Applied Intelligence (2021) 52:11089–11102
https://doi.org/10.1007/s10489-021-03028-4

A fuzzy rough set approach to hierarchical feature selection based on Hausdorff distance

Zeyu Qiu[1,2,3] · **Hong Zhao**[1,2]

Accepted: 5 November 2021 / Published online: 20 January 2022

Abstract
With increases in feature dimensions and the emergence of hierarchical class structures, hierarchical feature selection has become an important data preprocessing step in machine learning. A variety of effective feature selection methods based on granular computing and hierarchical information have been proposed. The fuzzy rough set method is an effective granular computing method for dealing with uncertainty. However, it is time-consuming because the distance calculations are only based on single samples. In this paper, we propose a fuzzy rough set approach using the Hausdorff distance of the sample set for hierarchical feature selection. This integrates the benefits of sample granularity and class hierarchical granularity. Firstly, the general feature selection task is decomposed into coarse-grained and fine-grained tasks according to the hierarchical structure of the data's semantic labels. This allows a large and difficult classification task to be divided into several small and controllable subtasks. Then, the Hausdorff distance-based fuzzy rough set method is used to select the best feature subset in each coarse- and fine-grained subtask. Unlike single-sample-based distance calculation, Hausdorff distance calculation uses a sample set of different classes. The new model greatly reduces the computational complexity of classification. Finally, we use the top-down support vector machine classifier to experimentally verify the effectiveness of the proposed methods on five hierarchical datasets. Compared with five existing feature selection algorithms in terms of three evaluation metrics, the proposed method provides the highest average accuracy and much lower running time. In particular, on the F194 dataset, our method takes the least time to improve the F_H indicator by 2% compared with that of the second-best algorithm.

Keywords Granular computing · Hierarchical feature selection · Fuzzy rough set · Hierarchical classification

1 Introduction

Feature selection is a prevalent data preprocessing procedure in machine learning. It is mainly used to avoid dimensional disasters and reduce the difficulty of various complex tasks. In the current era of "big data", many datasets have semantic hierarchies [12]. For example, ImageNet is a hierarchical image dataset with 1,000 major classes and 20,000 minor classes [11]. In the process of feature selection, different levels of features can be selected according to the required granularity on a given dataset. For example, one can distinguish a horse from a dog based on contours; however, distinguishing different breeds of dogs requires observation of the characteristics of the eyes, nose, tail, hair, etc. In such cases, hierarchical feature selection is necessary. The fuzzy rough set is an advanced theory commonly involved in feature selection and its effectiveness has been proven by many successful applications in science and engineering [44]. It has gradually become a "hot spot" in the field of artificial intelligence in recent years. This paper applies the fuzzy rough set concept to hierarchical feature selection and aims to solve the dimensional-disaster and data-redundancy problems common to big data applications.

Hierarchical feature selection has been extensively studied. Ding et al. [14] exploited the language differences between numbered arguments and un-numbered arguments to construct a semantic role classifier for text classification based on hierarchical feature selection tactics. Ruvolo

✉ Hong Zhao
hongzhaocn@163.com

[1] School of Computer Science, Minnan Normal University, Zhangzhou, Fujian, 363000, China

[2] Key Laboratory of Data Science and Intelligence Application, Fujian Province University, Zhangzhou, Fujian, 363000, China

[3] Fujian Key Laboratory of Granular Computing and Application, Minnan Normal University, Zhangzhou, Fujian, 363000, China

et al. [30] combined learned feature selection with hierarchical temporal aggregation for audio classification. Cheng et al. [9] invented a valid hierarchical feature selection technique for various computer vision tasks based on image segmentation and superpixel extraction. Feature selection methods that use fuzzy rough set have also become a research focus. Jensen et al. [18] proposed an approach of feature selection via the fuzzy rough set theory based on fuzzy similarity relations, which greatly reduces dimensionality while retaining classification accuracy. Hu et al. [17] creatively applied kernel functions to fuzzy rough set models and defined two kinds of kernelized fuzzy rough set. In addition, Zhao et al. [43] released an improved feature selection method dependent on sibling nodes for classification by fuzzy rough set, which is more advanced than flat feature selection. However, hierarchical feature selection tasks and fuzzy rough set methods are both time-consuming. Hence, combining them in a practical way is desirable but remains challenging.

Many things in the real world have levels, such as a government's hierarchical management of a country. This hierarchical thinking has brought great benefits to human society, and computers can gain similar benefits from hierarchical processes. This paper proposes a fuzzy rough set model for hierarchical feature selection using a semantic hierarchical data structure and fuzzy rough set knowledge. Multiple feature selection tasks are completed through layer-by-layer refinement of tasks under the premise of maintaining the feature selection's original goal. Then, we calculate the similarity of different classes according to Hausdorff distance to approximate decisions for a single task. Also, a threshold T is introduced to reduce the calculation time required to select the best feature subset. On the one hand, the hierarchical method allows the classifier to classify level-by-level, giving the traditional machine-learning algorithm better performance. On the other hand, using Hausdorff distance based on the sample set, rather than a Gaussian kernel based only on a single sample for fuzzy rough calculations, dramatically reduces the computational complexity. Our proposed algorithm can achieve excellent results with highly-dimensional multi-sample data. This reduces the occurrence of "dimensional disasters" and achieves the best classification performance.

In summary, our work has the following four main points: (1) A hierarchical feature selection algorithm built on fuzzy rough set is first proposed for dimension-reduction tasks. (2) We apply Hausdorff distance to measure the similarity among heterogeneous samples in fuzzy rough calculation processes. (3) We describe a strategy that introduces a threshold T to improve the efficiency of fuzzy rough calculations. (4) Finally, experiments are used to demonstrate the superiority of our algorithm in hierarchical feature selection tasks compared with the most recent existing algorithms. As far as we know, this is one of the few studies dedicated to reducing the time complexity of fuzzy rough calculations to have achieved significant results.

The remaining arrangements are as below. Section 2 provides a brief overview of fuzzy rough set. Section 3 describes our hierarchical feature selection methodology in detail. In Section 4, the effectiveness and superiority of our algorithm are verified via three groups of experiments. In conclusion, a summary and future research directions are provided in Section 5.

2 Preliminaries

In this section, we briefly review current knowledge of fuzzy rough sets. Rough set theory was originally proposed by Pawlak [26]. It is a tool for handling problems of incomplete and uncertain knowledge. However, the development and application of rough set theory have been limited by their strict equivalence. In order to deal with the limitations of rough sets, the fuzzy rough set theory was proposed by Dubois [15]. The fuzzy rough sets can solve problems involving fuzzy and uncertain knowledge, and has been successfully applied to feature selection [36], fuzzy decisions [3], fuzzy recognition [35], fuzzy clustering [42], and fuzzy control [32].

Let U be the set of objects, and R on $U \times U$ be a fuzzy relationship on U. For a fuzzy relationship R on U: (1) Said R is reflexive, if $R(x, x) = 1, (\forall x \in U)$. (2) Said R is symmetric, if $R(x, y) = R(y, x), (\forall x, y \in U)$. (3) Said R is transitive, if $R(x, z) \geq R(x, y) \wedge R(y, z), (\forall x, y, z \in U)$. If R meets the above conditions, then R is the fuzzy equivalent relationship.

Definition 1 A fuzzy information system is a tuple $\langle U, R\rangle$, where R is reflexive and a fuzzy relationship on U. If R is a fuzzy equivalence relationship [39], then the tuple $\langle U, R\rangle$ is a fuzzy equivalence relationship information system.

Definition 2 Let $\langle U, R\rangle$ be a fuzzy equivalence relationship information system, where R is a fuzzy equivalence relation on the universe U. $\forall X \in F(U)$, the lower approximation $\underline{R}X$ and upper approximation $\overline{R}X$ of X in space $\langle U, R\rangle$ are a bunch of fuzzy sets on U, and the membership function is [15]:

$$\underline{R}X(x) = \inf_{y \in U} \max(X(y), 1 - R(x, y)), x \in U, \tag{1}$$

$$\overline{R}X(x) = \sup_{y \in U} \min(X(y), R(x, y)), x \in U, \tag{2}$$

where $X(y)$ is the membership degree of element y belonging to fuzzy set X, and its value indicates the degree to which y belongs to fuzzy set X.

3 Hierarchical feature selection based on fuzzy rough set

We describe the overall model framework in Section 3.1 and elaborate on its two parts in Sections 3.2 and 3.3. These subsections describe our model comprehensively.

3.1 Model overview

The proposed model can be clearly described in two phases, as shown in Fig. 1. The first phase is the process of granulating tasks and data. The second phase involves feature selection using the improved fuzzy rough set algorithm for each task separately.

The first phase decomposes the general task of feature selection into multiple subtasks according to the semantic hierarchical structure of data labels. A semantic hierarchical structure is a tree structure constructed according to the semantic labels in the data. For example, *huskies* belong to both the *dog* and *mammal* categories, which are two different levels of category. This structure is generally constructed by generating new labels after clustering the data. There are many published datasets with semantic structures [13, 28]. We divide the subtasks of the data according to the given class hierarchical semantic structure. First, the root node is used as a separate subtask to perform the feature selection process. Then, each non-leaf node in the next layer is also regarded as an independent subtask, and so on, until all nodes in the next layer are leaf nodes.

Taking the Bridge dataset as an example [4], the process of the first phase is shown in the top half of Fig. 1. Our goal is to find the Bridge dataset's smallest feature subset in a general task, which can classify samples into six classes (*Wood*, *Suspen*, *Arch*, *Cantilv*, *Cont-t*, and *Simple*). Moreover, there is a basis for three classes (*Cantilv*, *Cont-t*, and *Simple*) to be clustered to as a *Truss* class semantically, because they have some common characteristics. Subsequently, the general task can be decomposed into two subtasks. Subtask 1 is to select features from the Bridge dataset to classify data into four classes (*Wood*, *Suspen*, *Arch*, and *Truss*). Subtask 2 is to select features from the *Truss* dataset to continue classifying into three classes (*Cantilv*, *Cont-t*, and *Simple*).

In the second phase, the feature selection algorithm completes each subtask independently using fuzzy rough calculations. These subtasks can be performed simultaneously on different threads. The fuzzy rough calculation process is shown in the second half of Fig. 1. The input data for each subtask is a subset $U^{(k)}$, where U is the original dataset, k represents the k^{th} subtask, and $U^{(k)}$ is the dataset of the

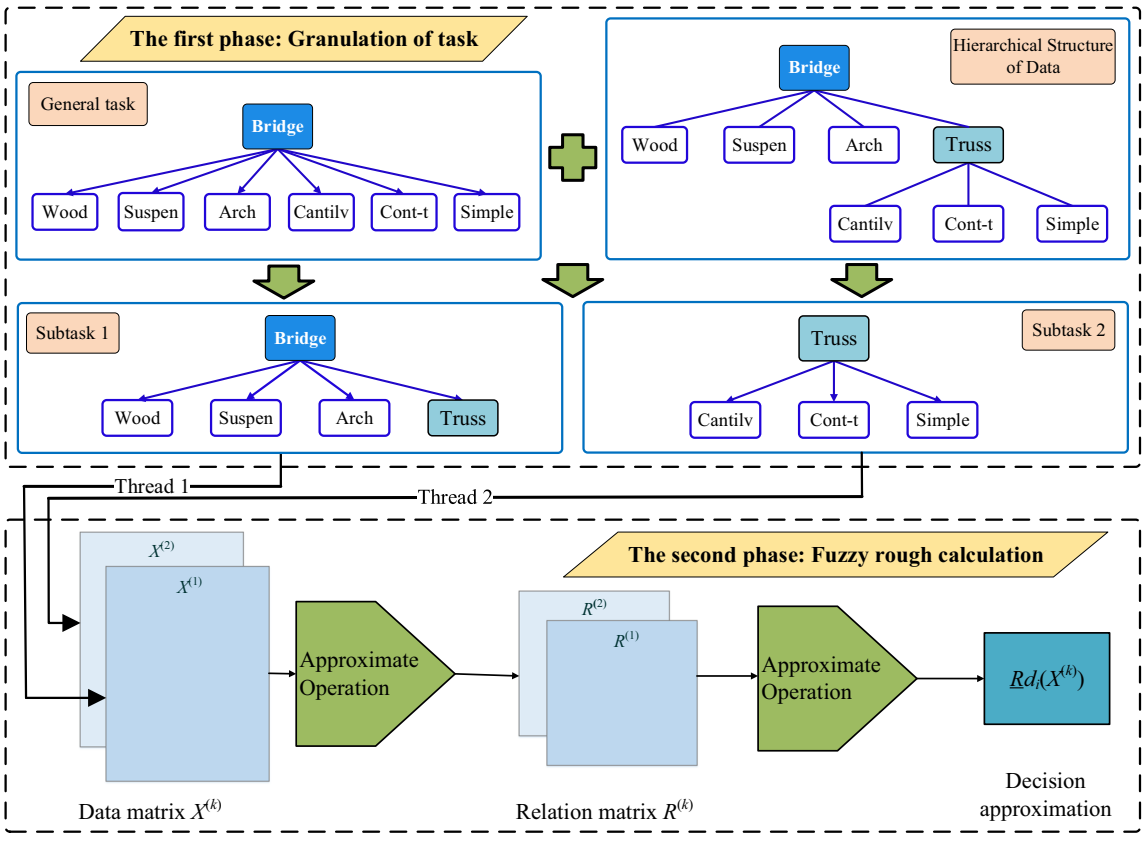

Fig. 1 Model framework

k^{th} subtask. Then, the similarity between any two classes on a subtask is calculated using the Hausdorff distance to form a relation matrix $R^{(k)}$, which is symmetric. By calculating $R^{(k)}$, we can then obtain the lower approximate membership degree of each sample set on the given feature set. Finally, the optimal feature subset is screened using the lower approximation concept of fuzzy rough sets to approximate the decision. However, the complexity of fuzzy rough calculations has always been a weakness of rough set theory. This paper provides some improved strategies that can vastly decrease calculation time.

3.2 Granulation of task and data

Before feature selection, we granulate the task according to the tree structure. For a complete task, we construct coarse-grained tasks at each non-leaf node of the tree structure. In this way, our task is to perform feature selection on coarse-grained tasks first and then on fine-grained tasks.

According to the class hierarchical tree structure of the dataset, feature selection is carried out from top to bottom. At the beginning of a task, the root node's task is assigned to its child nodes, and each child node continues to assign its task to its own child nodes until its next node is a leaf node. Because the dataset of the upper task contains the dataset of the lower task, our task is proceeds from coarse-grained to fine-grained. We call the upper task the *coarse-grained task*, and the lower task the *fine-grained task*. They constitute the whole hierarchical feature selection task.

For a clearer demonstration, we use Example 1 from the Bridge dataset, as shown in Fig. 2, to illustrate the process of task granulation [4, 33].

Example 1 The Bridge dataset is a classic dataset with six categories (*Wood*, *Suspen*, *Arch*, *Cantilv*, *Cont-t*, and *Simple*). They can be clustered into a class called *Truss* because of the similarity among the last three categories. For the convenience of describing the algorithm, we number each node in the tree. These numbers are the formal representation of the labels of the sample, as shown in Fig. 3. Therefore, we build a coarse-grained subtask on non-leaf node 8 and a fine-grained subtask on non-leaf node 7. Then, we can perform feature selection on these two subtasks separately. Task granulation is a process of gradual refinement from top to bottom. Different tasks should correspond to different datasets, so the original data should be split and changed accordingly. We call this process *data granulation*. By contrast, data granulation is a process of gradual coarsening from bottom to top.

A dataset is divided into several subsets according to the hierarchical tree structure. For a dataset $U = \{(x_i, l_i)\}_{i=1}^{N}$, $x_i = \{f_1, f_2, ..., f_n\}$ is a sample, where f_j represents a feature of x_i, l_i is the label of the sample x_i, and N is the number of samples. In this section, we split the dataset U into $U^{(k)} = \{\{x_i, l_i\}_{i=1}^{m_k}\}$ according to the tree structure, where $U^{(k)}$ is the sample set of the k^{th} subtask, l_i is the label of x_i for the k^{th} subtask, and m_k is the number of samples of $U^{(k)}$. To provide an intuitive understanding, we use Example 2 to introduce the data granulation approach.

Example 2 The Bridge dataset can be divided into two subsets corresponding to two subtasks for feature selection (coarse- and fine-grained). Figure 3 shows the subsets created with partial data from the Bridge dataset. Each row in the figure represents a sample, while the last column represents its labels. There are four types of labels in the Bridge1 dataset, which correspond to the second-level nodes of the hierarchical tree. The other subset, *Truss*, has three kinds of labels corresponding to the third-layer nodes on the hierarchical tree. Each subset requires an independent feature selection task, which we called a "subtask" before. The tasks can be executed simultaneously using a parallel pool because they are independent of each other, which greatly reduces time consumption. We adopt this method in the experiments.

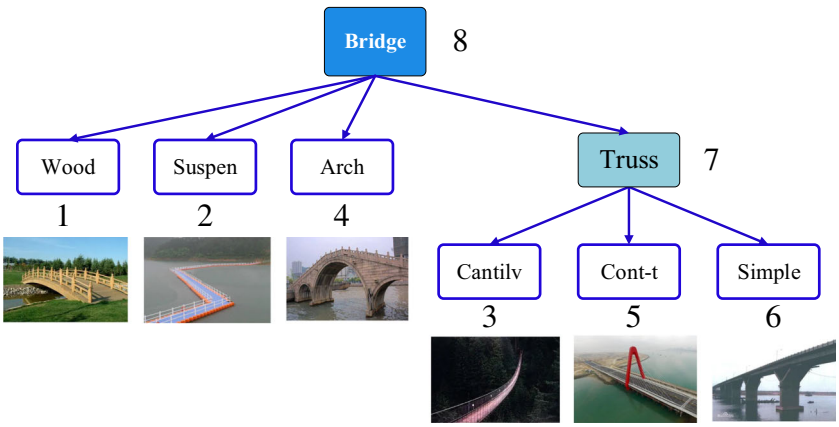

Fig. 2 Hierarchy tree of Bridge

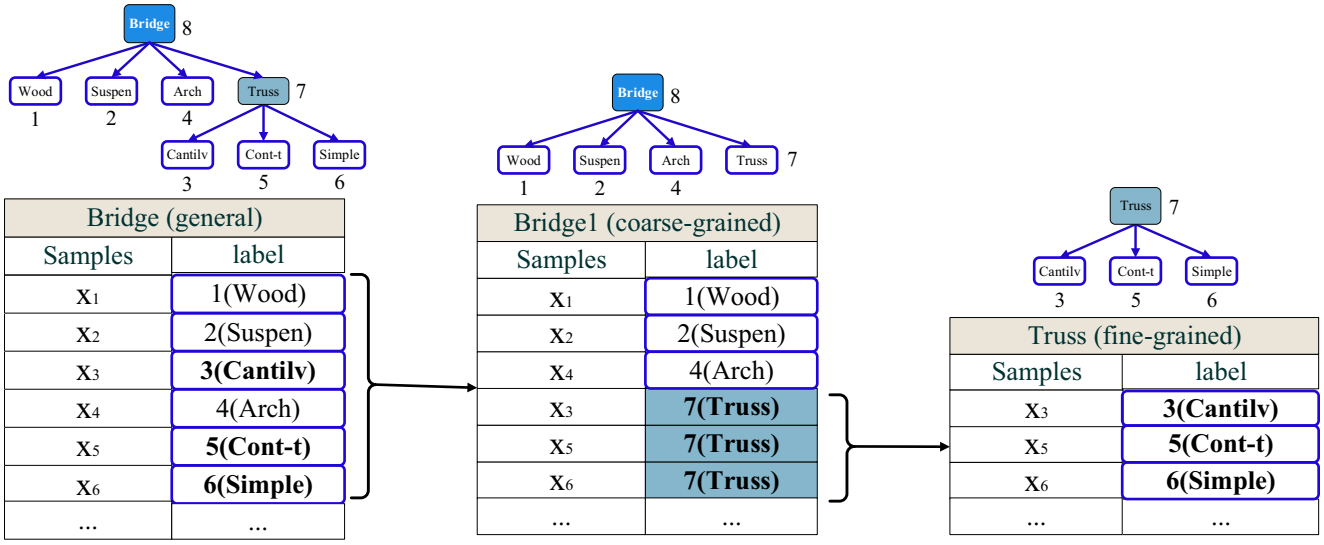

Fig. 3 Create sub-dataset from Bridge

A traditional feature selection algorithm does not need to consider the structural relationship of the data for small datasets. A subset of features with conspicuous classification capabilities can be selected under a single granularity framework. However, as the data grows rapidly, data labels are not independent of each other but have a certain structure, such as a semantic hierarchy [22]. This structure is manifested as samples being summarized into coarse-grained classes from fine-grained classes according to the data's semantic labels.

The granular computing method used in this article can effectively establish a user-centric concept based on the external world and simplify our understanding of the world [2, 40]. Likewise, in the process of solving problems, taking the appropriate granularity as the processing object not only ensures a satisfactory solution but also improves the problem-solving efficiency. In this paper, the hierarchical semantic structure of the data labels is used to granulate routine tasks and achieve feature selection in hierarchical datasets at different granularities.

3.3 Hierarchical fuzzy rough calculation based on Hausdorff distance

We propose a fuzzy rough set model based on a sample set, which is different from a traditional fuzzy rough set, which is only based on a single sample. On the one hand, fuzzy rough set models based on a single sample are strongly dependent on the quality of the samples; individual bad samples often reduce accuracy. Meanwhile, fuzzy rough set models based on sample sets are less affected by individual samples. On the other hand, when calculating the importance of each feature, cumbersome calculation of the lower approximations of single samples is no longer required. Instead, the lower approximation of the set of samples with the same labels is calculated, which greatly reduces the computational complexity. In addition, we introduce a threshold T to eliminate the features in each round of feature selection. By adjusting the threshold T, the calculation time and accuracy can be balanced, making it suitable for different scenarios.

Let $\langle U, R\rangle$ be a fuzzy approximation space, where the fuzzy set $X \subseteq F(U)$. For any subset $X_i \subseteq U$ meets that X_i is the sample subset with the same labels. The fuzzy lower approximation $\underline{R}X(X_i)$ of fuzzy set X with respect to $\langle U, R\rangle$ can be written as:

$$\begin{aligned} \underline{R}X(X_i) &= \inf_{X_j \subseteq U} \max\left(X(X_j), 1 - e^{H(X_i, X_j)}\right), \\ \overline{R}X(X_i) &= \sup_{X_j \subseteq U} \min\left(X(X_j), e^{H(X_i, X_j)}\right), \end{aligned} \tag{3}$$

where X_j is the sample subset with the same labels, $H(X_i, X_j)$ is the Hausdorff distance between the subset X_i and the subset X_j, and $X(X_j)$ can be understood as the degree to which the set X_j is included in the fuzzy set X, $X(X_j)$ can be obtained by calculating the average of the membership degrees of all elements in X_j to the fuzzy set X.

The Hausdorff distance is an efficient method for calculating the similarity between two point sets. We take two point sets $A = \{a_1, ..., a_m\}$ and $B = \{b_1, ..., b_n\}$, where m and n are the numbers of points A and B, respectively. We use $a_1, ..., a_m$ to represent the m points in set A, and $b_1, ..., b_n$ to represent the n points in set B. The Hausdorff distance formula can be written as:

$$H(A, B) = max\left\{h(A, B), h(B, A)\right\}, \tag{4}$$

where

$$h(A, B) = \max_{a \in A} \min_{b \in B} \| a - b \|, \tag{5}$$

where $\| \cdot \|$ indicates the normal form for calculating the distance; e.g., Euclidean norm. In this paper, we have slightly modified Hausdoff distance. For two sets A and B, we first calculate the mean point $\bar{a}$ of set A, and calculate the Euclidean distance between $\bar{a}$ with each point in set B, and retain the minimum value. Then calculate the mean point $\bar{b}$ of set B, calculate the Euclidean distance between $\bar{b}$ with each point in set A, and retain the minimum value. Finally, compare the smaller of the two retained small values as the distance between set A and set B, the time complexity is $O(n)$.

Let $U^{(k)} \in \mathbf{R}^{m_k \times n}$ be a input subset of the k^{th} subtask, where m_k is the number of samples and n is the number of features. We use $x_1, x_2, \cdots, x_{c_k}$ to represent the samples without labels of the k^{th} subtask, let $X^{(k)} = \{x_1, x_2, \cdots, x_{m_k}\}$, which represents the sample sets on the k^{th} subtask. Let $d_1, d_2, \cdots, d_{c_k}$ be the class set of $X^{(k)}$, where d_i is one class of samples labeled with i, and c_k is the number of classes of $X^{(k)}$. Let $\langle U^{(k)}, C, D^{(k)} \rangle$ be a decision system [8], where C is a non-empty set of condition attributes, and $D^{(k)}$ is the decision attribute of k^{th} subtask.

For all $x \in U^{(k)}$ and if R is a fuzzy similarity relation, then we have

$$d_i(x) = \begin{cases} 0, & x \notin \{d_i\} \\ 1, & x \in \{d_i\}, \end{cases} \tag{6}$$

where $x \in \{d_i\}$ indicates that the label of x is i. Then, the fuzzy rough approximations based on a sample set are computed as:

$$\underline{R} d_i(X_i^{(k)}) = \min_{i \neq j} (1 - e^{H(X_i^{(k)}, X_j^{(k)})}), \tag{7}$$

where $X_i^{(k)}$ is the sample set with label i of the k^{th} subtask, $X_j^{(k)}$ is the sample set with label j of the k^{th} subtask.

Following the fuzzy rough approximations definition, we can obtain the dependence of decision attribute $D^{(k)}$ on the condition attribute subset $B \subseteq C$ under the condition of the fuzzy rough set:

$$\gamma_B(D^{(k)}) = \frac{\sum_{i=1}^{c_k} \underline{R} d_i(X_i^{(k)})}{| U^{(k)} |}, \tag{8}$$

where $| \cdot |$ represents the number of samples and c_k is the number of classes of the k^{th} subtask.

Let C and $D^{(k)}$ be the condition attribute and decision attribute of the fuzzy rough set, respectively. The importance $\sigma_a(D^{(k)})$ of attribute $a \subseteq C$ with regard to $D^{(k)}$ can be written as:

$$\sigma_a(D^{(k)}) = \gamma_{B \cup a}(D^{(k)}) - \gamma_B(D^{(k)}). \tag{9}$$

A hierarchical feature selection algorithm based on fuzzy rough sets (HFRS) is proposed in Algorithm 1. The algorithm selects a feature subset for each subtask. A subset $U^{(k)}$ for the k^{th} subtask independently constitutes a decision system $\langle U^{(k)}, C, D^{(k)} \rangle$, where $U^{(k)}$ is the universe that represents the sample sets of the k^{th} subtask and C denotes the conditional feature set, and $D^{(k)}$ represents decision features of the k^{th} subtask. It starts from the empty selected set B, then gradually adds a feature $a \in C$ that helps in decision-making. Each time we try to add a feature a into the selected set B to help the decision, it must calculate the importance σ of the feature a according to Formula (9). If σ is greater than a given threshold T, we add the feature to the selected feature set B; otherwise, we skip this step and continue to try to add the next feature.

Algorithm 1 Hierarchical Feature Selection Based on Fuzzy Rough Set.

Input: subset $U^{(k)}$, the number of selected features nf
Output: selected feature set B

```
Take the U^(k) as a decision system ⟨U^(k), C, D^(k)⟩, B = ∅;
B = B ∪ {a}, where γ_a according to Algorithm 2 is maximum for a ∈ C;
C = C − B;
for each a ∈ C do
    Compute γ_{B∪a}(D^(k)) according to Algorithm 2;
    σ_a(D^(k)) = γ_{B∪a}(D^(k)) − γ_B(D^(k));
    if σ_a(D^(k)) > T then
        B = B ∪ {a};
    end if
end for
if length(B) < nf then
    C = C − B;
    for i = 1 to (nf − length(B)) do
        B = B∪{a}, where γ_a according to Algorithm 2 is maximum for a ∈ C;
    end for
else
    for i = 1 to (length(B) − nf) do
        B = B − {a}, where σ_a according to Algorithm 2 is minimum for a ∈ B;
    end for
end if
return B;
```

The Hausdorff distance is used to calculate the similarity between each class and the heterogeneous classes in the universe when calculating the feature dependency γ.

Algorithm 2 shows the specific calculation procedure. According to Equation (3), the class with the lowest dissimilarity to the current class is taken as the lower approximation. Furthermore, the lower approximation of all classes is the degree of dependence of the current attribute set.

Algorithm 2 Computing Dependencies γ.

Input: decision system $\langle U^{(k)}, C, D^{(k)} \rangle$, feature $a \in C$
Output: importance γ of feature $a \in C$

$\gamma = 0$;
Compute the number of classes of $U^{(k)}$ record as nc;
for $i = 1$ to nc **do**
  Select samples $X_i^{(k)}$ with label i;
  for $j = 1$ to nc **do**
    if $j \neq i$ **then**
      Select samples $X_j^{(k)}$ with label j;
      Compute $R(X_i^{(k)}, X_j^{(k)}) = e^{H(X_i^{(k)}, X_j^{(k)})}$;
    end if
  end for
  Compute $\underline{R}d_i(X^{(k)}) = \min_{i \neq j}(1 - e^{H(X_i^{(k)}, X_j^{(k)})})$;
  $\gamma = \gamma + \underline{R}d_i(X^{(k)})$;
end for
$\gamma = \gamma / nc$;
return γ;

Generally speaking, the lower approximation can be understood as the minimum dissimilarity between heterogeneous samples. The upper approximation is understood as the maximum similarity between similar sample sets. In the past, the similarity between samples was extracted from the original data via a Gaussian kernel to form the relationship matrix $R^{(k)}$ in fuzzy rough set theory [17]. Then, a fuzzy approximation operator is used to approximate the decision $D^{(k)}$. However, in fuzzy rough set algorithms, it is costly to employ a Gaussian kernel function to estimate the distances among samples. The time complexity requires $O(n^2)$, assuming that n is the number of samples. In our approach, the difference is that we employ the Hausdorff distance [1] to judge the similarity among classes rather than samples. The calculation time complexity is $O(n)$. If we adopt the Hausdorff distance as the similarity measure, the relation matrix $R^{(k)} = H(X_i^{(k)}, X_j^{(k)})$ contains the distance between any two classes in the dataset $X^{(k)}$, where $X_i^{(k)}$ and $X_j^{(k)}$ represent two sample sets of different classes in the data. It is obvious that the relations between classes based on Hausdorff distances satisfy the properties of reflexivity, symmetry, and transitivity defined in Section 2, as well as the Gaussian kernel. In fuzzy classification, the relation matrix $R^{(k)}$ characterizes the fuzzy equivalence relationship between classes.

4 Experiment

The experiments described in this section elaborate on the performance of our algorithm. First, all the datasets and evaluation metrics involved in the experiment are described. Then, we compare HFRS with seven current feature selection algorithms. Subsequently, comparisons are made between using HFRS with different parameter T values. Finally, the running time of the various algorithms are compared.

4.1 Dataset description

In our experiments[1], five representative public datasets are adopted to demonstrate the performance of the various feature selection algorithms. These datasets include three image datasets—VOC, ILSVRC, and Car196—and two protein datasets—DD and F194. Information on the datasets is summarized in Table 1 and the datasets are briefly described as follows.

1) VOC [16] contains 12,283 samples of 20 classes. These classes have a 5-layer tree structure. Figure 4 shows the hierarchy tree.
2) ILSVRC [23] contains data from the Large Scale Visual Recognition Challenge competition. It contains 24,191 images of 57 classes.
3) Car196 [31] is a dataset containing fine-grained images of vehicles. The Car196 dataset contains 196 classes of cars with 4,096 features. These classes have a three-layer tree structure.
4) DD [13] is a dataset including 3,020 protein sequences for training and 605 protein sequences for testing. It has 27 classes with a 3-layer tree structure.
5) F194 [38] is a dataset with 194 classes containing 8,525 protein sequences. Each sequence is characterized by 473 features.

4.2 Comparison algorithm and evaluation metrics

We set up three sets of comparative experiments to compare the proposed method with the recent feature selection algorithms PCC, FSNM, PLS, ILFS, BDFS, DSSA, and HGSO.

- Baseline: All original features are selected.
- The PCC method helps explain the relationships between features and response variables. It selects feature subsets by measuring the linear correlations between variables [10].

[1]Datasets and code used in this research have been explained and uploaded to GitHub. They are accessible at: https://github.com/fhqxa/APIN-HFRS.

Table 1 Dataset description

Name	Train	Test	Features	Classes	Layers
VOC	7,178	5,105	1,000	20	5
ILSVRC	12,346	11,845	4,096	57	4
Car196	8,144	7,541	4,096	196	3
DD	3,020	605	473	27	3
F194	7,105	1,420	473	194	3

- The FSNM is an efficient feature selection algorithm that assembles a loss function with regularization of $l_{2,1}$-norm minimization to improve its performance [25].
- The PLS algorithm selects features based on partial least squares, which achieves leading performance on small highly-dimensional samples [41].
- The ILFS algorithm is based on probabilistic latent graphs. It implements a ranking process by enumerating whole subsets of features as paths of a graph [29].
- The BDFS algorithm selects features by constructing structured sparse subspace learning modules to solve the long-standing subspace sparsity problem [37].
- The SSA is a single-objective optimization algorithm, which is inspired by the group behavior of seaweed when sailing and foraging in the ocean. DSSA enhances the diversity of SSA solutions and develops a local search algorithm to improve the use of SSA [34].
- The HGSO is a dimensionality reduction method that uses the Henry Gas Solubility Optimization algorithm to select important features to improve classification accuracy [24].

Our comparison methods are representative methods proposed in the past ten years or three years, and they are known to many researchers in the field of feature selection. In particular, they are all quite competitive in the quality and speed of selecting feature subsets [29, 34, 37]. Among them PCC, PLS, and ILFS are filter methods, according to some indicators of each feature, to select highly relevant features; FSNM and BDFS are embedded methods, which evaluate the pros and cons of feature subsets according to the error on the learner; and DSSA and HGSO are wrapper methods, which integrate the process of feature selection with the training process of the learner, and automatically perform feature selection. We verify the effectiveness of our method by comparing it with these three types of methods.

To evaluate the performance of the different algorithms in feature subset selection, the following three metrics are used [20]:

1) *Accuracy* - The degree to which the average value of the results of many experiments performed under certain conditions is consistent with the real value. It is used to measure the degree of system error.

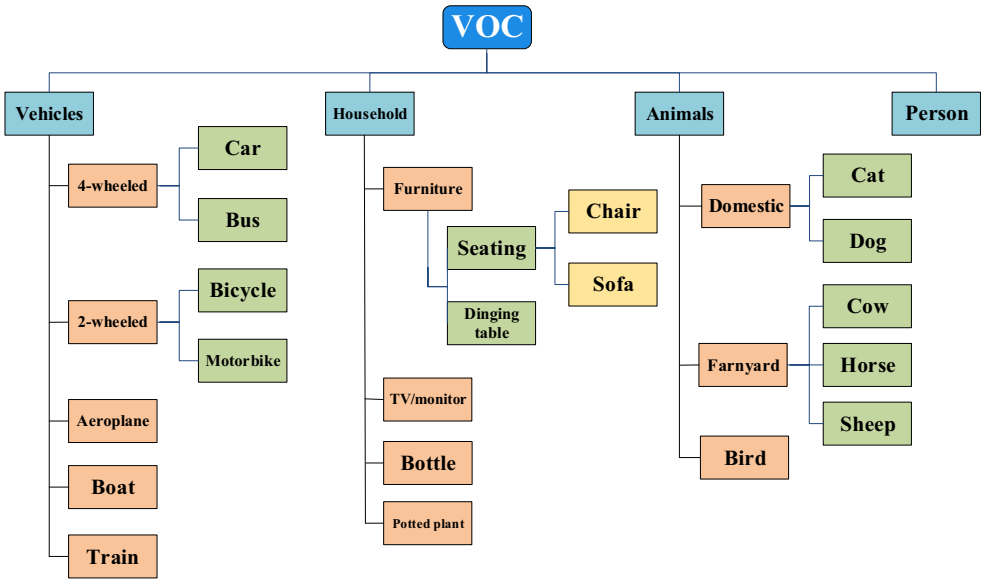

Fig. 4 Hierarchy tree of VOC

2) TIE - A measure of the distance between a sample's predicted category and the real category on a hierarchical tree. TIE is represented below:

$$TIE(Y, \hat{Y}) = \sum |E_h(y, \hat{y})|, \tag{10}$$

where Y denotes the true label set, $\hat{Y}$ denotes the predicted label set, and $y \in Y$ indicates the real label of sample x, while $\hat{y} \in \hat{Y}$ indicates the predicted label of sample x. $E_h(y, \hat{y})$ represents the set of edges from $\hat{y}$ to y and $|\cdot|$ is expressed as the number of elements of the set.

3) The F_H - value is proposed to evaluate overall precision and recall. Hierarchical precision P_H and recall R_H can be expressed as:

$$P_H = \frac{|\hat{Y}_{aug} \cap Y_{aug}|}{|\hat{Y}_{aug}|}, R_H = \frac{|\hat{Y}_{aug} \cap Y_{aug}|}{|Y_{aug}|}, \tag{11}$$

where $\hat{Y}_{aug}$ and Y_{aug} defined in [19] are an augmentation of $\hat{Y}$ and Y, they attempt to acquire the hierarchical relationship of classes. Defining F_H is similar to F_1 in general classification task. F_H is formulated as follows:

$$F_H = \frac{2 \cdot P_H \cdot R_H}{P_H + R_H}. \tag{12}$$

In the experiments, training datasets are used for the feature selection tasks. Test datasets with selected features are used for hierarchical classification, adopting the support vector machine classifier ($C = 1$) using a top-down method with ten-fold cross-validation by LIBSVM [6, 7]. This process goes from top-to-bottom and proceeds layer-by-layer until reaching a leaf node along the hierarchy tree. Finally, we use the labels predicted in fine-grained tasks to calculate the evaluation metrics.

4.3 Comparison between HFRS and other feature selection algorithms

In this work, we report on the performance of the different methods. For the sake of experimental fairness, we vary the number of selected features while holding the other parameters constant, and use the average classification metrics as the ultimate result.

Figure 5 illustrates the classification accuracies of the different feature selection methods applied to the five datasets. Since using HFRS with various values of T may have varying efficiency, we adjust parameter T and show the optimum results.

According to Fig. 5, we make the following observations:

(1) The accuracies of the different feature selection methods in classifying the different datasets increase significantly with the number of selected features;

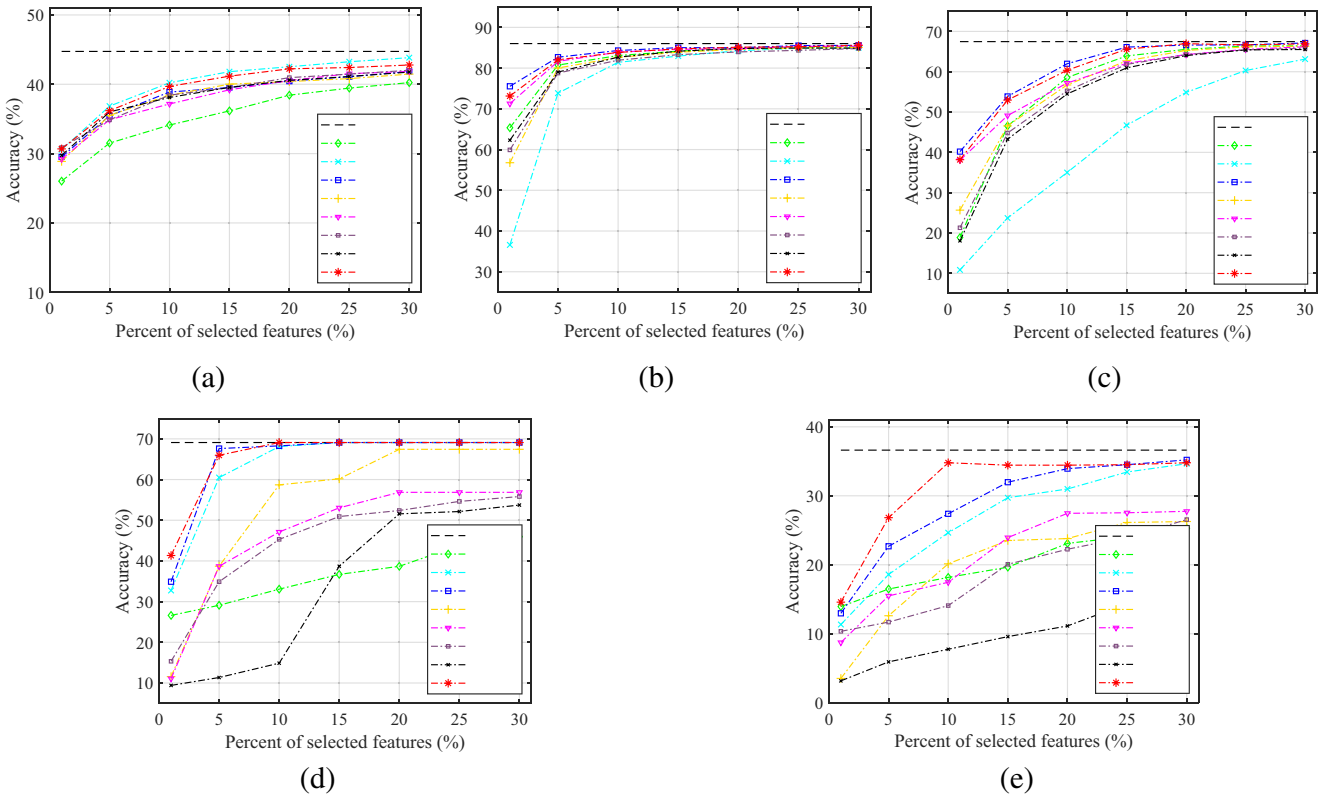

Fig. 5 Accuracy of the different feature sets using different feature selection methods. (a) VOC; (b) ILSVRC; (c) Car196; (d) DD; (e) F194

however, the specific performances differed. For datasets VOC, DD, and F194, the classification accuracy of PCC in terms of the algorithm's minimum, maximum, and convergence rate are far less than those of other algorithms with more selected features. For datasets ILSVRC and Car196, a similar situation occurs with the FSNM algorithm.

(2) Methods ILFS, BDFS, HGSO, and BDFS all have achieved good performance on the image dataset, but have not achieved satisfactory results on the protein dataset. This may be because these methods are not applicable to protein datasets.

(3) In contrast, HFRS and PLS achieve high accuracy on all five datasets, and an increasing trend of accuracy is evident. Overall, HFRS is better because it can always obtain comparable or higher accuracy than the other methods.

To better compare the specific performance of the different algorithms, the number of features selected via different methods for a specific dataset should be consistent [5]. For the VOC dataset, we use 200 selected features. For the ILSVRC and Car196 datasets, we select 820 features, which is 20% of the total features. However, the experiment shows that for the two protein datasets (DD and F194) with 10% of the features selected (i.e., 50 features) is best. We adjust the other super-parameters of each algorithm to ensure that they provide the best results on specific datasets.

Tables 2, 3 and 4 reveal the results of our experiments, where the "↓" indicates "the smaller the better", and the "↑" indicates "the larger the better". RANKS indicates the average ranking of each algorithm in classifying the five datasets.

Accuracy on five datasets of different feature selection methods is listed in Table 2. From this table, we can get the following observations.

(1) HFRS achieves the highest accuracy on the four datasets and second place on the VOC dataset. In particular, it is 0.85% higher than the second place on the F194 dataset.

(2) The results of HFRS on image and protein datasets are relatively stable, the average rank on the five datasets is 1.2. While other methods always only perform well on a certain type of dataset or a certain dataset. For example, the accuracy of the PCC algorithm on the ILSVRC dataset is 84.80%, while it is only 22.17% on the F194 dataset.

TIE is a metric used to measure the severity of mispredictions. The larger its value, the more serious the algorithm's error. Therefore, the smaller the value of TIE, the better the algorithm. The TIE results on five datasets by different feature selection methods are listed in Table 3. From this table, we can get the following observations.

(1) The TIE of HFRS is lower than other methods on the four datasets, only slightly higher than the FSNM method on the VOC dataset, and ranking first overall. This shows that the error label given by HFRS in the misclassified samples deviates from the true label to the smallest extent.

(2) HFRS and PLS methods have the same accuracy on the DD dataset, but the TIE (51 edges) of HFRS is 1.6 edges lower than the TIE (52.6 edges) of PLS. This shows that HFRS can not only reduce the deviation between misclassified labels and real labels, but also ensure the accuracy of classification.

The F_H results on five datasets of different feature selection methods are listed in Table 4. From this table, we can get that HFRS achieves the best results on the Car196, DD, and F194 datasets, and the second-best results on the VOC and ILSVRC datasets. Especially on the F194 dataset, it improves the F_H by 2% compared with that of the second-best algorithm.

4.4 Comparison of HFRS with different threshold values *T*

We also test the sensitivity of the model to the threshold T based on classification accuracy. The value of T balances

Table 2 Accuracy on five datasets by different feature selection methods (%). ↑

Model	VOC	ILSVRC	Car196	DD	F194	RANKS
PCC	38.43	84.80	65.56	33.06	22.17	6.2
FSNM	**42.53**	84.90	59.22	68.11	30.99	3.6
PLS	40.45	85.00	66.61	69.10	33.94	2.8
ILFS	40.39	84.60	65.56	67.43	23.80	5.4
BDFS	40.55	84.73	64.26	56.86	27.46	5.2
DSSA	40.96	83.97	65.26	52.39	22.25	5.6
HGSO	40.57	84.82	64.01	51.57	11.13	6.0
HFRS	42.23	**85.13**	**66.96**	**69.10**	**34.79**	1.2

The number in bold represents the best result of different methods

Table 3 TIE on five datasets by different feature selection methods (edges). ↓

Model	VOC	ILSVRC	Car196	DD	F194	RANKS
PCC	1,181.5	405.4	821.8	111.0	292.8	5.8
FSNM	**1,094.6**	415.8	1,174.2	51.6	295.4	3.8
PLS	1,149.3	397.6	791.2	52.6	266.6	2.2
ILFS	1,121.9	406.6	836.6	64.8	291.8	4.6
BDFS	1,133.7	401.4	867	86.6	299.4	5.6
DSSA	1,122.4	425.6	868.6	92.4	280.8	5.8
HGSO	1,135.2	402.6	871.8	178.6	426.6	6.8
HFRS	1,098.9	**392.8**	**783.6**	**51.0**	**241.6**	1.2

The number in bold represents the best result of different methods

the accuracy and running time of the HFRS method. The larger the value, the longer the running time, because it requires more cycles to select features according to Algorithm 1. The smaller the value of T, the shorter the running time of our method. However, the selected features cannot be guaranteed to be the optimal subset of features. Therefore, accuracy decreases with decreases in the T value. The VOC, DD, and F194 datasets are employed in the experiments, with T values set to {0.1, 0.05, 0.01, 0.005, 0.001, 0.0005} and the other parameters held constant. The number of selected features is the same as specified in the previous section.

Accuracy and running time on three datasets with different threshold T are listed in Table 5. From this table, we can get the following observations.

(1) On the VOC dataset, the accuracies of HFRS with T = 0.1, 0.05, 0.01, 0.005, 0.001, and 0.0005 are very close. On the DD dataset, the accuracy decreases as T decreases. For instance, the accuracy of HFRS when $T = 0.1$ is 69.10%, which is 1.83% better than it of HFRS when $T = 0.0005$.
(2) On the F194 dataset, as the value of T decreases, the accuracy first increases and then decreases. More features irrelevant to classification are filtered as the value of T decreases, so the classification accuracy increases. As the T value continues to decrease, some features that are helpful for classification are also discarded, so the classification accuracy decreases.
(3) In terms of running time, there are some similar cases in all three datasets. HFRS takes a long time when T is high and, as T decreases, its time consumption is much lower. This result is consistent with our initial analysis. Parameter T balances the performance and time consumption of the HFRS algorithm. Adjusting the T value can make HFRS achieve better performance in a shorter time.

4.5 Running time comparison

We compare the calculation time of HFRS and the five other algorithms on the five representative datasets VOC, ILSVRC, Car196, DD, and F194. All experiments are implemented on a computer with four 2.27 GHz processors and 16 GB of available RAM.

In past research, feature selection and attribute-reduction algorithms based on the rough set method have performed weakly compared with other algorithms in terms of computation time. Because the calculation of each feature's

Table 4 F_H on five datasets by different feature selection methods (%). ↑

Model	VOC	ILSVRC	Car196	DD	F194	RANKS
PCC	64.81	95.72	81.84	69.41	65.63	4.8
FSNM	**67.39**	95.61	82.29	85.79	65.33	3.4
PLS	65.87	**95.80**	82.51	85.51	68.71	2.6
ILFS	66.58	95.71	81.51	82.15	65.75	4.0
BDFS	66.26	95.76	80.84	76.14	64.86	4.8
DSSA	66.55	95.51	80.80	74.55	67.04	4.0
HGSO	66.37	95.75	80.73	50.80	49.93	4.8
HFRS	67.30	95.79	**82.68**	**85.95**	**70.73**	1.4

The number in bold represents the best result of different methods

Table 5 Accuracy and running time on three datasets with different threshold T

T	Accuracy (%)			Running time (s)		
	VOC	DD	F194	VOC	DD	F194
0.1	40.22	69.10	30.99	437.09	4.92	35.64
0.05	41.43	68.61	30.70	450.37	4.89	36.09
0.01	41.43	67.95	30.35	477.85	5.35	35.71
0.005	40.94	67.95	32.61	474.42	5.07	31.76
0.001	41.18	67.45	30.14	285.79	3.03	34.22
0.0005	41.27	67.27	25.21	69.77	2.39	16.73

importance is very time-consuming, many cycles are required to calculate the feature importance when selecting features [21, 27].

Table 6 reports the results of running time by different methods. From Table 6, we can get the following conclusions.

(1) HFRS significantly reduces the running time compared to some other excellent methods, such as BDFS, DSSA, and HGSO. For example, on the Car196 dataset, HFRS takes 3900s running time, while BDFS takes 35303s, which is more than 10 times that of HFRS.
(2) Although PCC, PLS and ILFS outperform HFRS in running time, their classification performance is not as good as HFRS. Therefore, HFRS is a balanced method in terms of running time and classification performance.

4.6 Discussion and analysis

This paper proposes a hierarchical feature selection method that uses a hierarchical tree structure of a dataset's classes to continuously granulate tasks. After experimental verification of the advantages of our method, this section discusses the value of the model at the application level.

(1) In the era of big data, many massive data processing tasks involve hierarchical classification, such as social network data analysis and weather forecasting. Therefore, in this paper, we use hierarchical task granulation. Any task that is beyond the available computing power can be granulated into smaller sub-tasks layer-by-layer according to the hierarchical structure of the dataset's classes.
(2) In big data scenarios, many features are often redundant. This may lead to misclassification. Therefore, feature selection plays an important role in classification. Feature selection can be used in face recognition, big data classification, target detection, target tracking, and other important fields. It not only reduces the difficulty of the task but also improves the processing speed.

5 Conclusions and future work

This paper proposed a feature selection method inspired by the concepts of hierarchical classification and reduced-attribute algorithms built on fuzzy rough set. We use the dataset's hierarchical structure to decompose the original feature selection task into tasks with different granularities from coarse to fine. In each task, we use a fuzzy rough

Table 6 Running time by different methods on five datasets (s). ↓

Model	VOC	ILSVRC	Car196	DD	F194
PCC	0.33	1.95	0.83	0.04	0.08
FSNM	193.48	1,681.1	329.05	13.63	171.79
PLS	0.51	6.96	3.13	0.11	0.76
ILFS	7.67	175.26	251.24	0.8	2.82
BDFS	1,687.6	40,006	35,303	79.44	131.85
DSSA	1,122.4	5,515	20,941	202.7	493.7
HGSO	961.3	11,501	46,417	110.7	464.1
HFRS	30.48	4,545.3	3,900	0.39	2.43

set method to select features, with the distances between samples calculated as Hausdorff distances instead of by the traditional Gaussian kernel. We also introduced a threshold T to reduce unnecessary cycles in the feature selection process, thus greatly reducing running time. Through three groups of comparative experiments, the superiority of this algorithm is verified. At the same time, some new problems are identified. In reality, hierarchical datasets are mostly incomplete and have numerous missing values. In future, we will attempt to use fuzzy rough sets for feature selection and to classify incomplete datasets.

Acknowledgements This work was supported by the Natural Science Foundation of Fujian Province under Grant No. 2021J011003.

References

1. Aksoy S, Nowak K, Purvine E, Young S (2019) Relative hausdorff distance for network analysis. Appl Netw Sci 4(1):80–105
2. Bargiela A, Pedrycz W (2016) Granular computing. In: Fuzzy logic, systems, artificial neural networks, and learning systems, pp 43–66
3. Blanco Mesa F, Merigó J, Gil Lafuente A (2017) Fuzzy decision making: a bibliometric-based review. J Intell Fuzzy Syst 32(3):2033–2050
4. Cai R, Qiao J, Zhang K, Zhang Z, Hao Z (2018) Causal discovery from discrete data using hidden compact representation. In: Advances in neural information processing systems, pp 2666–2674
5. Cai Z, Zhu W (2018) Multi-label feature selection via feature manifold learning and sparsity regularization. Int J Machine Learn Cybern 9(8):1321–1334
6. Cerri R, de Carvalho A (2010) New top-down methods using SVMs for hierarchical multilabel classification problems. In: International joint conference on neural networks, pp 1–8
7. Cesa-Bianchi N, Gentile C, Zaniboni L (2006) Hierarchical classification: combining bayes with SVM. In: International conference on machine learning, pp 177–184
8. Chen D, Zhao S (2010) Local reduction of decision system with fuzzy rough sets. Fuzzy Sets Syst 161(13):1871–1883
9. Cheng M, Liu Y, Hou Q, Bian J, Torr P, Hu S, Tu Z (2016) HFS: hierarchical feature selection for efficient image segmentation. In: European conference on computer vision, pp 867–882
10. Coelho F, Braga A, Verleysen M (2010) Multi-objective semi-supervised feature selection and model selection based on pearson's correlation coefficient. In: Iberoamerican congress on pattern recognition, pp 509–516
11. Deng J, Dong W, Socher R, Li L, Li K, Li F (2009) Imagenet: a large-scale hierarchical image database. In: IEEE Conference on computer vision and pattern recognition, pp 248–255
12. Deng Z (2018) An efficient structure for fast mining high utility itemsets. Appl Intell pp(48) 3161–3177
13. Ding C, Dubchak I (2001) Multi-class protein fold recognition using support vector machines and neural networks. Bioinformatics 17(4):349–358
14. Ding W, Chang B (2008) Improving chinese semantic role classification with hierarchical feature selection strategy. In: Empirical methods in natural language processing, pp 324–333
15. Dubois D, Prade H (1990) Rough fuzzy sets and fuzzy rough sets. Int J Gen Syst 17(2):191–209
16. Everingham M, Van Gool L, Williams C, Winn J, Zisserman A (2010) The pascal visual object classes (VOC) challenge. Int J Comput Vis 88(2):303–338
17. Hu Q, Yu D, Pedrycz W, Chen D (2011) Kernelized fuzzy rough sets and their applications. IEEE Trans Knowl Data Eng 23(11):1649–1667
18. Jensen R, Shen Q (2009) New approaches to fuzzy-rough feature selection. IEEE Trans Fuzzy Syst 17(4):824–838
19. Kononenko I (1994) Estimating attributes: analysis and extensions of relief. In: European conference on machine learning, pp 171–182
20. Kosmopoulos A, Partalas I, Gaussier E, Paliouras G, Androutsopoulos I (2015) Evaluation measures for hierarchical classification: a unified view and novel approaches. Data Min Knowl Disc 29(3):820–865
21. Kryszkiewicz M (1998) Rough set approach to incomplete information systems. Inf Sci 112(1-4):39–49
22. Kuipers B (2000) The spatial semantic hierarchy. Artif Intell 119(1-2):191–233
23. Liu X, Zhao H (2019) Hierarchical feature extraction based on discriminant analysis. Appl Intell 49(7):2780–2792
24. Nabil N, Essam H, Kashif H (2020) An efficient henry gas solubility optimization for feature selection. Expert Syst Appl 152(3):364–372
25. Nie F, Huang H, Cai X, Ding C (2010) Efficient and robust feature selection via joint $\ell_{2,1}$-norms minimization. In: Advances in neural information processing systems, pp 1813–1821
26. Pawlak Z (1982) Rough sets. Int J Comput Inform Sci 11(5):341–356
27. Qian Y, Liang J, Pedrycz W, Dang C (2010) Positive approximation: an accelerator for attribute reduction in rough set theory. Artif Intell 174(9–10):597–618
28. Reich Y, Fenves S (1989) Integration of generic learning tasks. Engineering Design Research Center, Carnegie Mellon University, Pittsburgh 24(1):1–12
29. Roffo G, Melzi S, Castellani U, Vinciarelli A (2017) Infinite latent feature selection: a probabilistic latent graph-based ranking approach. In: IEEE International conference on computer vision, pp 1398–1406
30. Ruvolo P, Fasel I, Movellan J (2010) A learning approach to hierarchical feature selection and aggregation for audio classification. Pattern Recogn Lett 31(12):1535–1542
31. Sohn K (2016) Improved deep metric learning with multi-class n-pair loss objective. In: Advances in neural information processing systems, pp 1857–1865
32. Sun K, Mou S, Qiu J, Wang T, Gao H (2018) Adaptive fuzzy control for nontriangular structural stochastic switched nonlinear systems with full state constraints. IEEE Trans Fuzzy Syst 27(8):1587–1601
33. Tang W, Mao K (2007) Feature selection algorithm for mixed data with both nominal and continuous features. Pattern Recogn Lett 28(5):563–571
34. Tubishat M, Ja'afar S, Alswaitti M, Mirjalili S, Idris N (2021) Dynamic salp swarm algorithm for feature selection. Expert Syst Appl 164(7):873–887
35. Wang N, Li W, Jiang T, Lv S (2017) Physical layer spoofing detection based on sparse signal processing and fuzzy recognition. IET Signal Process 11(5):640–646
36. Wang S, Zhu W (2018) Sparse graph embedding unsupervised feature selection. IEEE Trans Syst Man Cybern Syst 48(3):329–341
37. Wang Z, Nie F, Tian L, Wang R, Li X (2020) Discriminative feature selection via a structured sparse subspace learning module. In: International joint conference on artificial intelligence, pp 3009–3015

38. Wei L, Liao M, Gao X, Zou Q (2015) An improved protein structural classes prediction method by incorporating both sequence and structure information. IEEE Trans Nanobiosci 14(4):339–349
39. Xu W, Sun W, Liu Y, Zhang W (2013) Fuzzy rough set models over two universes. Int J Machine Learn Cybern 4(6):631–645
40. Yao Y (2016) A triarchic theory of granular computing. Granular Computing 1(2):145–157
41. You W, Yang Z, Ji G (2014) PLS-Based recursive feature elimination for high-dimensional small sample. Knowl-Based Syst 55:15–28
42. Zhang X (2018) Pythagorean fuzzy clustering analysis: a hierarchical clustering algorithm with the ratio index-based ranking methods. Int J Intell Syst 33(9):1798–1822
43. Zhao H, Wang P, Hu Q, Zhu P (2019) Fuzzy rough set based feature selection for large-scale hierarchical classification. IEEE Trans Fuzzy Syst 27(10):1891–1903
44. Zhu W (2009) Relationship among basic concepts in covering-based rough sets. Inf Sci 179(14):2478–2486

Zeyu Qiu is currently a M.S. student with the School of Computer Science, Minnan Normal University, Zhangzhou, China. His current research interests include data mining and machine learning for hierarchical feature selection and hierarchical classification.

Hong Zhao received the Ph.D degree from Tianjin University, Tianjin, China, in 2019. She is currently a Professor of the School of Computer Science, Minnan Normal University, Zhangzhou, China. She has authored over 40 journal and conference papers in the areas of granular computing based machine learning and cost-sensitive learning. Her current research interests include rough sets, granular computing, and data mining for hierarchical classificaion.