of classes [13]. In 2009, a workshop was organized for the PASCAL 2 large-scale hierarchical text classification challenge [14]. This workshop discussed the problems and challenges of large-scale hierarchical classification.

It has been reported that hierarchical methods produce better performance than flat classification techniques [15], [16]. Deng *et al.* [17] studied large-scale categorization using a category distance measure based on the WordNet hierarchy. They derived a hierarchy-aware cost function for classification and obtained more informative classification results. Moreover, a hierarchical structure makes it feasible to apply greedy algorithms for large-scale classification. Wei *et al.* [18] adapted a greedy algorithm for multilabel classification on tree-structured hierarchies using subtree optimization. The aforementioned methods are based on a predefined hierarchy. Some other studies [19] have focused on the construction of a hierarchical structure to deal with large-scale classification. For instance, a visual hierarchical structure has been constructed to organize large numbers of classes, and a learning algorithm was developed to train hierarchical classifiers [20]. These hierarchical approaches can

Manuscript received October 1, 2016; revised October 4, 2017, February 16, 2018, and June 3, 2018; accepted October 24, 2018. Date of publication; date of current version. This work was supported in part by the National Natural Science Foundation of China under Grant 61703196, Grant 61432011, Grant U1435212, Grant 61732011, and Grant 91746107; and in part by the Natural Science Foundation of Fujian Province under Grant 2018J01549. *(Corresponding author: Qinghua Hu.)*

H. Zhao is with the School of Computer Software, College of Intelligence and Computing, Tianjin University, Tianjin 300354, China, and also with the School of Computer Science, Minnan Normal University, Zhangzhou 363000, China (e-mail: hongzhaocn@163.com).

P. Wang is with the School of Mathematics and with the School of Computer Software, College of Intelligence and Computing, Tianjin University, Tianjin 300354, China (e-mail: wang_ping@tju.edu.cn).

Q. H. Hu and P. F. Zhu are with the College of Intelligence and Computing, Tianjin University, and also with the Tianjin Key Laboratory of Machine Learning, Tianjin 300354, China (e-mail: huqinghua@tju.edu.cn; zhupengfei@tju.edu.cn).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TFUZZ.2019.2892349

achieve competitive results in terms of both classification accuracy and computational efficiency.

A hierarchical class structure provides some external knowledge of the classes and is helpful not only for classifier training but also feature selection. However, few feature selection approaches for hierarchical class structures have been proposed. Hierarchical feature selection can split the problem into a set of smaller classification problems, each using its own feature set [21]. Freeman *et al.* [22] presented a method for joint feature selection and hierarchical classifier design using genetic algorithms, whereas Song *et al.* [23] proposed a feature selection method for hierarchical text classification. In these works, each child classification selects the best features considering the hierarchical class structure. They improve the accuracy of each classification task, but also reduce the feature dimension.

The theory of fuzzy rough sets is an effective mathematical tool for describing the inconsistency between attributes and decisions, and it is widely used in feature selection and attribute reduction [24]–[26]. In recent years, research on fuzzy rough sets can be categorized into two classes. First, many researchers have discussed the expansion of the fuzzy rough set model. In 2010, Chen *et al.* [27] introduced the concept of local reduction with fuzzy rough sets for a decision system. In 2011, Hu *et al.* [28] integrated kernel functions with fuzzy rough set models and proposed two types of kernelized fuzzy rough sets. In the second class, several different attribute reduction and feature selection methods using fuzzy rough sets have been proposed for different types of datasets [29]. For example, Zhao *et al.* [30] handled noisy datasets using fuzzy rough sets by proposing a robust method of dimension reduction. Another example is the application to decision systems with both symbolic and numerical conditional attributes by composing classical rough set and fuzzy rough set models [31]. In 2015, Chen *et al.* [32] studied the dynamic relation between granules, because data from different applications may evolve with time, that is, the objects, attributes, and attribute values may change dynamically.

The models and applications of fuzzy rough sets have been discussed in a comprehensive manner in recent decades [33]–[35]. These studies have focused almost exclusively on datasets with binary classification or multiclass tasks [36]–[38]. Few studies have considered datasets with high-dimensional classes, especially those with hierarchical class structures. In the era of big data, there may be hundreds of class labels, and the hierarchical structure of the classes is often available. This hierarchical data structure reflects the relationship among classes and is helpful for feature selection and classifier training. However, fuzzy rough set-based feature selection using the hierarchical structure has not been systematically studied.

In this paper, we propose a fuzzy rough set model for hierarchical classification and develop the corresponding feature selection algorithm. First, we embed the hierarchical structure into fuzzy rough sets and redefine the lower and upper approximations using an inclusive strategy and a sibling strategy for the hierarchical classification. The properties of the fuzzy rough sets for hierarchical classification are discussed. Second, we discuss the feature evaluation and feature searching strategies for hierarchical feature selection. In hierarchical classification, we can reduce the search domain for the nearest sample using the predefined class hierarchy. This analysis provides a new viewpoint to extend fuzzy rough sets in hierarchical applications. Finally, a feature selection algorithm is designed for the hierarchical feature selection problem. We use sibling nodes to compute the nearest samples, resulting in an efficient algorithm design. Moreover, some resampling strategies are also considered to accelerate the algorithm. Support vector machines (SVM), k-nearest neighbors (KNN), naive Bayes (NB) classifiers, and three hierarchical measures are used to test the performances of flat and hierarchical feature selection. We report the results of several experiments to demonstrate that the proposed algorithm outperforms the flat algorithms in terms of efficiency and accuracy.

This paper is organized as follows. In Section II, we present some preliminaries on fuzzy rough sets. Then, we introduce the model of fuzzy rough sets for hierarchical classification in Section III. We design a hierarchical feature selection algorithm in Section IV. In Section V, we introduce the evaluation measures for hierarchical feature selection algorithms. In Section VI, we present experimental results and analyze the effectiveness of the hierarchical feature selection algorithm. Finally, in Section VII, we conclude this paper.

II. Preliminaries

In this section, we review the notation for rough sets and fuzzy rough sets.

A. *Rough Sets*

Decision systems are fundamental in data mining and machine learning. Let $I = \langle U, C, D\rangle$ be a decision system, where U is a nonempty set of finite objects (the universe), C is a set of conditional attributes, and D is a set of decision attributes. For each $a \in C \cup D$, $I_a : U \rightarrow V_a$. Set V_a is the value set of attribute a, and I_a is an information function for each attribute a.

R is an equivalence relation on U calculated by

$$\mathrm{IND}(R) = \{(x, y) \in U \times U | \forall a \in R, a(x) = a(y)\} \quad (1)$$

where x and y are indiscernible by attributes from R when $(x, y) \in \mathrm{IND}(R)$. The equivalence relation partitions the universe into a family of disjoint subsets called equivalence classes. The equivalence class including x is denoted by $[x]_R$. We call $\mathrm{AS} = \langle U, R\rangle$ an approximation space. For any $X \subseteq U$, two subsets of objects, called lower and upper approximations of X in $\langle U, R\rangle$, are defined as [39]

$$\underline{R}X = \{[x]_R | [x]_R \subseteq X\} \quad (2)$$

$$\overline{R}X = \{[x]_R | [x]_R \cap X \neq \emptyset\}. \quad (3)$$

If $\underline{R}X \neq \overline{R}X$, X is a rough set in the approximation space; otherwise, we say that X is definable.

The rough set theory described above can deal with datasets that contain discrete values [39], [40]. However, most datasets contain numerical attributes. The model of fuzzy rough sets is an extended model to address this problem [41]. The theory of fuzzy rough sets offers an effective way to model the vagueness and imprecision presented in numerical data [28].

B. Fuzzy Rough Sets

Let U be a nonempty and finite set of objects, and R be a fuzzy binary relation on U. We call FAS $= \langle U, R\rangle$ a fuzzy approximation space, where R is a fuzzy equivalence relation. $\forall x, y, z \in U$, we have the following:

1) reflexivity: $R(x, x) = 1$;
2) symmetry: $R(x, y) = R(y, x)$; and
3) min–max transitivity: $\min_y (R(x, y), R(y, z)) \le R(x, z)$.

More generally, we say that R is a fuzzy T-equivalence relation if for $\forall x, y, z \in U$, R satisfies reflexivity, symmetry, and T-transitivity, that is, $T(R(x, y), R(y, z)) \le R(x, z)$.

Given fuzzy approximation space FAS $= \langle U, R\rangle$ and fuzzy subset $X \subseteq U$, fuzzy rough sets can be summarized as the following four operators [42]:

$$\begin{aligned}\underline{R_S}X(x) &= \inf_{y\in U} S(N(R(x, y)), X(y))\\ \overline{R_T}X(x) &= \sup_{y\in U} T(R(x, y), X(y))\\ \underline{R_\vartheta}X(x) &= \inf_{y\in U} \vartheta(R(x, y), X(y))\\ \overline{R_\sigma}X(x) &= \sup_{y\in U} \sigma(N(R(x, y)), X(y)),\end{aligned} \tag{4}$$

where T, S, ϑ, and σ denote the fuzzy triangular norm (T-norm), fuzzy triangular conorm (T-conorm), T-residuated implication, and its dual, respectively, and N is a negator. The standard negator is defined as $N(x) = 1 - x$. Several fuzzy operators and their properties were introduced in [43]. Some typical fuzzy operators are listed as follows: $S_M(a, b) = \max(a, b)$,

$$T_M(a, b) = \min(a, b), \vartheta_M(a, b) = \begin{cases}1, & a \le b\\ b, & a > b\end{cases},$$

$$\sigma_M(a, b) = \begin{cases}0, & a \ge b\\ b, & a < b\end{cases}.$$

Let $I = \langle U, C, D\rangle$ be a decision system, where U is a universe of objects, C is a nonempty set of conditional attributes with numerical values, and D is the decision attribute that divides the samples into subset $\{d_1, d_2, \ldots, d_l\}$. For all $x \in U$ and if R is a fuzzy similarity relation, then we have

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

Then, the fuzzy rough approximations are computed as

$$\begin{aligned}\underline{R_S}d_i(x) &= \inf_{y\notin d_i} (1 - R(x, y))\\ \overline{R_T}d_i(x) &= \sup_{y\in d_i} R(x, y)\\ \underline{R_\vartheta}d_i(x) &= \inf_{y\notin d_i} (\sqrt{1 - R^2(x, y)})\\ \overline{R_\sigma}d_i(x) &= \sup_{y\in d_i}(1 - \sqrt{1 - R^2(x, y)}).\end{aligned} \tag{6}$$

The lower and upper approximations use an equivalence relation to granulate the universe and generate Boolean elemental granules [28] in rough sets. A fuzzy rough set [41] is defined by two fuzzy sets, fuzzy lower and upper approximations defined in (6) that are obtained by extending the corresponding crisp rough set notions defined previously in (2) and (3) [24].

TABLE I
DESCRIPTION OF SYMBOLS USED THROUGHOUT THIS PAPER

Symbol	Meaning
$pos(x)$	The set of samples with the same class of x
$neg(x)$	The set of negative samples of x
$anc(d_u)$	The set of ancestor categories of class d_u
$des(d_u)$	The set of descendant categories of class d_u
$sib(d_u)$	The set of sibling categories of class d_u
$LCA(d_u, d_v)$	Lowest common ancestor of classes d_u and d_v
$\hat{D}$, D	Sets of predicted and true classes
$\hat{D}_{aug}$, D_{aug}	Augmented Sets of predicted and true classes
B	The selected feature subset

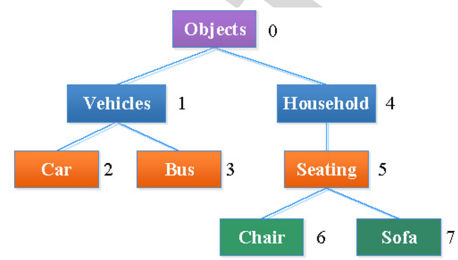

Fig. 1. Example of a tree-based hierarchical class structure.

III. FUZZY ROUGH SETS FOR HIERARCHICAL CLASSIFICATION

A number of learning algorithms have been developed based on fuzzy rough sets [44], [45]. Large-scale data are not only a rich source of information but also produce complex class structures, such as hierarchies. It is interesting and challenging to exploit such structures in modeling.

A. Hierarchical Classification

In this study, we are interested in a tree-based hierarchical class structure. In all cases, the hierarchy imposes a parent-child relationship among the classes, which implies that an instance belonging to a specific class also belongs to all its ancestor classes. Table I describes the most frequent symbols used throughout this paper.

A taxonomy is thus typically defined as a pair $(D, \prec)$, where D is the set of all classes and "$\prec$" represents the "is-a" relationship, which is the *subclass-of* relationship with the following properties [13]:

1) Asymmetry: if $d_i \prec d_j$ then $d_j \nprec d_i$ for every $d_i, d_j \in D$.
2) Antireflexivity: $d_i \nprec d_i$ for every $d_i \in D$.
3) Transitivity: if $d_i \prec d_j$ and $d_j \prec d_k$, then $d_i \prec d_k$ for every $d_i, d_j, d_k \in D$.

An example of a tree-based hierarchical class structure is shown in Fig. 1. The root node *Objects* is not the real class of each sample.

Example 1: In Fig. 1, we can obtain asymmetry and transitivity of a tree-based hierarchical class structure as follows:

1) Asymmetry: *Chair* is a *Seating*, but *Seating* is not a *Chair*.
2) Transitivity: *Chair* is a *Seating* and *Seating* is a *Household*. We can know that *Chair* is a *Household*.

TABLE II
THREE STRATEGIES TO DEFINE POSITIVE AND NEGATIVE SAMPLES

Method	Positive samples	Negative samples
Exclusive strategy [46]	A	Not A
Inclusive strategy [46]	$A + des(A)$	Not $[A + des(A)]$
Sibling strategy [47]	A	$sib(A)$

TABLE III
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}
A	0	0.12	0.19	0.37	0.45	0.49	0.31	0.62	0.35	0.81	0.89	0.92
D	d_1	d_1	d_2	d_2	d_3	d_3	d_4	d_4	d_5	d_5	d_6	d_6

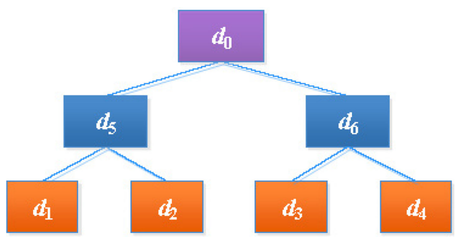

Fig. 2. Tree structure of example data.

B. Flat Classification and Hierarchical Classification

In fuzzy rough sets, the fuzzy lower approximation depends on the nearest sample y from different classes of x. For convenience, we call samples with the same class as x positive samples and call those from different classes as x negative samples. The search scope of negative samples plays a crucial role in defining the lower approximation of fuzzy rough sets. There are several ways to define the positive samples and negative samples for training binary classifiers. We can use these strategies to compute the fuzzy lower approximation and fuzzy upper approximation. Table II gives three strategies to define positive and negative samples, and they are exclusive, inclusive, and sibling strategies.

In flat classification, we do not consider the relationship among different classes. Therefore, the negative samples are not A if the positive sample is A. We call this an exclusive strategy [46], as described in the first row of Table II. Thus, only samples explicitly labeled with A as their most specific class are selected as positive samples, and everything else is considered as negative samples.

Given a classification task, we have 12 samples listed in Table III. Each sample is characterized by a condition attribute A. d_1, d_2, d_3, d_4, d_5, and d_6 are six classes.

The positive class is the class of sample x_i, and the negative class is the class different from x_i. Compared with hierarchical classification, the flat classification approach is the simplest one that does not consider the hierarchy of the class.

Hierarchical problems are particularly prevalent in large-scale datasets. We are interested in approaches that cope with a predefined class hierarchy. Fig. 2 shows the tree structure of D_{tree}, where D_{tree} is a tree-based hierarchical class with values d_1, d_2, d_3, d_4, d_5, and d_6 in Table III.

According to the tree-based hierarchical class structure, there is an "is-a" relationship between the parent and child nodes to describe the parent-child relationship. The descendant categories of x are positive samples; therefore, it is not necessary to consider these samples when the lower approximation is computed. We call this an inclusive strategy [46], as described in the second row of Table II, where $\text{des}(A)$ denotes descendant categories of class A.

Based on the tree-based hierarchical class structure, sibling nodes with the same parent have a high fuzzy similarity degree. Therefore, it may be effective to search for negative samples within only the sibling nodes called the sibling strategy. The sibling strategy [47] is listed in the third row of Table II, where $\text{sib}(A)$ denotes sibling categories of class A. We can use this hierarchical information to decrease the search scope of the negative samples and reduce the algorithm's complexity.

We use the following example to compare the exclusive strategy with flat classes and the inclusive and sibling strategies with hierarchical classes.

Example 2: Continuing with Example 1, we give an intuitive interpretation of different positive and negative samples in Fig. 1.

We have the following results according to different strategies.

1) Exclusive strategy: The positive sample is *Chair* if we let A be *Chair*. That is, $\text{pos}(A) = \{5\}$. The negative samples are not *Chair*, that is, $\text{neg}(A) = \{1, 2, 3, 4, 5, 7\}$.
2) Inclusive strategy: The positive samples are *Seating*, *Chair*, and *Sofa*, that is, $\text{pos}(A) = \{5, 6, 7\}$. The negative samples are $\text{neg}(A) = \{1, 2, 3, 4\}$.
3) Sibling strategy: The positive sample is *Chair* if we let A be *Chair*. The negative samples are $\text{sib}(A) = \{7\}$.

In fuzzy rough sets, the fuzzy lower approximation of a sample is computed from the nearest sample to x_i in classes different from x_i, which means the nearest negative sample. In this tree hierarchical structure, the nearest sample is in the descendant, ancestor, and sibling categories. From Table II, the descendant categories are usually positive samples. Therefore, we use the sibling strategy to select negative samples. For example, the nearest negative sample to *Chair* is *Sofa*, which is consistent with an intuitive interpretation.

C. Fuzzy Rough Sets for Hierarchical Classification

Classification is one of the most important problems in data mining, machine learning, and statistical pattern recognition. Related research has focused on flat classification problems, which are standard binary or multiclass classification problems [48]. The lower approximation of classical fuzzy rough sets is the minimum distance of a sample from the different classes, and the upper approximation is the maximum distance in the same class [49]. Generally, we focus on traditional datasets with nonhierarchical classes. Therefore, the same classes of x exclude every instance except for those that have exactly the same class as x (and not those that are more general or more specific).

Nowadays, in some important applications, there are several hierarchical classification problems. The hierarchy defines an inheritance (IS-A) relationship between the class nodes, where each class is a special case of its parent class [46]. Any class is a special case of each ancestor class, where an ancestor is any class along the path from the class to the root of the hierarchy. Now, we consider the fuzzy lower approximation of classification for hierarchical classes.

The tree-based hierarchical class structure can be formulated as $\langle U, C, D_{\text{tree}}\rangle$, where U is a universal set of objects, C is a nonempty set of conditional attributes, and D_{tree} is the decision attribute that divides the samples into subsets $\{d_1, d_2, \ldots, d_l\}$. l is the number of classes. D_{tree} satisfies a pair $(D_{\text{tree}}, \prec)$, which is introduced in Section III-A. R is a fuzzy similarity relation on U generated with features $B \subseteq C$.

There are several methods for defining the set of positive (same) and negative (different) classes in Table II. We can use these strategies to define the approximation of fuzzy rough sets for hierarchical classification. Traditional classification deals with nonhierarchical classes, which is flat classification. We call this the exclusive strategy. The lower and upper approximations are defined in (6).

When inclusive strategy is considered, for all $x \in U$, we have

$$d_i(x) = \begin{cases} 0, & x \notin \{\text{des}(d_i) \cup d_i\} \\ 1, & x \in \{\text{des}(d_i) \cup d_i\} \end{cases}. \tag{7}$$

The fuzzy rough approximations are defined as

$$\begin{aligned} \underline{R_S}_{\text{inclusive}} d_i(x) &= \inf_{y \notin \{\text{des}(d_i) \cup d_i\}} (1 - R(x,y)) \\ \overline{R_T}_{\text{inclusive}} d_i(x) &= \sup_{y \in \{\text{des}(d_i) \cup d_i\}} R(x,y) \\ \underline{R_\vartheta}_{\text{inclusive}} d_i(x) &= \inf_{y \notin \{\text{des}(d_i) \cup d_i\}} (\sqrt{1 - R^2(x,y)}) \\ \overline{R_\sigma}_{\text{inclusive}} d_i(x) &= \sup_{y \in \{\text{des}(d_i) \cup d_i\}} (1 - \sqrt{1 - R^2(x,y)}). \end{aligned} \tag{8}$$

When sibling strategy is considered, for all $x \in U$, we have

$$d_i(x) = \begin{cases} 0, & x \in \{\text{sib}(d_i)\} \\ 1, & x \in \{d_i\} \end{cases}. \tag{9}$$

The fuzzy rough approximations are defined as

$$\begin{aligned} \underline{R_S}_{\text{sibling}} d_i(x) &= \inf_{y \in \{\text{sib}(d_i)\}} (1 - R(x,y)) \\ \overline{R_T}_{\text{sibling}} d_i(x) &= \sup_{y \in \{d_i\}} R(x,y) \\ \underline{R_\vartheta}_{\text{sibling}} d_i(x) &= \inf_{y \in \{\text{sib}(d_i)\}} (\sqrt{1 - R^2(x,y)}) \\ \overline{R_\sigma}_{\text{sibling}} d_i(x) &= \sup_{y \in \{d_i\}} (1 - \sqrt{1 - R^2(x,y)}). \end{aligned} \tag{10}$$

Several properties of the fuzzy rough sets for hierarchical classification are as follows. Compared with the exclusive strategy, we have the following propositions when we consider the sibling strategy.

Proposition 1: Given $\langle U, C, D_{\text{tree}}\rangle$, R is a fuzzy similarity relation induced by $B \subseteq C$. Let d_i be a class of samples labeled with i, for $x \in U$

$$\begin{aligned} \underline{R_S}_{\text{sibling}} d_i(x) &\geq \underline{R_S} d_i(x) \\ \underline{R_\vartheta}_{\text{sibling}} d_i(x) &\geq \underline{R_\vartheta} d_i(x). \end{aligned} \tag{11}$$

Proof: Suppose that y_{si} is the sample with class $y_{si} \in \text{sib}(d_i)$, such that $\underline{R_S}_{\text{sibling}} d_i(x) = 1 - R(x, y_{si})$. Suppose that y_{ex} is the sample with class $y_{ex} \in D_{\text{tree}} \backslash d_i$, such that $\underline{R_S} d_i(x) = 1 - R(x, y_{ex})$. Since $\text{sib}(d_i) \subseteq D_{\text{tree}} \backslash d_i$, we have $R(x, y_{si}) \leq R(x, y_{ex})$. Therefore, $\underline{R_S}_{\text{sibling}} d_i(x) \geq \underline{R_S} d_i(x)$. Analogically, we can also obtain $\underline{R_\vartheta}_{\text{sibling}} d_i(x) \geq \underline{R_\vartheta} d_i(x)$. ■

Proposition 2: Given $\langle U, C, D_{\text{tree}}\rangle$, R is a fuzzy similarity relation induced by $B \subseteq C$. If d_i is a class of samples labeled with i and $x \in U$, we have

$$\begin{aligned} \overline{R_T}_{\text{sibling}} d_i(x) &= \overline{R_T} d_i(x) \\ \overline{R_\sigma}_{\text{sibling}} d_i(x) &= \overline{R_\sigma} d_i(x). \end{aligned} \tag{12}$$

Proof: Since $\overline{R_T} d_i(x) = \sup_{y \in d_i} R(x,y)$ and $\overline{R_T}_{\text{sibling}} d_i(x) = \sup_{y \in d_i} R(x,y)$. Therefore, $\overline{R_T}_{\text{sibling}} d_i(x) = \overline{R_T} d_i(x)$. Analogically, we can also obtain $\overline{R_\sigma}_{\text{sibling}} d_i(x) = \overline{R_\sigma} d_i(x)$. ■

The sibling strategy and inclusive strategy have different positive and negative sample definitions. We have the following proposition when we consider the sibling strategy and inclusive strategy, respectively.

Proposition 3: Given $\langle U, C, D_{\text{tree}}\rangle$, R is a fuzzy similarity relation induced by $B \subseteq C$. Let d_i be a class of samples labeled with i, for $x \in U$

$$\begin{aligned} \underline{R_S}_{\text{sibling}} d_i(x) &\geq \underline{R_S}_{\text{inclusive}} d_i(x) \\ \overline{R_T}_{\text{sibling}} d_i(x) &\leq \overline{R_T}_{\text{inclusive}} d_i(x) \\ \underline{R_\vartheta}_{\text{sibling}} d_i(x) &\geq \underline{R_\vartheta}_{\text{inclusive}} d_i(x) \\ \overline{R_\sigma}_{\text{sibling}} d_i(x) &\leq \overline{R_\sigma}_{\text{inclusive}} d_i(x). \end{aligned} \tag{13}$$

Proof: Suppose that y_{si} is the sample with class from $\text{sib}(d_i)$, such that $\underline{R_S}_{\text{sibling}} d_i(x) = 1 - R(x, y_{si})$. Suppose that y_{in} is the sample with class from $D_{\text{tree}} \backslash \{\text{des}(d_i) \cup d_i\}$, such that $\underline{R_S}_{\text{inclusive}} d_i(x) = 1 - R(x, y_{\text{in}})$. Since $\text{sib}(d_i) \subseteq D_{\text{tree}} \backslash \{\text{des}(d_i) \cup d_i\}$, we have $R(x, y_{si}) \leq R(x, y_{\text{in}})$. Thus, $\underline{R_S}_{\text{sibling}} d_i(x) \geq \underline{R_S}_{\text{inclusive}} d_i(x)$. Analogically, we can also obtain $\underline{R_\vartheta}_{\text{sibling}} d_i(x) \geq \underline{R_\vartheta}_{\text{inclusive}} d_i(x)$.

Suppose that y_{si} is the sample with class from d_i, such that $\overline{R_T}_{\text{sibling}} d_i(x) = R(x, y_{si})$. Suppose that y_{in} is the sample with class from $\{\text{des}(d_i) \cup d_i\}$, such that $\overline{R_T}_{\text{inclusive}} d_i(x) = R(x, y_{\text{in}})$. Since $d_i \subseteq \{\text{des}(d_i) \cup d_i\}$, we have $R(x, y_{si}) \leq R(x, y_{\text{in}})$. Thus, $\overline{R_T}_{\text{sibling}} d_i(x) \leq \overline{R_T}_{\text{inclusive}} d_i(x)$. Analogically, we can also obtain $\overline{R_\sigma}_{\text{sibling}} d_i(x) \leq \overline{R_\sigma}_{\text{inclusive}} d_i(x)$. ■

According to Propositions 2 and 3, we can obtain

$$\begin{aligned} \overline{R_T} d_i(x) &\leq \overline{R_T}_{\text{inclusive}} d_i(x) \\ \overline{R_\sigma} d_i(x) &\leq \overline{R_\sigma}_{\text{inclusive}} d_i(x). \end{aligned} \tag{14}$$

Proposition 4: Given $\langle U, C, D_{\text{tree}}\rangle$, R is a fuzzy similarity relation induced by $B \subseteq C$. Let d_i be a class of samples labeled with i, for $x \in U$

$$\begin{aligned} \underline{R_S}_{\text{inclusive}} d_i(x) &\geq \underline{R_S} d_i(x) \\ \underline{R_\vartheta}_{\text{inclusive}} d_i(x) &\geq \underline{R_\vartheta} d_i(x). \end{aligned} \tag{15}$$

Proof: Suppose that y_{in} is the sample with class from $D_{\text{tree}} \backslash \{\text{des}(d_i) \cup d_i\}$, such that $\underline{R_S}_{\text{inclusive}} d_i(x) = 1 - R(x, y_{\text{in}})$. Suppose that y_{ex} is the sample with class $y_{ex} \in D_{\text{tree}} \backslash d_i$, such that $\underline{R_S} d_i(x) = 1 - R(x, y_{ex})$. Since $D_{\text{tree}} \backslash \{\text{des}(d_i) \cup d_i\} \subseteq D_{\text{tree}} \backslash d_i$, we have $R(x, y_{\text{in}}) \leq R(x, y_{ex})$. Thus, $\underline{R_S}_{\text{inclusive}} d_i(x) \geq \underline{R_S} d_i(x)$. Analogically, we can also obtain $\underline{R_\vartheta}_{\text{inclusive}} d_i(x) \geq \underline{R_\vartheta} d_i(x)$. ■

Proposition 5: Given $\langle U, C, D_{\text{tree}}\rangle$, R_1 and R_2 are two fuzzy similarity relations induced by B_1 and B_2, respectively, and $R_1 \subseteq R_2$. Let d_i be a class of samples labeled with i, for $x \in U$

$$\begin{aligned} \underline{R_{1S}}_{\text{sibling}} d_i(x) &\geq \underline{R_{2S}}_{\text{sibling}} d_i(x) \\ \overline{R_{1T}}_{\text{sibling}} d_i(x) &\leq \overline{R_{2T}}_{\text{sibling}} d_i(x) \\ \underline{R_{1\vartheta}}_{\text{sibling}} d_i(x) &\geq \underline{R_{2\vartheta}}_{\text{sibling}} d_i(x) \\ \overline{R_{1\sigma}}_{\text{sibling}} d_i(x) &\leq \overline{R_{2\sigma}}_{\text{sibling}} d_i(x). \end{aligned} \tag{16}$$

Proof: The proof is straightforward. ■

We give the following example to compare the computation among three strategies on the intermediate nodes. For simplification, we use the model defined with T-norm and T-conorm operators. For comparing with the flat algorithm in [28], we use the same function, the Gaussian function, to compute fuzzy similarity relations R, and the parameter σ is set to 0.2

$$R(x, y) = \exp\left(-\frac{||x - y||^2}{\sigma}\right), \tag{17}$$

where $||x - y||$ is the distance between x and y.

Example 3: We give an example of computing fuzzy lower approximation based on different strategies with the data listed in Table III. We select x_3 with class d_2 to compute the lower approximation. For exclusive strategy

$$\begin{aligned} \underline{R_S} d_2(x_3) &= \inf_{y \notin \{d_2\}} (1 - R(x_3, y)) \\ &= \inf_{y \in \{d_1, d_3, d_4, d_5, d_6\}} (1 - R(x_3, y)) \\ &= 1 - \exp\left(-\frac{||x_3 - x_2||^2}{0.2}\right) = 0.0242. \end{aligned} \tag{18}$$

As to the inclusive strategy

$$\begin{aligned} \underline{R_S}_{\text{inclusive}} d_2(x_3) &= \inf_{y \notin \{\text{des}(d_2) \cup d_2\}} (1 - R(x_3, y)) \\ &= \inf_{y \notin \{d_2, d_1, d_3\}} (1 - R(x_3, y)) \\ &= \inf_{y \in \{d_4, d_5, d_6\}} (1 - R(x_3, y)) \\ &= 1 - \exp\left(-\frac{||x_3 - x_7||^2}{0.2}\right) = 0.0695. \end{aligned} \tag{19}$$

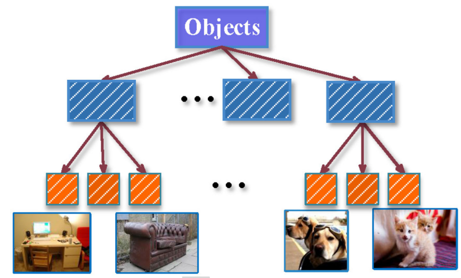

Fig. 3. Example of sibling relationship.

As to the sibling strategy

$$\begin{aligned} \underline{R_S}_{\text{sibling}} d_2(x_3) &= \inf_{y \in \{\text{sib}(d_2)\}} (1 - R(x_3, y)) \\ &= \inf_{y \in \{d_5\}} (1 - R(x_1, y)) \\ &= 1 - \exp\left(-\frac{||x_3 - x_9||^2}{0.2}\right) = 0.1201. \end{aligned} \tag{20}$$

We have $\underline{R_S}_{\text{sibling}} d_i(x) \geq \underline{R_S}_{\text{inclusive}} d_i(x) \geq \underline{R_S} d_i(x)$.

In this example, we should compute the samples $y \in \{d_1, d_3, d_4, d_5, d_6\}$ when we use the exclusive strategy and the samples $y \in \{d_4, d_5, d_6\}$ when we consider the inclusive strategy. We need to compute the samples $y \in \{d_5\}$ for the sibling strategy. This can significantly reduce the computation time, especially for large datasets.

IV. Hierarchical Feature Selection

Feature selection is an indispensable preprocessing step of high-dimensional data classification [50], and can help to identify redundant or correlated features [51]. Fuzzy rough set theory is an effective method for selecting feature subsets using the dependencies between the decision and condition attributes. These dependencies can identify effective features for classification. The two main steps in any feature selection algorithm are feature evaluation and the search strategy.

The inclusive strategy and sibling strategy discussed above have their own advantages. The inclusive strategy reduces the computational complexity when we consider the intermediate nodes. In this paper, we consider the leaf nodes to be real classes and use the sibling strategy to select the feature subset. The minimum distance of a sample from different classes is a critical factor in feature selection. Fig. 3 shows the hierarchical structure of classes. In this hierarchical structure, there are common characteristics among the sibling classes because they share a parent node. Thus, we select the nearest negative samples from the sibling nodes, which is consistent with an intuitive interpretation.

Definition 1: Given a hierarchical classification problem $\langle U, C, D_{\text{tree}}\rangle$, R is the $T-$ equivalence relation on U computed with the distance function $R(x, y)$ in the feature space $B \subseteq C$. $D_{\text{tree}} = \{d_0, d_1, d_2, \ldots, d_l\}$, where d_0 is the root of the tree and it is not the real class. U is divided into $\{d_1, d_2, \ldots, d_l\}$ with the decision attribute, where l is the number of classes. The fuzzy positive region of D_{tree} in term of B is defined as

$$\text{POS}^S_{B_{\text{sibling}}}(D_{\text{tree}}) = \cup_{i=1}^{l} \underline{R_S}_{\text{sibling}} d_i. \tag{21}$$

Definition 2: Given a classification problem $\langle U, C, D_{\text{tree}}\rangle$, R is the T-equivalence relation on U computed with the distance function $R(x, y)$ in the feature space $B \subseteq C$, and U is divided into $\{d_1, d_2, \ldots, d_l\}$ with the decision attribute, where l is the number of classes. The quality of the classification approximation is defined as

$$\gamma^S_{B_{\text{sibling}}}(D_{\text{tree}}) = \frac{|\cup_{i=1}^{l} \underline{R_S}_{\text{sibling}} d_i|}{|U|}. \tag{22}$$

As $\underline{R_S}_{\text{sibling}} d_i(x) = \inf_{y \in \text{sib}(d_i)} (1 - R(x, y))$, we get that

$$|\cup_{i=1}^{l} \underline{R_S}_{\text{sibling}} d_i| = \sum_{j=1}^{|U|} \sum_{i=1}^{l} \underline{R_S}_{\text{sibling}} d_i(x_j). \tag{23}$$

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

$$\begin{aligned} \sum_{j=1}^{|U|} \sum_{i=1}^{l} \underline{R_S}_{\text{sibling}} d_i(x_j) &= \sum_{j=1}^{|U|} \underline{R_S}_{\text{sibling}} d(x_j) \\ &= \sum_{j=1}^{|U|} \inf_{x_j \in d, y \in \text{sib}(d)} (1 - R(x_j, y)) \end{aligned} \tag{24}$$

where d is the class label of x_j.

The coefficients of classification quality reflect the approximation ability of the approximation space or the ability of the granulated space induced by feature subset B to characterize the decision [28]. These coefficients can evaluate the condition attribute with degree $\gamma^S_B(D_{\text{tree}})$, and reflect the dependence between the decision and condition attributes. The monotonicity approximations are given by Theorem 1, which applies to both sibling strategy and inclusive strategy.

Theorem 1: Given a hierarchical classification problem $\langle U, C, D_{\text{tree}}\rangle$, R_1 and R_2 are two fuzzy similarity relations induced by B_1 and B_2, respectively, and $R_1 \subseteq R_2$, we have

$$\text{POS}^S_{B_1}(D_{\text{tree}}) \subseteq \text{POS}^S_{B_2}(D_{\text{tree}}). \tag{25}$$

Proof: Let d_i be a class of samples labeled with i, for $x \in U$, we have $\underline{R_{1S}} d_i(x) \geq \underline{R_{2S}} d_i(x)$ since $R_1 \subseteq R_2$. We can derive that $\text{POS}^S_{B_1}(D_{\text{tree}}) \subseteq \text{POS}^S_{B_2}(D_{\text{tree}})$ since $\text{POS}^S_B(D_{\text{tree}}) = \cup_{i=1}^{l} \underline{R_S} d_i$. ■

According to Definition 2 and Theorem 1, we have

$$\gamma^S_{B_1}(D_{\text{tree}}) \leq \gamma^S_{B_2}(D_{\text{tree}}). \tag{26}$$

In a feature selection algorithm, feature evaluation quantifies how good the feature subset is, and search strategies are used to identify the optimal feature subset. First, we evaluate each feature according to its dependence coefficient and rank them in terms of feature quality. Then, we select the best feature and delete redundant features to further reduce the computation time.

A fuzzy rough sets based feature selection algorithm for hierarchical classification (FFS-HC) is illustrated in Algorithm 1.

Algorithm 1 A fuzzy Rough Sets Based Feature Selection Algorithm for Hierarchical Classification (FFS-HC).

Input: $\langle U, C, D_{\text{tree}}\rangle$
Output: A feature subset B
1: $B = \emptyset$; $CD = \emptyset$;
//Addition
2: $CA = C$;
3: **while** $(\gamma^S_C(D_{\text{tree}}) - \gamma^S_B(D_{\text{tree}}) < \delta)$) **do**
4: **for** each $a \in CA$ **do**
5: Compute $\gamma^S_{a \cup B}(D_{\text{tree}})$ according to SSFE;
6: **end for**//Delete the redundant features
7: **if** $B == \emptyset$ **then**
8: **for** each $a \in CA$ **do**
9: Select feature a_{del} is smaller than the average $\gamma^S_a(D_{\text{tree}})$;
10: $CD = CD \cup a_{\text{del}}$;
11: **end for**
12: $CA = CA - CD$;
13: **end if**
14: Select a' with the maximal $\gamma^S_{a' \cup B}(D_{\text{tree}})$;
15: $B = B \cup \{a'\}$;
16: $CA = CA - \{a'\}$;
17: **end while**
18: return B;

The sibling strategy based feature evaluation (SSFE) of FFS-HC is provided in line 5 in Algorithm 1, and the specific implementation of SSFE is illustrated in Algorithm 2. D_{tree} is a tree-based hierarchical structure of the classes, and it is a global variable that should be explicitly initialized.

We use a sibling-based relief algorithm to find the optimal feature subset for comparing the flat feature selection with the proposed hierarchical feature selection. The complexity of the relief algorithm will become unacceptable when the number of records in the dataset increases to a large scale. In general, the size of the search space for the feature selection algorithm is $2^{|C|}$. Algorithm 1 deals with this issue effectively by deleting redundant features to reduce the search space.

We consider two strategies in Algorithms 1 and 2 for reducing the search space. First, we can reduce the computing space by using the sibling strategy, which is listed from lines 3–9 in Algorithm 2. This strategy can reduce the computation time significantly. Second, we compute the dependence of each feature only once. We then delete the redundant features in the first round, as described from lines 7–13 in Algorithm 1.

V. Evaluation Measures

The proposed method is to deal with hierarchical classification, which is different from flat classification. Accordingly, the evaluation measures for the FFS-HC algorithm should be different. Measures were introduced to evaluate hierarchical classification in [13].

Example 4: Fig. 1 shows the hierarchical classification subtree of visual object classes (VOC) classification. We assume that the true class for a test instance is *Car* and that two classification systems output *Bus* (Case 1) and *Sofa* (Case 2) as the

Algorithm 2 Sibling Strategy Based Feature Evaluation (SSFE).

```
Input: ⟨U, C, D_tree⟩, r = 0, and B
Output: r
for i = 1 to |U| do
    Compute decision d_i of sample x_i;
    Select samples X_sib with class sib(d_i);
    if length(X_sib) == 0 then
        Random select samples out of d_i as X_sib;
    end if
    for each y ∈ X_sib do
        Compute 1 − R(x_i, y);
    end for
    Select y' such that R_S_sibling d_i(x_i) = 1 − R(x_i, y');
    r = r + 1 − R(x_i, y');
end for
r = r/|U|;
return r;
```

predicted classes. These two errors are the same using flat evaluation measures, and these two systems are punished equally. However, Case 2 is more severe because it makes a prediction in a different and unrelated subtree. Thus, the punishment for Case 2 should be larger than that for Case 1.

In some cases, a sample can be classified into more than one class in the hierarchy. The pair-based measure and set-based measure are two main hierarchical evaluation measures.

A. Pair-Based Measures

As stated above, different classification errors result in different levels of penalty. In our model, this penalty is defined by the tree distance, which is called the *tree-induced error* (TIE) in [52]. The TIE is computed by predicting label d_v when the correct label is d_u

$$\mathrm{TIE}(d_u, d_v) = |E_H(d_u, d_v)| \tag{27}$$

where $E_H(d_u, d_v)$ is the set of edges along the path from d_u to d_v in the hierarchy, and $|\cdot|$ denotes the count of elements. That is, $\mathrm{TIE}(d_u, d_v)$ is defined to be the number of edges along the path from d_u to d_v in the tree of D. $\mathrm{TIE}(d_u, d_u) = 0$, $\mathrm{TIE}(d_u, d_v) = \mathrm{TIE}(d_v, d_u)$, and the triangle inequality always holds with equality.

Example 5: Continuing with Example 4, the true class for a test instance is *Car*. The predicted class with *Sofa* is punished $\mathrm{TIE}(2, 7) = 5$, which is larger than the punishment $\mathrm{TIE}(2, 3) = 2$ for the predicted class with *Bus*.

B. Set-Based Measures

Pair-based measures consider only a pair of predicted and true classes. Unlike pair-based measures, set-based measures take into account the entire sets of predicted and true classes, including their ancestors or descendants.

Set-based measures have the following two distinct phases:

1) the augmentation of D and $\hat{D}$ with information on the hierarchy; and
2) the calculation of a cost measure based on the augmented sets.

The augmentation of D and $\hat{D}$ is a crucial step that attempts to capture the hierarchical relations of the classes. There are different measures based on different augmented approaches for the sets of predicted and true classes. We select the measure that the sets are augmented with the ancestors of the true and predicted classes [3], [53] as follows:

$$\begin{aligned} D_{\text{aug}} &= D \cup \mathrm{anc}(D) \\ \hat{D}_{\text{aug}} &= \hat{D} \cup \mathrm{anc}(\hat{D}). \end{aligned} \tag{28}$$

Hierarchical precision and recall are defined as follows:

$$\begin{aligned} P_H &= \frac{|\hat{D}_{\text{aug}} \cap D_{\text{aug}}|}{|\hat{D}_{\text{aug}}|} \\ R_H &= \frac{|\hat{D}_{\text{aug}} \cap D_{\text{aug}}|}{|D_{\text{aug}}|} \end{aligned} \tag{29}$$

where $|\cdot|$ denotes the count of elements. The F_1-measure is defined as follows:

$$F_H = \frac{2 \cdot P_H \cdot R_H}{P_H + R_H}. \tag{30}$$

Continuing with Example 4, we can compute the hierarchical precision, recall, and F_1-measure of two cases.

Case 1: In Fig. 1, let $D = \{2\}$ and $\hat{D} = \{3\}$, which means that the true class of a test instance is *Car* and the predicted class is *Bus:* $D_{\text{aug}} = \{2, 1, 0\}$ and $\hat{D}_{\text{aug}} = \{3, 1, 0\}$; $P_H = 0.67$, $R_H = 0.67$, and $F_H = 0.67$.

Case 2: In Fig. 1, let $D = \{2\}$ and $\hat{D} = \{7\}$, which means that the true class for a test instance is *Car* and the predicted class is *Sofa:* $D_{\text{aug}} = \{2, 1, 0\}$ and $\hat{D}_{\text{aug}} = \{7, 5, 4, 0\}$; $P_H = 0.25$, $R_H = 0.33$, and $F_H = 0.29$.

C. Lowest Common Ancestor (LCA) F_1 Measure

The set-based measure adds all the ancestors, and it has over penalizing errors that occur to nodes with many ancestors. Kosmopoulos *et al.* [13] proposed LCA measures to deal with this problem. The concept of LCA was defined in graph theory [54]. The LCA of two nodes d_u and d_v of a tree D, $\mathrm{LCA}(d_u, d_v)$, is defined as the lowest node in D (furthest from the root), which is an ancestor of both d_u and d_v [13]. For example, in Fig. 1, $\mathrm{LCA}(d_u, d_v) = 1$ if $d_u = 2$ and $d_v = 3$, which means that the LCA of *Car* and *Bus* is *vehicles*.

Example 6: In Fig. 1, let $D = \{6\}$ and $\hat{D} = \{7\}$. The LCA of *Chair* and *Sofa* is only the node *Seating*. Thus, based on LCA method, $D_{\text{aug}} = \{6, 5\}$ and $\hat{D}_{\text{aug}} = \{7, 5\}$. $P_{\text{LCA}} = 0.5$, $R_{\text{LCA}} = 0.5$, and $F_{\text{LCA}} = 0.5$. However, based on hierarchal method, $D_{\text{aug}} = \{6, 5, 4, 0\}$ and $\hat{D}_{\text{aug}} = \{7, 5, 4, 0\}$. $P_H = 0.75$, $R_H = 0.75$, and $F_H = 0.75$.

According to Example 6, redundant nodes can lead to fluctuations in P_{LCA}, R_{LCA}, and F_{LCA}. Thus they should be removed.

TABLE IV
DATA DESCRIPTION

No.	Datasets	Data type	$\|U\|$	$\|C\|$	$\|d\|$	Node	Leaf	Depth
1	*Bridges* [54]	Num&Sym	108	12	6	7	6	2
2	*SAIAPR* [55]	Image	99526	512	256	256	200	5
3	*VOC* [56]	Image	12283	1000	20	30	20	4
4	*News20* [57]	Text	18846	26214	20	27	20	3

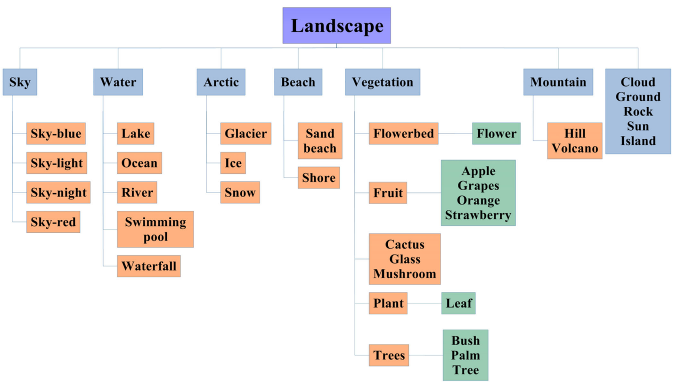

Fig. 4. Hierarchy of landscape branch of the *SAIAPR* dataset.

VI. EXPERIMENTAL ANALYSIS

In this section, we first introduce four datasets used in our experiments. We then compare the proposed hierarchical feature selection with the flat feature selection proposed in [28]. All the numerical experiments are implemented in MATLAB R2014b and executed on an Intel Core i7-3770 running at 3.40 GHz with 16.0 GB memory and a 64-bit Windows 7 operating system. We select the feature subsets on the training sets and test them on the test sets using an SVM, a KNN, and NB classifiers, respectively. For the SVM classifier, ten-fold cross-validation is performed using a linear kernel and $c = 1$. For the KNN classifier, we set parameter $k = 5$ for the class decision based on the preliminary experiments.

A. Datasets

Four datasets are used in the experiments. Basic statistics for these datasets are provided in Table IV.

The first dataset is *Bridges* that is from the University of California-Irvine library [55].

The second dataset is *SAIAPR*, which is an extension of IAPR TC-12 collection. Each image has been manually segmented and the resultant regions have been annotated according to a predefined vocabulary of labels; the vocabulary is organized according to a hierarchy of concepts. According to [56], an object can be in one of six main branches: "animal," "landscape," "man-made," "human," "food," or "other." Fig. 4 shows the "landscape" branch of the hierarchy.

We use portions of the samples (1000, 5000, and 10 000) as a training set to select the feature subset, and use 5000, 10 000, and all samples as the test set to evaluate the effectiveness of the selected feature subset. According to Algorithm 1, 41 features are first selected from 512 features in three training sets containing 1000, 5000, and 10 000 samples, respectively; these features share some attributes. The number of shared attributes is listed in Table V. For example, the feature subset selected from 5000 samples has 32 features that are identical to those in the feature subset selected from 10 000 samples. The running time when using 5000, 10 000, and all samples to test the 41 features selected in different subsets are 53, 190, and 13 500 s, respectively. This demonstrates that using a portion of the samples to approximate the dependence coefficient of the samples can essentially reduce the running time.

TABLE V
NUMBER OF SHARING ATTRIBUTES

	1,000	5,000	10,000
1,000	41		
5,000	32	41	
10,000	35	32	41

TABLE VI
FLAT CLASSIFICATION ACCURACY (SVM)

SVM	1,000	5,000	10,000
All Samples	21.40±0.08	21.28±0.18	21.42±0.20
10,000 Samples	20.57±0.74	20.43±0.57	20.71±0.64
5,000 Samples	19.51±1.68	19.17±1.53	19.47±1.41

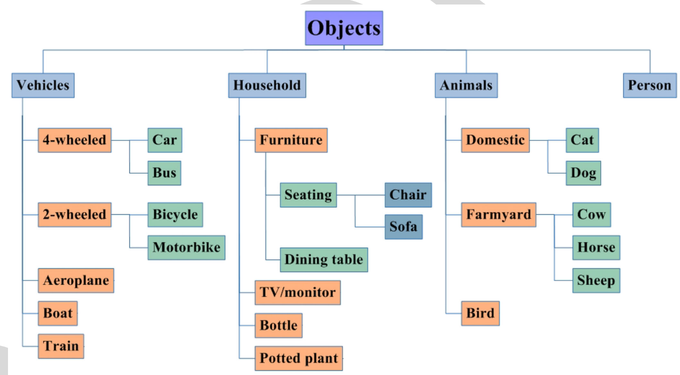

Fig. 5. Hierarchy of the VOC dataset.

The results of flat SVM classification accuracy using different sample subsets listed in Table VI confirm that it is not necessary to use all samples to select features. In this study, we use 5000 samples to select a feature subset under the basic premise of not affecting the classification accuracy.

The third dataset is PASCAL VOC, which is a benchmark in visual object category recognition and detection that provides the vision and machine learning communities with a standard dataset of images and annotations [57]. Fig. 5 shows the hierarchy of VOC. In Table IV, there are 7178 samples for the training dataset and 5105 samples for the testing dataset of PASCAL VOC [57].

Finally, the fourth dataset is *News20* corpus, which was collected and originally used for document classification by Lang [58]. This dataset includes 18 446 messages collected from 20 different Netnews newsgroups. One thousand messages from each of the 20 newsgroups were chosen at random and partitioned by newsgroup name. The list of newsgroups from which the messages were chosen is shown in Fig. 6. We use the "by-date" version, which contains 951 documents evenly distributed across 20 classes. After stemming and stop word removal, this corpus contains 26 214 distinct terms [59].

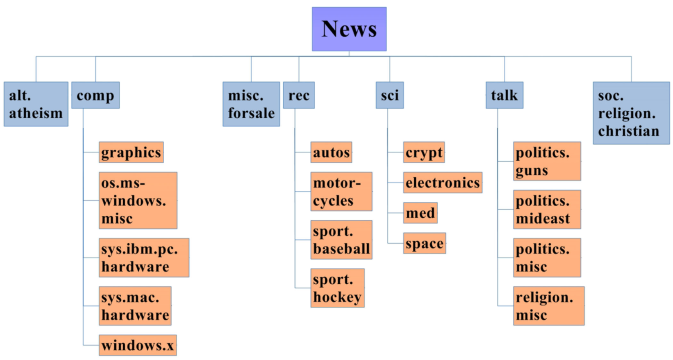

Fig. 6. Hierarchy of the *News20* dataset.

TABLE VII
FLAT EVALUATION ON DIFFERENT DATASETS

(a) *Bridges*

Percentage	SVM		KNN (k=5)		NB	
	Flat	Hierarchical	Flat	Hierarchical	Flat	Hierarchical
100.00%	63.14		56.41		65.56	
63.64%	59.23	**65.95**	52.85	**59.24**	63.80	**64.80**
54.55%	54.74	**62.14**	56.33	**57.09**	**64.45**	61.17
18.18%	53.91	**55.65**	53.74	**53.91**	**57.50**	53.91

(b) *SAIAPR*

Percentage	SVM		KNN (k=5)		NB	
	Flat	Hierarchical	Flat	Hierarchical	Flat	Hierarchical
100.00%	21.10		19.83		6.32	
19.00%	15.28	**20.58**	17.76	**19.48**	**7.34**	6.40
8.01%	14.31	**19.60**	17.28	**18.90**	5.38	**7.42**
4.10%	10.70	**16.84**	10.42	**17.18**	6.32	**7.54**

(c) *VOC*

Percentage	SVM		KNN (k=5)		NB	
	Flat	Hierarchical	Flat	Hierarchical	Flat	Hierarchical
100.00%	48.07		38.72		27.09	
22.00%	38.04	**44.23**	27.44	**37.06**	15.61	**27.09**
6.00%	33.53	**39.51**	24.64	**36.12**	15.05	**29.38**
2.00%	30.30	**34.14**	21.80	**32.14**	15.22	**30.50**

(d) *News20*

Percentage	SVM		KNN (k=5)		NB	
	Flat	Hierarchical	Flat	Hierarchical	Flat	Hierarchical
100.00%	49.60		7.25		31.43	
1.91%	—	**40.03**	—	**20.19**	—	**32.15**
0.15%	**25.32**	23.74	**21.33**	15.75	**27.21**	22.89
0.11%	21.65	**22.79**	**19.02**	17.43	22.81	**22.87**

B. Flat Evaluation

The performance evaluation measures of previous learning algorithms are those commonly used to describe the classification accuracy of SVM, KNN, and NB methods. We refer to these measures as flat evaluations because they do not consider the hierarchical classes. We first use classification accuracy listed in Table VII to visually compare the results of the proposed algorithm with those from a flat algorithm on different datasets. The best performance on each measure is highlighted in bold.

From Table VII, we can identify the changes in accuracy with different numbers of selected features. We can also observe that the performance of the features selected by the hierarchical method is better than that of the flat method. In Table VII(a), it is clear that using 63.64% of features gives better performance than using all features on SVM and KNN classifiers. This means that we can obtain a set of representative features using only the

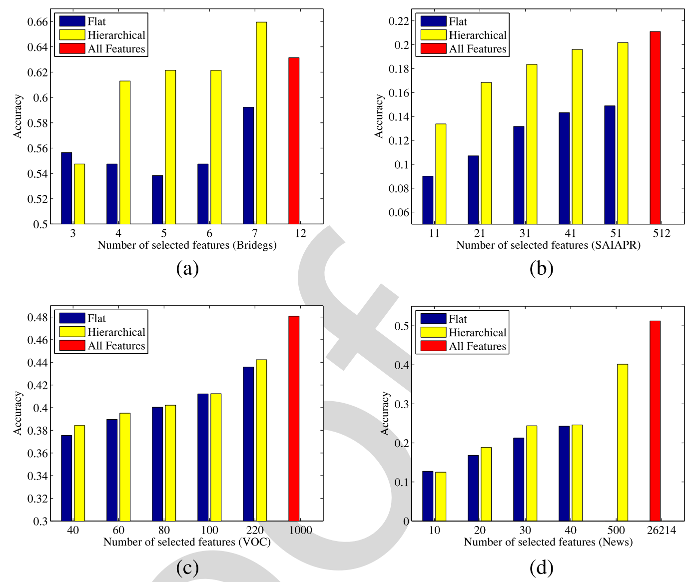

Fig. 7. Comparison of accuracy between flat and hierarchical strategies. (a) *Bridges*. (b) *SAIAPR*. (c) VOC. (d) *News20*.

sibling samples. These results prove the effectiveness of the hierarchical selection method proposed in this paper.

There are 26 214 features in the *News20* dataset. The flat feature selection method takes almost three hours to select a feature. It could not output its results within several days when we select 500 features (1.91%× 26 214). Thus, we use "—" to denote this condition in Table VII. In addition, from Table VII, we can observe that the performance of KNN is not great. The dataset of *News20* is relatively sparse and may be inherently difficult to learn, as evidenced by the relatively poor performance with all features. The accuracy of KNN is only 7.25% when all features have been selected. Thus, KNN is not suitable for this dataset. The accuracy of SVM classification is 40.03% when we select 1.91% of features using the hierarchy method.

Fig. 7 compares the accuracy of SVM between flat and hierarchical strategies on different datasets. The results of the experiments show that our algorithm performs well with different numbers of condition attributes.

C. Hierarchical Evaluation

We use SVM to evaluate our algorithm because the usual measure of performance for such classifiers is the accuracy rate. However, in hierarchical application problems, the output of the hierarchical algorithm is part of the hierarchical classes, which is different from the case of flat classes. Thus, we also use hierarchical evaluation to evaluate the performance of our algorithm. Table VIII presents the results of the hierarchical and flat algorithms on different datasets evaluated by the TIE, Hierarchical F_1, and LCA F_1 measures.

We use TIE to consider some different errors caused by the hierarchy. The "↓" after TIE indicates "the smaller the better." Hierarchical F_1 and LCA F_1 are set-based measures. The "↑" after the set-based measures indicates "the larger the better." We describe the results of these three measures on four datasets in Table VIII. In terms of effectiveness, hierarchical feature selection gives better performance than that of flat feature selection.

TABLE VIII
HIERARCHICAL EVALUATION ON DIFFERENT DATASETS

(a) *Bridges*

Percentage	TIE ↓		Hierarchical F_1 ↑		LCA F_1 ↑	
	Flat	Hierarchical	Flat	Hierarchical	Flat	Hierarchical
100%		10.3		0.81		0.79
63.64%	11.7	**9.5**	0.79	**0.83**	0.77	**0.81**
54.55%	12.6	**10.9**	0.78	**0.81**	0.75	**0.79**
18.18%	12.9	**12.3**	0.78	**0.79**	0.74	**0.75**

(b) *SAIAPR*

Percentage	TIE ↓		Hierarchical F_1 ↑		LCA F_1 ↑	
	Flat	Hierarchical	Flat	Hierarchical	Flat	Hierarchical
100%		1748		0.55		0.48
19.0%	2146	**1768**	0.46	**0.54**	0.42	**0.48**
8.01%	2199	**1807**	0.45	**0.53**	0.41	**0.47**
4.10%	1913	**1885**	0.47	**0.51**	0.41	**0.45**

(c) *VOC*

Percentage	TIE ↓		Hierarchical F_1 ↑		LCA F_1 ↑	
	Flat	Hierarchical	Flat	Hierarchical	Flat	Hierarchical
100%		962		0.70		0.67
22.0%	1244	**1033**	0.65	**0.67**	0.60	**0.65**
6.00%	1347	**1126**	0.62	**0.63**	0.57	**0.61**
2.00%	1447	**1237**	**0.60**	0.58	0.54	**0.58**

(d) *News20*

Percentage	TIE ↓		Hierarchical F_1 ↑		LCA F_1 ↑	
	Flat	Hierarchical	Flat	Hierarchical	Flat	Hierarchical
100%		152		0.72		0.70
1.91%	—	**186**	—	**0.66**	—	**0.63**
0.15%	238	**231**	**0.57**	**0.57**	**0.54**	**0.54**
0.11%	250	**233**	0.55	**0.56**	0.52	**0.53**

TABLE IX
AVERAGE NUMBER OF SAMPLES IN THE SEARCH SPACE

Dataset	Instances	Flat	Hierarchical
Bridges	108	82 (75.9%)	30 (27.8%)
SAIAPR	99526	97542 (98.0%)	6473 (6.5%)
VOC	7178	6503 (90.6%)	276 (3.9%)
News20	11314	10743 (95.0%)	1774 (15.7%)

The results demonstrate that our algorithm provides an efficient solution to finding a better subset of the features.

In terms of the three measures in Table VIII, we observe the following:

1) The value of TIE is related to the scale of the hierarchical structure of classes.
2) The value of LCA F_1 is less than that of Hierarchical F_1. This is because having many common ancestors tends to overpenalize errors. LCA F_1 can avoid this type of error.
3) These three measures for the quantitative hierarchical comparison results are consistent with the flat comparison results.

D. *Comparison of Efficiencies Between Flat and Hierarchical Strategies*

We now study the computational complexity of the flat and hierarchical strategies. Table IX lists the average number of samples in the search space when we compute the lower and upper approximations.

For example, there are 7178 samples in VOC training dataset. The flat feature selection algorithm requires 6503 computations to select one feature. This is 90.6% of the size of VOC training dataset. In contrast, the hierarchical strategy can select one feature from only 276 computations, which is only 3.9% of all the samples. The computational load is reduced from 98.0% to 6.5% on *SAIAPR*. *SAIAPR* has 256 classes, and the sibling strategy is an effective method for datasets with more classes. These statistics lead us to the conclusion that the hierarchical strategy clearly reduces the computational complexity. Example 7 gives an intuitive understanding of the search space of the sibling strategy.

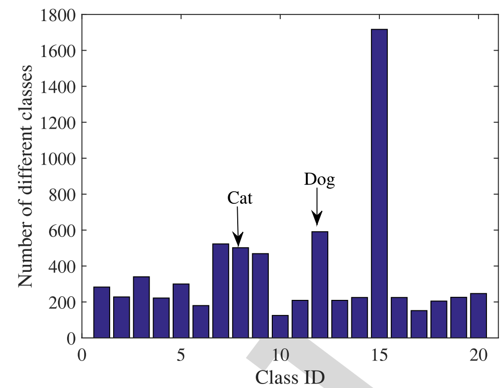

Fig. 8. Number of different classes in VOC dataset.

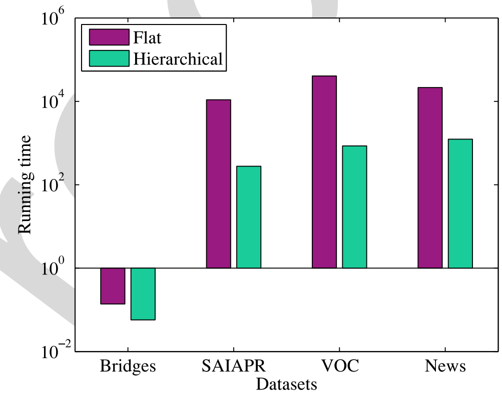

Fig. 9. Running time comparison of the first feature selection between flat and hierarchical strategies.

Example 7: Fig. 5 shows a hierarchical structure of 20 classes. The *Dog* and *Cat* classes have a sibling relationship in this hierarchical structure. Fig. 8 shows the number of different classes in VOC training dataset. Using the exclusive strategy, the negative samples of a *Cat* are all non-*Cat* samples. In contrast, when we use the sibling strategy, the negative samples of a *Cat* are *Dog* samples.

We compare the efficiency of the flat algorithm and hierarchical algorithm. The running times for selecting the first feature for both algorithms are shown in Fig. 9, where the unit of the running time is second. From the results, we note that the hierarchical algorithm is an efficient algorithm in terms of the running time.

The deleting strategy works well on large datasets. Table X shows the comparison of running time used in selecting the first feature and selecting other features. The running time for selecting the first feature is 278.35s on *SAIAPR*. There is a significant reduction from 278.35 to 41.43s.

TABLE X
RUNNING TIME (S)

Dataset	First	Average	Percentage
Bridges	0.058	0.011	28.8%
SAIAPR	278.35	41.43	14.9%
VOC	857.17	309.00	36.1%
News20	1238.65	410.56	33.2%

VII. Conclusions and Future Work

We have proposed a fuzzy rough set based feature selection algorithm for large-scale hierarchical classification. Based on the complicated data structure of modern datasets, we proposed a hierarchical feature selection method by considering the sibling strategy. We used the sibling nodes as the nearest samples from different classes to compute the fuzzy lower approximation and evaluate the features. Two accelerating strategies were employed in the proposed algorithm. In addition, flat and hierarchical evaluations were used to evaluate the effectiveness of the algorithm. Our advantage in terms of practical application is that we control the error rate artificially using the given hierarchical class structure. Experimental results indicate the efficiency and effectiveness of the proposed algorithm. In particular, the proposed algorithm improves the classification performance by selecting the most relevant feature subset. In summary, this study suggests new research trends concerning fuzzy rough sets and hierarchical feature selection problems.

The current implementation of the algorithm just considers tree structures of class labels. In fact, there are other complex structures in practices, such as directed acyclic graphs [18] and chain structures [60]. In the future, we will discuss feature selection algorithms for such tasks. In addition, the proposed algorithm just selects some informative features from the original set. However, discriminant information sometimes hides in the lower-dimensional combination of the high-dimensional features, where feature mapping or feature extraction is preferred. However, the proposed algorithm cannot achieve this objective. We are going to design techniques for hierarchical feature extraction in the future.

References

[1] S. Dumais and H. Chen, “Hierarchical classification of web content,” in *Proc. 23rd Annu. Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, 2000, pp. 256–263.

[2] J. Deng, W. Dong, R. Socher, L. J. Li, K. Li, and L. Fei, “Imagenet: A large-scale hierarchical image database,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2009, pp. 248–255.

[3] J. Struyf, S. Džeroski, H. Blockeel, and A. Clare, *Hierarchical Multi-Classification With Predictive Clustering Trees in Functional Genomics*. New York, NY, USA: Springer, 2005.

[4] Z. L. Cai and W. Zhu, “Multi-label feature selection via feature manifold learning and sparsity regularization,” *Int. J. Mach. Learn. Cybern.*, vol. 9, no. 8, pp. 1321–1334, 2018.

[5] J. H. Dai, B. jie Wei, X. H. Zhang, and Q. L. Zhang, “Uncertainty measurement for incomplete interval-valued information systems based on α-weak similarity,” *Knowl.-Based Syst.*, vol. 136, pp. 159–171, 2017.

[6] S. P. Wang and W. Zhu, “Sparse graph embedding unsupervised feature selection,” *IEEE Trans. Syst. Man Cybern. Syst.*, vol. 48, no. 3, pp. 329–341, Mar. 2018.

[7] X. D. Wu, X. Q. Zhu, G. Q. Wu, and W. Ding, “Data mining with big data,” *IEEE Trans. Knowl. Data Eng.*, vol. 26, no. 1, pp. 97–107, Jan. 2014.

[8] Y. L. Chen, H. W. Hu, and K. Tang, “Constructing a decision tree from data with hierarchical class labels,” *Expert Syst. Appl.*, vol. 36, no. 3, pp. 4838–4847, 2009.

[9] S. Gopal and Y. M. Yang, “Hierarchical Bayesian inference and recursive regularization for large-scale classification,” *ACM Trans. Knowl. Discovery Data*, vol. 9, no. 3, pp. 1–23, 2015.

[10] M. Ashburner *et al.*, “Gene ontology: Tool for the unification of biology,” *Nature Genetics*, vol. 25, no. 1, pp. 25–29, 2000.

[11] N. Nourani-Vatani, R. López-Sastre, and S. Williams, “Structured output prediction with hierarchical loss functions for seafloor imagery taxonomic categorization,” in *Pattern Recognition and Image Analysis*. New York, NY, USA: Springer, 2015, pp. 173–183.

[12] A. X. Sun and E. P. Lim, “Hierarchical text classification and evaluation,” in *Proc. IEEE Int. Conf. Data Mining*, 2001, pp. 521–528.

[13] A. Kosmopoulos, I. Partalas, E. Gaussier, G. Paliouras, and I. Androutsopoulos, “Evaluation measures for hierarchical classification: A unified view and novel approaches,” *Data Mining Knowl. Discovery*, vol. 29, no. 3, pp. 820–865, 2015.

[14] A. Kosmopoulos, E. Gaussier, G. Paliouras, and S. Aseervatham, “The ECIR 2010 large scale hierarchical classification workshop,” *ACM SIGIR Forum*, vol. 44, no. 1, pp. 23–32, 2010.

[15] J. Deng, S. Satheesh, A. C. Berg, and L. Fei, “Fast and balanced: Efficient label tree learning for large scale object recognition,” in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2011, pp. 567–575.

[16] C. Freeman, “Feature selection and hierarchical classifier design with applications to human motion recognition,” Ph.D. dissertation, Dept. Elect. Comput. Eng., Univ. Waterloo, Waterloo, ON, Canada, 2014.

[17] J. Deng, A. C. Berg, K. Li, and L. Fei, “What does classifying more than 10,000 image categories tell us?” in *Proc. Eur. Conf. Comput. Vis.*, 2010, pp. 71–84.

[18] B. Wei and J. T. Kwok, “Multi-label classification on tree- and dag-structured hierarchies,” in *Proc. Int. Conf. Mach. Learn.*, 2011, pp. 17–24.

[19] S. Zhao, Y. Han, Q. Zou, and Q. Hu, “Hierarchical support vector machine based structural classification with fused hierarchies,” *Neurocomputing*, vol. 214, pp. 86–92, 2016.

[20] J. P. Fan, J. Zhang, K. Z. Mei, J. Y. Peng, and L. Gao, “Cost-sensitive learning of hierarchical tree classifiers for large-scale image classification and novel category detection,” *Pattern Recognit.*, vol. 48, no. 5, pp. 1673–1687, 2015.

[21] C. Freeman, D. Kulić, and O. Basir, “Feature-selected tree-based classification,” *IEEE Trans. Cybern.*, vol. 43, no. 6, pp. 1990–2004, Dec. 2013.

[22] C. Freeman, D. Kulić, and O. Basir, “Joint feature selection and hierarchical classifier design,” in *Proc. Int. Conf. Syst., Man, Cybern.*, 2011, pp. 1728–1734.

[23] J. Song, P. Zhang, S. Qin, and J. Gong, “A method of the feature selection in hierarchical text classification based on the category discrimination and position information,” in *Proc. Int. Conf. Ind. Informat.-Comput. Technol., Intell. Technol., Ind. Inf. Integration*, 2015, pp. 132–135.

[24] R. Jensen and N. Mac Parthal áin, “Towards scalable fuzzy–rough feature selection,” *Inf. Sci.*, vol. 323, pp. 1–15, 2015.

[25] J. H. Li, Y. Ren, C. L. Mei, Y. H. Qian, and X. B. Yang, “A comparative study of multigranulation rough sets and concept lattices via rule acquisition,” *Knowl.-Based Syst.*, vol. 91, pp. 152–164, 2016.

[26] C. Z. Wang *et al.*, “A fitting model for feature selection with fuzzy rough sets,” *IEEE Trans. Fuzzy Syst.*, vol. 25, no. 4, pp. 741–753, Aug. 2017.

[27] D. G. Chen and S. Y. Zhao, “Local reduction of decision system with fuzzy rough sets,” *Fuzzy Sets Syst.*, vol. 161, no. 13, pp. 1871–1883, 2010.

[28] Q. H. Hu, D. R. Yu, W. Pedrycz, and D. G. Chen, “Kernelized fuzzy rough sets and their applications,” *IEEE Trans. Knowl. Data Eng.*, vol. 23, no. 11, pp. 1649–1667, Nov. 2011.

[29] J. H. Dai, H. Hu, W. Z. Wu, Y. H. Qian, and D. B. Huang, “Maximal discernibility pairs based approach to attribute reduction in fuzzy rough sets,” *IEEE Trans. Fuzzy Syst.*, vol. 26, no. 4, pp. 2174–2187, Aug. 2018.

[30] S. Y. Zhao, H. Chen, C. P. Li, M. Y. Zhai, and X. Y. Du, “RFRR: Robust fuzzy rough reduction,” *IEEE Trans. Fuzzy Syst.*, vol. 21, no. 5, pp. 825–841, Oct. 2013.

[31] D. G. Chen and Y. Y. Yang, “Attribute reduction for heterogeneous data based on the combination of classical and fuzzy rough set models,” *IEEE Trans. Fuzzy Syst.*, vol. 22, no. 5, pp. 1325–1334, Oct. 2014.

[32] H. M. Chen, T. R. Li, C. Luo, S. J. Horng, and G. Y. Wang, “A decision-theoretic rough set approach for dynamic data mining,” *IEEE Trans. Fuzzy Syst.*, vol. 23, no. 6, pp. 1958–1970, Dec. 2015.

[33] E. Ramentol *et al.*, “IFROWANN: Imbalanced fuzzy-rough ordered weighted average nearest neighbor classification,” *IEEE Trans. Fuzzy Syst.*, vol. 23, no. 5, pp. 1622–1637, Oct. 2015.

[34] W. H. Xu, Q. R. Wang, and X. T. Zhang, "Multi-granulation fuzzy rough sets in a fuzzy tolerance approximation space," *Int. J. Fuzzy Syst.*, vol. 13, no. 4, pp. 246–259, 2011.

[35] W. Z. Wu and Y. Leung, "Theory and applications of granular labelled partitions in multi-scale decision tables," *Inf. Sci.*, vol. 181, no. 18, pp. 3878–3897, 2011.

[36] X. D. Yue, Y. F. Chen, D. Q. Miao, and J. Qian, "Tri-partition neighborhood covering reduction for robust classification," *Int. J. Approximate Reasoning*, vol. 83, pp. 371–384, 2017.

[37] C. Z. Wang, Q. H. Hu, X. Z. Wang, D. G. Chen, Y. H. Qian, and D. Zhe, "Feature selection based on neighborhood discrimination index," *IEEE Trans. Neural Netw. Learning Syst.*, vol. 29, no. 7, pp. 2986–2999, Jul. 2018.

[38] W. Z. Wu, Y. Qian, T. J. Li, and S. M. Gu, "On rule acquisition in incomplete multi-scale decision tables," *Inf. Sci.*, vol. 378, no. C, pp. 282–302, 2016.

[39] Z. Pawlak, *Rough Sets: Theoretical Aspects of Reasoning About Data*. Boston, MA, USA: Kluwer, 1991.

[40] W. Zhu, "Relationship among basic concepts in covering-based rough sets," *Inf. Sci.*, vol. 179, no. 14, pp. 2478–2486, 2009.

[41] D. Dubois and H. Prade, "Putting rough sets and fuzzy sets together," in *Intelligent Decision Support*. New York, NY, USA: Springer, 1992, pp. 203–232.

[42] L. D'eer, N. Verbiest, C. Cornelis, and L. Godo, "A comprehensive study of implicator-conjunctor based and noise-tolerant fuzzy rough sets: Definitions, properties and robustness analysis," *Fuzzy Sets Syst.*, vol. 275, pp. 1–38, 2015.

[43] D. S. Yeung, D. G. Chen, E. C. Tsang, J. W. Lee, and W. Xi Zhao, "On the generalization of fuzzy rough sets," *IEEE Trans. Fuzzy Syst.*, vol. 13, no. 3, pp. 343–361, Jun. 2005.

[44] S. Vluymans, D. S. Tarragó, Y. Saeys, C. Cornelis, and F. Herrera, "Fuzzy rough classifiers for class imbalanced multi-instance data," *Pattern Recognit.*, vol. 53, pp. 36–45, 2016.

[45] S. Vluymans, C. Cornelis, F. Herrera, and Y. Saeys, "Multi-label classification using a fuzzy rough neighborhood consensus," *Inf. Sci.*, vols. 433–434, pp. 96–114, 2018.

[46] R. Eisner, B. Poulin, D. Szafron, P. Lu, and R. Greiner, "Improving protein function prediction using the hierarchical structure of the gene ontology," in *Proc. IEEE Symp. Comput. Intell. Bioinformat. Comput. Biol.*, 2005, pp. 1–10.

[47] M. Ceci and D. Malerba, "Classifying web documents in a hierarchy of categories: A comprehensive study," *J. Intell. Inf. Syst.*, vol. 28, no. 1, pp. 37–78, 2007.

[48] C. N. Silla and A. A. Freitas, "A survey of hierarchical classification across different application domains," *Data Mining Knowl. Discovery*, vol. 22, no. 1/2, pp. 31–72, 2011.

[49] Q. H. Hu, L. Zhang, S. An, D. Zhang, and D. R. Yu, "On robust fuzzy rough set models," *IEEE Trans. Fuzzy Syst.*, vol. 20, no. 4, pp. 636–651, Aug. 2012.

[50] A. Çakmak Pehlivanlı, "A novel feature selection scheme for high-dimensional data sets: Four-staged feature selection," *J. Appl. Statist.*, vol. 43, pp. 1140–1154, 2014.

[51] I. Guyon and A. Elisseeff, "An introduction to variable and feature selection," *J. Mach. Learn. Res.*, vol. 3, pp. 1157–1182, 2003.

[52] O. Dekel, J. Keshet, and Y. Singer, "Large margin hierarchical classification," in *Proc. Int. Conf. Mach. Learn.*, 2004, pp. 1–8.

[53] L. Cai and T. Hofmann, "Exploiting known taxonomies in learning overlapping concepts," in *Proc. Int. Joint Conf. Artif. Intell.*, 2007, pp. 714–719.

[54] A. V. Aho, J. E. Hopcroft, and J. D. Ullman, "On finding lowest common ancestors in trees," *SIAM J. Comput.*, vol. 5, no. 1, pp. 115–132, 1976.

[55] C. L. Blake and C. J. Merz, "UCI repository of machine learning databases," 1998. [Online]. Available: http://www.ics.uci.edu/∼mlearn/mlrepository.html

[56] H. J. Escalante *et al.*, "The segmented and annotated IAPR TC-12 benchmark," *Comput. Vis. Image Understanding*, vol. 114, no. 4, pp. 419–428, 2010.

[57] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes (VOC) challenge," *Int. J. Comput. Vis.*, vol. 88, no. 2, pp. 303–338, 2010.

[58] K. Lang, "Newsweeder: Learning to filter netnews," in *Proc. 20th Int. Conf. Mach. Learn.*, 1995, pp. 331–339.

[59] D. Cai, X. F. He, W. V. Zhang, and J. W. Han, "Regularized locality preserving indexing via spectral regression," in *Proc. 16th ACM Conf. Inf. Knowl. Manage.*, 2007, pp. 741–750.

[60] J. Knorn, A. Rabe, V. C. Radeloff, T. Kuemmerle, J. Kozak, and P. Hostert, "Land cover mapping of large areas using chain classification of neighboring landsat satellite images," *Remote Sens. Environ.*, vol. 113, no. 5, pp. 957–964, 2009.

Hong Zhao received the M.S. degree in computer application from Liaoning Normal University, Dalian, China, in 2006. She is currently a Ph.D. student in software engineering with the School of Computer Software, College of Intelligence and Computing, Tianjin University.

She is also a Professor with the School of Computer Science, Minnan Normal University, Zhangzhou, China. She has authored more than 40 journal and conference papers in the areas of granular computing based machine learning and cost-sensitive learning. Her research interests include rough sets, granular computing, and data mining for hierarchical classification.

Ping Wang received the B.S., M.S., and Ph.D. degrees in computer science from Tianjin University, Tianjin, China, in 1988, 1991, and 1998, respectively.

She is currently a Professor with the School of Mathematics, Tianjin University. She is also a Ph.D. Supervisor with the School of Computer Software, College of Intelligence and Computing, Tianjin University. Her research interests include image processing and machine learning.

Qinghua Hu (SM'13) received the B.S. and M.S. degrees in power engineering and received his Ph.D. degree in control science and engineering.

He was a Postdoctoral Fellow with the Department of Computing, Hong Kong Polytechnic University, Hong Kong, from 2009 to 2011. He is currently a Full Professor and the Vice Director of the College of Intelligence and Computing, Tianjin University, Tianjin, China. He has authored about 200 peer reviewed journal or conference papers in the areas of granular computing based machine learning, reasoning with uncertainty, pattern recognition, and fault diagnosis. His research interests include rough sets, granular computing, and data mining for classification and regression.

Prof. Hu was the Program Committee Co-Chair of the International Conference on Rough Sets and Current Trends in Computing in 2010, the Chinese Rough Set and Soft Computing Society in 2012 and 2014, and the International Conference on Rough Sets and Knowledge Technology, the International Conference on Machine Learning and Cybernetics in 2014, and the General Co-Chair of International Joint Conference on Rough Sets 2015. He is currently the PC-Co-Chairs of China Conference on Machine Learning (CCML) 2017 and Chinese Conference on Computer Vision 2017. He is currently an Associate Editor for the IEEE TRANSACTIONS ON FUZZY SYSTEMS.

Pengfei Zhu received the B.S. degree in power engineering, M.S. degree in power machinery and engineering, and Ph.D degree in computer vision.

He is currently an Associate Professor with the College of Intelligence and Computing, Tianjin University, Tianjin, China. He has authored or coauthored more than 30 papers in International Conference on Computer Vision, Conference on Computer Vision and Pattern Recognition, European Conference on Computer Vision, Association for the Advancement of Artificial Intelligence, International Joint Conferences on Artificial Intelligence, and the IEEE TRANSACTIONS ON INFORMATION FORENSICS AND SECURITY, and IEEE TRANSACTIONS ON IMAGE PROCESSING. His research interests include machine learning and computer vision.

Dr. Zhu is the Local Arrangement Chair for the International Joint Conference on Rough Sets 2015, and the Chinese Conference on Computer Vision 2017.

Fuzzy Rough Set Based Feature Selection for Large-Scale Hierarchical Classification

Hong Zhao, Ping Wang, Qinghua Hu, *Senior Member, IEEE*, and Pengfei Zhu

***Abstract*—The classification of high-dimensional tasks remains a significant challenge for machine learning algorithms. Feature selection is considered to be an indispensable preprocessing step in high-dimensional data classification. In the era of big data, there may be hundreds of class labels, and the hierarchical structure of the classes is often available. This structure is helpful in feature selection and classifier training. However, most current techniques do not consider the hierarchical structure. In this paper, we design a feature selection strategy for hierarchical classification based on fuzzy rough sets. First, a fuzzy rough set model for hierarchical structures is developed to compute the lower and upper approximations of classes organized with a class hierarchy. This model is distinguished from existing techniques by the hierarchical class structure. A hierarchical feature selection problem is then defined based on the model. The new model is more practical than existing feature selection approaches, as many real-world tasks are naturally cast in terms of hierarchical classification. A feature selection algorithm based on sibling nodes is proposed, and this is shown to be more efficient and more versatile than flat feature selection. Compared with the flat feature selection algorithm, the computational load of the proposed algorithm is reduced from 98.0% to 6.5%, while the classification performance is improved on the *SAIAPR* dataset. The related experiments also demonstrate the effectiveness of the hierarchical algorithm.**

***Index Terms*—Feature selection, fuzzy rough sets, granular computing, hierarchical classification.**

I. Introduction

IN THE era of big data, we can observe the following new trends in classification learning.

1) The number of samples continues to increase. We now have abundant datasets for model training.
2) The number of features used to describe the samples has increased from tens to hundreds of thousands, resulting in high-dimensional tasks.
3) The number of class labels is also becoming larger and larger. There are several hundred class labels in some classification tasks, and the class labels form a hierarchical structure, e.g., large-scale web categorization [1], image recognition [2], and gene classification [3].

The number of features is a crucial factor affecting the performance of a classifier. Feature selection aims to select a subset of features to decrease the time complexity, reduce the storage burden, and improve the generalization ability of classification [4]–[6]. This has a significant impact on both the running time and accuracy of the subsequent processing steps. Thus, it is highly desirable to develop effective algorithms that can select informative features from the raw data [7].

Various feature selection algorithms have been developed to select features for binary classification or multiclass tasks. However, there are complex classification structures in real-world applications, where the class labels to be predicted are hierarchically related [8]. Many real-world knowledge systems use a hierarchical scheme to organize their data, particularly ImageNet, Wikipedia [9], Internet web content, biological data [10], geographical data [11], and text data [12]. Hierarchical classification is an increasingly popular method that addresses the problem of classifying items into a hierarchy of classes [13]. In 2009, a workshop was organized for the PASCAL 2 large-scale hierarchical text classification challenge [14]. This workshop discussed the problems and challenges of large-scale hierarchical classification.

It has been reported that hierarchical methods produce better performance than flat classification techniques [15], [16]. Deng *et al.* [17] studied large-scale categorization using a category distance measure based on the WordNet hierarchy. They derived a hierarchy-aware cost function for classification and obtained more informative classification results. Moreover, a hierarchical structure makes it feasible to apply greedy algorithms for large-scale classification. Wei *et al.* [18] adapted a greedy algorithm for multilabel classification on tree-structured hierarchies using subtree optimization. The aforementioned methods are based on a predefined hierarchy. Some other studies [19] have focused on the construction of a hierarchical structure to deal with large-scale classification. For instance, a visual hierarchical structure has been constructed to organize large numbers of classes, and a learning algorithm was developed to train hierarchical classifiers [20]. These hierarchical approaches can

Manuscript received October 1, 2016; revised October 4, 2017, February 16, 2018, and June 3, 2018; accepted October 24, 2018. Date of publication; date of current version. This work was supported in part by the National Natural Science Foundation of China under Grant 61703196, Grant 61432011, Grant U1435212, Grant 61732011, and Grant 91746107; and in part by the Natural Science Foundation of Fujian Province under Grant 2018J01549. *(Corresponding author: Qinghua Hu.)*

H. Zhao is with the School of Computer Software, College of Intelligence and Computing, Tianjin University, Tianjin 300354, China, and also with the School of Computer Science, Minnan Normal University, Zhangzhou 363000, China (e-mail: hongzhaocn@163.com).

P. Wang is with the School of Mathematics and with the School of Computer Software, College of Intelligence and Computing, Tianjin University, Tianjin 300354, China (e-mail: wang_ping@tju.edu.cn).

Q. H. Hu and P. F. Zhu are with the College of Intelligence and Computing, Tianjin University, and also with the Tianjin Key Laboratory of Machine Learning, Tianjin 300354, China (e-mail: huqinghua@tju.edu.cn; zhupengfei@tju.edu.cn).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TFUZZ.2019.2892349

achieve competitive results in terms of both classification accuracy and computational efficiency.

A hierarchical class structure provides some external knowledge of the classes and is helpful not only for classifier training but also feature selection. However, few feature selection approaches for hierarchical class structures have been proposed. Hierarchical feature selection can split the problem into a set of smaller classification problems, each using its own feature set [21]. Freeman *et al.* [22] presented a method for joint feature selection and hierarchical classifier design using genetic algorithms, whereas Song *et al.* [23] proposed a feature selection method for hierarchical text classification. In these works, each child classification selects the best features considering the hierarchical class structure. They improve the accuracy of each classification task, but also reduce the feature dimension.

The theory of fuzzy rough sets is an effective mathematical tool for describing the inconsistency between attributes and decisions, and it is widely used in feature selection and attribute reduction [24]–[26]. In recent years, research on fuzzy rough sets can be categorized into two classes. First, many researchers have discussed the expansion of the fuzzy rough set model. In 2010, Chen *et al.* [27] introduced the concept of local reduction with fuzzy rough sets for a decision system. In 2011, Hu *et al.* [28] integrated kernel functions with fuzzy rough set models and proposed two types of kernelized fuzzy rough sets. In the second class, several different attribute reduction and feature selection methods using fuzzy rough sets have been proposed for different types of datasets [29]. For example, Zhao *et al.* [30] handled noisy datasets using fuzzy rough sets by proposing a robust method of dimension reduction. Another example is the application to decision systems with both symbolic and numerical conditional attributes by composing classical rough set and fuzzy rough set models [31]. In 2015, Chen *et al.* [32] studied the dynamic relation between granules, because data from different applications may evolve with time, that is, the objects, attributes, and attribute values may change dynamically.

The models and applications of fuzzy rough sets have been discussed in a comprehensive manner in recent decades [33]–[35]. These studies have focused almost exclusively on datasets with binary classification or multiclass tasks [36]–[38]. Few studies have considered datasets with high-dimensional classes, especially those with hierarchical class structures. In the era of big data, there may be hundreds of class labels, and the hierarchical structure of the classes is often available. This hierarchical data structure reflects the relationship among classes and is helpful for feature selection and classifier training. However, fuzzy rough set-based feature selection using the hierarchical structure has not been systematically studied.

In this paper, we propose a fuzzy rough set model for hierarchical classification and develop the corresponding feature selection algorithm. First, we embed the hierarchical structure into fuzzy rough sets and redefine the lower and upper approximations using an inclusive strategy and a sibling strategy for the hierarchical classification. The properties of the fuzzy rough sets for hierarchical classification are discussed. Second, we discuss the feature evaluation and feature searching strategies for hierarchical feature selection. In hierarchical classification, we can reduce the search domain for the nearest sample using the predefined class hierarchy. This analysis provides a new viewpoint to extend fuzzy rough sets in hierarchical applications. Finally, a feature selection algorithm is designed for the hierarchical feature selection problem. We use sibling nodes to compute the nearest samples, resulting in an efficient algorithm design. Moreover, some resampling strategies are also considered to accelerate the algorithm. Support vector machines (SVM), k-nearest neighbors (KNN), naive Bayes (NB) classifiers, and three hierarchical measures are used to test the performances of flat and hierarchical feature selection. We report the results of several experiments to demonstrate that the proposed algorithm outperforms the flat algorithms in terms of efficiency and accuracy.

This paper is organized as follows. In Section II, we present some preliminaries on fuzzy rough sets. Then, we introduce the model of fuzzy rough sets for hierarchical classification in Section III. We design a hierarchical feature selection algorithm in Section IV. In Section V, we introduce the evaluation measures for hierarchical feature selection algorithms. In Section VI, we present experimental results and analyze the effectiveness of the hierarchical feature selection algorithm. Finally, in Section VII, we conclude this paper.

II. Preliminaries

In this section, we review the notation for rough sets and fuzzy rough sets.

A. *Rough Sets*

Decision systems are fundamental in data mining and machine learning. Let $I = \langle U, C, D\rangle$ be a decision system, where U is a nonempty set of finite objects (the universe), C is a set of conditional attributes, and D is a set of decision attributes. For each $a \in C \cup D$, $I_a : U \to V_a$. Set V_a is the value set of attribute a, and I_a is an information function for each attribute a.

R is an equivalence relation on U calculated by

$$\mathrm{IND}(R) = \{(x, y) \in U \times U | \forall a \in R, a(x) = a(y)\} \quad (1)$$

where x and y are indiscernible by attributes from R when $(x, y) \in \mathrm{IND}(R)$. The equivalence relation partitions the universe into a family of disjoint subsets called equivalence classes. The equivalence class including x is denoted by $[x]_R$. We call $\mathrm{AS} = \langle U, R\rangle$ an approximation space. For any $X \subseteq U$, two subsets of objects, called lower and upper approximations of X in $\langle U, R\rangle$, are defined as [39]

$$\underline{R}X = \{[x]_R | [x]_R \subseteq X\} \quad (2)$$

$$\overline{R}X = \{[x]_R | [x]_R \cap X \neq \emptyset\}. \quad (3)$$

If $\underline{R}X \neq \overline{R}X$, X is a rough set in the approximation space; otherwise, we say that X is definable.

The rough set theory described above can deal with datasets that contain discrete values [39], [40]. However, most datasets contain numerical attributes. The model of fuzzy rough sets is an extended model to address this problem [41]. The theory of fuzzy rough sets offers an effective way to model the vagueness and imprecision presented in numerical data [28].

B. Fuzzy Rough Sets

Let U be a nonempty and finite set of objects, and R be a fuzzy binary relation on U. We call FAS $= \langle U, R\rangle$ a fuzzy approximation space, where R is a fuzzy equivalence relation. $\forall x, y, z \in U$, we have the following:

1) reflexivity: $R(x,x) = 1$;
2) symmetry: $R(x,y) = R(y,x)$; and
3) min–max transitivity: $\min_y (R(x,y), R(y,z)) \le R(x,z)$.

More generally, we say that R is a fuzzy T-equivalence relation if for $\forall x, y, z \in U$, R satisfies reflexivity, symmetry, and T-transitivity, that is, $T(R(x,y), R(y,z)) \le R(x,z)$.

Given fuzzy approximation space FAS $= \langle U, R\rangle$ and fuzzy subset $X \subseteq U$, fuzzy rough sets can be summarized as the following four operators [42]:

$$\begin{aligned}
\underline{R_S}X(x) &= \inf_{y\in U} S(N(R(x,y)), X(y))\\
\overline{R_T}X(x) &= \sup_{y\in U} T(R(x,y), X(y))\\
\underline{R_\vartheta}X(x) &= \inf_{y\in U} \vartheta(R(x,y), X(y))\\
\overline{R_\sigma}X(x) &= \sup_{y\in U} \sigma(N(R(x,y)), X(y)),
\end{aligned}\tag{4}$$

where T, S, ϑ, and σ denote the fuzzy triangular norm (T-norm), fuzzy triangular conorm (T-conorm), T-residuated implication, and its dual, respectively, and N is a negator. The standard negator is defined as $N(x) = 1 - x$. Several fuzzy operators and their properties were introduced in [43]. Some typical fuzzy operators are listed as follows: $S_M(a,b) = \max(a,b)$,

$$T_M(a,b) = \min(a,b), \vartheta_M(a,b) = \begin{cases} 1, & a \le b \\ b, & a > b \end{cases},$$

$$\sigma_M(a,b) = \begin{cases} 0, & a \ge b \\ b, & a < b \end{cases}.$$

Let $I = \langle U, C, D\rangle$ be a decision system, where U is a universe of objects, C is a nonempty set of conditional attributes with numerical values, and D is the decision attribute that divides the samples into subset $\{d_1, d_2, \ldots, d_l\}$. For all $x \in U$ and if R is a fuzzy similarity relation, then we have

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

Then, the fuzzy rough approximations are computed as

$$\begin{aligned}
\underline{R_S}d_i(x) &= \inf_{y\notin d_i} (1 - R(x,y))\\
\overline{R_T}d_i(x) &= \sup_{y\in d_i} R(x,y)\\
\underline{R_\vartheta}d_i(x) &= \inf_{y\notin d_i} (\sqrt{1 - R^2(x,y)})\\
\overline{R_\sigma}d_i(x) &= \sup_{y\in d_i} (1 - \sqrt{1 - R^2(x,y)}).
\end{aligned}\tag{6}$$

The lower and upper approximations use an equivalence relation to granulate the universe and generate Boolean elemental granules [28] in rough sets. A fuzzy rough set [41] is defined by two fuzzy sets, fuzzy lower and upper approximations defined in (6) that are obtained by extending the corresponding crisp rough set notions defined previously in (2) and (3) [24].

TABLE I
DESCRIPTION OF SYMBOLS USED THROUGHOUT THIS PAPER

Symbol	Meaning
$pos(x)$	The set of samples with the same class of x
$neg(x)$	The set of negative samples of x
$anc(d_u)$	The set of ancestor categories of class d_u
$des(d_u)$	The set of descendant categories of class d_u
$sib(d_u)$	The set of sibling categories of class d_u
$LCA(d_u, d_v)$	Lowest common ancestor of classes d_u and d_v
$\hat{D}$, D	Sets of predicted and true classes
$\hat{D}_{aug}$, D_{aug}	Augmented Sets of predicted and true classes
B	The selected feature subset

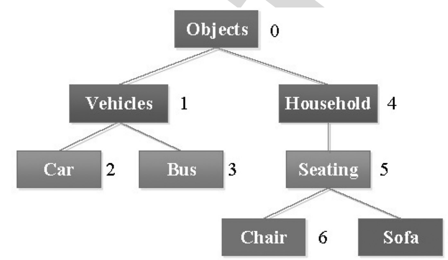

Fig. 1. Example of a tree-based hierarchical class structure.

III. Fuzzy Rough Sets for Hierarchical Classification

A number of learning algorithms have been developed based on fuzzy rough sets [44], [45]. Large-scale data are not only a rich source of information but also produce complex class structures, such as hierarchies. It is interesting and challenging to exploit such structures in modeling.

A. Hierarchical Classification

In this study, we are interested in a tree-based hierarchical class structure. In all cases, the hierarchy imposes a parent-child relationship among the classes, which implies that an instance belonging to a specific class also belongs to all its ancestor classes. Table I describes the most frequent symbols used throughout this paper.

A taxonomy is thus typically defined as a pair $(D, \prec)$, where D is the set of all classes and "$\prec$" represents the "is-a" relationship, which is the *subclass-of* relationship with the following properties [13]:

1) Asymmetry: if $d_i \prec d_j$ then $d_j \nprec d_i$ for every $d_i, d_j \in D$.
2) Antireflexivity: $d_i \nprec d_i$ for every $d_i \in D$.
3) Transitivity: if $d_i \prec d_j$ and $d_j \prec d_k$, then $d_i \prec d_k$ for every $d_i, d_j, d_k \in D$.

An example of a tree-based hierarchical class structure is shown in Fig. 1. The root node *Objects* is not the real class of each sample.

Example 1: In Fig. 1, we can obtain asymmetry and transitivity of a tree-based hierarchical class structure as follows:

1) Asymmetry: *Chair* is a *Seating*, but *Seating* is not a *Chair*.
2) Transitivity: *Chair* is a *Seating* and *Seating* is a *Household*. We can know that *Chair* is a *Household*.

TABLE II
THREE STRATEGIES TO DEFINE POSITIVE AND NEGATIVE SAMPLES

Method	Positive samples	Negative samples
Exclusive strategy [46]	A	Not A
Inclusive strategy [46]	$A + des(A)$	Not $[A + des(A)]$
Sibling strategy [47]	A	$sib(A)$

TABLE III
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}
A	0	0.12	0.19	0.37	0.45	0.49	0.31	0.62	0.35	0.81	0.89	0.92
D	d_1	d_1	d_2	d_2	d_3	d_3	d_4	d_4	d_5	d_5	d_6	d_6

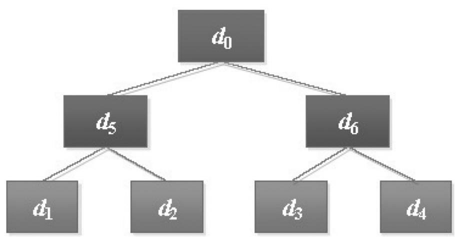

Fig. 2. Tree structure of example data.

B. Flat Classification and Hierarchical Classification

In fuzzy rough sets, the fuzzy lower approximation depends on the nearest sample y from different classes of x. For convenience, we call samples with the same class as x positive samples and call those from different classes as x negative samples. The search scope of negative samples plays a crucial role in defining the lower approximation of fuzzy rough sets. There are several ways to define the positive samples and negative samples for training binary classifiers. We can use these strategies to compute the fuzzy lower approximation and fuzzy upper approximation. Table II gives three strategies to define positive and negative samples, and they are exclusive, inclusive, and sibling strategies.

In flat classification, we do not consider the relationship among different classes. Therefore, the negative samples are not A if the positive sample is A. We call this an exclusive strategy [46], as described in the first row of Table II. Thus, only samples explicitly labeled with A as their most specific class are selected as positive samples, and everything else is considered as negative samples.

Given a classification task, we have 12 samples listed in Table III. Each sample is characterized by a condition attribute A. d_1, d_2, d_3, d_4, d_5, and d_6 are six classes.

The positive class is the class of sample x_i, and the negative class is the class different from x_i. Compared with hierarchical classification, the flat classification approach is the simplest one that does not consider the hierarchy of the class.

Hierarchical problems are particularly prevalent in large-scale datasets. We are interested in approaches that cope with a predefined class hierarchy. Fig. 2 shows the tree structure of D_{tree}, where D_{tree} is a tree-based hierarchical class with values d_1, d_2, d_3, d_4, d_5, and d_6 in Table III.

According to the tree-based hierarchical class structure, there is an "is-a" relationship between the parent and child nodes to describe the parent-child relationship. The descendant categories of x are positive samples; therefore, it is not necessary to consider these samples when the lower approximation is computed. We call this an inclusive strategy [46], as described in the second row of Table II, where $\text{des}(A)$ denotes descendant categories of class A.

Based on the tree-based hierarchical class structure, sibling nodes with the same parent have a high fuzzy similarity degree. Therefore, it may be effective to search for negative samples within only the sibling nodes called the sibling strategy. The sibling strategy [47] is listed in the third row of Table II, where $\text{sib}(A)$ denotes sibling categories of class A. We can use this hierarchical information to decrease the search scope of the negative samples and reduce the algorithm's complexity.

We use the following example to compare the exclusive strategy with flat classes and the inclusive and sibling strategies with hierarchical classes.

Example 2: Continuing with Example 1, we give an intuitive interpretation of different positive and negative samples in Fig. 1.

We have the following results according to different strategies.

1) Exclusive strategy: The positive sample is *Chair* if we let A be *Chair*. That is, $\text{pos}(A) = \{5\}$. The negative samples are not *Chair*, that is, $\text{neg}(A) = \{1, 2, 3, 4, 5, 7\}$.
2) Inclusive strategy: The positive samples are *Seating*, *Chair*, and *Sofa*, that is, $\text{pos}(A) = \{5, 6, 7\}$. The negative samples are $\text{neg}(A) = \{1, 2, 3, 4\}$.
3) Sibling strategy: The positive sample is *Chair* if we let A be *Chair*. The negative samples are $\text{sib}(A) = \{7\}$.

In fuzzy rough sets, the fuzzy lower approximation of a sample is computed from the nearest sample to x_i in classes different from x_i, which means the nearest negative sample. In this tree hierarchical structure, the nearest sample is in the descendant, ancestor, and sibling categories. From Table II, the descendant categories are usually positive samples. Therefore, we use the sibling strategy to select negative samples. For example, the nearest negative sample to *Chair* is *Sofa*, which is consistent with an intuitive interpretation.

C. Fuzzy Rough Sets for Hierarchical Classification

Classification is one of the most important problems in data mining, machine learning, and statistical pattern recognition. Related research has focused on flat classification problems, which are standard binary or multiclass classification problems [48]. The lower approximation of classical fuzzy rough sets is the minimum distance of a sample from the different classes, and the upper approximation is the maximum distance in the same class [49]. Generally, we focus on traditional datasets with nonhierarchical classes. Therefore, the same classes of x exclude every instance except for those that have exactly the same class as x (and not those that are more general or more specific).

Nowadays, in some important applications, there are several hierarchical classification problems. The hierarchy defines an inheritance (IS-A) relationship between the class nodes, where each class is a special case of its parent class [46]. Any class is a special case of each ancestor class, where an ancestor is any class along the path from the class to the root of the hierarchy. Now, we consider the fuzzy lower approximation of classification for hierarchical classes.

The tree-based hierarchical class structure can be formulated as $\langle U, C, D_{\text{tree}}\rangle$, where U is a universal set of objects, C is a nonempty set of conditional attributes, and D_{tree} is the decision attribute that divides the samples into subsets $\{d_1, d_2, \ldots, d_l\}$. l is the number of classes. D_{tree} satisfies a pair $(D_{\text{tree}}, \prec)$, which is introduced in Section III-A. R is a fuzzy similarity relation on U generated with features $B \subseteq C$.

There are several methods for defining the set of positive (same) and negative (different) classes in Table II. We can use these strategies to define the approximation of fuzzy rough sets for hierarchical classification. Traditional classification deals with nonhierarchical classes, which is flat classification. We call this the exclusive strategy. The lower and upper approximations are defined in (6).

When inclusive strategy is considered, for all $x \in U$, we have

$$d_i(x) = \begin{cases} 0, & x \notin \{\text{des}(d_i) \cup d_i\} \\ 1, & x \in \{\text{des}(d_i) \cup d_i\} \end{cases}. \tag{7}$$

The fuzzy rough approximations are defined as

$$\begin{aligned} \underline{R_S}_{\text{inclusive}} d_i(x) &= \inf_{y \notin \{\text{des}(d_i) \cup d_i\}} (1 - R(x,y)) \\ \overline{R_T}_{\text{inclusive}} d_i(x) &= \sup_{y \in \{\text{des}(d_i) \cup d_i\}} R(x,y) \\ \underline{R_\vartheta}_{\text{inclusive}} d_i(x) &= \inf_{y \notin \{\text{des}(d_i) \cup d_i\}} (\sqrt{1 - R^2(x,y)}) \\ \overline{R_\sigma}_{\text{inclusive}} d_i(x) &= \sup_{y \in \{\text{des}(d_i) \cup d_i\}} (1 - \sqrt{1 - R^2(x,y)}). \end{aligned} \tag{8}$$

When sibling strategy is considered, for all $x \in U$, we have

$$d_i(x) = \begin{cases} 0, & x \in \{\text{sib}(d_i)\} \\ 1, & x \in \{d_i\} \end{cases}. \tag{9}$$

The fuzzy rough approximations are defined as

$$\begin{aligned} \underline{R_S}_{\text{sibling}} d_i(x) &= \inf_{y \in \{\text{sib}(d_i)\}} (1 - R(x,y)) \\ \overline{R_T}_{\text{sibling}} d_i(x) &= \sup_{y \in \{d_i\}} R(x,y) \\ \underline{R_\vartheta}_{\text{sibling}} d_i(x) &= \inf_{y \in \{\text{sib}(d_i)\}} (\sqrt{1 - R^2(x,y)}) \\ \overline{R_\sigma}_{\text{sibling}} d_i(x) &= \sup_{y \in \{d_i\}} (1 - \sqrt{1 - R^2(x,y)}). \end{aligned} \tag{10}$$

Several properties of the fuzzy rough sets for hierarchical classification are as follows. Compared with the exclusive strategy, we have the following propositions when we consider the sibling strategy.

Proposition 1: Given $\langle U, C, D_{\text{tree}}\rangle$, R is a fuzzy similarity relation induced by $B \subseteq C$. Let d_i be a class of samples labeled with i, for $x \in U$

$$\begin{aligned} \underline{R_S}_{\text{sibling}} d_i(x) &\geq \underline{R_S} d_i(x) \\ \underline{R_\vartheta}_{\text{sibling}} d_i(x) &\geq \underline{R_\vartheta} d_i(x). \end{aligned} \tag{11}$$

Proof: Suppose that y_{si} is the sample with class $y_{si} \in \text{sib}(d_i)$, such that $\underline{R_S}_{\text{sibling}} d_i(x) = 1 - R(x, y_{si})$. Suppose that y_{ex} is the sample with class $y_{ex} \in D_{\text{tree}} \backslash d_i$, such that $\underline{R_S} d_i(x) = 1 - R(x, y_{ex})$. Since $\text{sib}(d_i) \subseteq D_{\text{tree}} \backslash d_i$, we have $R(x, y_{si}) \leq R(x, y_{ex})$. Therefore, $\underline{R_S}_{\text{sibling}} d_i(x) \geq \underline{R_S} d_i(x)$. Analogically, we can also obtain $\underline{R_\vartheta}_{\text{sibling}} d_i(x) \geq \underline{R_\vartheta} d_i(x)$. ∎

Proposition 2: Given $\langle U, C, D_{\text{tree}}\rangle$, R is a fuzzy similarity relation induced by $B \subseteq C$. If d_i is a class of samples labeled with i and $x \in U$, we have

$$\begin{aligned} \overline{R_T}_{\text{sibling}} d_i(x) &= \overline{R_T} d_i(x) \\ \overline{R_\sigma}_{\text{sibling}} d_i(x) &= \overline{R_\sigma} d_i(x). \end{aligned} \tag{12}$$

Proof: Since $\overline{R_T} d_i(x) = \sup_{y \in d_i} R(x,y)$ and $\overline{R_T}_{\text{sibling}} d_i(x) = \sup_{y \in d_i} R(x,y)$. Therefore, $\overline{R_T}_{\text{sibling}} d_i(x) = \overline{R_T} d_i(x)$. Analogically, we can also obtain $\overline{R_\sigma}_{\text{sibling}} d_i(x) = \overline{R_\sigma} d_i(x)$. ∎

The sibling strategy and inclusive strategy have different positive and negative sample definitions. We have the following proposition when we consider the sibling strategy and inclusive strategy, respectively.

Proposition 3: Given $\langle U, C, D_{\text{tree}}\rangle$, R is a fuzzy similarity relation induced by $B \subseteq C$. Let d_i be a class of samples labeled with i, for $x \in U$

$$\begin{aligned} \underline{R_S}_{\text{sibling}} d_i(x) &\geq \underline{R_S}_{\text{inclusive}} d_i(x) \\ \overline{R_T}_{\text{sibling}} d_i(x) &\leq \overline{R_T}_{\text{inclusive}} d_i(x) \\ \underline{R_\vartheta}_{\text{sibling}} d_i(x) &\geq \underline{R_\vartheta}_{\text{inclusive}} d_i(x) \\ \overline{R_\sigma}_{\text{sibling}} d_i(x) &\leq \overline{R_\sigma}_{\text{inclusive}} d_i(x). \end{aligned} \tag{13}$$

Proof: Suppose that y_{si} is the sample with class from $\text{sib}(d_i)$, such that $\underline{R_S}_{\text{sibling}} d_i(x) = 1 - R(x, y_{si})$. Suppose that y_{in} is the sample with class from $D_{\text{tree}} \backslash \{\text{des}(d_i) \cup d_i\}$, such that $\underline{R_S}_{\text{inclusive}} d_i(x) = 1 - R(x, y_{\text{in}})$. Since $\text{sib}(d_i) \subseteq D_{\text{tree}} \backslash \{\text{des}(d_i) \cup d_i\}$, we have $R(x, y_{si}) \leq R(x, y_{\text{in}})$. Thus, $\underline{R_S}_{\text{sibling}} d_i(x) \geq \underline{R_S}_{\text{inclusive}} d_i(x)$. Analogically, we can also obtain $\underline{R_\vartheta}_{\text{sibling}} d_i(x) \geq \underline{R_\vartheta}_{\text{inclusive}} d_i(x)$.

Suppose that y_{si} is the sample with class from d_i, such that $\overline{R_T}_{\text{sibling}} d_i(x) = R(x, y_{si})$. Suppose that y_{in} is the sample with class from $\{\text{des}(d_i) \cup d_i\}$, such that $\overline{R_T}_{\text{inclusive}} d_i(x) = R(x, y_{\text{in}})$. Since $d_i \subseteq \{\text{des}(d_i) \cup d_i\}$, we have $R(x, y_{si}) \leq R(x, y_{\text{in}})$. Thus, $\overline{R_T}_{\text{sibling}} d_i(x) \leq \overline{R_T}_{\text{inclusive}} d_i(x)$. Analogically, we can also obtain $\overline{R_\sigma}_{\text{sibling}} d_i(x) \leq \overline{R_\sigma}_{\text{inclusive}} d_i(x)$. ∎

According to Propositions 2 and 3, we can obtain

$$\begin{aligned} \overline{R_T} d_i(x) &\leq \overline{R_T}_{\text{inclusive}} d_i(x) \\ \overline{R_\sigma} d_i(x) &\leq \overline{R_\sigma}_{\text{inclusive}} d_i(x). \end{aligned} \tag{14}$$

Proposition 4: Given $\langle U, C, D_{\text{tree}}\rangle$, R is a fuzzy similarity relation induced by $B \subseteq C$. Let d_i be a class of samples labeled with i, for $x \in U$

$$\underline{R_S}_{\text{inclusive}} d_i(x) \geq \underline{R_S} d_i(x)$$
$$\underline{R_\vartheta}_{\text{inclusive}} d_i(x) \geq \underline{R_\vartheta} d_i(x). \tag{15}$$

Proof: Suppose that y_{in} is the sample with class from $D_{\text{tree}}\backslash\{\text{des}(d_i) \cup d_i\}$, such that $\underline{R_S}_{\text{inclusive}} d_i(x) = 1 - R(x, y_{\text{in}})$. Suppose that y_{ex} is the sample with class $y_{ex} \in D_{\text{tree}}\backslash d_i$, such that $\underline{R_S} d_i(x) = 1 - R(x, y_{ex})$. Since $D_{\text{tree}}\backslash\{\text{des}(d_i) \cup d_i\} \subseteq D_{\text{tree}}\backslash d_i$, we have $R(x, y_{\text{in}}) \leq R(x, y_{ex})$. Thus, $\underline{R_S}_{\text{inclusive}} d_i(x) \geq \underline{R_S} d_i(x)$. Analogically, we can also obtain $\underline{R_\vartheta}_{\text{inclusive}} d_i(x) \geq \underline{R_\vartheta} d_i(x)$. ■

Proposition 5: Given $\langle U, C, D_{\text{tree}}\rangle$, R_1 and R_2 are two fuzzy similarity relations induced by B_1 and B_2, respectively, and $R_1 \subseteq R_2$. Let d_i be a class of samples labeled with i, for $x \in U$

$$\underline{R_{1S}}_{\text{sibling}} d_i(x) \geq \underline{R_{2S}}_{\text{sibling}} d_i(x)$$
$$\overline{R_{1T}}_{\text{sibling}} d_i(x) \leq \overline{R_{2T}}_{\text{sibling}} d_i(x)$$
$$\underline{R_{1\vartheta}}_{\text{sibling}} d_i(x) \geq \underline{R_{2\vartheta}}_{\text{sibling}} d_i(x)$$
$$\overline{R_{1\sigma}}_{\text{sibling}} d_i(x) \leq \overline{R_{2\sigma}}_{\text{sibling}} d_i(x). \tag{16}$$

Proof: The proof is straightforward. ■

We give the following example to compare the computation among three strategies on the intermediate nodes. For simplification, we use the model defined with T-norm and T-conorm operators. For comparing with the flat algorithm in [28], we use the same function, the Gaussian function, to compute fuzzy similarity relations R, and the parameter σ is set to 0.2

$$R(x, y) = \exp\left(-\frac{||x - y||^2}{\sigma}\right), \tag{17}$$

where $||x - y||$ is the distance between x and y.

Example 3: We give an example of computing fuzzy lower approximation based on different strategies with the data listed in Table III. We select x_3 with class d_2 to compute the lower approximation. For exclusive strategy

$$\begin{aligned}\underline{R_S} d_2(x_3) &= \inf_{y \notin \{d_2\}} (1 - R(x_3, y)) \\ &= \inf_{y \in \{d_1, d_3, d_4, d_5, d_6\}} (1 - R(x_3, y)) \\ &= 1 - \exp\left(-\frac{||x_3 - x_2||^2}{0.2}\right) = 0.0242.\end{aligned} \tag{18}$$

As to the inclusive strategy

$$\begin{aligned}\underline{R_S}_{\text{inclusive}} d_2(x_3) &= \inf_{y \notin \{\text{des}(d_2) \cup d_2\}} (1 - R(x_3, y)) \\ &= \inf_{y \notin \{d_2, d_1, d_3\}} (1 - R(x_3, y)) \\ &= \inf_{y \in \{d_4, d_5, d_6\}} (1 - R(x_3, y)) \\ &= 1 - \exp\left(-\frac{||x_3 - x_7||^2}{0.2}\right) = 0.0695.\end{aligned} \tag{19}$$

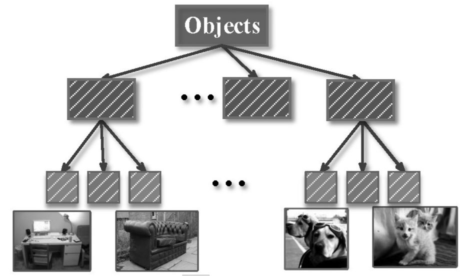

Fig. 3. Example of sibling relationship.

As to the sibling strategy

$$\begin{aligned}\underline{R_S}_{\text{sibling}} d_2(x_3) &= \inf_{y \in \{\text{sib}(d_2)\}} (1 - R(x_3, y)) \\ &= \inf_{y \in \{d_5\}} (1 - R(x_1, y)) \\ &= 1 - \exp\left(-\frac{||x_3 - x_9||^2}{0.2}\right) = 0.1201.\end{aligned} \tag{20}$$

We have $\underline{R_S}_{\text{sibling}} d_i(x) \geq \underline{R_S}_{\text{inclusive}} d_i(x) \geq \underline{R_S} d_i(x)$.

In this example, we should compute the samples $y \in \{d_1, d_3, d_4, d_5, d_6\}$ when we use the exclusive strategy and the samples $y \in \{d_4, d_5, d_6\}$ when we consider the inclusive strategy. We need to compute the samples $y \in \{d_5\}$ for the sibling strategy. This can significantly reduce the computation time, especially for large datasets.

IV. Hierarchical Feature Selection

Feature selection is an indispensable preprocessing step of high-dimensional data classification [50], and can help to identify redundant or correlated features [51]. Fuzzy rough set theory is an effective method for selecting feature subsets using the dependencies between the decision and condition attributes. These dependencies can identify effective features for classification. The two main steps in any feature selection algorithm are feature evaluation and the search strategy.

The inclusive strategy and sibling strategy discussed above have their own advantages. The inclusive strategy reduces the computational complexity when we consider the intermediate nodes. In this paper, we consider the leaf nodes to be real classes and use the sibling strategy to select the feature subset. The minimum distance of a sample from different classes is a critical factor in feature selection. Fig. 3 shows the hierarchical structure of classes. In this hierarchical structure, there are common characteristics among the sibling classes because they share a parent node. Thus, we select the nearest negative samples from the sibling nodes, which is consistent with an intuitive interpretation.

Definition 1: Given a hierarchical classification problem $\langle U, C, D_{\text{tree}}\rangle$, R is the $T-$ equivalence relation on U computed with the distance function $R(x, y)$ in the feature space $B \subseteq C$. $D_{\text{tree}} = \{d_0, d_1, d_2, \ldots, d_l\}$, where d_0 is the root of the tree and it is not the real class. U is divided into $\{d_1, d_2, \ldots, d_l\}$ with the decision attribute, where l is the number of classes. The fuzzy positive region of D_{tree} in term of B is defined as

$$\text{POS}^S_{B_{\text{sibling}}}(D_{\text{tree}}) = \cup_{i=1}^{l} \underline{R_S}_{\text{sibling}} d_i. \tag{21}$$

Definition 2: Given a classification problem $\langle U, C, D_{\text{tree}}\rangle$, R is the T-equivalence relation on U computed with the distance function $R(x, y)$ in the feature space $B \subseteq C$, and U is divided into $\{d_1, d_2, \ldots, d_l\}$ with the decision attribute, where l is the number of classes. The quality of the classification approximation is defined as

$$\gamma^S_{B_{\text{sibling}}}(D_{\text{tree}}) = \frac{|\cup_{i=1}^{l} \underline{R_S}_{\text{sibling}} d_i|}{|U|}. \tag{22}$$

As $\underline{R_S}_{\text{sibling}} d_i(x) = \inf_{y \in \text{sib}(d_i)} (1 - R(x, y))$, we get that

$$|\cup_{i=1}^{l} \underline{R_S}_{\text{sibling}} d_i| = \sum_{j=1}^{|U|} \sum_{i=1}^{l} \underline{R_S}_{\text{sibling}} d_i(x_j). \tag{23}$$

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

$$\begin{aligned}\sum_{j=1}^{|U|} \sum_{i=1}^{l} \underline{R_S}_{\text{sibling}} d_i(x_j) &= \sum_{j=1}^{|U|} \underline{R_S}_{\text{sibling}} d(x_j) \\ &= \sum_{j=1}^{|U|} \inf_{x_j \in d, y \in \text{sib}(d)} (1 - R(x_j, y))\end{aligned} \tag{24}$$

where d is the class label of x_j.

The coefficients of classification quality reflect the approximation ability of the approximation space or the ability of the granulated space induced by feature subset B to characterize the decision [28]. These coefficients can evaluate the condition attribute with degree $\gamma^S_B(D_{\text{tree}})$, and reflect the dependence between the decision and condition attributes. The monotonicity approximations are given by Theorem 1, which applies to both sibling strategy and inclusive strategy.

Theorem 1: Given a hierarchical classification problem $\langle U, C, D_{\text{tree}}\rangle$, R_1 and R_2 are two fuzzy similarity relations induced by B_1 and B_2, respectively, and $R_1 \subseteq R_2$, we have

$$\text{POS}^S_{B_1}(D_{\text{tree}}) \subseteq \text{POS}^S_{B_2}(D_{\text{tree}}). \tag{25}$$

Proof: Let d_i be a class of samples labeled with i, for $x \in U$, we have $\underline{R_{1S}} d_i(x) \geq \underline{R_{2S}} d_i(x)$ since $R_1 \subseteq R_2$. We can derive that $\text{POS}^S_{B_1}(D_{\text{tree}}) \subseteq \text{POS}^S_{B_2}(D_{\text{tree}})$ since $\text{POS}^S_B(D_{\text{tree}}) = \cup_{i=1}^{l} \underline{R_S} d_i$. ■

According to Definition 2 and Theorem 1, we have

$$\gamma^S_{B_1}(D_{\text{tree}}) \leq \gamma^S_{B_2}(D_{\text{tree}}). \tag{26}$$

In a feature selection algorithm, feature evaluation quantifies how good the feature subset is, and search strategies are used to identify the optimal feature subset. First, we evaluate each feature according to its dependence coefficient and rank them in terms of feature quality. Then, we select the best feature and delete redundant features to further reduce the computation time.

A fuzzy rough sets based feature selection algorithm for hierarchical classification (FFS-HC) is illustrated in Algorithm 1.

Algorithm 1 A fuzzy Rough Sets Based Feature Selection Algorithm for Hierarchical Classification (FFS-HC).

Input: $\langle U, C, D_{\text{tree}}\rangle$
Output: A feature subset B
1: $B = \emptyset$; $CD = \emptyset$;
//Addition
2: $CA = C$;
3: **while** $(\gamma^S_C(D_{\text{tree}}) - \gamma^S_B(D_{\text{tree}}) < \delta)$) **do**
4: **for** each $a \in CA$ **do**
5: Compute $\gamma^S_{a\cup B}(D_{\text{tree}})$ according to SSFE;
6: **end for**//Delete the redundant features
7: **if** $B == \emptyset$ **then**
8: **for** each $a \in CA$ **do**
9: Select feature a_{del} is smaller than the average $\gamma^S_a(D_{\text{tree}})$;
10: $CD = CD \cup a_{\text{del}}$;
11: **end for**
12: $CA = CA - CD$;
13: **end if**
14: Select a' with the maximal $\gamma^S_{a'\cup B}(D_{\text{tree}})$;
15: $B = B \cup \{a'\}$;
16: $CA = CA - \{a'\}$;
17: **end while**
18: return B;

The sibling strategy based feature evaluation (SSFE) of FFS-HC is provided in line 5 in Algorithm 1, and the specific implementation of SSFE is illustrated in Algorithm 2. D_{tree} is a tree-based hierarchical structure of the classes, and it is a global variable that should be explicitly initialized.

We use a sibling-based relief algorithm to find the optimal feature subset for comparing the flat feature selection with the proposed hierarchical feature selection. The complexity of the relief algorithm will become unacceptable when the number of records in the dataset increases to a large scale. In general, the size of the search space for the feature selection algorithm is $2^{|C|}$. Algorithm 1 deals with this issue effectively by deleting redundant features to reduce the search space.

We consider two strategies in Algorithms 1 and 2 for reducing the search space. First, we can reduce the computing space by using the sibling strategy, which is listed from lines 3–9 in Algorithm 2. This strategy can reduce the computation time significantly. Second, we compute the dependence of each feature only once. We then delete the redundant features in the first round, as described from lines 7–13 in Algorithm 1.

V. Evaluation Measures

The proposed method is to deal with hierarchical classification, which is different from flat classification. Accordingly, the evaluation measures for the FFS-HC algorithm should be different. Measures were introduced to evaluate hierarchical classification in [13].

Example 4: Fig. 1 shows the hierarchical classification subtree of visual object classes (VOC) classification. We assume that the true class for a test instance is *Car* and that two classification systems output *Bus* (Case 1) and *Sofa* (Case 2) as the

Algorithm 2 Sibling Strategy Based Feature Evaluation (SSFE).

```
Input: ⟨U, C, D_tree⟩, r = 0, and B
Output: r
for i = 1 to |U| do
    Compute decision d_i of sample x_i;
    Select samples X_sib with class sib(d_i);
    if length(X_sib) == 0 then
        Random select samples out of d_i as X_sib;
    end if
    for each y ∈ X_sib do
        Compute 1 − R(x_i, y);
    end for
    Select y' such that R_S_sibling d_i(x_i) = 1 − R(x_i, y');
    r = r + 1 − R(x_i, y');
end for
r = r/|U|;
return r;
```

predicted classes. These two errors are the same using flat evaluation measures, and these two systems are punished equally. However, Case 2 is more severe because it makes a prediction in a different and unrelated subtree. Thus, the punishment for Case 2 should be larger than that for Case 1.

In some cases, a sample can be classified into more than one class in the hierarchy. The pair-based measure and set-based measure are two main hierarchical evaluation measures.

A. Pair-Based Measures

As stated above, different classification errors result in different levels of penalty. In our model, this penalty is defined by the tree distance, which is called the *tree-induced error* (TIE) in [52]. The TIE is computed by predicting label d_v when the correct label is d_u

$$\mathrm{TIE}(d_u, d_v) = |E_H(d_u, d_v)| \tag{27}$$

where $E_H(d_u, d_v)$ is the set of edges along the path from d_u to d_v in the hierarchy, and $|\cdot|$ denotes the count of elements. That is, $\mathrm{TIE}(d_u, d_v)$ is defined to be the number of edges along the path from d_u to d_v in the tree of D. $\mathrm{TIE}(d_u, d_u) = 0$, $\mathrm{TIE}(d_u, d_v) = \mathrm{TIE}(d_v, d_u)$, and the triangle inequality always holds with equality.

Example 5: Continuing with Example 4, the true class for a test instance is *Car*. The predicted class with *Sofa* is punished $\mathrm{TIE}(2, 7) = 5$, which is larger than the punishment $\mathrm{TIE}(2, 3) = 2$ for the predicted class with *Bus*.

B. Set-Based Measures

Pair-based measures consider only a pair of predicted and true classes. Unlike pair-based measures, set-based measures take into account the entire sets of predicted and true classes, including their ancestors or descendants.

Set-based measures have the following two distinct phases:

1) the augmentation of D and $\hat{D}$ with information on the hierarchy; and
2) the calculation of a cost measure based on the augmented sets.

The augmentation of D and $\hat{D}$ is a crucial step that attempts to capture the hierarchical relations of the classes. There are different measures based on different augmented approaches for the sets of predicted and true classes. We select the measure that the sets are augmented with the ancestors of the true and predicted classes [3], [53] as follows:

$$\begin{aligned} D_{\text{aug}} &= D \cup \mathrm{anc}(D) \\ \hat{D}_{\text{aug}} &= \hat{D} \cup \mathrm{anc}(\hat{D}). \end{aligned} \tag{28}$$

Hierarchical precision and recall are defined as follows:

$$\begin{aligned} P_H &= \frac{|\hat{D}_{\text{aug}} \cap D_{\text{aug}}|}{|\hat{D}_{\text{aug}}|} \\ R_H &= \frac{|\hat{D}_{\text{aug}} \cap D_{\text{aug}}|}{|D_{\text{aug}}|} \end{aligned} \tag{29}$$

where $|\cdot|$ denotes the count of elements. The F_1-measure is defined as follows:

$$F_H = \frac{2 \cdot P_H \cdot R_H}{P_H + R_H}. \tag{30}$$

Continuing with Example 4, we can compute the hierarchical precision, recall, and F_1-measure of two cases.

Case 1: In Fig. 1, let $D = \{2\}$ and $\hat{D} = \{3\}$, which means that the true class of a test instance is *Car* and the predicted class is *Bus:* $D_{\text{aug}} = \{2, 1, 0\}$ and $\hat{D}_{\text{aug}} = \{3, 1, 0\}$; $P_H = 0.67$, $R_H = 0.67$, and $F_H = 0.67$.

Case 2: In Fig. 1, let $D = \{2\}$ and $\hat{D} = \{7\}$, which means that the true class for a test instance is *Car* and the predicted class is *Sofa:* $D_{\text{aug}} = \{2, 1, 0\}$ and $\hat{D}_{\text{aug}} = \{7, 5, 4, 0\}$; $P_H = 0.25$, $R_H = 0.33$, and $F_H = 0.29$.

C. Lowest Common Ancestor (LCA) F_1 Measure

The set-based measure adds all the ancestors, and it has over penalizing errors that occur to nodes with many ancestors. Kosmopoulos *et al.* [13] proposed LCA measures to deal with this problem. The concept of LCA was defined in graph theory [54]. The LCA of two nodes d_u and d_v of a tree D, $\mathrm{LCA}(d_u, d_v)$, is defined as the lowest node in D (furthest from the root), which is an ancestor of both d_u and d_v [13]. For example, in Fig. 1, $\mathrm{LCA}(d_u, d_v) = 1$ if $d_u = 2$ and $d_v = 3$, which means that the LCA of *Car* and *Bus* is *vehicles*.

Example 6: In Fig. 1, let $D = \{6\}$ and $\hat{D} = \{7\}$. The LCA of *Chair* and *Sofa* is only the node *Seating*. Thus, based on LCA method, $D_{\text{aug}} = \{6, 5\}$ and $\hat{D}_{\text{aug}} = \{7, 5\}$. $P_{\text{LCA}} = 0.5$, $R_{\text{LCA}} = 0.5$, and $F_{\text{LCA}} = 0.5$. However, based on hierarchal method, $D_{\text{aug}} = \{6, 5, 4, 0\}$ and $\hat{D}_{\text{aug}} = \{7, 5, 4, 0\}$. $P_H = 0.75$, $R_H = 0.75$, and $F_H = 0.75$.

According to Example 6, redundant nodes can lead to fluctuations in P_{LCA}, R_{LCA}, and F_{LCA}. Thus they should be removed.

TABLE IV
DATA DESCRIPTION

No.	Datasets	Data type	$\|U\|$	$\|C\|$	$\|d\|$	Node	Leaf	Depth
1	*Bridges* [54]	Num&Sym	108	12	6	7	6	2
2	*SAIAPR* [55]	Image	99526	512	256	256	200	5
3	*VOC* [56]	Image	12283	1000	20	30	20	4
4	*News20* [57]	Text	18846	26214	20	27	20	3

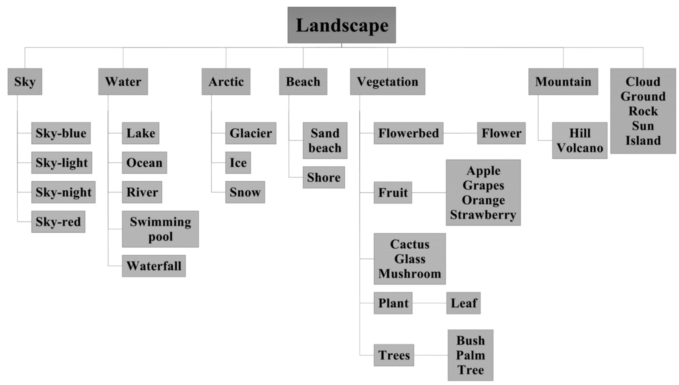

Fig. 4. Hierarchy of landscape branch of the *SAIAPR* dataset.

VI. EXPERIMENTAL ANALYSIS

In this section, we first introduce four datasets used in our experiments. We then compare the proposed hierarchical feature selection with the flat feature selection proposed in [28]. All the numerical experiments are implemented in MATLAB R2014b and executed on an Intel Core i7-3770 running at 3.40 GHz with 16.0 GB memory and a 64-bit Windows 7 operating system. We select the feature subsets on the training sets and test them on the test sets using an SVM, a KNN, and NB classifiers, respectively. For the SVM classifier, ten-fold cross-validation is performed using a linear kernel and $c = 1$. For the KNN classifier, we set parameter $k = 5$ for the class decision based on the preliminary experiments.

A. Datasets

Four datasets are used in the experiments. Basic statistics for these datasets are provided in Table IV.

The first dataset is *Bridges* that is from the University of California-Irvine library [55].

The second dataset is *SAIAPR*, which is an extension of IAPR TC-12 collection. Each image has been manually segmented and the resultant regions have been annotated according to a predefined vocabulary of labels; the vocabulary is organized according to a hierarchy of concepts. According to [56], an object can be in one of six main branches: "animal," "landscape," "man-made," "human," "food," or "other." Fig. 4 shows the "landscape" branch of the hierarchy.

We use portions of the samples (1000, 5000, and 10 000) as a training set to select the feature subset, and use 5000, 10 000, and all samples as the test set to evaluate the effectiveness of the selected feature subset. According to Algorithm 1, 41 features are first selected from 512 features in three training sets containing 1000, 5000, and 10 000 samples, respectively; these features share some attributes. The number of shared attributes is listed in Table V. For example, the feature subset selected from 5000 samples has 32 features that are identical to those in the feature subset selected from 10 000 samples. The running time when using 5000, 10 000, and all samples to test the 41 features selected in different subsets are 53, 190, and 13 500 s, respectively. This demonstrates that using a portion of the samples to approximate the dependence coefficient of the samples can essentially reduce the running time.

The results of flat SVM classification accuracy using different sample subsets listed in Table VI confirm that it is not necessary to use all samples to select features. In this study, we use 5000 samples to select a feature subset under the basic premise of not affecting the classification accuracy.

The third dataset is PASCAL VOC, which is a benchmark in visual object category recognition and detection that provides the vision and machine learning communities with a standard dataset of images and annotations [57]. Fig. 5 shows the hierarchy of VOC. In Table IV, there are 7178 samples for the training dataset and 5105 samples for the testing dataset of PASCAL VOC [57].

Finally, the fourth dataset is *News20* corpus, which was collected and originally used for document classification by Lang [58]. This dataset includes 18 446 messages collected from 20 different Netnews newsgroups. One thousand messages from each of the 20 newsgroups were chosen at random and partitioned by newsgroup name. The list of newsgroups from which the messages were chosen is shown in Fig. 6. We use the "by-date" version, which contains 951 documents evenly distributed across 20 classes. After stemming and stop word removal, this corpus contains 26 214 distinct terms [59].

TABLE V
NUMBER OF SHARING ATTRIBUTES

	1,000	5,000	10,000
1,000	41		
5,000	32	41	
10,000	35	32	41

TABLE VI
FLAT CLASSIFICATION ACCURACY (SVM)

SVM	1,000	5,000	10,000
All Samples	21.40±0.08	21.28±0.18	21.42±0.20
10,000 Samples	20.57±0.74	20.43±0.57	20.71±0.64
5,000 Samples	19.51±1.68	19.17±1.53	19.47±1.41

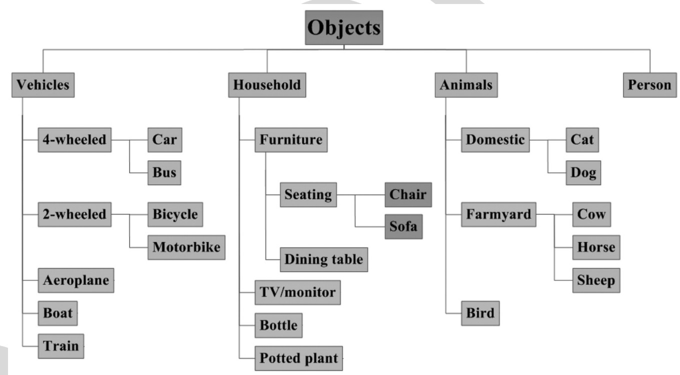

Fig. 5. Hierarchy of the VOC dataset.

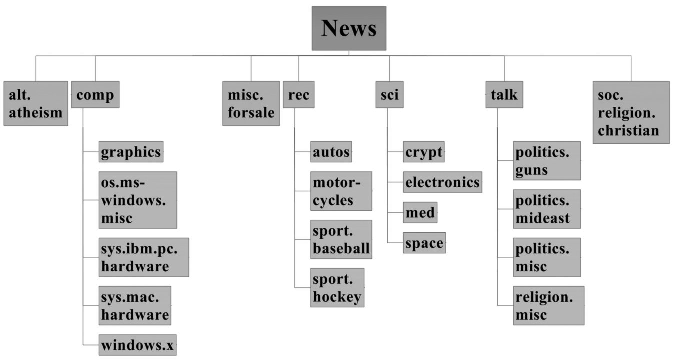

Fig. 6. Hierarchy of the *News20* dataset.

TABLE VII
FLAT EVALUATION ON DIFFERENT DATASETS

(a) *Bridges*

Percentage	SVM		KNN (k=5)		NB	
	Flat	Hierarchical	Flat	Hierarchical	Flat	Hierarchical
100.00%	63.14		56.41		65.56	
63.64%	59.23	**65.95**	52.85	**59.24**	63.80	**64.80**
54.55%	54.74	**62.14**	56.33	**57.09**	**64.45**	61.17
18.18%	53.91	**55.65**	53.74	**53.91**	**57.50**	53.91

(b) *SAIAPR*

Percentage	SVM		KNN (k=5)		NB	
	Flat	Hierarchical	Flat	Hierarchical	Flat	Hierarchical
100.00%	21.10		19.83		6.32	
19.00%	15.28	**20.58**	17.76	**19.48**	**7.34**	6.40
8.01%	14.31	**19.60**	17.28	**18.90**	5.38	**7.42**
4.10%	10.70	**16.84**	10.42	**17.18**	6.32	**7.54**

(c) *VOC*

Percentage	SVM		KNN (k=5)		NB	
	Flat	Hierarchical	Flat	Hierarchical	Flat	Hierarchical
100.00%	48.07		38.72		27.09	
22.00%	38.04	**44.23**	27.44	**37.06**	15.61	**27.09**
6.00%	33.53	**39.51**	24.64	**36.12**	15.05	**29.38**
2.00%	30.30	**34.14**	21.80	**32.14**	15.22	**30.50**

(d) *News20*

Percentage	SVM		KNN (k=5)		NB	
	Flat	Hierarchical	Flat	Hierarchical	Flat	Hierarchical
100.00%	49.60		7.25		31.43	
1.91%	—	**40.03**	—	**20.19**	—	**32.15**
0.15%	**25.32**	23.74	**21.33**	15.75	**27.21**	22.89
0.11%	21.65	**22.79**	**19.02**	17.43	22.81	**22.87**

B. Flat Evaluation

The performance evaluation measures of previous learning algorithms are those commonly used to describe the classification accuracy of SVM, KNN, and NB methods. We refer to these measures as flat evaluations because they do not consider the hierarchical classes. We first use classification accuracy listed in Table VII to visually compare the results of the proposed algorithm with those from a flat algorithm on different datasets. The best performance on each measure is highlighted in bold.

From Table VII, we can identify the changes in accuracy with different numbers of selected features. We can also observe that the performance of the features selected by the hierarchical method is better than that of the flat method. In Table VII(a), it is clear that using 63.64% of features gives better performance than using all features on SVM and KNN classifiers. This means that we can obtain a set of representative features using only the sibling samples. These results prove the effectiveness of the hierarchical selection method proposed in this paper.

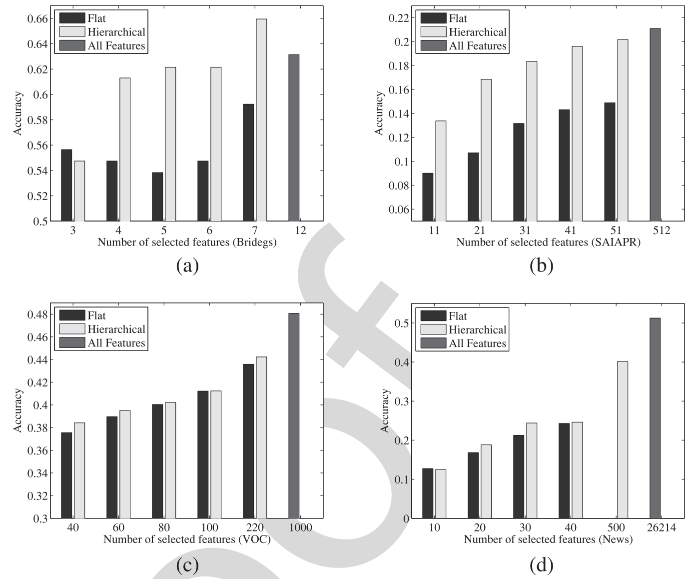

Fig. 7. Comparison of accuracy between flat and hierarchical strategies. (a) *Bridges*. (b) *SAIAPR*. (c) VOC. (d) *News20*.

There are 26 214 features in the *News20* dataset. The flat feature selection method takes almost three hours to select a feature. It could not output its results within several days when we select 500 features (1.91%× 26 214). Thus, we use "—" to denote this condition in Table VII. In addition, from Table VII, we can observe that the performance of KNN is not great. The dataset of *News20* is relatively sparse and may be inherently difficult to learn, as evidenced by the relatively poor performance with all features. The accuracy of KNN is only 7.25% when all features have been selected. Thus, KNN is not suitable for this dataset. The accuracy of SVM classification is 40.03% when we select 1.91% of features using the hierarchy method.

Fig. 7 compares the accuracy of SVM between flat and hierarchical strategies on different datasets. The results of the experiments show that our algorithm performs well with different numbers of condition attributes.

C. Hierarchical Evaluation

We use SVM to evaluate our algorithm because the usual measure of performance for such classifiers is the accuracy rate. However, in hierarchical application problems, the output of the hierarchical algorithm is part of the hierarchical classes, which is different from the case of flat classes. Thus, we also use hierarchical evaluation to evaluate the performance of our algorithm. Table VIII presents the results of the hierarchical and flat algorithms on different datasets evaluated by the TIE, Hierarchical F_1, and LCA F_1 measures.

We use TIE to consider some different errors caused by the hierarchy. The "↓" after TIE indicates "the smaller the better." Hierarchical F_1 and LCA F_1 are set-based measures. The "↑" after the set-based measures indicates "the larger the better." We describe the results of these three measures on four datasets in Table VIII. In terms of effectiveness, hierarchical feature selection gives better performance than that of flat feature selection.

TABLE VIII
HIERARCHICAL EVALUATION ON DIFFERENT DATASETS

(a) *Bridges*

Percentage	TIE ↓		Hierarchical F_1 ↑		LCA F_1 ↑	
	Flat	Hierarchical	Flat	Hierarchical	Flat	Hierarchical
100%		10.3		0.81		0.79
63.64%	11.7	**9.5**	0.79	**0.83**	0.77	**0.81**
54.55%	12.6	**10.9**	0.78	**0.81**	0.75	**0.79**
18.18%	12.9	**12.3**	0.78	**0.79**	0.74	**0.75**

(b) *SAIAPR*

Percentage	TIE ↓		Hierarchical F_1 ↑		LCA F_1 ↑	
	Flat	Hierarchical	Flat	Hierarchical	Flat	Hierarchical
100%		1748		0.55		0.48
19.0%	2146	**1768**	0.46	**0.54**	0.42	**0.48**
8.01%	2199	**1807**	0.45	**0.53**	0.41	**0.47**
4.10%	1913	**1885**	0.47	**0.51**	0.41	**0.45**

(c) *VOC*

Percentage	TIE ↓		Hierarchical F_1 ↑		LCA F_1 ↑	
	Flat	Hierarchical	Flat	Hierarchical	Flat	Hierarchical
100%		962		0.70		0.67
22.0%	1244	**1033**	0.65	**0.67**	0.60	**0.65**
6.00%	1347	**1126**	0.62	**0.63**	0.57	**0.61**
2.00%	1447	**1237**	**0.60**	0.58	0.54	**0.58**

(d) *News20*

Percentage	TIE ↓		Hierarchical F_1 ↑		LCA F_1 ↑	
	Flat	Hierarchical	Flat	Hierarchical	Flat	Hierarchical
100%		152		0.72		0.70
1.91%	—	**186**	—	**0.66**	—	**0.63**
0.15%	238	**231**	**0.57**	**0.57**	**0.54**	**0.54**
0.11%	250	**233**	0.55	**0.56**	0.52	**0.53**

TABLE IX
AVERAGE NUMBER OF SAMPLES IN THE SEARCH SPACE

Dataset	Instances	Flat	Hierarchical
Bridges	108	82 (75.9%)	30 (27.8%)
SAIAPR	99526	97542 (98.0%)	6473 (6.5%)
VOC	7178	6503 (90.6%)	276 (3.9%)
News20	11314	10743 (95.0%)	1774 (15.7%)

The results demonstrate that our algorithm provides an efficient solution to finding a better subset of the features.

In terms of the three measures in Table VIII, we observe the following:

1) The value of TIE is related to the scale of the hierarchical structure of classes.
2) The value of LCA F_1 is less than that of Hierarchical F_1. This is because having many common ancestors tends to overpenalize errors. LCA F_1 can avoid this type of error.
3) These three measures for the quantitative hierarchical comparison results are consistent with the flat comparison results.

D. Comparison of Efficiencies Between Flat and Hierarchical Strategies

We now study the computational complexity of the flat and hierarchical strategies. Table IX lists the average number of samples in the search space when we compute the lower and upper approximations.

For example, there are 7178 samples in VOC training dataset. The flat feature selection algorithm requires 6503 computations to select one feature. This is 90.6% of the size of VOC training dataset. In contrast, the hierarchical strategy can select one feature from only 276 computations, which is only 3.9% of all the samples. The computational load is reduced from 98.0% to 6.5% on *SAIAPR*. *SAIAPR* has 256 classes, and the sibling strategy is an effective method for datasets with more classes. These statistics lead us to the conclusion that the hierarchical strategy clearly reduces the computational complexity. Example 7 gives an intuitive understanding of the search space of the sibling strategy.

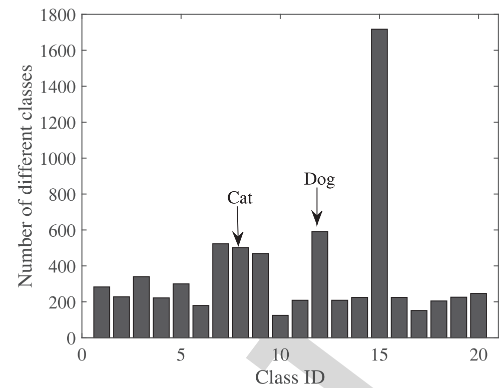

Fig. 8. Number of different classes in VOC dataset.

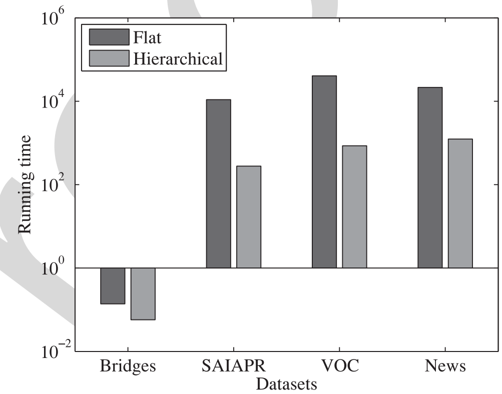

Fig. 9. Running time comparison of the first feature selection between flat and hierarchical strategies.

Example 7: Fig. 5 shows a hierarchical structure of 20 classes. The *Dog* and *Cat* classes have a sibling relationship in this hierarchical structure. Fig. 8 shows the number of different classes in VOC training dataset. Using the exclusive strategy, the negative samples of a *Cat* are all non-*Cat* samples. In contrast, when we use the sibling strategy, the negative samples of a *Cat* are *Dog* samples.

We compare the efficiency of the flat algorithm and hierarchical algorithm. The running times for selecting the first feature for both algorithms are shown in Fig. 9, where the unit of the running time is second. From the results, we note that the hierarchical algorithm is an efficient algorithm in terms of the running time.

The deleting strategy works well on large datasets. Table X shows the comparison of running time used in selecting the first feature and selecting other features. The running time for selecting the first feature is 278.35s on *SAIAPR*. There is a significant reduction from 278.35 to 41.43s.

TABLE X
RUNNING TIME (S)

Dataset	First	Average	Percentage
Bridges	0.058	0.011	28.8%
SAIAPR	278.35	41.43	14.9%
VOC	857.17	309.00	36.1%
News20	1238.65	410.56	33.2%

VII. CONCLUSIONS AND FUTURE WORK

We have proposed a fuzzy rough set based feature selection algorithm for large-scale hierarchical classification. Based on the complicated data structure of modern datasets, we proposed a hierarchical feature selection method by considering the sibling strategy. We used the sibling nodes as the nearest samples from different classes to compute the fuzzy lower approximation and evaluate the features. Two accelerating strategies were employed in the proposed algorithm. In addition, flat and hierarchical evaluations were used to evaluate the effectiveness of the algorithm. Our advantage in terms of practical application is that we control the error rate artificially using the given hierarchical class structure. Experimental results indicate the efficiency and effectiveness of the proposed algorithm. In particular, the proposed algorithm improves the classification performance by selecting the most relevant feature subset. In summary, this study suggests new research trends concerning fuzzy rough sets and hierarchical feature selection problems.

The current implementation of the algorithm just considers tree structures of class labels. In fact, there are other complex structures in practices, such as directed acyclic graphs [18] and chain structures [60]. In the future, we will discuss feature selection algorithms for such tasks. In addition, the proposed algorithm just selects some informative features from the original set. However, discriminant information sometimes hides in the lower-dimensional combination of the high-dimensional features, where feature mapping or feature extraction is preferred. However, the proposed algorithm cannot achieve this objective. We are going to design techniques for hierarchical feature extraction in the future.

REFERENCES

[1] S. Dumais and H. Chen, "Hierarchical classification of web content," in *Proc. 23rd Annu. Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, 2000, pp. 256–263.

[2] J. Deng, W. Dong, R. Socher, L. J. Li, K. Li, and L. Fei, "Imagenet: A large-scale hierarchical image database," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2009, pp. 248–255.

[3] J. Struyf, S. Džeroski, H. Blockeel, and A. Clare, *Hierarchical Multi-Classification With Predictive Clustering Trees in Functional Genomics*. New York, NY, USA: Springer, 2005.

[4] Z. L. Cai and W. Zhu, "Multi-label feature selection via feature manifold learning and sparsity regularization," *Int. J. Mach. Learn. Cybern.*, vol. 9, no. 8, pp. 1321–1334, 2018.

[5] J. H. Dai, B. jie Wei, X. H. Zhang, and Q. L. Zhang, "Uncertainty measurement for incomplete interval-valued information systems based on α-weak similarity," *Knowl.-Based Syst.*, vol. 136, pp. 159–171, 2017.

[6] S. P. Wang and W. Zhu, "Sparse graph embedding unsupervised feature selection," *IEEE Trans. Syst. Man Cybern. Syst.*, vol. 48, no. 3, pp. 329–341, Mar. 2018.

[7] X. D. Wu, X. Q. Zhu, G. Q. Wu, and W. Ding, "Data mining with big data," *IEEE Trans. Knowl. Data Eng.*, vol. 26, no. 1, pp. 97–107, Jan. 2014.

[8] Y. L. Chen, H. W. Hu, and K. Tang, "Constructing a decision tree from data with hierarchical class labels," *Expert Syst. Appl.*, vol. 36, no. 3, pp. 4838–4847, 2009.

[9] S. Gopal and Y. M. Yang, "Hierarchical Bayesian inference and recursive regularization for large-scale classification," *ACM Trans. Knowl. Discovery Data*, vol. 9, no. 3, pp. 1–23, 2015.

[10] M. Ashburner *et al.*, "Gene ontology: Tool for the unification of biology," *Nature Genetics*, vol. 25, no. 1, pp. 25–29, 2000.

[11] N. Nourani-Vatani, R. López-Sastre, and S. Williams, "Structured output prediction with hierarchical loss functions for seafloor imagery taxonomic categorization," in *Pattern Recognition and Image Analysis*. New York, NY, USA: Springer, 2015, pp. 173–183.

[12] A. X. Sun and E. P. Lim, "Hierarchical text classification and evaluation," in *Proc. IEEE Int. Conf. Data Mining*, 2001, pp. 521–528.

[13] A. Kosmopoulos, I. Partalas, E. Gaussier, G. Paliouras, and I. Androutsopoulos, "Evaluation measures for hierarchical classification: A unified view and novel approaches," *Data Mining Knowl. Discovery*, vol. 29, no. 3, pp. 820–865, 2015.

[14] A. Kosmopoulos, E. Gaussier, G. Paliouras, and S. Aseervatham, "The ECIR 2010 large scale hierarchical classification workshop," *ACM SIGIR Forum*, vol. 44, no. 1, pp. 23–32, 2010.

[15] J. Deng, S. Satheesh, A. C. Berg, and L. Fei, "Fast and balanced: Efficient label tree learning for large scale object recognition," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2011, pp. 567–575.

[16] C. Freeman, "Feature selection and hierarchical classifier design with applications to human motion recognition," Ph.D. dissertation, Dept. Elect. Comput. Eng., Univ. Waterloo, Waterloo, ON, Canada, 2014.

[17] J. Deng, A. C. Berg, K. Li, and L. Fei, "What does classifying more than 10,000 image categories tell us?" in *Proc. Eur. Conf. Comput. Vis.*, 2010, pp. 71–84.

[18] B. Wei and J. T. Kwok, "Multi-label classification on tree- and dag-structured hierarchies," in *Proc. Int. Conf. Mach. Learn.*, 2011, pp. 17–24.

[19] S. Zhao, Y. Han, Q. Zou, and Q. Hu, "Hierarchical support vector machine based structural classification with fused hierarchies," *Neurocomputing*, vol. 214, pp. 86–92, 2016.

[20] J. P. Fan, J. Zhang, K. Z. Mei, J. Y. Peng, and L. Gao, "Cost-sensitive learning of hierarchical tree classifiers for large-scale image classification and novel category detection," *Pattern Recognit.*, vol. 48, no. 5, pp. 1673–1687, 2015.

[21] C. Freeman, D. Kulić, and O. Basir, "Feature-selected tree-based classification," *IEEE Trans. Cybern.*, vol. 43, no. 6, pp. 1990–2004, Dec. 2013.

[22] C. Freeman, D. Kulić, and O. Basir, "Joint feature selection and hierarchical classifier design," in *Proc. Int. Conf. Syst., Man, Cybern.*, 2011, pp. 1728–1734.

[23] J. Song, P. Zhang, S. Qin, and J. Gong, "A method of the feature selection in hierarchical text classification based on the category discrimination and position information," in *Proc. Int. Conf. Ind. Informat.-Comput. Technol., Intell. Technol., Ind. Inf. Integration*, 2015, pp. 132–135.

[24] R. Jensen and N. Mac Parthal áin, "Towards scalable fuzzy–rough feature selection," *Inf. Sci.*, vol. 323, pp. 1–15, 2015.

[25] J. H. Li, Y. Ren, C. L. Mei, Y. H. Qian, and X. B. Yang, "A comparative study of multigranulation rough sets and concept lattices via rule acquisition," *Knowl.-Based Syst.*, vol. 91, pp. 152–164, 2016.

[26] C. Z. Wang *et al.*, "A fitting model for feature selection with fuzzy rough sets," *IEEE Trans. Fuzzy Syst.*, vol. 25, no. 4, pp. 741–753, Aug. 2017.

[27] D. G. Chen and S. Y. Zhao, "Local reduction of decision system with fuzzy rough sets," *Fuzzy Sets Syst.*, vol. 161, no. 13, pp. 1871–1883, 2010.

[28] Q. H. Hu, D. R. Yu, W. Pedrycz, and D. G. Chen, "Kernelized fuzzy rough sets and their applications," *IEEE Trans. Knowl. Data Eng.*, vol. 23, no. 11, pp. 1649–1667, Nov. 2011.

[29] J. H. Dai, H. Hu, W. Z. Wu, Y. H. Qian, and D. B. Huang, "Maximal discernibility pairs based approach to attribute reduction in fuzzy rough sets," *IEEE Trans. Fuzzy Syst.*, vol. 26, no. 4, pp. 2174–2187, Aug. 2018.

[30] S. Y. Zhao, H. Chen, C. P. Li, M. Y. Zhai, and X. Y. Du, "RFRR: Robust fuzzy rough reduction," *IEEE Trans. Fuzzy Syst.*, vol. 21, no. 5, pp. 825–841, Oct. 2013.

[31] D. G. Chen and Y. Y. Yang, "Attribute reduction for heterogeneous data based on the combination of classical and fuzzy rough set models," *IEEE Trans. Fuzzy Syst.*, vol. 22, no. 5, pp. 1325–1334, Oct. 2014.

[32] H. M. Chen, T. R. Li, C. Luo, S. J. Horng, and G. Y. Wang, "A decision-theoretic rough set approach for dynamic data mining," *IEEE Trans. Fuzzy Syst.*, vol. 23, no. 6, pp. 1958–1970, Dec. 2015.

[33] E. Ramentol *et al.*, "IFROWANN: Imbalanced fuzzy-rough ordered weighted average nearest neighbor classification," *IEEE Trans. Fuzzy Syst.*, vol. 23, no. 5, pp. 1622–1637, Oct. 2015.

[34] W. H. Xu, Q. R. Wang, and X. T. Zhang, “Multi-granulation fuzzy rough sets in a fuzzy tolerance approximation space,” *Int. J. Fuzzy Syst.*, vol. 13, no. 4, pp. 246–259, 2011.

[35] W. Z. Wu and Y. Leung, “Theory and applications of granular labelled partitions in multi-scale decision tables,” *Inf. Sci.*, vol. 181, no. 18, pp. 3878–3897, 2011.

[36] X. D. Yue, Y. F. Chen, D. Q. Miao, and J. Qian, “Tri-partition neighborhood covering reduction for robust classification,” *Int. J. Approximate Reasoning*, vol. 83, pp. 371–384, 2017.

[37] C. Z. Wang, Q. H. Hu, X. Z. Wang, D. G. Chen, Y. H. Qian, and D. Zhe, “Feature selection based on neighborhood discrimination index,” *IEEE Trans. Neural Netw. Learning Syst.*, vol. 29, no. 7, pp. 2986–2999, Jul. 2018.

[38] W. Z. Wu, Y. Qian, T. J. Li, and S. M. Gu, “On rule acquisition in incomplete multi-scale decision tables,” *Inf. Sci.*, vol. 378, no. C, pp. 282–302, 2016.

[39] Z. Pawlak, *Rough Sets: Theoretical Aspects of Reasoning About Data*. Boston, MA, USA: Kluwer, 1991.

[40] W. Zhu, “Relationship among basic concepts in covering-based rough sets,” *Inf. Sci.*, vol. 179, no. 14, pp. 2478–2486, 2009.

[41] D. Dubois and H. Prade, “Putting rough sets and fuzzy sets together,” in *Intelligent Decision Support*. New York, NY, USA: Springer, 1992, pp. 203–232.

[42] L. D’eer, N. Verbiest, C. Cornelis, and L. Godo, “A comprehensive study of implicator-conjunctor based and noise-tolerant fuzzy rough sets: Definitions, properties and robustness analysis,” *Fuzzy Sets Syst.*, vol. 275, pp. 1–38, 2015.

[43] D. S. Yeung, D. G. Chen, E. C. Tsang, J. W. Lee, and W. Xi Zhao, “On the generalization of fuzzy rough sets,” *IEEE Trans. Fuzzy Syst.*, vol. 13, no. 3, pp. 343–361, Jun. 2005.

[44] S. Vluymans, D. S. Tarragó, Y. Saeys, C. Cornelis, and F. Herrera, “Fuzzy rough classifiers for class imbalanced multi-instance data,” *Pattern Recognit.*, vol. 53, pp. 36–45, 2016.

[45] S. Vluymans, C. Cornelis, F. Herrera, and Y. Saeys, “Multi-label classification using a fuzzy rough neighborhood consensus,” *Inf. Sci.*, vols. 433–434, pp. 96–114, 2018.

[46] R. Eisner, B. Poulin, D. Szafron, P. Lu, and R. Greiner, “Improving protein function prediction using the hierarchical structure of the gene ontology,” in *Proc. IEEE Symp. Comput. Intell. Bioinformat. Comput. Biol.*, 2005, pp. 1–10.

[47] M. Ceci and D. Malerba, “Classifying web documents in a hierarchy of categories: A comprehensive study,” *J. Intell. Inf. Syst.*, vol. 28, no. 1, pp. 37–78, 2007.

[48] C. N. Silla and A. A. Freitas, “A survey of hierarchical classification across different application domains,” *Data Mining Knowl. Discovery*, vol. 22, no. 1/2, pp. 31–72, 2011.

[49] Q. H. Hu, L. Zhang, S. An, D. Zhang, and D. R. Yu, “On robust fuzzy rough set models,” *IEEE Trans. Fuzzy Syst.*, vol. 20, no. 4, pp. 636–651, Aug. 2012.

[50] A. Çakmak Pehlivanlı, “A novel feature selection scheme for high-dimensional data sets: Four-staged feature selection,” *J. Appl. Statist.*, vol. 43, pp. 1140–1154, 2014.

[51] I. Guyon and A. Elisseeff, “An introduction to variable and feature selection,” *J. Mach. Learn. Res.*, vol. 3, pp. 1157–1182, 2003.

[52] O. Dekel, J. Keshet, and Y. Singer, “Large margin hierarchical classification,” in *Proc. Int. Conf. Mach. Learn.*, 2004, pp. 1–8.

[53] L. Cai and T. Hofmann, “Exploiting known taxonomies in learning overlapping concepts,” in *Proc. Int. Joint Conf. Artif. Intell.*, 2007, pp. 714–719.

[54] A. V. Aho, J. E. Hopcroft, and J. D. Ullman, “On finding lowest common ancestors in trees,” *SIAM J. Comput.*, vol. 5, no. 1, pp. 115–132, 1976.

[55] C. L. Blake and C. J. Merz, “UCI repository of machine learning databases,” 1998. [Online]. Available: http://www.ics.uci.edu/∼mlearn/mlrepository.html

[56] H. J. Escalante *et al.*, “The segmented and annotated IAPR TC-12 benchmark,” *Comput. Vis. Image Understanding*, vol. 114, no. 4, pp. 419–428, 2010.

[57] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman, “The pascal visual object classes (VOC) challenge,” *Int. J. Comput. Vis.*, vol. 88, no. 2, pp. 303–338, 2010.

[58] K. Lang, “Newsweeder: Learning to filter netnews,” in *Proc. 20th Int. Conf. Mach. Learn.*, 1995, pp. 331–339.

[59] D. Cai, X. F. He, W. V. Zhang, and J. W. Han, “Regularized locality preserving indexing via spectral regression,” in *Proc. 16th ACM Conf. Inf. Knowl. Manage.*, 2007, pp. 741–750.

[60] J. Knorn, A. Rabe, V. C. Radeloff, T. Kuemmerle, J. Kozak, and P. Hostert, “Land cover mapping of large areas using chain classification of neighboring landsat satellite images,” *Remote Sens. Environ.*, vol. 113, no. 5, pp. 957–964, 2009.

Hong Zhao received the M.S. degree in computer application from Liaoning Normal University, Dalian, China, in 2006. She is currently a Ph.D. student in software engineering with the School of Computer Software, College of Intelligence and Computing, Tianjin University.

She is also a Professor with the School of Computer Science, Minnan Normal University, Zhangzhou, China. She has authored more than 40 journal and conference papers in the areas of granular computing based machine learning and cost-sensitive learning. Her research interests include rough sets, granular computing, and data mining for hierarchical classification.

Ping Wang received the B.S., M.S., and Ph.D. degrees in computer science from Tianjin University, Tianjin, China, in 1988, 1991, and 1998, respectively.

She is currently a Professor with the School of Mathematics, Tianjin University. She is also a Ph.D. Supervisor with the School of Computer Software, College of Intelligence and Computing, Tianjin University. Her research interests include image processing and machine learning.

Qinghua Hu (SM’13) received the B.S. and M.S. degrees in power engineering and received his Ph.D. degree in control science and engineering.

He was a Postdoctoral Fellow with the Department of Computing, Hong Kong Polytechnic University, Hong Kong, from 2009 to 2011. He is currently a Full Professor and the Vice Director of the College of Intelligence and Computing, Tianjin University, Tianjin, China. He has authored about 200 peer reviewed journal or conference papers in the areas of granular computing based machine learning, reasoning with uncertainty, pattern recognition, and fault diagnosis. His research interests include rough sets, granular computing, and data mining for classification and regression.

Prof. Hu was the Program Committee Co-Chair of the International Conference on Rough Sets and Current Trends in Computing in 2010, the Chinese Rough Set and Soft Computing Society in 2012 and 2014, and the International Conference on Rough Sets and Knowledge Technology, the International Conference on Machine Learning and Cybernetics in 2014, and the General Co-Chair of International Joint Conference on Rough Sets 2015. He is currently the PC-Co-Chairs of China Conference on Machine Learning (CCML) 2017 and Chinese Conference on Computer Vision 2017. He is currently an Associate Editor for the IEEE TRANSACTIONS ON FUZZY SYSTEMS.

Pengfei Zhu received the B.S. degree in power engineering, M.S. degree in power machinery and engineering, and Ph.D degree in computer vision.

He is currently an Associate Professor with the College of Intelligence and Computing, Tianjin University, Tianjin, China. He has authored or coauthored more than 30 papers in International Conference on Computer Vision, Conference on Computer Vision and Pattern Recognition, European Conference on Computer Vision, Association for the Advancement of Artificial Intelligence, International Joint Conferences on Artificial Intelligence, and the IEEE TRANSACTIONS ON INFORMATION FORENSICS AND SECURITY, and IEEE TRANSACTIONS ON IMAGE PROCESSING. His research interests include machine learning and computer vision.

Dr. Zhu is the Local Arrangement Chair for the International Joint Conference on Rough Sets 2015, and the Chinese Conference on Computer Vision 2017.